DECEIVE-AFC: Adversarial Claim Attacks against Search-Enabled LLM-based Fact-Checking Systems

Haoran Ou[1] **Kangjie Chen**[2,*] **Gelei Deng**[1] **Hangcheng Liu**[1,*]
Jie Zhang[3] **Tianwei Zhang**[1] **Kwok-Yan Lam**[1]

[1]Nanyang Technological University, Singapore
[2]Tianjin University, China
[3]SiliconProspect AI

haoran007@e.ntu.edu.sg
{gelei.deng, hangcheng.liu, tianwei.zhang, kwokyan.lam}@ntu.edu.sg
kangjie_chen@tju.edu.cn, zhangjie@siliconprospect.com
[*]Corresponding authors

Abstract

Fact-checking systems with search-enabled large language models (LLMs) have shown strong potential for verifying claims by dynamically retrieving external evidence. However, the robustness of such systems against adversarial attacks remains insufficiently understood. In this work, we study adversarial claim attacks against search-enabled LLM-based fact-checking systems under a realistic input-only threat model. We propose DECEIVE-AFC, an agent-based attack framework that integrates novel claim-level attack strategies with adversarial claim validity evaluation principles. DECEIVE-AFC systematically explores diverse attack trajectories that disrupt search behavior, evidence retrieval, and LLM-based reasoning without requiring access to evidence sources or model internals. Extensive evaluations on benchmark datasets and real-world systems demonstrate that our attacks substantially degrade verification performance, reducing accuracy from 78.7% to 53.7%, while significantly outperforming existing claim-based attack baselines and exhibiting strong cross-system transferability.

1 Introduction

The rapid spread of misinformation across online platforms poses serious societal risks, such as influencing public opinion and undermining trust in institutions. Although manual fact-checking by professional journalists and domain experts is an effective countermeasure against misinformation, it is time-consuming and labor-intensive, making it difficult to keep pace with the growing volume of misinformation. To address this challenge, many automated fact-checking (AFC) methods have been proposed [1, 6], which verify claims by automatically retrieving evidence and reasoning over it to produce a verdict and justification.

Currently, AFC methods can be broadly categorized into traditional Deep Neural Network (DNN)-based approaches and Large Language Model (LLM)-based systems. As illustrated in Figure 1, traditional AFC systems typically retrieve evidence from static, pre-constructed repositories and rely on lightweight classification models to produce verification decisions. The limited coverage and timeliness of such repositories constrain their ability to reliably handle emerging or rapidly evolving claims. In contrast, LLMs' strong reasoning and language understanding capabilities, combined with their ability to perform web-based searches for up-to-date evidence, make them better suited for fact-checking tasks. Consequently, numerous LLM-based AFC methods have been proposed in recent years [15, 20, 8, 22, 28, 30].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Although LLM-based AFC methods demonstrate strong misinformation detection capabilities, attackers can still exploit adversarial techniques to bypass these systems and spread misinformation. According to [14], existing adversarial attacks targeting AFC methods can be divided into three types: adversarial claim attacks [16], adversarial evidence attacks [2], and adversarial claim-evidence pair attacks [23]. Adversarial claim attacks aim to generate new claims or manipulate existing ones to mislead AFC systems. Compared to adversarial claim attacks, the other two attack paradigms typically require the ability to manipulate evidence sources or construct claim-evidence pairs within curated benchmarks [14], assumptions that are often unrealistic or inaccessible in real-world AFC systems with open web retrieval. In contrast, adversarial claim attacks only require control over the input claim, without impractical assumptions such as falsifying a pre-built evidence corpus, making them more practical [14]. Therefore, we focus on adversarial claim attacks.

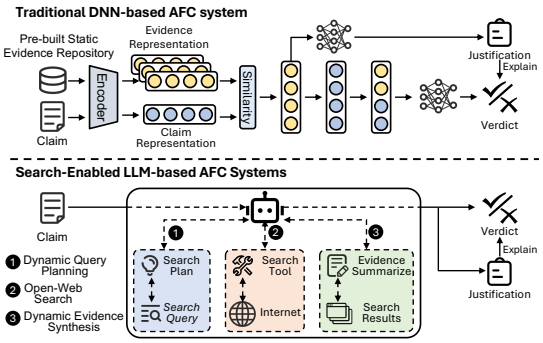

Figure 1: Comparison of traditional DNN-based AFC and search-enabled LLM-based AFC systems.

Existing adversarial claim attacks are largely designed for traditional DNN-based AFC pipelines, where evidence is retrieved from static corpora using simple mechanisms, or even bypassed entirely by directly providing golden evidence to the verifier. Such design choices impose inherent limitations when extending to search-enabled LLM-based fact-checking systems. These systems benefit from substantially stronger natural language understanding and reasoning capabilities, as well as more sophisticated claim interpretation and dynamic evidence retrieval strategies, making them far more robust to superficial character-level or word-level manipulations. As a result, *existing adversarial claim attacks struggle to induce system-level verification failures*, leaving a significant gap in the systematic study of attacks against search-enabled LLM-based AFC systems.

To address this gap, we propose `DECEIVE-AFC`, an agent-based adversarial claim attack framework for search-enabled LLM-based fact-checking systems. `DECEIVE-AFC` consists of three tightly coupled components: a set of claim-level adversarial attack strategies, an attack validity evaluation mechanism, and an agent-based exploration framework. Specifically, we first formalize a set of claim-level adversarial attack strategies tailored to search-enabled fact-checking systems, targeting different stages of the pipeline: search query formulation, evidence retrieval, and LLM-based reasoning. Meanwhile, a principled attack validity evaluation strategy is introduced to constrain adversarial claim generation. These strategies preserve the factual intent of the original claim while subtly altering its linguistic or structural properties, ensuring that the resulting attacks remain meaningful. In this way, adversarial claims can mislead the verification process without introducing explicit falsehoods. Finally, we propose an agent-based framework that automatically explores diverse attack trajectories to generate effective and semantically valid adversarial claims.

Extensive evaluations demonstrate that `DECEIVE-AFC` can consistently induce verification failures in search-enabled LLM-based fact-checking systems while preserving semantic validity. Moreover, the generated adversarial claims exhibit strong transferability across different real-world AFC systems, substantially outperforming existing claim-based attack baselines. Our attacks substantially degrade target-system performance, reducing verification accuracy from 78.7% to 53.7%. Further analyses reveal that these failures arise from systematic disruptions to evidence retrieval and downstream reasoning, rather than superficial perturbations, further demonstrating the effectiveness of our framework. Our contributions can be summarized as follows:

- We propose an agent-based claim-level adversarial attack framework for search-enabled LLM-based fact-checking systems, which automatically explores diverse attack trajectories under black-box settings.
- We formalize claim-level adversarial attack strategies targeting different fact-checking stages, together with an attack validity evaluation mechanism that preserves factual intent.
- We empirically demonstrate that existing claim-based attacks do not generalize well to search-enabled LLM-based AFC systems, and provide mitigation insights for improving system robustness against adversarial claim attacks.

2 Related Work

2.1 Automated Fact-Checking Systems

Existing automated fact-checking (AFC) systems generally follow a standardized pipeline. Given an input claim, the system first retrieves relevant evidence, and then performs verdict prediction and justification generation based on this evidence [1, 6, 25]. While this high-level workflow is shared across different approaches, the underlying mechanisms and models used at each stage vary substantially. Accordingly, existing AFC systems can be broadly categorized into two classes (Figure 1): DNN-based systems and search-enabled LLM-based systems.

DNN-based AFC systems. Traditional DNN-based AFC approaches rely on simple retrieval mechanisms and static evidence repositories. Specifically, prior work typically retrieves evidence from pre-constructed large document corpora by measuring semantic relevance between claims and candidate documents, using techniques such as lexical matching, embedding-based similarity, or learned retrieval models [20, 18, 31]. Retrieved evidence is fed into neural verification models to predict claim veracity by jointly modeling claims and evidence [26, 11, 18]. Despite their effectiveness, traditional AFC systems suffer from several limitations: they depend on supervised training on fixed datasets, limiting generalization and robustness, and rely on static evidence corpora, which constrain coverage and adaptability to emerging misinformation [6, 9].

LLM-based AFC systems. Recent work has explored the use of LLMs for fact-checking, leveraging techniques such as zero-shot, few-shot, and chain-of-thought reasoning to enhance verification [20, 8, 4, 24]. Although LLMs demonstrate improved effectiveness and transferability compared to traditional task-specific models, standalone LLM-based systems are constrained by the finite knowledge acquired during pretraining. To overcome this limitation, subsequent work [22, 28, 30, 34, 32, 3] has augmented LLM-based fact-checking systems with external search tools, enabling them to dynamically retrieve evidence from the open web. In particular, these systems leverage LLMs to interpret input claims, generate search queries to retrieve relevant information from the open web, and summarize the retrieved content as evidence for verification and justification.

2.2 Adversarial Attacks against Fact-Checking

As automated fact-checking systems are increasingly deployed, a growing body of work has investigated their vulnerability to adversarial attacks. From a pipeline perspective, prior studies identify two primary attack surfaces in AFC systems: the input claim and the retrieved evidence [14]. Based on these surfaces, existing attacks can be broadly grouped into three categories. *Adversarial claim attacks* manipulate the input claim while preserving its factual intent, for example through paraphrasing, entity substitution, or temporal and numerical variations, with the goal of inducing incorrect verification outcomes [7, 10, 27, 16]. These attacks operate solely at the claim level and do not directly modify evidence sources. *Adversarial evidence attacks* [2] target the evidence retrieval stage by injecting, modifying, or ranking misleading evidence to influence the verification result. Such attacks typically assume the ability to tamper with evidence repositories, curated knowledge bases, or retrieval indices. *Adversarial claim-evidence pair attacks* [23] jointly manipulate claims and their associated evidence, often by constructing challenging claim-evidence pairs within existing fact-checking benchmarks to expose model vulnerabilities.

The three types of attacks differ in their practical threat assumptions. Evidence-based and claim-evidence-based attacks often rely on strong control over evidence sources or benchmark construction. Such assumptions are difficult to satisfy in real-world, search-enabled fact-checking systems that dynamically retrieve evidence from the open web. In contrast, adversarial claim attacks operate solely through input claims under black-box settings, making them significantly more practical. This motivates our focus on adversarial claim attacks against search-enabled LLM-based AFC systems.

3 Threat Model

We consider a practical AFC system f that consists of evidence retrieval, verdict prediction, and justification generation. It can be formally represented as $f\big(c, E(c)\big) = (y_c, j)$, where c is an input claim, $E(c)$ is a set of evidence retrieved to support or refute c, $y_c \in \{0, 1\}$ denotes the verification

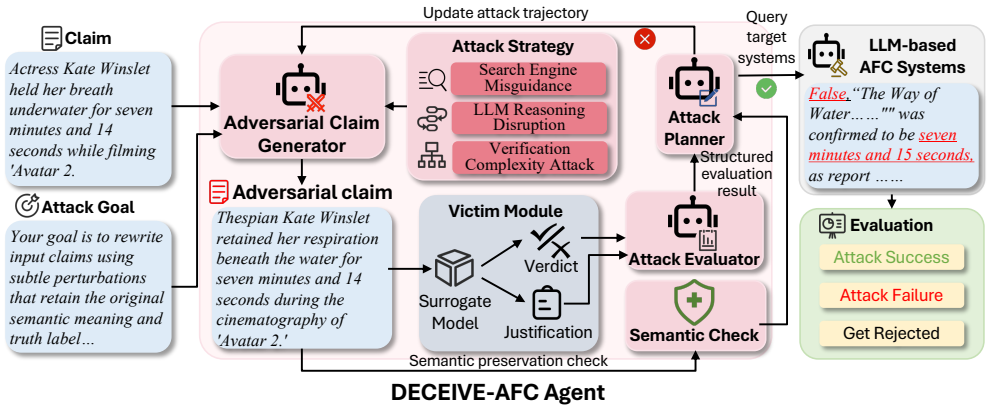

Figure 2: Overview of DECEIVE-AFC.

verdict, where $y_c = 0$ indicates that the claim is true and $y_c = 1$ otherwise, and j is the corresponding generated justification.

Attack goals. We consider a practical adversarial claim attack scenario, in which an attacker perturbs an input claim c into an adversarial claim c' to mislead the automated fact-checking system. A valid attack should simultaneously satisfy the following three conditions:

- **Attack effectiveness.** The attack should reliably induce incorrect verification outcomes in the target AFC system, thereby compromising its fact-checking capability.
- **Factual content preservation.** c' should preserve the original semantics of c, i.e., $\mathrm{Sim}(c, c') \geq \tau$, where $\mathrm{Sim}(\cdot, \cdot)$ measures the semantic similarity between two claims (a higher value indicates greater similarity), and τ is a threshold for preserving semantic similarity. This constraint prevents semantic drift, ensuring that adversarial claims reflect the original factual content and that the verification remains meaningful.
- **High attack stealthiness.** c' should induce relevant justifications. If the generated justification is completely meaningless or irrelevant, the attack can be easily detected through human inspection.

Attacker's capabilities. To ensure the practicality of the attack, we assume that the attacker can only access the inputs and outputs of the target AFC system, without any knowledge of its internals (e.g., evidence sources). This assumption is consistent with the typical deployment of commercial AFC systems. Nevertheless, the attacker can still generate c' by querying a surrogate AFC system and leveraging transferability to launch the attack, a strategy widely studied in other domains.

4 Methodology

To systematically investigate adversarial claim attacks against search-enabled LLM-based fact-checking systems, we present an agent-based adversarial evaluation framework (Figure 2). At a high level, DECEIVE-AFC iteratively generates and evaluates adversarial claims, using evaluation feedback to guide subsequent attack exploration. We first introduce claim-level attack strategies that manipulate input claims to influence evidence retrieval and verification (Section 4.1). Next, we specify an attack validity evaluation strategy that constrains adversarial claim generation under our threat model (Section 4.2). Finally, we design an agent-based pipeline to automatically discover effective adversarial claims (Section 4.3).

4.1 Adversarial Claim Attack Strategy

Although attackers have no access to the internal models or retrieval mechanisms, the input claim serves as the sole interface through which the entire pipeline is activated. As a result, carefully crafted adversarial claims can implicitly influence multiple stages of the fact-checking process, shaping how evidence is retrieved and how the final verification decision is made. Motivated by this observation, we propose a set of adversarial claim attack strategies for search-enabled fact-checking systems,

aiming to induce incorrect verification outcomes under black-box settings. By targeting vulnerabilities at different stages of the AFC pipeline, we categorize adversarial claim attacks into three classes.

1. Search engine misguidance. A key distinction of search-enabled AFC systems is the integration of an online search component, which interprets claims into search queries and retrieves external evidence. This introduces a failure mode in which changes to the input claim can lead to different retrieved evidence. To exploit this vulnerability, we design search misguidance strategies that systematically modify the lexical and structural features of the claim to steer query generation and retrieval outcomes while preserving the overall factual intent of the claim. As a result, irrelevant, low-quality, or misleading evidence is more likely to be retrieved, leading subsequent verification to rely on less reliable evidence. This strategy includes four sub-strategies: (1) *low-frequency synonym introduction*, (2) *non-standard entity referencing*, (3) *redundant background information injection*, and (4) *keyword dispersion*.

2. LLM reasoning disruption. After retrieval, such AFC systems rely on LLMs to jointly reason over the input claim and retrieved evidence to produce final predictions and justifications. This stage introduces a failure mode where increased linguistic complexity or irrelevant signals can disrupt the alignment between the claim and retrieved evidence during reasoning. To exploit this, we design reasoning disruption strategies that modify the syntactic and discourse structures of the claim, increasing cognitive load and introducing ambiguity without altering factual content. These perturbations distract attention and weaken claim–evidence alignment, thereby increasing the likelihood of incorrect or inconsistent reasoning outcomes. This strategy includes four sub-strategies: (1) *injecting factually irrelevant but valid statements*, (2) *increasing syntactic complexity*, (3) *introducing conditional or speculative phrasing*, and (4) *employing double negation structures*.

3. Verification complexity attack. Search-enabled AFC systems rely on multi-step retrieval and reasoning to verify complex claims. This creates a failure mode where errors in intermediate steps can propagate and lead to incorrect predictions. To exploit this, we design verification complexity attacks that transform simple claims into multi-hop reasoning problems by introducing implicit dependencies. This increases the complexity of the verification pipeline, making it more brittle as errors in intermediate steps can propagate. This is achieved through two sub-strategies: (1) *decomposing explicit entities into indirect references*, and (2) *rephrasing atomic facts as compound relational statements*. We detail the descriptions of all strategies in Appendix A.

4.2 Attack Validity Evaluation Strategy

In this section, we describe how adversarial claim generation is constrained to ensure attack validity and how attack success is evaluated within our framework. Specifically, attack success is assessed at two stages of the pipeline: during the iterative refinement process and in the final evaluation stage. We adopt distinct evaluation criteria at each stage to reflect their different objectives and constraints.

Attack validity evaluation during refinement. During iterative refinement, attack validity is evaluated using strict criteria in a fully automated manner. Specifically, we first require the verification verdict to be incorrect, ensuring that the attack induces a decision error. We then use semantic similarity measures [17] to assess the consistency between the benign and adversarial claims, preventing obvious semantic drift or topic changes that would invalidate the attack. In addition, we employ Natural Language Inference (NLI) [29] to examine the logical relationship between benign and adversarial claims, ensuring that the adversarial claim preserves the original factual intent rather than altering it into a contradictory fact. Finally, we leverage an LLM to assess whether the generated justification remains relevant to the adversarial claim, thereby filtering out obvious topic drift.

Attack evaluation in the final stage. In this stage, we adopt the same criteria but with relaxed thresholds. While an incorrect verdict remains mandatory, we relax the semantic similarity and NLI-based constraints: (1) the similarity threshold between benign and adversarial claims is lowered, and (2) the NLI requirement is weakened from strict bi-directional entailment to the absence of explicit contradiction. Because misleading AFC systems often require subtle semantic shifts to induce incorrect fact-checking decisions, such shifts may reduce embedding-based similarity or NLI scores, even when the core factual intent is preserved. Human validation is applied only at this final evaluation stage to borderline cases that do not satisfy the strict automatic criteria, and is never used during adversarial claim generation or iterative refinement. Detailed annotation protocols are provided in Appendix B. This evaluation strategy balances strict automated optimization with reliable

final evaluation, while preserving the original factual intent of successful adversarial claims. We also analyze the sensitivity and reliability of these validity criteria in Appendix C.

4.3 Agent-based Attack Trajectory Exploration Framework

Building upon the adversarial claim attack strategies (Section 4.1) as well as the attack validity and evaluation criteria (Section 4.2), we design DECEIVE-AFC, which can systematically discover effective adversarial claims under a realistic black-box scenario. DECEIVE-AFC models the attack process as a closed-loop interaction among multiple components, each responsible for a distinct role in generation, evaluation, and refinement.

Adversarial Claim Generator. Conditioned on the selected attack strategy or feedback from previous iterations, the Generator is responsible for initializing candidate adversarial claims or refining existing ones. For each attack instance, the Generator performs multi-round refinement and retains context from previously generated claim variants and feedback within the same conversation, ensuring consistent and coherent modification of the claim.

Victim Module. To simulate the behavior of a realistic AFC system, we employ a surrogate fact-checking model as the victim model during refinement. This design avoids direct interaction with proprietary or rate-limited target systems, enabling efficient and repeated querying while remaining consistent with black-box settings. Given a benign claim or candidate adversarial claim, the Victim Module queries the surrogate model and returns the predicted verification verdict with corresponding justification. Each interaction is performed independently, where no historical context or previous claim variants are retained across queries, preventing unintended contamination from earlier interactions and preserving fairness.

Attack Evaluator. Given the verification verdicts and justifications generated for both benign and adversarial claims, the Evaluator performs a structured analysis of the AFC system's behavioral changes under adversarial perturbations. It produces diagnostic signals for subsequent refinement, such as verdict flips and justification shifts. Each evaluation is conducted independently without access to prior evaluation history, ensuring an unbiased assessment.

Semantic Consistency Check. In addition to the Evaluator, we employ a lightweight semantic consistency check to assess the validity of adversarial claims by checking for excessive semantic drift between benign and adversarial inputs. It filters out adversarial claims that degenerate into semantically unrelated or contradictory statements, thereby preserving the original factual content. The specific semantic evaluation criteria follow those defined in Section 4.2.

Attack Planner. The Attack Planner uses feedback from the Evaluator and semantic validity outcomes from the Semantic Consistency Check to guide the attack process. Specifically, it follows deterministic routing rules based on structured evaluation signals, including attack success, justification shift, evidence sufficiency, and semantic consistency. Based on these signals, it decides whether to terminate or continue refinement. The process stops when the attack successfully induces an incorrect verdict and satisfies semantic validity, or reaches the maximum refinement rounds. If refinement continues, the Planner determines the next action, including (i) further optimizing within the current strategy, (ii) switching to a different sub-strategy within the same strategy family, or (iii) transitioning to a different high-level strategy. Specifically, it continues within the current strategy when partial progress is observed, switches strategies when the current approach shows limited effect, and prioritizes retrieval-oriented strategies when evidence insufficiency is detected. The resulting decision is passed to the Generator to guide subsequent claim generation, enabling adaptive exploration of attack trajectories. A detailed example of the Planner's feedback-driven routing process is provided in Appendix D.

5 Experiments

5.1 Setup

Target models. We evaluate DECEIVE-AFC's performance on three representative search-enabled LLM-based fact-checking systems, which serve as the victim models in our experiments, including (1) HiSS [34], (2) LEMMA [32], and (3) DEFAME [3]. These systems all build upon LLMs for fact-checking and dynamically retrieve external evidence from the open web. Detailed descriptions are provided in Appendix F.1.

Table 1: Verification performance of the surrogate victim model under benign and adversarial claims.

Setting	Acc	Prec	Rec	F1	ASR ↑
Benign	90.5%	95.1%	85.4%	90.0%	–
DECEIVE-AFC	57.4%	61.1%	40.1%	48.4%	36.6%

Table 2: Comparison of attack transferability under single- and multi-surrogate optimization.

Target AFC	Single Surrogate	Multi-Surrogate
DEFAME	31.4%	32.1%
HiSS	19.5%	19.8%
LEMMA	18.8%	19.5%

Baselines. We select six representative attack approaches. Four are claim-level perturbation methods from FACTEVAL [16]: *LEET*, *Homoglyph*, *Character Swap*, and *Phonetic*, covering visual obfuscation, character-level noise, and phonetic variation. We also include two semantic-level LLM-based attack baselines that aim to mislead the AFC system [5]: *Paraphrase*, which performs synonym-based rewriting while preserving semantic meaning, and *Style rewriting*, which rewrites claims in alternative styles (e.g., sensational or news-reporting) to enhance plausibility. Detailed descriptions of each baseline are provided in Appendix F.2.

Dataset. Our experiments are conducted on the MOCHEG dataset [33], a representative benchmark for evaluating automated fact-checking (AFC) systems. It contains diverse claims collected from multiple fact-checking sources, covering a wide range of real-world topics. In our evaluation, we adopt a claim manipulation setting, where DECEIVE-AFC transforms benign claims into adversarial variants. Attacks are performed on the test split of MOCHEG, and samples labeled as *Not Enough Information* (NEI) are excluded to align with our threat model. After filtering, the final evaluation set contains 1,642 samples, including 817 positive and 825 negative claims.

Metrics. To evaluate the effectiveness of adversarial claim attacks against search-enabled LLM-based fact-checking systems, we report following metrics. We also report accuracy, precision, recall, and F1 score to assess the verification performance of target models under benign and adversarial inputs.

- *Attack Success Rate (ASR).* The proportion of adversarial claims that successfully cause the target AFC system to produce an incorrect verification verdict under the validity constraints (Section 4.2).
- *ROUGE-1/2/L.* These metrics measure lexical overlap between benign and adversarial retrieved evidence/justification at the unigram, bigram, and longest common subsequence levels [13].
- *MAUVE.* It measures distribution-level similarity between sets of retrieved evidence/justification, reflecting shifts in evidence retrieval patterns under adversarial attacks [21].
- *SummaC (avg).* It measures semantic agreement between benign and adversarial retrieved evidence/justification, where higher values indicate greater consistency [12].

Configuration. We evaluate DECEIVE-AFC under a unified black-box setting, where different model components are assigned roles for adversarial claim generation, evaluation, and validation. Both the Generator and Evaluator agents are instantiated using GPT-4o, while the surrogate fact-checking system is implemented with a search-enabled LLM that integrates external web retrieval. Adversarial claims are iteratively refined under semantic preservation constraints with a fixed iteration budget. Full implementation details, model choice, and validation criteria are provided in Appendix G.

5.2 Main Results

Surrogate model performance. Table 1 shows the surrogate model's performance under benign and adversarial claims. While achieving high accuracy on benign inputs, performance drops significantly under adversarial claims, with accuracy decreasing from 90.5% to 57.4%, corresponding to an ASR of 36.6%. We further analyze the refinement process and observe that successful attacks not only flip verdicts but also induce shifts in generated justifications, indicating disruption of reasoning and evidence interpretation. Detailed analysis is provided in Appendix I. The high performance under benign inputs is expected and consistent with prior work [19], as GPT-4o-search-preview can often retrieve authoritative fact-checking articles containing the original claims and their verdicts.

Table 3: Comparison of adversarial claim attack methods across different target fact-checking systems. "Benign" denotes verification performance on original (non-adversarial) claims. ASR is computed under the validity constraints described in Section 4.2.

Attack Method	HiSS					LEMMA					DEFAME				
	Acc	Prec	Rec	F1	ASR ↑	Acc	Prec	Rec	F1	ASR ↑	Acc	Prec	Rec	F1	ASR ↑
Benign	77.5%	80.8%	70.8%	75.4%	–	66.1%	72.8%	64.9%	62.3%	–	78.7%	86.9%	76.9%	81.1%	–
LEET	–	–	–	–	–	55.2%	69.2%	55.1%	44.8%	12.1%	70.6%	70.9%	70.6%	70.5%	8.9%
Homoglyph	69.1%	67.1%	67.2%	67.1%	8.5%	59.7%	67.8%	54.9%	45.2%	6.7%	71.9%	71.2%	73.4%	72.3%	7.1%
Character Swap	70.1%	70.5%	68.5%	69.5%	7.6%	59.1%	58.7%	60.5%	59.5%	7.2%	73.1%	72.4%	74.1%	73.2%	5.8%
Phonetic	73.4%	84.9%	54.2%	66.2%	4.3%	57.5%	69.9%	57.5%	49.6%	8.8%	71.9%	71.9%	71.8%	71.8%	6.9%
Paraphrase	68.9%	70.8%	63.6%	67.0%	8.7%	57.9%	58.6%	52.6%	55.3%	8.4%	67.5%	67.4%	67.1%	67.2%	11.7%
Style-rewriting	68.3%	70.4%	62.7%	66.3%	9.4%	57.1%	57.8%	51.2%	54.3%	9.2%	66.4%	66.5%	65.5%	66.0%	12.9%
DECEIVE-AFC	61.9%	62.2%	59.9%	61.1%	**19.5%**	51.1%	50.6%	79.6%	61.9%	**18.8%**	53.7%	55.5%	58.4%	50.3%	**31.4%**

Table 4: Similarity between benign and adversarial evidence and justifications on target AFC systems (averaged). R-1/2/L denotes ROUGE-1/2/L. Lower values indicate larger shifts.

Case	Evidence			Justification		
	R-1/2/L ↓	MAUVE ↓	SummaC ↓	R-1/2/L ↓	MAUVE ↓	SummaC ↓
Failed attacks	0.323 / 0.172 / 0.241	0.622	0.032	0.582 / 0.374 / 0.427	0.896	-0.063
Successful attacks	0.245 / 0.121 / 0.176	0.449	-0.061	0.478 / 0.267 / 0.330	0.653	-0.184

Therefore, while such models are not intended for standalone deployment, their strong performance, practical efficiency, and moderate cost make them suitable surrogate models for large-scale adversarial evaluation in realistic black-box settings.

Multi-surrogate optimization. We examine whether optimizing adversarial claims against a single surrogate limits their transferability. We construct a multi-surrogate variant that randomly selects one of GPT-4o-search, Gemini-2.5-Flash-search, and Perplexity at each refinement iteration, while keeping the remaining attack procedure unchanged. Table 2 compares the transfer performance of single- and multi-surrogate optimization. Multi-surrogate optimization yields only modest ASR improvements across the three target systems. On DEFAME, for example, ASR increases from 31.4% to 32.1%, while the average attack cost increases from $9.2 to $11.5 per 100 samples and the execution time increases from 97.4 s to 121.3 s. These results suggest that introducing additional surrogate diversity provides only marginal improvements in attack transferability, while incurring additional computational cost and latency.

Target search-enabled LLM-based systems performance. Table 3 compares different attacks across three target systems. Our method consistently achieves the highest ASR, demonstrating strong effectiveness and transferability. Perturbation-based methods yield limited degradation and often produce invalid or unreadable inputs. In particular, LEET leads to frequent refusals in HiSS due to severe distortions. Thus, its results on HiSS are not reported. LLM-based rewriting methods perform slightly better but remain substantially weaker than our approach, indicating that naive rewriting is insufficient to mislead search-enabled AFC systems.

Evidence and justification shift analysis. We analyze the similarity between retrieved evidence and generated justifications from benign and adversarial claims on target AFC systems, averaged across models and grouped by successful and failed attacks (Table 4). Successful attacks consistently yield lower similarity in both retrieved evidence and generated justifications, while failures remain closer to the benign case. Notably, shifts are more pronounced in justifications, indicating that small perturbations in retrieval can lead to amplified deviations in justification. Overall, DECEIVE-AFC operates across multiple stages of the AFC pipeline, affecting both retrieval and downstream reasoning. The results for each target AFC system are provided in Appendix H.

Attack strategy comparison. To evaluate the effectiveness of our attack strategies, we apply the same iterative framework to baseline methods. DEFAME is selected as the representative target due to its strong performance on benign inputs. We consider two representative baselines with the strongest performance: LEET, a rule-based perturbation method, and Style-rewriting, an LLM-based semantic rewriting approach. Results in Table 5 show that baseline methods achieve limited gains under iterative optimization, indicating that attack effectiveness also depends on the underlying strategy rather than optimization alone. LEET incurs low cost but achieves very low ASR, while Style-rewriting, despite a similar cost to our method, yields substantially lower ASR. These results

Table 5: Effect of iterative optimization on baseline attacks.

Method	ASR ↑	Cost (USD/100) ↓	Time (s) ↓
LEET (optimized)	10.1%	3.6	47.4
Style-rewriting (optimized)	15.4%	8.9	92.3
DECEIVE-AFC	**31.4%**	**9.2**	**97.4**

Table 6: Ablation study of attack strategies and exploration mechanisms.

Method	ASR ↑	Cost (USD / 100)	Time (s)
Search engine misguidance	23.6%	10.3	110.2
LLM reasoning disruption	23.4%	10.6	113.7
Verification complexity attack	22.9%	10.5	112.6
Random exploration	16.7%	11.4	128.8
DECEIVE-AFC	**31.4%**	**9.2**	**97.4**

show that DECEIVE-AFC achieves substantially higher attack effectiveness at a comparable cost. Additional experiments on human validation, evaluator consistency, dataset generalization, and the underlying LLM are provided in Appendix E.

5.3 Ablation Study

We analyze contributions of four components in DECEIVE-AFC: (1) adversarial claim attack strategies, (2) agent-based exploration, (3) the semantic consistency check, and (4) the number of refinement rounds. We evaluate (1) and (2) on DEFAME to measure their impact on attack effectiveness, while (3) and (4) are analyzed on the surrogate model to isolate their roles during optimization.

Adversarial claim attack strategy. Each strategy is evaluated individually under the same iterative framework. As shown in Table 6 (rows 1–3), all strategies achieve relatively high ASR but underperform the full framework. This suggests that different strategies capture complementary attack opportunities by targeting distinct vulnerabilities in the claim interpretation and verification process. Their combination allows the framework to explore a broader range of adversarial claim modifications than any individual strategy.

Agent-based exploration. We construct a random variant where attack strategies are selected randomly at each refinement step without using feedback from the Evaluator. As shown in Table 6 (row 4), it achieves substantially lower ASR, even worse than single-strategy optimization. This demonstrates the importance of using intermediate evaluation feedback to adaptively guide strategy selection and subsequent refinement toward more effective adversarial claims.

Semantic consistency check. We compare the full framework with a variant without the semantic consistency check. As shown in Table 7, removing this check slightly increases label flips (i.e., incorrect predictions) but reduces ASR, because some flipped verdicts result from changes to the claim's factual meaning. This indicates that the semantic consistency check stabilizes optimization by preventing semantically invalid attacks. We further analyze the effect of refinement rounds (component (4)), observing a trade-off between effectiveness and efficiency. Detailed analysis results are provided in Appendix J.

6 Potential Mitigation

Our experimental results demonstrate that adversarial claim attacks can successfully flip the final verification verdict. Meanwhile, generated justifications are seemingly relevant and coherent to the verdict, increasing the difficulty of identification. Incorrect decisions are no longer accompanied by obviously flawed reasoning, making such failures harder to recognize and potentially increasing their downstream impact. Even the search-enabled, LLM-based AFC systems, which are commonly regarded as more robust due to their multi-stage verification pipelines, remain susceptible to such attacks. To mitigate these risks, we discuss two mitigation strategies for improving the robustness of search-enabled LLM-based AFC systems.

Table 7: Ablation study on the effect of the semantic consistency check

Setting	Label Flip Rate	Attack Success Rate
Full framework	42.1%	36.6%
w/o semantic consistency check	43.3%	23.1%

The first is *verification model fine-tuning*. Most existing fact-checking systems rely on general base LLMs, lacking safety alignment to handle adversarially crafted claims. By fine-tuning base LLMs on paired benign claims and their adversarial variants, the model can learn common adversarial perturbation patterns. This helps the model distinguish superficial linguistic changes from factual alterations and better interpret the underlying claim semantics. To examine the effectiveness of this mitigation, we additionally conduct adversarial fine-tuning on the verification model in DEFAME. After fine-tuning, the ASR decreases from 31.4% to 25.6%, while clean verification accuracy changes from 78.7% to 76.9%. These results demonstrate the potential of adversarial fine-tuning as a practical mitigation, while highlighting the remaining robustness gap and the need to balance adversarial robustness against clean verification accuracy.

The second is *interpretation and retrieval validation*. On the other hand, our results underscore the importance of strengthening the fact-checking pipeline itself. In search-enabled, LLM-based AFC systems, the verification process typically involves interpreting the input claim, generating search queries, and verifying the claim based on retrieved evidence. Adversarial claim manipulations can disrupt these stages, leading to the use of misleading, noisy, or low-credibility sources, even when the LLM is reasoning correctly. Adversarial-aware validation at these stages may mitigate such risks by checking semantic consistency between the original claim and generated queries, as well as the relevance and credibility of retrieved evidence.

7 Limitations and Ethical Considerations

Limitations. Despite its effectiveness, our framework relies on LLM-based rewriting during iterative refinement, incurring higher computational cost than lightweight rule-based methods. However, such methods perform poorly in search-enabled AFC systems. Compared to LLM-based baselines, our method achieves substantially better effectiveness at a comparable cost. As LLM capabilities improve and costs decrease, the efficiency of our framework is expected to further improve.

Ethical Considerations. This work investigates adversarial attacks on AFC systems to expose vulnerabilities and improve robustness. While the proposed methods may be misused to generate misleading claims, our goal is to inform the development of more robust verification systems. To reduce risks, we do not deploy the system or release harmful content in real-world applications. Any released artifacts will follow responsible disclosure practices with appropriate safeguards.

8 Conclusion

We present a systematic study of adversarial claim attacks against search-enabled LLM-based fact-checking systems and propose `DECEIVE-AFC`, an agent-based framework that combines tailored attack strategies with validity constraints. Our approach generates semantically consistent adversarial claims that effectively degrade fact-checking performance. Experiments on multiple real-world systems show that `DECEIVE-AFC` outperforms strong baselines and exhibits consistent cross-system transferability. The results highlight that search-enabled AFC systems remain vulnerable to carefully crafted claims that indirectly affect retrieval and verification. We further discuss mitigation strategies toward more robust and trustworthy fact-checking systems.

Acknowledgments and Disclosure of Funding

This research is supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Infocomm Media Development Authority.

References

[1] Mubashara Akhtar, Michael Schlichtkrull, Zhijiang Guo, Oana Cocarascu, Elena Simperl, and Andreas Vlachos. Multimodal automated fact-checking: A survey. *arXiv preprint arXiv:2305.13507*, 2023.

[2] Nicholas Boucher, Ilia Shumailov, Ross Anderson, and Nicolas Papernot. Bad characters: Imperceptible nlp attacks. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1987–2004. IEEE, 2022.

[3] Tobias Braun, Mark Rothermel, Marcus Rohrbach, and Anna Rohrbach. Defame: Dynamic evidence-based fact-checking with multimodal experts. *arXiv preprint arXiv:2412.10510*, 2024.

[4] Kevin Matthe Caramancion. News verifiers showdown: a comparative performance evaluation of chatgpt 3.5, chatgpt 4.0, bing ai, and bard in news fact-checking. In *2023 IEEE Future Networks World Forum (FNWF)*, pages 1–6. IEEE, 2023.

[5] Canyu Chen and Kai Shu. Can llm-generated misinformation be detected? *arXiv preprint arXiv:2309.13788*, 2023.

[6] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. A survey on automated fact-checking. *Transactions of the association for computational linguistics*, 10:178–206, 2022.

[7] Christopher Hidey, Tuhin Chakrabarty, Tariq Alhindi, Siddharth Varia, Kriste Krstovski, Mona Diab, and Smaranda Muresan. Deseption: Dual sequence prediction and adversarial examples for improved fact-checking. *arXiv preprint arXiv:2004.12864*, 2020.

[8] Beizhe Hu, Qiang Sheng, Juan Cao, Yuhui Shi, Yang Li, Danding Wang, and Peng Qi. Bad actor, good advisor: Exploring the role of large language models in fake news detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 22105–22113, 2024.

[9] Linmei Hu, Siqi Wei, Ziwang Zhao, and Bin Wu. Deep learning for fake news detection: A comprehensive survey. *AI open*, 3:133–155, 2022.

[10] Youngwoo Kim and James Allan. Fever breaker's run of team nbauzdrlqg. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 99–104, 2019.

[11] Neema Kotonya and Francesca Toni. Explainable automated fact-checking: A survey. *arXiv preprint arXiv:2011.03870*, 2020.

[12] Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177, 2022. doi: 10.1162/tacl_a_00453. URL `https://aclanthology.org/2022.tacl-1.10/`.

[13] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[14] Fanzhen Liu, Sharif Abuadbba, Kristen Moore, Surya Nepal, Cecile Paris, Jia Wu, Jian Yang, and Quan Z Sheng. Adversarial attacks against automated fact-checking: A survey. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 22979–23001, 2025.

[15] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35, 2023.

[16] Mamta Mamta and Oana Cocarascu. Facteval: Evaluating the robustness of fact verification systems in the era of large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10647–10660, 2025.

[17] Lingling Meng, Runqing Huang, and Junzhong Gu. A review of semantic similarity measures in wordnet. *International Journal of Hybrid Information Technology*, 6(1):1–12, 2013.

[18] Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. Automated fact-checking for assisting human fact-checkers. *arXiv preprint arXiv:2103.07769*, 2021.

[19] Haoran Ou, Gelei Deng, Xingshuo Han, Jie Zhang, Xinlei He, Han Qiu, Shangwei Guo, and Tianwei Zhang. Holmes: Automated fact check with large language models. *arXiv preprint arXiv:2505.03135*, 2025.

[20] Liangming Pan, Xiaobao Wu, Xinyuan Lu, Luu Anh Tuan, William Yang Wang, Min-Yen Kan, and Preslav Nakov. Fact-checking complex claims with program-guided reasoning. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 6981–7004, 2023.

[21] Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: measuring the gap between neural text and human text using divergence frontiers. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

[22] Peng Qi, Zehong Yan, Wynne Hsu, and Mong Li Lee. Sniffer: Multimodal large language model for explainable out-of-context misinformation detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13052–13062, 2024.

[23] Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. Towards debiasing fact verification models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3419–3425, 2019.

[24] Sahar Tahmasebi, Eric Müller-Budack, and Ralph Ewerth. Multimodal misinformation detection using large vision-language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2189–2199, 2024.

[25] James Thorne and Andreas Vlachos. Automated fact checking: Task formulations, methods and future directions. *arXiv preprint arXiv:1806.07687*, 2018.

[26] James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. The fact extraction and verification (fever) shared task. *arXiv preprint arXiv:1811.10971*, 2018.

[27] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Evaluating adversarial attacks against multiple fact verification systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2944–2953, 2019.

[28] Longzheng Wang, Xiaohan Xu, Lei Zhang, Jiarui Lu, Yongxiu Xu, Hongbo Xu, Minghao Tang, and Chuang Zhang. Mmidr: Teaching large language model to interpret multimodal misinformation via knowledge distillation. *arXiv preprint arXiv:2403.14171*, 2024.

[29] Shuohang Wang and Jing Jiang. Learning natural language inference with lstm. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1442–1451, 2016.

[30] Xiuxian Wang, Lanjun Wang, Yuting Su, Hongshuo Tian, Guoqing Jin, and An-An Liu. Few-shot in-context learning for implicit semantic multimodal content detection and interpretation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[31] Lianwei Wu, Linyong Wang, and Yongqiang Zhao. Unified evidence enhancement inference framework for fake news detection. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 6541–6549, 2024.

[32] Keyang Xuan, Li Yi, Fan Yang, Ruochen Wu, Yi R. Fung, and Heng Ji. Lemma: Towards lvlm-enhanced multimodal misinformation detection with external knowledge augmentation, 2024. URL `https://arxiv.org/abs/2402.11943`.

[33] Barry Menglong Yao, Aditya Shah, Lichao Sun, Jin-Hee Cho, and Lifu Huang. End-to-end multimodal fact-checking and explanation generation: A challenging dataset and models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2733–2743, 2023.

[34] Xuan Zhang and Wei Gao. Towards llm-based fact verification on news claims with a hierarchical step-by-step prompting method. *arXiv preprint arXiv:2310.00305*, 2023.

Appendix

A Detailed Attack Strategy

In this section, we provide detailed descriptions of the adversarial claim attack strategies used in our framework.

1. Search Engine Misguidance. These strategies aim to indirectly influence the evidence retrieval stage by modifying lexical features of the claim, including four sub-categories:

- *Low-frequency synonym introduction.* This strategy replaces common nouns, verbs, or descriptors in the claim with low-frequency, formal, or uncommon synonyms. While semantic equivalence is preserved, the modified wording reduces alignment with typical search queries, thereby weakening keyword-based retrieval performance.
- *Non-standard entity referencing.* Instead of explicitly mentioning named entities, this strategy refers to them indirectly through descriptive phrases, metonymy, or role-based identifiers. Although the referenced entity remains unambiguous to humans, the absence of canonical entity names can significantly affect search result relevance.
- *Redundant background information injection.* This strategy inserts additional but factually irrelevant background details into the claim. These perturbations do not alter the claim's truth value but introduce distracting keywords that may bias retrieval toward tangential or less informative evidence sources.
- *Keyword dispersion.* This strategy restructures the claim by spreading core keywords across multiple clauses or diluting them with connective phrases. By lowering keyword concentration, this strategy weakens the search engine's ability to correctly identify and prioritize relevant evidence.

2. LLM Reasoning Disruption. These strategies target the verification stage by increasing the cognitive and reasoning complexity of the claim, including four sub-categories:

- *Injecting factually irrelevant but valid statements.* This strategy adds neutral, factually correct statements before or after the core claim. Although unrelated to the verification target, such statements can distract the model's attention and interfere with its ability to focus on the relevant factual content.
- *Increasing syntactic complexity.* The claim is rewritten using embedded clauses, nested sentence structures, or parenthetical expressions. These syntactic transformations preserve meaning but increase parsing difficulty, which can impair reasoning accuracy in complex verification scenarios.
- *Introducing conditional or speculative phrasing.* This strategy embeds the claim within conditional or hypothetical constructions that do not alter its truth value. Such framing can obscure the model's interpretation of factual assertions, leading to uncertainty or misclassification.
- *Employing double negation structures.* The claim is rewritten using double negation or logically redundant constructions. Although logically equivalent to the original statement, these formulations increase reasoning load and can confuse entailment-based verification mechanisms.

3. Verification Complexity Attack. This strategy targets multi-hop reasoning by transforming direct factual statements into more complex relational structures, including two sub-categories:

- *Decomposing explicit entities into indirect references.* This strategy replaces direct entity mentions with indirect descriptions involving intermediate concepts or attributes. As a result, the system must infer entity identity through multiple reasoning steps rather than relying on explicit mentions.
- *Rephrasing atomic facts as compound relational statements.* Simple factual assertions are reformulated as compound statements involving multiple relationships or dependencies. While the

Table 8: Sensitivity analysis of the semantic similarity threshold.

Threshold	Label Flip Rate	ASR	Validity Rate
0.90	24.8%	20.6%	100.0%
0.85	34.9%	29.3%	100.0%
0.80	36.7%	31.4%	99.3%
0.75	38.2%	34.5%	97.1%
0.70	41.4%	36.2%	94.4%
0.65	42.9%	30.3%	82.8%

underlying fact remains unchanged, verification now requires aggregating information across multiple relational hops.

B Human Validation Protocol

Human validation is applied only during the final evaluation as a complementary step to handle borderline cases that do not satisfy the strict automatic criteria in Section 4.2. It is not involved in adversarial claim generation or iterative refinement, both of which remain fully automated. To minimize subjectivity, we adopt a structured and rule-based annotation protocol:

- **(1) Semantic consistency.** Annotators verify that the adversarial claim preserves the original factual intent and does not introduce contradictions.
- **(2) Interpretability.** Annotators ensure that the adversarial claim remains understandable and natural for human readers.
- **(3) Independent evaluation.** Each case is reviewed by multiple annotators.
- **(4) Agreement filtering.** Only cases with annotator agreement are retained.

Based on our analysis of the experiment results, only a small proportion of cases (9.4%) fall below the strict thresholds and require human validation. Thus, human assessment is limited to a small subset of borderline cases in the final evaluation, while the attack generation and refinement process remains fully automated. This design provides a balance between automation and reliability, ensuring robust evaluation without heavy reliance on human judgment.

C Additional Analysis of Attack Validity Evaluation

C.1 Threshold Sensitivity

We analyze the sensitivity of attack evaluation to the semantic similarity threshold used to constrain adversarial claims. We vary the threshold from 0.90 to 0.65 while keeping all other settings fixed. As shown in Table 8, relaxing the threshold generally increases the label flip rate and ASR by allowing greater flexibility in claim rewriting. However, overly relaxed thresholds reduce attack validity due to increased semantic drift. In particular, lowering the threshold from 0.70 to 0.65 increases the label flip rate from 41.4% to 42.9%, but decreases the validity rate from 94.4% to 82.8% and the resulting ASR from 36.2% to 30.3%. These results indicate that attack effectiveness is not simply driven by relaxing the semantic constraint, but reflects a trade-off between successful perturbation and factual-content preservation.

C.2 Reliability of Automatic Validity Evaluation

We further examine whether the automatic validity evaluation can distinguish meaning-preserving rewrites from subtle meaning-altering perturbations. In addition to semantic similarity, our evaluation uses NLI-based consistency checking to identify changes that remain lexically similar but alter the factual meaning. Table 9 shows representative controlled examples. A synonym-based rewrite is accepted because it maintains high semantic similarity and entailment consistency, whereas antonym substitution, negation, and entity substitution are rejected due to contradictions detected by NLI, despite their high semantic similarity.

Table 9: Representative examples for automatic validity evaluation. Semantic similarity alone may remain high for meaning-altering perturbations, while NLI identifies the resulting contradictions.

Perturbation	Semantic Similarity	NLI Relation	Decision
Synonym substitution	0.922	Entailment	Accept
Antonym substitution	0.990	Contradiction	Reject
Negation	0.999	Contradiction	Reject
Entity substitution	0.946	Contradiction	Reject

We additionally manually inspect the adversarial claims that pass the automatic validity criteria, which account for 90.6% of the evaluated cases. Among the 1,488 automatically accepted claims, we observe no meaning-altering cases missed by the automatic evaluation (0/1,488). Together, these results show that combining semantic similarity with NLI helps prevent high-similarity but meaning-altering perturbations from being incorrectly accepted.

D Illustrative Example: Feedback-Driven Multi-Round Attack

We present a representative example to illustrate how the Attack Planner adaptively refines attack strategies based on structured feedback. The Planner implements a deterministic routing policy based on structured signals from the Evaluator and Semantic Consistency Check. It terminates when the attack succeeds while satisfying semantic consistency, continues or refines the current strategy when partial progress is observed, switches strategies after repeated ineffective attempts, and prioritizes retrieval-oriented strategies when the feedback indicates insufficient or degraded evidence.

Case background. *Original claim (x):* "In a July 2020 press conference, Goya Foods CEO Robert Unanue said, 'We are all truly blessed' to have a leader like Donald Trump." *Ground-truth label:* True. *Attack goal:* Induce the AFC system to predict False while preserving semantic consistency.

Round 1

- **Strategy:** Search engine misguidance (Low-frequency synonym introduction)
- **Key modification:** "press conference" $\rightarrow$ "press briefing".
- **AFC output:** *True* (the verdict is not flipped).
- **Evaluator signals:** `case_type = resistant_strong`, indicating that the system retrieves complete supporting evidence and remains robust to the perturbation.
- **Routing rationale:** Simple lexical substitution is insufficient because the search component still retrieves strong evidence through fuzzy matching and related terms.
- **Planner decision:** Switch to an LLM reasoning disruption strategy.

Round 4

- **Strategy:** LLM reasoning disruption (Injecting factually irrelevant but valid statements).
- **Key modification:** Adds an unrelated factual clause about the company's anniversary.
- **AFC output:** *True* (the verdict is not flipped).
- **Evaluator signals:** `case_type = resistant_but_degraded`, indicating that the attack begins to degrade justification quality but remains insufficient to flip the verdict.
- **Routing rationale:** The added distraction weakens evidence organization, but the core evidence still supports the original claim.
- **Planner decision:** Refine the attack direction toward a more precise lexical modification.

Round 6

- **Strategy:** Search engine misguidance (Low-frequency synonym introduction)
- **Key modification:** "We are all truly blessed" $\rightarrow$ "We are all exceptionally favored".

- **AFC output:** *False* (the verdict is successfully flipped).
- **Evaluator signals:** `attack_success = yes`; the retrieved evidence supports the original event but conflicts with the modified quotation.
- **Routing rationale:** The attack succeeds because the modified quoted phrase changes the exact-match evidence basis while preserving high semantic similarity.
- **Planner decision:** Terminate the attack.

Summary. This example demonstrates how the planner dynamically adapts attack strategies based on feedback signals. Instead of applying blind perturbations, the planner uses structured evaluation signals reflecting the target system's response (e.g., *resistant_strong* vs. degraded states) and accordingly adjusts the attack trajectory. In this example, it shifts from ineffective lexical perturbations to reasoning-level disruption, and finally to precise semantic modifications that exploit subtle vulnerabilities. This adaptive routing enables efficient exploration of the attack space, allowing the attack to succeed within a limited number of refinement rounds without exhaustive search.

E Additional Robustness and Generalization Experiments

E.1 Effect of Human Validation

To examine whether the reported attack effectiveness depends on human validation, we additionally evaluate successful attacks using only the automatic validity criteria, without human assessment of borderline cases. As shown in Table 10, removing human validation results in only marginal changes in ASR across all three target AFC systems: from 31.4% to 31.2% on DEFAME, from 19.5% to 19.2% on HiSS, and from 18.8% to 18.7% on LEMMA. These results indicate that the measured attack effectiveness is largely preserved under fully automatic evaluation and is not materially dependent on human validation.

Table 10: Comparison of ASR under the original evaluation protocol and automatic-only evaluation without human validation.

Target AFC	Original	Automatic Only
DEFAME	31.4%	31.2%
HiSS	19.5%	19.2%
LEMMA	18.8%	18.7%

E.2 Cross-Model Evaluator Consistency

We further examine whether using the same model family for the Generator and Evaluator introduces model-specific coupling into the refinement process. We replace the GPT-4o Evaluator with Gemini-2.5-Flash while keeping the remaining framework unchanged, and compare their evaluation outputs on the same generated cases. The two Evaluators achieve 98.3% agreement, indicating that the structured evaluation signals are highly consistent across different model families. This suggests that the Evaluator's feedback is not strongly dependent on using the same model as the Generator.

E.3 Generalization to AVeriTeC

To evaluate whether the attack effectiveness generalizes beyond Mocheg, we additionally conduct experiments using AVeriTeC. Table 11 reports the ASR of Style-Rewriting and `DECEIVE-AFC` on both datasets under the same evaluation setting.

Table 11: Attack success rates on Mocheg and AVeriTeC.

Method	Mocheg	AVeriTeC
Style-Rewriting	12.9%	11.6%
`DECEIVE-AFC`	31.4%	28.7%

The results show a similar trend across the two datasets. While the ASR of both methods decreases slightly on AVeriTeC, `DECEIVE-AFC` achieves 28.7% ASR compared with 11.6% for Style-Rewriting. This additional evaluation indicates that the observed attack effectiveness is not limited to claims drawn from Mocheg and remains substantial on a different fact-checking dataset.

E.4 Robustness to the Underlying LLM

We further examine whether the attack effectiveness is sensitive to the underlying LLM used by the target AFC system. Specifically, we replace GPT-4o in DEFAME with Gemini-2.5-Flash while keeping the remaining fact-checking pipeline unchanged. The resulting ASR is 30.2%, compared with 31.4% when using GPT-4o. The small difference suggests that the observed vulnerability is not specific to the original underlying LLM and persists when DEFAME is instantiated with a different model family.

F Detailed Experiment Settings

F.1 Target Models.

This section provides descriptions of the target fact-checking systems evaluated in our experiments.

HiSS. It is an LLM-based fact-checking approach that leverages in-context learning for claim verification. Its core design introduces a Hierarchical Step-by-Step prompting strategy, which decomposes an input claim into multiple subclaims and verifies each subclaim through a sequence of question-answering steps.

LEMMA. It is a misinformation detection framework built upon large language models. The method enhances LLM-based reasoning by explicitly augmenting it with external knowledge. Given a claim, LEMMA generates multiple search queries to retrieve relevant external evidence. This retrieved knowledge is then incorporated into the LLM's reasoning process to support verification decisions.

DEFAME. It is a modular, zero-shot, search-enabled fact-checking system for open-domain claim verification. It adopts a multi-stage pipeline that dynamically selects tools and search depth to retrieve evidence from the open web. DEFAME performs end-to-end verification by jointly reasoning over claims and retrieved evidence.

F.2 Baselines

This appendix provides detailed descriptions of the baseline claim perturbations adopted from FACTEVAL [16]. The LLM-based attack baselines are based on prior work [5].

LEET Perturbation. LEET perturbation replaces alphabetic characters with visually similar numbers or symbols (e.g., $A \rightarrow 4$, $E \rightarrow 3$). This writing style is commonly observed in online communities and adversarial settings. The perturbation primarily targets tokenization and lexical matching, while remaining easily interpretable by human readers.

Homoglyph Perturbation. Homoglyph perturbation substitutes characters with visually confusable Unicode counterparts (e.g., $n \rightarrow \tilde{n}$), based on standardized Unicode confusable character mappings. This attack introduces subtle visual ambiguity at the character level and has been shown to be particularly effective in misleading fact verification models without altering the perceived meaning of the claim.

Character Swap Perturbation. Character Swap perturbation randomly swaps adjacent characters within words, simulating common typographical errors frequently observed in real-world text inputs. This perturbation introduces lightweight spelling noise while largely preserving word readability and sentence-level semantics.

Phonetic Perturbation. Phonetic perturbation modifies words according to human-written phonetic variations, reflecting spelling patterns influenced by pronunciation differences, especially among non-native speakers. A fixed perturbation budget is applied to alter a subset of words in the claim, introducing realistic orthographic variations while maintaining semantic consistency.

Paraphrase. Paraphrase reformulates the input claim by altering its lexical and syntactic structure while preserving its original semantic meaning. This is typically achieved using LLM-based

Table 12: Decoding configurations for different model roles.

Role	Model	Temperature	Top-P
Generator	GPT-4o	0.8	1.0
Evaluator	GPT-4o	0.0	1.0
Victim	GPT-4o-search-preview	1.0	1.0

Table 13: Impact of model choice on attack success rate (ASR).

Setting	ASR ↑
GPT-4o	31.4%
Gemini-2.5-flash	29.3%

prompting to produce semantically equivalent variants. It aims to modify the surface form of the claim without changing its factual content, thereby targeting systems that rely on lexical matching or shallow textual patterns.

Style Rewriting. Style Rewriting transforms the input claim into alternative stylistic forms (e.g., sensational, formal, or news-reporting styles) while maintaining its semantic content. This is implemented via LLM-based rewriting with prompts that control tone, narrative style, or discourse structure. It aims to increase the plausibility and perceived credibility of the claim, potentially affecting how the AFC system interprets and processes the input.

G Implementation and Configuration Details

G.1 Model Instantiation

The Generator Agent and Evaluator Agent are both instantiated using GPT-4o. The surrogate fact-checking system is implemented using GPT-4o-search-preview, which combines LLM-based reasoning with external web search. We use different model parameters for generation and evaluation to balance diversity and stability. As shown in Table 12, the temperature is set to 0.8 for the Generator to promote linguistic diversity during adversarial claim generation, while the Evaluator uses a temperature of 0 to ensure deterministic and reproducible evaluation. For all models, Top-P is set to 1.0. The maximum response length is limited to 1000 tokens.

G.2 Model Choices

To examine the impact of model choice, we conduct additional experiments by replacing GPT-4o with Gemini-2.5-flash as the Generator and Evaluator, while keeping all other settings unchanged. DEFAME is used as the target AFC system. Gemini-2.5-flash is a weaker model compared to GPT-4o based on public LLM rankings[1]. The results are summarized in Table 13. We observe that using a weaker model leads to a moderate decrease in attack success rate. However, the attack remains highly effective, indicating that the performance gains primarily stem from the proposed attack strategies and framework design, rather than reliance on a specific model choice.

G.3 Semantic Consistency Check and Validation Criteria

Semantic similarity between the original and adversarial claims is computed using Sentence-BERT (all-mpnet-base-v2). Natural Language Inference consistency is assessed using DeBERTa-large-MNLI. During the refinement stage, adversarial claims are required to satisfy a semantic similarity threshold of 0.85 and bidirectional entailment under NLI. For the final evaluation, the similarity threshold is relaxed to 0.7, and the NLI constraint is weakened to the absence of explicit contradiction. These thresholds are used as the default settings in our experiments; their sensitivity is further analyzed in Appendix C.1. For borderline cases that do not satisfy the strict automatic criteria but

[1] `https://huggingface.co/spaces/lmarena-ai/arena-leaderboard`

Table 14: Evidence and justification shift analysis across target AFC systems. Lower scores indicate larger shifts from benign outputs.

Target	Output	Setting	ROUGE-1	ROUGE-2	ROUGE-L	MAUVE	SummaC
HiSS	Evidence	Success	0.253	0.128	0.183	0.447	-0.047
		Failure	0.308	0.159	0.226	0.563	0.021
	Justification	Success	0.492	0.287	0.337	0.752	-0.102
		Failure	0.581	0.371	0.422	0.907	-0.018
LEMMA	Evidence	Success	0.257	0.133	0.192	0.463	-0.043
		Failure	0.312	0.166	0.229	0.589	0.018
	Justification	Success	0.506	0.296	0.351	0.766	-0.094
		Failure	0.571	0.363	0.412	0.896	-0.024
DEFAME	Evidence	Success	0.226	0.098	0.154	0.437	-0.089
		Failure	0.349	0.191	0.266	0.713	0.052
	Justification	Success	0.436	0.219	0.301	0.442	-0.356
		Failure	0.594	0.387	0.447	0.884	-0.147

satisfy the relaxed final-evaluation criteria, human validation is conducted only during the final evaluation to ensure preservation of factual intent.

G.4 Refinement Budget and Strategy Space

Each attack instance is refined for at most 10 iterations. The attack strategy space consists of all combinations of strategy categories defined in Section 4.1. Only adversarial claims that satisfy the semantic preservation criteria are used to replace the original claims in the constructed evaluation dataset for testing on the target models.

H Evidence and Justification Shift Analysis

We provide detailed per-model results of the evidence and justification shift analysis in Table 14. Across all target models, successful attacks consistently exhibit lower similarity than failed attacks, indicating larger deviations from the benign case. Moreover, this shift is more pronounced in justifications than in evidence.

This pattern reflects the different roles of retrieval and reasoning in the AFC pipeline. Retrieval is affected by query perturbations and thus changes gradually. Justification reflects the reasoning process and can shift more abruptly once the decision boundary is crossed. Consequently, failed attacks may already show perturbed evidence but remain close in justification, while successful attacks exhibit substantial justification shifts that lead to incorrect predictions. The overall shift magnitude also aligns with attack effectiveness, with higher-ASR models (e.g., DEFAME) showing stronger deviations.

I Refinement Analysis

This section provides detailed analyses of the refinement process on the surrogate model, including refinement case distributions and justification similarity analysis. Since generated justifications reflect the reasoning and evidence retrieval behavior of AFC systems, analyzing justification shifts helps characterize how adversarial attacks alter the verification process during iterative refinement.

Refinement cases analysis. Figure 3 demonstrates the outcome distributions of successful and failed adversarial cases during optimizations. Successful cases predominantly arise from shifts or degradations in justification and evidence reasoning. In contrast, failures are largely due to strong model resistance or degraded reasoning without verdict flips, with a small fraction invalidated by semantic constraints. This suggests that effective attacks rely on manipulating justification and evidence interpretation.

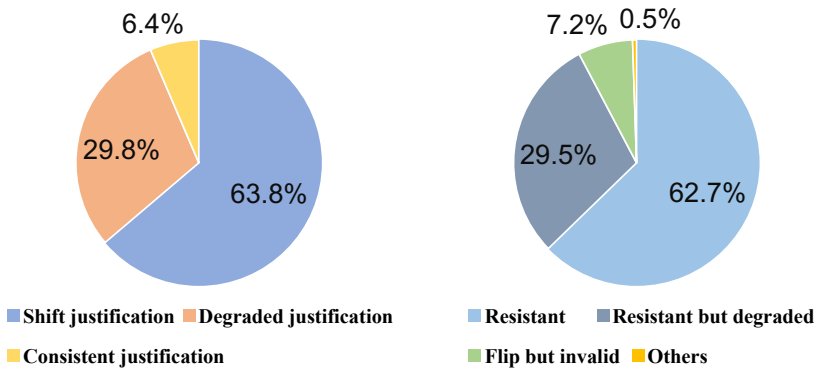

Figure 3: Outcome analysis of successful (left) and failed (right) adversarial optimizations.

Table 15: Justification similarity between success/failure claims and their original benign counterparts.

Setting	ROUGE-1	ROUGE-2	ROUGE-L	MAUVE	SummaC
Attack Success	0.470	0.257	0.324	0.385	-0.389
Attack Failure	**0.617**	**0.423**	**0.465**	**0.945**	**-0.193**

Justification analysis. We compare the similarity between justifications generated for successful/failed adversarial claims and their original benign counterparts (Table 15). Overall, justifications associated with attack failures exhibit substantially higher similarity across all metrics, indicating that unsuccessful attacks tend to preserve reasoning patterns closer to the benign case. In contrast, successful attacks produce justifications that are significantly less similar, reflecting pronounced shifts in evidence usage and reasoning structure. This gap suggests that effective adversarial claims primarily succeed by inducing meaningful deviations in the justification process, highlighting that our proposed attack strategy is a key factor underlying successful attacks.

J Effect of Refinement Rounds

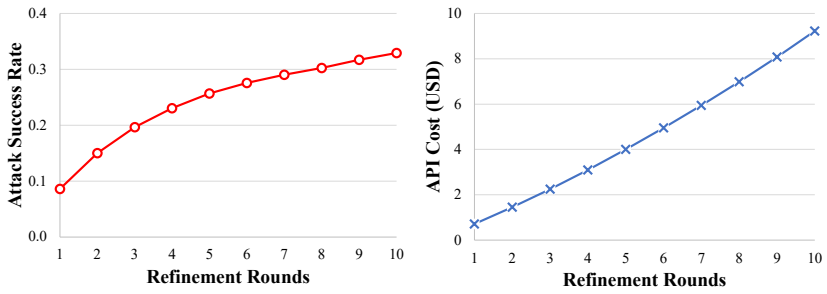

Figure 4: Impact of the maximum number of refinement rounds on ASR (left) and cost (right).

This section provides additional analysis on the impact of refinement rounds on attack performance and efficiency. Specifically, we investigate the effect of varying the number of refinement rounds from 1 to 10, keeping all other components fixed. We record the ASR and the refinement cost (100 samples per) as shown in Figure 4. Our results reveal a clear trade-off between optimization effectiveness and efficiency. Increasing the number of refinement rounds improves attack success rates, but with diminishing returns beyond a certain threshold. Excessive refinement can lead to higher computational costs and occasional over-optimization, where improvements become marginal. These findings suggest that a moderate number of refinement rounds strikes a balance between attack performance and efficiency.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the proposed multi-stage adversarial attack framework, its application to search-enabled AFC systems, and its effectiveness across multiple target models. These claims are supported by experimental results and analysis, and are scoped to the black-box setting considered in this work.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations related to computational cost introduced by LLM-based rewriting during iterative refinement. It clarifies the trade-off between effectiveness and efficiency, and situates the approach within a reasonable evaluation budget.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present theoretical analyses or formal proofs. It focuses on empirical evaluation of adversarial attacks.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides detailed descriptions of the proposed framework, including attack strategies, model roles, model configurations, dataset usage, and evaluation metrics. These details provide sufficient information to reproduce the experimental setup.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The dataset used in this work is publicly available. Given safety considerations, we will release the code for reproducing the experiments upon publication in a controlled manner to mitigate potential misuse risks, consistent with our ethical considerations.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides detailed experimental settings, including dataset construction, model roles and configurations, parameters, attack procedures, and evaluation metrics. Additional implementation details are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experiments are conducted using fixed model configurations and evaluation protocols without repeated runs. Therefore, no error bars or statistical significance tests are reported. All methods are evaluated under identical conditions, ensuring a fair comparison and minimizing variability from stochastic generation.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the computational cost of the proposed framework, including API usage and inference time, and compares it with baseline methods. All experiments are conducted under a unified setting with a fixed evaluation budget. Additional analyses, such as the trade-off between refinement rounds and efficiency, are also provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work complies with the NeurIPS Code of Ethics. The study uses publicly available datasets and does not involve human subjects or sensitive personal data. Potential risks and safety concerns are discussed in the Ethical Considerations paragraph in Section 7. Appropriate mitigation strategies are also discussed.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses both positive and negative societal impacts of the proposed approach. On the positive side, the study aims to identify vulnerabilities in search-enabled fact-checking systems, thus improving their robustness. On the negative side, the adversarial attack techniques could potentially be misused to generate misleading claims or facilitate disinformation. These risks and corresponding mitigation strategies are discussed in the Ethical Considerations paragraph in Section 7.

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: The proposed framework involves adversarial attack techniques that have potential dual-use risks. To mitigate misuse, we do not deploy the system in real-world applications or release harmful content. Code will be released in a controlled manner upon publication, with safeguards consistent with our ethical considerations.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The dataset used in this work (MOCHEG) is publicly available and properly cited. All assets are used in accordance with their intended purpose and publicly accessible terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: This work does not release new assets at submission time. The dataset used is publicly available, and the code will be released upon publication as described in the paper.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing or structured research with human subjects. All experiments are conducted using automated systems and publicly available datasets. Only limited human validation (9.4% of cases) mentioned in Appendix B is used for sanity checking without crowdsourcing or a formal human-subject study.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or structured research with human subjects. All experiments are conducted using automated systems and publicly available datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The proposed framework relies on LLMs as core components for adversarial claim generation, evaluation, and refinement. The roles of different LLM-based agents, including the Generator, Evaluator, and surrogate fact-checking system, are explicitly described in the methodology section 4.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.