Inconsistent Reasoning Attacks to Identify Weaknesses in Automatic Scientific Claim Verification Tools

Md Athikul Islam[0009−0007−9223−6852], Noel Ellison[0009−0001−4875−195X], Bishal Lakha[0009−0001−8234−4389], and Edoardo Serra[0000−0003−0689−5063]

Boise State University, Boise, Idaho, USA
{mdathikulislam, noelellison, bishallakha}@u.boisestate.edu
edoardoserra@boisestate.edu

Abstract. Scientific Claim Verification (SCV) tools are essential for evaluating the validity of scientific assertions, particularly within autonomous science. However, they often struggle to interpret complex scientific language and detect reasoning flaws, leading to potential misclassification. Adversarial attacks, particularly paraphrase attacks, reveal these weaknesses by rewording claims while maintaining their meaning. Paraphrase attacks are not the only way to identify weaknesses in SCV tools, but other existing methods often fail to preserve semantic equivalence, requiring extensive human filtering.
To address this, we define inconsistent reasoning attacks, a broader class of adversarial attack strategies that expose logical weaknesses in SCV systems. Using an evolutionary algorithm and large language models, this approach iteratively modifies claims to trigger misclassifications while maintaining logical inconsistencies. This method improves semantic accuracy and attack effectiveness, particularly for paraphrase-based attacks. Evaluation against a leading SCV system (MultiVerS) confirms persistent vulnerabilities, even though a retrieval-augmented generation (RAG) system with an Attack-Reflection mechanism shows potential in mitigating these issues. The findings emphasize the susceptibility of SCV systems to reasoning inconsistencies with a larger attack success rate than other attack techniques and highlight the Attack-Reflection mechanism as a promising defense.

Keywords: Automatic Scientific Claim · Verification Tools · Adversarial Attacks · Robustness · Large Language Models

1 Introduction

Scientific Claim Verification (SCV) tools are essential for assessing the validity of scientific claims, particularly in the emerging field of autonomous science or discovery [16, 2] and the fast-evolving landscape of social media [21]. However, these tools face significant challenges due to the complexity of scientific language, which requires access to up-to-date research and a deep understanding

of claims for accurate verification [27, 23]. MultiVerS [26], one of the most advanced SCV models, leverages multitask loss for rationale selection and label prediction using long-document transformers. Despite these advancements, SCV tools remain vulnerable to reasoning errors, leading to incorrect or inconsistent claim classifications.

Standard NLP adversarial attacks [10, 18, 6] can be used to evaluate the robustness of SCV tools by altering text while attempting to preserve its meaning. However, in the context of SCV, these approaches often fail because the semantics of scientific claims are highly sensitive to small changes. For instance, replacing "Nonsteroidal anti-inflammatory drugs are ineffective as cancer treatments" with "Nonsteroidal anti-inflammatory drugs are indispensable as cancer treatments" drastically alters the meaning despite a minor word substitution. While these methods may work well in general NLP tasks, they do not adequately capture the strict semantic precision required for scientific claims, leading to misclassifications by SCV tools. To address these limitations, *paraphrase attacks* have been proposed as a more targeted adversarial strategy [12]. These attacks attempt to generate reworded claims that maintain semantic equivalence, thereby exposing SCV systems' weaknesses when semantically identical claims receive different truthfulness classifications. However, despite their improvements over standard NLP adversarial attacks, existing paraphrase attack methods still struggle to preserve meaning fully. Many generated paraphrases subtly alter the scientific validity of a claim, necessitating human intervention to filter out invalid attacks. This reliance on manual validation reduces the scalability of paraphrase attacks and limits their application in automated adversarial testing.

While paraphrase attacks provide a valuable means of identifying SCV vulnerabilities, they are not the only way to reveal inconsistencies in reasoning. Beyond lexical or syntactic changes, SCV tools can also be challenged by logical inconsistencies that do not rely solely on paraphrasing. To better capture these weaknesses, we introduce a broader taxonomy of adversarial attacks, which we term *inconsistent reasoning attacks*. These attacks systematically expose flaws in SCV decision-making by manipulating claim logic rather than just altering surface-level text. In addition to paraphrase attacks, inconsistent reasoning attacks also include specific-to-general attacks, which broaden a claim's scope by making it more general or vague (e.g., replacing "car" with "vehicle" or "strongly related" with "related"), and general-to-specific attacks, which instead refine broad claims by replacing general terms with more precise ones (e.g., replacing "vehicle" with "car").

Another type of inconsistent reasoning attack is negation manipulation, where the logical structure of a claim is altered by introducing double negatives or modifying negation patterns, like adding "does not fail to" into an affirmative statement. Union attacks are another form of inconsistent reasoning attacks that merge multiple claims into a single statement, introducing logical complexity that can mislead SCV systems and result in misclassified claims. By challenging SCV tools through these various logical inconsistencies, rather than just surface-

level linguistic modifications, inconsistent reasoning attacks provide a more comprehensive framework for evaluating system vulnerabilities. Additionally, one of the key contributions of our approach is the ability to merge multiple types of reasoning attacks, compounding their effects to create stronger adversarial examples. This capability allows for more effective stress-testing of SCV tools by generating claims that simultaneously exploit multiple logical inconsistencies.

To generate these attacks, we develop a genetic algorithm leveraging large language models that iteratively evolve claims through mutation and crossover operations. This method ensures that inconsistencies in SCV classifications are systematically identified while minimizing invalid transformations. Unlike paraphrase-based approaches, our framework provides a more robust assessment of SCV weaknesses by uncovering reasoning errors rather than focusing solely on linguistic variation.

We evaluate these attacks against MultiVerS and a retrieval-augmented generation (RAG) system with Attack-Reflection mechanisms. Our findings indicate that MultiVerS, despite being one of the most robust SCV models, remains highly susceptible to inconsistent reasoning attacks. However, the RAG system with Attack-Reflection demonstrates promising potential in mitigating these vulnerabilities in zero-shot learning tasks, suggesting that integrating Attack-Reflection mechanisms could enhance SCV robustness against adversarial manipulations.

More specifically, this work presents the following key contributions:

- Beyond Paraphrase Attacks: We introduce *inconsistent reasoning attacks*, exposing logical inconsistencies in SCV decision-making beyond standard NLP adversarial techniques and paraphrase attacks.
- Evolutionary Attack Generation: We develop a *genetic algorithm* leveraging LLMs to iteratively craft adversarial claims that reveal reasoning failures in SCV tools. The main novelties include distinct attack strategies for each type of inconsistent reasoning attack and an LLM-based crossover operation.
- Multi-Type Inconsistency Attacks: Our method improves attack success rates, minimizes invalid transformations, and enables *attack merging* to create more challenging adversarial cases.
- SCV Tool Vulnerability Analysis: We assess MultiVerS, demonstrating its susceptibility to inconsistent reasoning attacks and exposing weaknesses.
- Mitigation via RAG with an Attack-Reflection Mechanism: We explore self-reflective mechanisms over attacks in RAG-based SCV models to address logical inconsistencies and enhance resilience against attacks.

2 Related Works

Scientific Claim Verification (SCV) tools aim to assess the truthfulness of claims by retrieving relevant research abstracts and analyzing supporting or refuting evidence. MultiVerS [26] is considered one of the most advanced and most effective SCV tools. MultiVerS retrieves relevant abstracts, selects rationale statements, and classifies the claim as SUPPORT or REFUTE. MultiVerS enhances prior methods by incorporating full abstracts using a long-document transformer

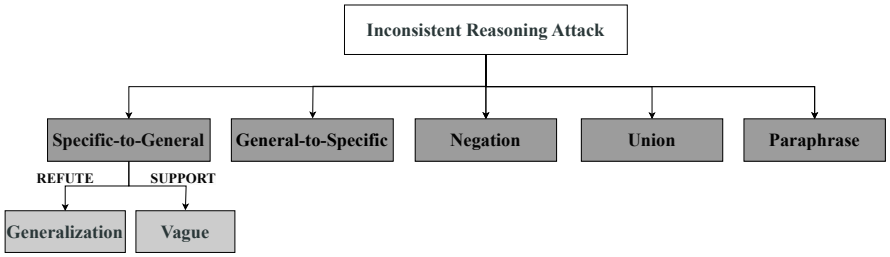


Fig. 1: Display of categories and subcategories of inconsistent reasoning attacks.

trained on the Semantic Scholar Open Research Corpus (S2ORC) [1], employing a multitask loss for rationale selection and label prediction.

The retrieval mechanism in SCV tools parallels Retrieval-Augmented Generation (RAG) [13], which enhances large language models (LLMs) by retrieving relevant document excerpts to improve factual grounding [5]. This mitigates hallucinations [8] and improves claim verification accuracy [11, 22, 3]. Unlike traditional SCV systems that rely on selecting static evidence, RAG-based approaches dynamically generate responses based on retrieved information [4], making them particularly suited for knowledge-intensive tasks such as scientific fact-checking [25, 7]. Despite advancements in SCV, adversarial attacks on NLP models reveal vulnerabilities in automated fact-checking. TextFooler [10] perturbs claims by replacing important words with synonyms based on word embeddings, but this often leads to semantic drift, particularly in scientific contexts. PWWS (Probabilistic Word Replacement Strategy) [18] prioritizes impactful word substitutions using WordNet yet struggles with specialized terminology. BAE (BERT-based Adversarial Examples) [6] introduces token insertions, deletions, and replacements using pre-trained transformers, improving fluency but still failing to preserve domain-specific meaning.

As highlighted in [12], these adversarial methods are inadequate for SCV due to their simplistic word-replacement strategies that fail to maintain the technical integrity of claims. While [12] proposed a paraphrase-based attack generation approach to enhance semantic fidelity, it still requires manual filtering, limiting scalability and application in real-world scenarios. While standard NLP attacks report higher success rates than those in [12], a qualitative analysis reveals that [12] produces a greater number of valid attacks that preserve semantics compared to standard NLP attacks. However, the evolutionary attack algorithm in [12] lacks crossover operations with only mutations, which reduces its overall effectiveness in searching the space of possible attacks.

Our approach advances beyond existing methods by employing a full-stack evolutionary attack model with crossover and diverse mutation techniques implemented through LLMs. The inclusion of the crossover enables better attack generation by combining different adversarial strategies, leading to more effective and automated adversarial testing for SCV tools. The semantic and logical consistency of our proposed attacks is ensured through the use of LLMs and

Table 1: Inconsistent Reasoning Attacks definitions and examples.

Category	**Definition**	**Example**
Generalization	Replace specific terms in the original claim with broader categories. Only considered when the original claim is refuted.	~~White Blood cells~~ **Blood cells** are NOT an important part of your immune system, which are positively correlated with detection and healing.
Vague	Substitute precise terms with more ambiguous language. Only considered when the original claim is supported.	White blood cells are an important part of your immune system, which are ~~positively~~ correlated with detection and healing.
General-to-Specific	Replace general terms with more specific ones.	~~White blood cells~~ **Neutrophils** are an important part of your immune system, which are positively correlated with detection and healing.
Negation	Introduce double negatives or reverse relational terms.	White blood cells are an important part of your immune system, which are ~~positively~~ **not negatively** correlated with detection and healing.
Union	Combine multiple claims into a single statement. Only considered when one claim is supported and the other is refuted.	White blood cells are an important part of your immune system **and neutrophils are not a type of red blood cell**.
Paraphrase	Rephrase the original claim while preserving its meaning.	White blood cells are an important part of your immune system, which are ~~positively correlated with detection and~~ good for healing.

a self-reflection process. Moreover, our experimental results demonstrate a significantly higher attack success rate compared to standard NLP attacks. Consequently, our approach outperforms the method proposed in [12], as our approach achieves a higher attack success rate than standard NLP attacks, whereas [12] does not. Additionally, in Section 5.7, we follow the same qualitative analysis conducted in [12], and our results indicate superior performance.

3 Description of Inconsistent Reasoning Attacks

This section introduces the types of inconsistent reasoning attacks, illustrated in Figure 1. Building on the concept of paraphrasing attacks defined in [12], we generalize to encompass a broader category of adversarial attacks. *An **inconsistent reasoning attack** occurs when an SCV tool assigns a label to an altered claim that is logically inconsistent with the label it assigned to the original claim.* A paraphrasing attack falls under this category since it involves a semantic rewording of the original claim that should not affect its label, yet the SCV tool assigns a different label. This discrepancy reveals an inconsistency in reasoning, as a change in label is unjustified when the meaning remains the same. In machine learning, traditional adversarial attacks introduce small perturbations in

order to alter labels. In contrast, paraphrasing attacks implement perturbations that can be substantial while still maintaining semantic equivalence. Some inconsistent reasoning attacks extend beyond semantic equivalence, increasing the level of perturbation even further. The types of inconsistent reasoning attacks depicted in Figure 1 comprise five attack types: specific-to-general, general-to-specific, negation, union, and paraphrase. These categories are not intended to be exhaustive and we anticipate that this research will inspire further categories of inconsistent reasoning attacks. Table 1 provides definitions and examples for each attack type.

Specific-to-general attacks involve replacing specific terms in the claim with more generic terminology. Generalization attacks, a subset of specific-to-general attacks, replace specific terms in the original claim with broader categories. This approach leverages the hierarchical structure inherent in many scientific claims, where concepts are organized from broader groups into narrower categories. For example, replacing "car" with "vehicle" since vehicle is the broader, higher-level group encompassing "cars" in addition to other vehicle types. Similar to the generalization example in Table 1 where "white blood cells" was replaced with "blood cells." These attacks succeed only when the original claim is refuted, as a refuted claim about a specific subset remains refuted at the broader level, as it is refuted for at least one subset. However, a supported statement for a specific subset may not be supported for all subsets encompassed by the broader level.

Another subset of specific-to-general attacks involves vague generalization, where precise terms are substituted with more ambiguous language. For example, removing the word "positively" from the phrase "positively correlated" in the vague example in Table 1. Unlike generalization attacks, vague generalization is only effective when the original claim is supported, as the broader wording still encompasses the specific case.

In contrast, general-to-specific attacks move in the opposite direction, replacing general terms with more specific ones. For instance, substituting "vehicle" with "car" makes the claim more precise. Similar to the general-to-specific example in Table 1 where "white blood cells" was replaced with "neutrophils", which is a specific type of white blood cell. These attacks are effective when the original claim is supported, as a statement supported at a general level remains valid for its specific subcategories. However, a refuted general claim may not necessarily apply to all its subgroups.

Negation attacks manipulate the logical structure of claims by introducing double negatives or reversing relational terms. Scientific statements often describe relationships between concepts, using paired terms such as increase/decrease, positive/negative, or rise/fall. A negation attack alters these relationships by replacing a term with its opposite and introducing negation, such as changing "increases" to "does not decrease". For example, when the negation attack in Table 1 replaces "positively" with the double negative of "not negatively". Some variants make the claim more ambiguous by removing directional indicators entirely, effectively converting it into a vague attack rather than a strict negation.

Algorithm 1 Inconsistent Reasoning Attacks

Input: SCV model m, LLM llm, Iterations R, Population size N_{pop}, Dataset DB
function generateAttack($claim, label$)
  $Pop_{list} \leftarrow \{(claim, label)\}$
  $mutationPrompts = [paraphrase, union, negation, generalToSpecific, generalization, vague]$
  $crossOverPrompt = crossOver$
  Filter $mutationPrompts$ based on the $label$'s value.
  for $iter = 1$ to R **do**
    $mutatedClaims \leftarrow mutatePop(mutationPrompts, Pop_{list})$
    $crossedClaims \leftarrow crossOverPop(crossOverPrompt, Pop_{list})$
    Add $mutatedClaims$ and $crossedClaims$ to Pop_{list}
    $Pop_{list} \leftarrow selection(Pop_{list})$
  end for
end function
function mutatePop($mutationPrompts, Pop_{list}$)
  $mutatedClaims \leftarrow \{\}$
  for $pop = 1$ to N_{pop} **do**
    $mutationPrompt \leftarrow$ Random selection from $mutationPrompts$
    $(claim, label) \leftarrow$ Random selection from Pop_{list}
    if $prompt = union$ **then**
      Extend $claim$ by a neighbor claim from the DB
    end if
    Pre-process $mutationPrompt$ using $claim$
    $invokeLLM(mutatedClaims, mutationPrompt, label)$
  end for
  Return $mutatedClaims$
end function
function crossOverPop($crossOverPrompt, Pop_{list}$)
  $crossedClaims \leftarrow \{\}$
  for $pop = 1$ to N_{pop} **do**
    $(claim1, label1) \leftarrow$ Random selection from Pop_{list}
    $(claim2, label2) \leftarrow$ Random selection from Pop_{list}
    Pre-process $crossOverPrompt$ using $claim1$ and $claim2$
    $invokeLLM(crossedClaims, crossOverPrompt, label)$
  end for
  Return $crossedClaims$
end function
function invokeLLM($claimList, prompt, label$)
  $newClaim \leftarrow llm(prompt)$
  $newLabel \leftarrow$m.$predict(newClaim)$
  if $newLabel \neq label$ **then**
    `attackSuccess` $\leftarrow$ `True`
    Exit function.
  end if
  Add $(newClaim, newLabel)$ to $claimList$
end function
function selection(Pop_{list})
  Score Pop_{list} using m.score(Pop_{list})
  Sort Pop_{list} by the scores (descending)
  Return $Pop_{list} \leftarrow Pop_{list}[: N_{pop}]$
end function

Another common transformation involves adding phrases like "does not fail to", which modifies the logical interpretation of the statement.

Union attacks merge a refuted original claim with a supported claim (retrieved from the training set) into a single statement. The resulting union claim should still be classified as refuted. These attacks are considered successful only if the final claim is misclassified as supported. The union of two supported claims resulting in a refuted classification would be evidence of a hallucination (likely resulting from linking two supported claims in an improper way), which would be considered an unsuccessful attack. Combining two refuted statements such that an SCV tool would classify the union as supported would be a difficult task and was therefore not considered as it would result in minimal, if any, successful attacks. An example of a union attack can be found in Table 1, where the supported claim "White blood cells are an important part of your immune system" is combined with the refuted claim "Neutrophils are not a type of red blood cell".

Paraphrase attacks rephrase a claim while preserving its meaning. For example, in Table 1, the claim "White blood cells are an important part of your immune system, which are positively correlated with detection and good for healing" is paraphrased by removing "...positively correlated with detection and...". While negation and paraphrasing attacks retain semantic equivalence, double negation can be seen as a form of paraphrasing attack. However, we classify negation separately due to its distinct nature [12].

4 Inconsistent Reasoning Attacks Generator

Inspired by genetic algorithms, the Inconsistent Reasoning Attacks (*IRA*) generator iteratively challenges the Scientific Claim Verification (SCV) target model, m, by generating adversarial claims. The attack operates over multiple **global iterations**, denoted as R, during which a population of claims undergoes systematic perturbations. Each iteration consists of two primary operations: **mutation** and **crossover**. These operations leverage an LLM to generate diverse claim variations that can mislead m. To ensure meaningful perturbations, a **self-reflection mechanism** refines the generated claims, evaluating their alignment with attack objectives. The generator continues iteratively until a perturbed claim successfully alters m's classification or the maximum number of iterations is reached. Algorithm 1 provides a step-by-step breakdown of this process. All prompts used in this paper are provided in Prompts.pdf (additional material).

4.1 Mutation Operation

The **mutation operation** introduces localized changes to an input claim, altering its structure or semantics while retaining its contextual essence. This step leverages a predefined set of mutation strategies, including paraphrasing, generalization, negation, and others, to generate adversarial claims. **Mutation**

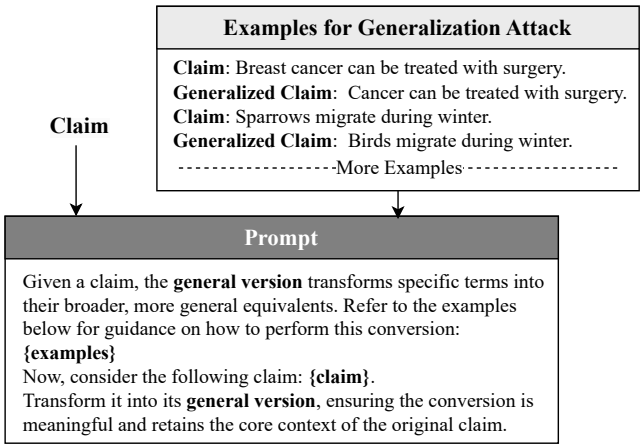


Fig. 2: Prompt design for generalization attacks.

Prompts: A predefined list of mutation strategies guides the LLM in generating new claims. Simple attacks such as paraphrasing require minimal modifications, while more complex mutations integrate additional contextual information through supplementary examples in the LLM prompt. Figure 2 illustrates an example prompt design for generalization attacks.
Process: A mutation strategy is randomly applied to a claim from the population. The LLM generates a new claim, which is evaluated against m. If the model's prediction changes, the attack is successful. This iterative process refines adversarial claims through self-reflection to maximize their impact.
Self-Reflection of LLM: To enhance adversarial claim generation, Inconsistent Reasoning Attacks employs a **self-reflection mechanism**, inspired by its effectiveness in problem-solving [19, 20] and reducing hallucinations [9]. The LLM-Reflector evaluates each mutated claim, refining it iteratively until it reaches a meaningful deviation or a maximum number of iterations, R_{sr}. Figure 3 illustrates this workflow, where the LLM-Generator perturbs claims and the LLM-Reflector evaluates and refines them.

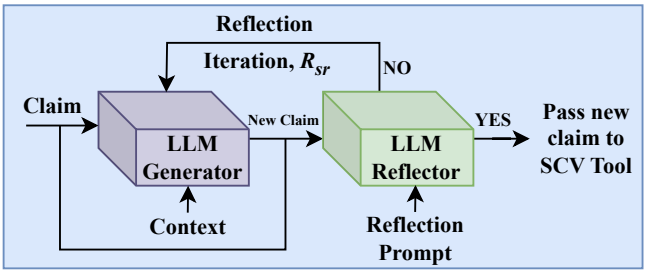


Fig. 3: Self-reflection of LLM.

4.2 Crossover Operation

The **crossover operation** merges two claims to create a hybrid adversarial claim, introducing novel perturbations that increase the likelihood of altering m's classification. The crossover mechanism is structured as follows:
Crossover Prompt A specialized prompt directs the LLM to combine semantic components from two distinct claims into a single hybrid claim.
Process Two claims are selected from the population, and the LLM synthesizes a new claim by integrating relevant aspects of both. The generated claim is then tested against m, and if its prediction is altered, the attack is deemed successful.

Like mutation, the crossover process benefits from **self-reflection**, refining the hybrid claim through iterative assessment and modification. The crossover's impact is further strengthened by the population selection mechanism, ensuring only the most effective claims persist across iterations.

4.3 Population Selection

The **Population Selection** mechanism manages the evolving set of adversarial claims throughout the attack process. It maintains an optimal population of claims by evaluating and ranking them based on their success in misleading m.

Each claim undergoes evaluation based on m's classification confidence and perturbation effectiveness. The claims are ranked in descending order of their impact, with only the top N_{pop} claims retained for subsequent iterations (Algorithm 1, lines 46-48). An attack iteration is considered successful if at least one perturbed claim generates a different label from m compared to the original claim. This signifies that the adversarial modification effectively misled the classification process. At each iteration, newly generated claims are evaluated and ranked, ensuring that only the most potent adversarial examples persist. The attack continues until either a successful perturbation is achieved or the maximum global iterations, R, are reached.

Through this iterative process of mutation, crossover, self-reflection, and population selection, Inconsistent Reasoning Attacks systematically generates robust adversarial claims that challenge m's decision boundary, demonstrating the efficacy of genetic-inspired attack mechanisms in adversarial NLP research.

5 Experiment

This section evaluates *IRA* against three state-of-the-art NLP attacks on two datasets and three victim models. We detail the datasets, baselines, victim models, implementation, evaluation metrics, analyses, ablation studies, and hyperparameters.

5.1 Datasets

We use two well-known datasets for scientific claim verification, summarized in Table 2:

Table 2: Dataset Splits with Class Distribution

Dataset	Train		Val		Test	
	Support	Refute	Support	Refute	Support	Refute
SciFact	332	173	124	64	100	100
HealthVer	3782	2411	533	391	671	425

SciFact[25] is a curated dataset of scientific claims annotated with abstracts. Claims are labeled as SUPPORT, REFUTE, or NOINFO. For SUPPORT and REFUTE labels, rationales and key statistics, such as the number of sentences per rationale and evidence abstracts per claim, are provided.
HealthVer[17] is a COVID-19-focused dataset for fact-checking health claims. It contains manually annotated claim-evidence pairs extracted from web snippets and verified against scientific literature. Each pair is categorized as SUPPORT, REFUTE, or NEUTRAL.

5.2 NLP Attack Methods and Victim Models

We compare *IRA* against the following three attack methods.[1]:
PWWS [18] uses WordNet[2] to generate synonym-based substitutions and prioritizes word replacements based on model probability shifts and word significance.
TextFooler [10] ranks word importance by measuring the cumulative probability change before and after its removal.
BAE-Attack [6] is a black-box attack that generates adversarial examples by applying contextual perturbations using a BERT-masked language model. BAE-Attack modifies text by masking tokens and replacing or inserting alternatives suggested by BERT-MLM.

For adversarial attacks, we evaluate the following SCV tools as victim models:
MultiVerS uses a Longformer encoder for claim verification at both abstract and sentence levels [26]. It selects rationales via three-way classification, discarding Not Enough Information (NEI) labels and retaining SUPPORT and REFUTE. Abstract retrieval involves gathering candidates and refining predictions with a neural re-ranking mechanism.
Retrieval-Augmented Generation (RAG) operates by indexing the entire corpus of abstracts [13]. For each fact-checking instance, the search query embedding is matched against the indexed corpus. The top-K documents retrieved are then provided, along with the query, to the LLM for final classification.
Attack-Reflection-RAG extends RAG by integrating additional information about potential attacks, thereby improving its adversarial robustness. This approach first processes the claim and queries the LLM for possible perturbations

[1] As explained in Section 2, we do not include a comparison with [12], as it underperforms based on the metrics defined in Section 5.3 for the considered NLP attacks.
[2] https://wordnet.princeton.edu/

introduced by an attack. If the LLM detects a potential adversarial modification, it supplies this information as additional context to the RAG model during inference, helping it make more informed predictions. The prompts used and further details are in Prompts.pdf in the additional material.

5.3 Metrics

To assess the effectiveness of *IRA* against the victim models, we use three metrics: **Clean Accuracy (Clean%)** represents the classification accuracy on the clean test dataset. This metric evaluates the performance of the victim model when it is not exposed to any adversarial attacks. A higher Clean% indicates that the model generalizes well to unseen data and can accurately classify clean instances. **Accuracy Under Attack (Aua%)** measures the model's accuracy in adversarial settings [14, 15, 28]. A higher Aua% indicates greater resilience to adversarial attacks, making it a crucial metric in adversarial robustness research.
Attack Success Rate (Suc%) represents the proportion of successfully altered texts out of the total attempted examples [14, 15, 28]. A lower Suc% indicates greater model robustness.

5.4 Implementation Settings

We generate the adversarial examples using NVIDIA GeForce RTX 4090. We employ the LLM LLaMA-3.1, accessed via the open-source library Ollama [3], for both retrieval-augmented generation (RAG) and adversarial generation. The clean accuracy (Clean%) is calculated across the entire test dataset. We reproduce all victim models using their respective open-source implementations and predefined hyperparameters. Our GitHub repository for reproducibility is available at [4].

5.5 Main Results

Table 3 presents the experimental results of *IRA* compared to state-of-the-art attacks PWWS, TextFooler, and BAE-Attack on MultiVerS and RAG-based SCV tools across two datasets, SciFact and HealthVer. Since the test set for SciFact is private [24], we report results on its development set, while for HealthVer, we use the test set. Across all models and datasets, *IRA* consistently achieves the highest Suc%, demonstrating its superior ability to degrade model performance. Specifically, for MultiVerS, *IRA* reduces Aua% to 29.26 on SciFact and 22.52 on HealthVer, surpassing PWWS, TextFooler, and BAE-Attack while maintaining a significantly higher Suc% (67.65 and 69.85). Similar trends hold for RAG, where *IRA* lowers accuracy to 18.09 on SciFact and 15.38 on HealthVer, with an Suc% of 77.18 and 78.63, respectively. For the Attack-Reflection-RAG model, *IRA* continues to demonstrate effectiveness. On SciFact, *IRA* reduces Aua% to 28.19

[3] https://ollama.com/
[4] https://github.com/atikbappy/inconsistency-reasoning

Table 3: Experimental results of *IRA* compared to three state-of-the-art textual attacks against MultiVerS and RAG-based SCV tools on two datasets, SciFact and HealthVer. The best performance is highlighted in bold.

Model	**Dataset**	**Clean%**	**PWWS**		**TextFooler**		**BAE-Attack**		***IRA***	
			Aua%	**Suc%**	**Aua%**	**Suc%**	**Aua%**	**Suc%**	**Aua%**	**Suc%**
MultiVerS	SciFact	90.43	55.85	38.24	58.51	35.29	82.45	8.82	**29.26**	**67.65**
	HealthVer	74.73	32.41	56.61	35.16	52.94	67.58	9.56	**22.52**	**69.85**
RAG	SciFact	79.26	30.85	61.07	33.51	57.72	69.15	12.75	**18.09**	**77.18**
	HealthVer	71.98	24.18	66.41	26.37	63.36	59.89	16.69	**15.38**	**78.63**
Attack-Reflection-RAG	SciFact	76.06	40.43	46.85	43.09	43.35	70.21	7.69	**28.19**	**62.94**
	HealthVer	70.33	37.91	46.09	42.86	39.06	62.63	10.93	**27.47**	**60.94**

with an Suc% of 62.94. On HealthVer, it lowers Aua% to 27.47 with an Suc% of 60.94, outperforming baselines. Notably, on HealthVer, where baseline attacks have success rates below 47%, *IRA* is the only method with a dominant impact. Overall, *IRA* improves Suc% by +35.17 on MultiVerS, +31.58 on RAG, and +29.61 on Attack-Reflection-RAG, confirming its superiority over prior attacks in degrading scientific claim verification models.

Attack-Reflection-RAG is highly resilient to attacks, except for MultiVerS outperforming it on SciFact—likely due to parameter estimation on the same test set. This anomaly aside, attack reflection warrants further investigation as a mitigation strategy.

5.6 Ablation Study

We perform an ablation study to assess the impact of individual attacks on *IRA*. Two experiments were conducted: excluding a specific attack to evaluate overall performance and isolating a single attack to analyze its contribution. Table 4

Table 4: Ablation study results showing both Aua% and Suc% for SciFact and HealthVer. "Excluded" means the attack was excluded, "Used" means only that attack was applied. The row "All" corresponds to the overall performance when all attacks are applied (or equivalently, when no attack is excluded).

Attack Type	**SciFact**				**HealthVer**			
	Excluded		**Used**		**Excluded**		**Used**	
	Aua%	**Suc%**	**Aua%**	**Suc%**	**Aua%**	**Suc%**	**Aua%**	**Suc%**
Paraphrase	55.32	38.82	53.72	41.17	40.11	46.33	40.11	46.32
Negation	37.77	58.23	78.19	14.11	26.37	64.71	67.58	10.29
General-to-Specific	29.79	67.06	87.23	3.52	29.67	60.29	60.44	19.11
Generalization	33.51	62.94	83.51	8.23	28.02	62.50	66.48	11.03
Vague	33.51	62.94	88.30	2.35	25.82	65.44	66.68	8.08
Union	41.49	54.12	83.51	7.64	31.32	58.09	57.69	22.79
All	**29.26**	**67.65**	**29.26**	**67.65**	**22.52**	**69.85**	**22.52**	**69.85**

presents the excluded and isolated attacks, along with their attack success rate (Suc%) and accuracy under attack (Aua%) for SciFact and HealthVer.

IRA is most effective when all attacks are combined, achieving the lowest Aua% (29.26% for SciFact, 22.52% for HealthVer) and the highest Suc% (67.65% and 69.85%, respectively), as shown in Table 4. This underscores the strength of a diversified attack strategy.

Among individual attacks, excluding *Paraphrase* causes the largest Aua% increase (around 26% for SciFact,18% for HealthVer), emphasizing its crucial role. Its standalone effectiveness is further confirmed by achieving the highest Aua% and Suc%.

5.7 Qualitative Analysis

For the qualitative analysis, we analyze 54 paraphrasing attacks that had been deemed successful by the model and show an example of a single attack that combined multiple inconsistent reasoning attacks in one.

Table 5: Results of the qualitative analysis to show the success of attacks and evaluate quality.

Quality Metric	**Count (%)**
Rejected Attacks	12 (22.2%)
Successful Attacks	42 (77.8%)
Quality of Successful Attacks	
High	28 (66.7%)
Medium	13 (31.0%)
Low	1 (2.4%)

Among the 54 paraphrasing attacks that were reviewed, we identified any successful attacks that were truly unsuccessful, that is, an attack where the logic of the original claim had not been maintained, and the classification from the SCV tool was correct in reversing its original label.

Attacks that were truly successful were evaluated, and the quality of the attack was ranked: high/medium/low. Table 5 presents the results of the qualitative analysis. Most of the attacks were truly successful (77.8%), and all but one was of high or medium quality.

Of the 54 paraphrasing attacks that were reviewed, 12 (22.2%) were rejected and deemed unsuccessful. The attacks were rejected for one of three reasons: 1) changed meaning, 2) hallucination, or 3) nonsensical statement.
An attack "changed meaning" if the meaning from the original claim had been altered or reversed in the attack. A "hallucination" occurred when the attack added new information that was contrary to the original claim. A "nonsensical statement" was an attack that contained nonsensical grammar, facts, or science. Of the 12 rejections, 7 were hallucinations (58.3%), 4 changed meaning (33.3%),

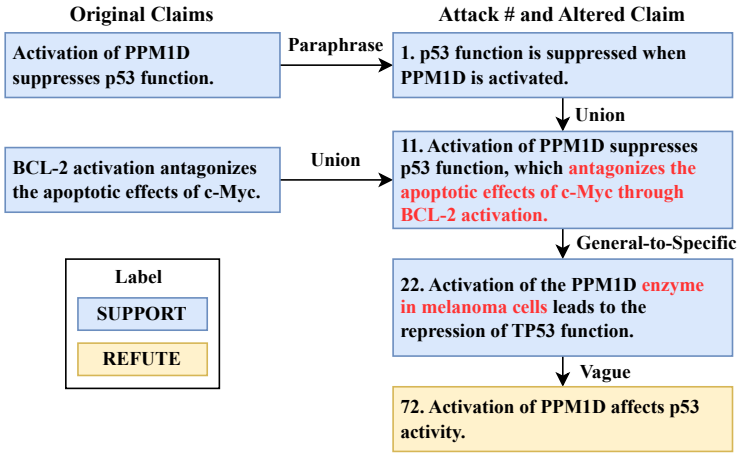


Fig. 4: Example of a generated combined inconsistent reasoning attacks

and 1 was a nonsensical statement (8.3%). Hallucinations were expected due to the greedy nature of inconsistent reasoning attacks, which optimize for modification of an original claim that reverses the classification of the SCV tool.

The single attack that combined multiple inconsistent reasoning attacks that were reviewed went through 72 iterations, applying an inconsistent reasoning attack at each iteration. Figure 4 displays 4 of the 72 intermediate states (the 1st, 11th, 22nd, and 72nd states).

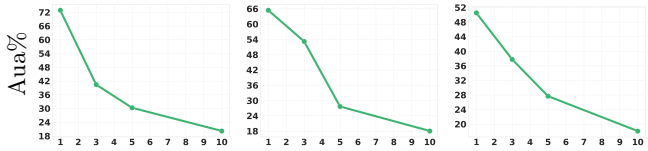
These 4 intermediate states include paraphrasing, union, general-to-specific, and vague inconsistent reasoning attacks. When possible, added or updated text is colored red. This figure shows how two original claims were joined (union), rephrased (paraphrase), made more specific (general-to-specific), and made more vague in order to form an attack claim that reversed the classification of the SCV tool from SUPPORT to REFUTE making this a successful attack.

5.8 Hyperparameter Analysis

Figure 5 illustrates the impact of key hyperparameters - Number of iterations (R), Number of Population (N_{pop}) and Number of Self-Reflection Iterations (R_{sr}) - on the performance of IRA, as measured by Aua%

Number of Iterations Increasing the number of iterations reduces the Aua% consistently, as shown in subfigure 5a. This shows that longer iterative processes allow the algorithm to explore more potential attack strategies, resulting in better performance. However, balancing performance improvement with the increased computational costs is crucial when choosing R.

Number of Population Subfigure 5b demonstrates that a larger population contributes to a steady decline in Aua%, indicating that a diverse candidate pool during optimization improves the overall outcome of IRA. While higher N_{pop}



(a) R (b) N_{pop} (c) R_{sr}

Fig. 5: Hyperparameter Analysis for *IRA*. 50 random samples from SciFact dataset were used to compute Aua% for given hyperparameters.

values lead to better results, they demand additional computational resources and can slow down the process.

Number of Self Reflection Iterations Subfigure 5c highlights a negative correlation between R_{sr} and Aua%. With more self-reflection iterations, LLMs can increasingly refine its strategies, leading to more effective attacks. Nonetheless, higher R_{sr} requires higher computational resources, so choosing a good trade-off is necessary.

6 Conclusion

Scientific Claim Verification (SCV) tools play a crucial role in assessing the validity of scientific claims; however, they remain vulnerable to logical inconsistencies. Adversarial attacks provide a powerful approach for identifying these vulnerabilities and enhancing SCV systems. In this study, we introduce inconsistent reasoning attacks, a broader class of adversarial manipulations encompassing five types: specific-to-general, general-to-specific, negation, union, and paraphrase. To detect and test these weaknesses, we develop an evolutionary algorithm that leverages large language models to generate effective adversarial attacks targeting logical inconsistencies.

Evaluation against MultiVerS reveals persistent vulnerabilities, while a retrieval-augmented generation (RAG) system with an Attack-Reflection mechanism shows promise in mitigating these issues. Strengthening SCV systems against such attacks is essential for ensuring their reliability. Future research should integrate self-reflection and adversarial robustness to improve SCV effectiveness.

References

1. Beltagy, I., Peters, M.E., Cohan, A.: Longformer: The long-document transformer (2020)
2. Coley, C.W., Eyke, N.S., Jensen, K.F.: Autonomous discovery in the chemical sciences part i: Progress. Angewandte Chemie International Edition **59**(51), 22858–22893 (2020)
3. Dey, A.U., Llabrés, A., Valveny, E., Karatzas, D.: Retrieval augmented verification: Unveiling disinformation with structured representations for zero-shot fact-checking of multi-modal social media posts. arXiv preprint arXiv:2404.10702 (2024)

4. Dmonte, A., Oruche, R., Zampieri, M., Calyam, P., Augenstein, I.: Claim verification in the age of large language models: A survey. arXiv preprint arXiv:2408.14317 (2024)
5. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., Wang, H.: Retrieval-augmented generation for large language models: A survey (2024)
6. Garg, S., Ramakrishnan, G.: BAE: BERT-based adversarial examples for text classification. In: Proc. of EMNLP 2020. pp. 6174–6181. ACL (Nov 2020)
7. Guan, J., Dodge, J., Wadden, D., Huang, M., Peng, H.: Language models hallucinate, but may excel at fact verification. arXiv preprint arXiv:2310.14564 (2023)
8. Huang, L., Yu, W., Ma, W., Zhong, W., et al.: A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. ACM Trans. Inf. Syst. (Nov 2024)
9. Ji, Z., Yu, T., Xu, Y., Lee, N., Ishii, E., Fung, P.: Towards mitigating LLM hallucination via self reflection. In: Findings of ACL: EMNLP 2023. pp. 1827–1843. ACL, Singapore (2023)
10. Jin, D., Jin, Z., Zhou, J.T., Szolovits, P.: Is BERT really robust? a strong baseline for nlp attack on text classification and entailment. In: Proc. AAAI. vol. 34, pp. 8018–8025 (2020)
11. Khaliq, M.A., Chang, P., Ma, M., Pflugfelder, B., Miletić, F.: Ragar, your falsehood radar: Rag-augmented reasoning for political fact-checking using multimodal llms. arXiv preprint arXiv:2404.12065 (2024)
12. Layne, J., Ratul, Q.E.A., Serra, E., Jajodia, S.: Analyzing robustness of automatic scientific claim verification tools against adversarial rephrasing attacks. ACM Trans. Intell. Syst. Technol. **15**(5) (Nov 2024)
13. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. Adv. Neural Inf. Process. Syst. **33**, 9459–9474 (2020)
14. Li, Z., Xu, J., Zeng, J., Li, L., Zheng, X., Zhang, Q., Chang, K.W., Hsieh, C.J.: Searching for an effective defender: Benchmarking defense against adversarial word substitution. In: Proc. 2021 Conf. Empirical Methods in NLP. pp. 3137–3147. ACL, Online and Punta Cana, Dominican Republic (2021)
15. Liu, Q., Zheng, R., Rong, B., Liu, J., Liu, Z., Cheng, Z., Qiao, L., Gui, T., Zhang, Q., Huang, X.: Flooding-X: Improving BERT's resistance to adversarial attacks via loss-restricted fine-tuning. In: Proc. 60th Ann. Meet. Assoc. Comput. Linguistics (Vol. 1: Long Papers). pp. 5634–5644. ACL, Dublin, Ireland (2022)
16. Maffettone, P.M., Friederich, P., Baird, S.G., Blaiszik, B., Brown, K.A., Campbell, S.I., Cohen, O.A., Davis, R.L., Foster, I.T., Haghmoradi, N., et al.: What is missing in autonomous discovery: open challenges for the community. Digital Discovery **2**(6), 1644–1659 (2023)
17. Mourad Sarrouti, Asma Ben Abacha, Y.M., Demner-Fushman, D.: Evidence-based fact-checking of health-related claims. In: EMNLP (2021)
18. Ren, S., Deng, Y., He, K., Che, W.: Generating natural language adversarial examples through probability weighted word saliency. In: Proceedings of the 57th annual meeting of the association for computational linguistics. pp. 1085–1097 (2019)
19. Renze, M., Guven, E.: Self-reflection in llm agents: Effects on problem-solving performance (2024)
20. Salehi, S.: Improving problem-solving through reflection. Stanford University (2018)

21. Sarrouti, M., Abacha, A.B., M'rabet, Y., Demner-Fushman, D.: Evidence-based fact-checking of health-related claims. In: Findings of the Association for Computational Linguistics: EMNLP 2021. pp. 3499–3512 (2021)
22. Singhal, R., Patwa, P., Patwa, P., Chadha, A., Das, A.: Evidence-backed fact checking using rag and few-shot in-context learning with llms. arXiv preprint arXiv:2408.12060 (2024)
23. Tan, N.Ö., Tandon, N., Wadden, D., Tafjord, O., Gahegan, M., Witbrock, M.: Faithful reasoning over scientific claims. In: Proceedings of the AAAI Symposium Series. vol. 3, pp. 263–272 (2024)
24. Wadden, D.: The multivers model: Source code. https://github.com/dwadden/multivers (2023), code associated with the MultiVerS paper. Accessed: 2025-02-10
25. Wadden, D., Lin, S., Lo, K., Wang, L.L., van Zuylen, M., Cohan, A., Hajishirzi, H.: Fact or fiction: Verifying scientific claims. In: Proc. of EMNLP 2020. pp. 7534–7550. ACL, Online (2020)
26. Wadden, D., Lo, K., Wang, L.L., Cohan, A., Beltagy, I., Hajishirzi, H.: MultiVerS: Improving scientific claim verification with weak supervision and full-document context. In: Findings of ACL: NAACL 2022. pp. 61–76. ACL, Seattle, United States (2022)
27. Wang, L.: Using machine learning to verify scientific claims. Artificial Intelligence in Science p. 120 (2023)
28. Wang, Z., Liu, Z., Zheng, X., Su, Q., Wang, J.: RMLM: A flexible defense framework for mitigating word-level adversarial attacks. In: Proc. of the 61st Annual Meeting of ACL (Vol. 1: Long Papers). pp. 2757–2774. ACL, Toronto, Canada (2023)