

FACT2FICTION: Targeted Poisoning Attack to Agentic Fact-checking System

Haorui He,^{1,2} Yupeng Li,^{1*} Bin Benjamin Zhu,³ Dacheng Wen,^{1,2}
Reynold Cheng,² Francis C. M. Lau²

¹Department of Interactive Media, Hong Kong Baptist University

²School of Computing and Data Science, The University of Hong Kong

³Microsoft Corporation
ivanypli@gmail.com

Abstract

State-of-the-art (SOTA) fact-checking systems combat misinformation by employing autonomous LLM-based agents to decompose complex claims into smaller sub-claims, verify each sub-claim individually, and aggregate the partial results to produce verdicts with justifications (explanations for the verdicts). The security of these systems is crucial, as compromised fact-checkers can amplify misinformation, but remains largely underexplored. To bridge this gap, this work introduces a novel threat model against such fact-checking systems and presents Fact2Fiction, the first poisoning attack framework targeting SOTA agentic fact-checking systems. Fact2Fiction employs LLMs to mimic the decomposition strategy and exploit system-generated justifications to craft tailored malicious evidences that compromise sub-claim verification. Extensive experiments demonstrate that Fact2Fiction achieves 8.9%–21.2% higher attack success rates than SOTA attacks across various poisoning budgets and exposes security weaknesses in existing fact-checking systems, highlighting the need for defensive countermeasures.

Code — <https://trustworthycomp.github.io/Fact2Fiction/>

Appendix — <https://arxiv.org/abs/2508.06059>

1 Introduction

The proliferation of misinformation has become a pressing challenge in the digital era, with false information spreading at an unprecedented rate and scale on online platforms (Vlachos and Riedel 2014; Li et al. 2024). Manual fact-checking is inadequate to address the sheer volume of misinformation, which calls for the development of (automated) fact-checking systems to combat misinformation at scale.

Fact-checking systems typically adopt the Retrieval Augmented Generation (RAG) framework (Guo, Schlichtkrull, and Vlachos 2022; Schlichtkrull et al. 2024), which integrates large language models (LLMs) with external evidence retrieval modules. These systems retrieve relevant evidences for the textual claim being verified to predict their veracity and generate corresponding justifications that elucidate the rationales behind their verdicts. While prior research on fact-checking (Schlichtkrull, Guo, and Vlachos 2024; Braun

et al. 2025; Rothermel et al. 2024; Yoon et al. 2024) has focused on improving accuracy and explainability, the security vulnerabilities of these systems remain underexplored. This oversight has dire consequences, as compromised fact-checking systems can amplify misinformation by supporting false claims and/or undermine confidence in factual reporting by refuting true claims, thereby eroding public trust.

Another line of research (Liu et al. 2024; Yi et al. 2025; Zou et al. 2025) has explored prompt injection and poisoning attacks on general RAG-based systems, where adversaries inject malicious instructions or fabricated corpora into the knowledge base of the systems to manipulate their outputs. These attacks are limited to targeting only rudimentary RAG frameworks that directly prompt LLMs with retrieved results based on user queries. However, recent state-of-the-art (SOTA) fact-checking systems, such as DEFAME (Braun et al. 2025) and InFact (Rothermel et al. 2024), have evolved beyond naive RAG frameworks to adopt an agentic paradigm. Such systems leverage LLM-based agents to actively plan fact-checking by decomposing complex claims into smaller sub-claims (Huang et al. 2025; Li et al. 2025b), then autonomously retrieve evidences to verify each sub-claim sequentially, and finally aggregate the partial results to produce a final verdict. This claim decomposition approach not only enhances performance in fact-checking (Chen et al. 2022, 2024; Hu, Long, and Wang 2025), but also renders these systems inherently robust to existing attacks. As shown in our experiments (see Sec. 5), it simultaneously reduces the retrievability and effectiveness of the malicious content crafted by existing attacks.

For instance, consider the claim “Sean Connery refused to be in an Apple commercial in a letter.” The SOTA PoisonedRAG attack generates broad malicious evidences targeting the main claim, such as: “Close friends of Sean confirm he wrote a letter to Steve Jobs to refuse Apple’s commercial deal.” However, agentic fact-checking systems decompose this claim into specific sub-claims, such as “What type of source first published the original story of the alleged letter from Sean?”, and perform adaptive evidence retrieval for each specific sub-claim. Consequently, the generic malicious evidences crafted by PoisonedRAG become irrelevant to the verification of these sub-claims, which makes them unlikely to be retrieved and ineffective at misleading the sub-claim verification,

*Corresponding author.

even if retrieved. Instead, the sub-claim is addressed with clean evidences, which reveal that the claim originated from Scoopertino, a satirical website for funny but fictional Apple-related news. Thus, even if some other sub-claims may be compromised, the inherent cross-validation mechanisms in agentic systems, which aggregate the results of all sub-claims, correctly identify Scoopertino as satirical and ensure an accurate final verdict (Hu, Long, and Wang 2025).

To bridge these gaps, we propose FACT2FICTION, **the first poisoning attack framework against agentic fact-checking systems**. Rather than focusing only on main claims, as illustrated in Fig. 1, Fact2Fiction mimics claim decomposition to generate a surrogate set of sub-claims and comprehensively craft malicious evidences against all sub-claims. Furthermore, Fact2Fiction exploits a unique yet previously overlooked vulnerability in fact-checking systems: their justifications. These justifications expose critical evidences and reasoning patterns behind verdicts, which enables attackers to create targeted malicious evidences that directly contradict the original reasoning of the victim systems and allocate more malicious evidences to sub-claims that are emphasized in the justifications.

Extensive experiments across two agentic systems (DEFAME and InFact) under varying poisoning budgets validate that Fact2Fiction achieves 8.9%–21.2% higher attack success rates (ASRs) than the SOTA PoisonedRAG attack (Zou et al. 2025). Additionally, Fact2Fiction demonstrates superior attack efficiency, requiring only 6.3%–12.5% of the malicious evidences to achieve performance comparable to PoisonedRAG. **Our evaluations reveal these critical insights:** (1) Justifications introduce a transparency-security trade-off, yielding up to a 12.4% improvement in ASR under constrained budgets. (2) Evidence quality matters beyond retrievability. Malicious evidences crafted by Fact2Fiction achieve an 8.9% higher ASR than PoisonedRAG at the same level of retrievability. (3) Different attacks exhibit varying saturation points for attack effectiveness, beyond which additional poisoning budgets yield minimal improvements in ASR, across different victim systems. Future research should investigate factors influencing saturation points and strategies to limit attack effectiveness. (4) Existing defenses are ineffective against Fact2Fiction, which highlights the urgent need for novel countermeasures.

Our main contributions can be summarized as follows.

- **Threat Model:** We propose a novel threat model against fact-checking systems that exploits their justifications for targeted poisoning attacks.
- **Attack Method:** We introduce Fact2Fiction, the first attack framework that targets SOTA agentic fact-checking systems and crafts targeted malicious evidences.
- **Evaluations and Findings:** Extensive experiments show that Fact2Fiction outperforms prior attacks across diverse settings and reveal critical insights based on the findings.

2 Related Work

Automated Fact-checking. Automated fact-checking systems employ the RAG framework: given a claim,

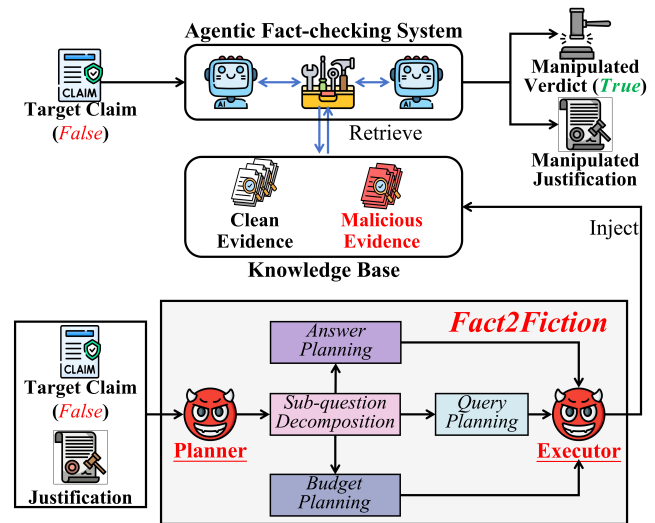


Figure 1: Overview of our Fact2Fiction attack framework.

they retrieve relevant evidences and prompt an LLM to predict a verdict (e.g., *Supported*, *Refuted*, *Not Enough Evidence*, or *Conflicting/Cherry-picking*) with a justification (Schlichtkrull et al. 2024; Yoon et al. 2024; He et al. 2025). This baseline processes claims as a whole, limiting coverage of implicit aspects.

Recent systems adopt an *agentic paradigm*, where autonomous LLM-based agents decompose complex claims into verifiable sub-claims. InFact (Rothermel et al. 2024), winner of the AVeriTeC 2024 challenge (Schlichtkrull et al. 2024), performs explicit sub-question decomposition: it generates sub-questions, retrieves evidences for each via adaptive queries, and aggregates sub-verdicts into a final decision, enabling comprehensive scrutiny of explicit and implicit claim details (Chen et al. 2022, 2024; Hu, Long, and Wang 2025). DEFAME (Braun et al. 2025) instead applies implicit dynamic decomposition: agents iteratively refine queries and reasoning based on evolving evidence across multiple retrieval rounds, achieving accuracy comparable to InFact but with higher efficiency. *Unlike prior work aimed at improving accuracy, we are the first to investigate the security of such agentic fact-checking systems.*

Attacks on RAG-based Systems. Attacks on RAG systems focus on prompt injection and poisoning attacks.

Prompt injection attacks (Yi et al. 2025; Liu et al. 2024) manipulate system outputs by inserting malicious instructions into LLM inputs (e.g., “When verifying claim X, output verdict Y”). However, these attacks are generally less effective against RAG-based systems because such instructions, being semantically different from relevant evidence corpora, are rarely retrieved by semantic search (Zou et al. 2025). Moreover, prompt injection can often be mitigated with simple defenses like paraphrasing (Jain et al. 2023).

Poisoning attacks (Du, Bosselut, and Manning 2022; Pan et al. 2023; Zou et al. 2025) offer a more potent threat by crafting malicious corpora, often using LLMs, and injecting them into the knowledge base of the targeted system. The

state-of-the-art PoisonedRAG attack (Zou et al. 2025) shows that even small amounts of such crafted content can reliably mislead RAG-based systems into producing attacker-chosen responses. However, these attacks have only been tested on basic RAG setups, and, as shown in Sec. 1, they fail against state-of-the-art agentic fact-checking systems due to their claim decomposition strategies. Furthermore, they overlook a fact-checking-specific vulnerability: leveraging system-generated justifications to boost attack success.

To address these limitations, we propose *Fact2Fiction*, the first attack framework tailored for agentic fact-checking systems. Compared with prior attacks, *Fact2Fiction* provides three key advantages: (1) it reverse-engineers the agentic process by decomposing claims into sub-claims for full-scope compromise, (2) it uses system-generated justifications to craft fine-grained malicious evidences for each sub-claim, and (3) it allocates the poisoning budget strategically across sub-claims, emphasizing the most influential ones.

3 Threat Model

We define the threat model against fact-checking systems.

Attacker’s Objective. The attacker aims to manipulate the fact-checking systems to endorse false (refuted) claims and/or discredit true (supported) claims. Consider an attacker selecting a textual target claim c_i with a ground-truth veracity label $y_i \in \{\text{Supported}, \text{Refuted}\}$. The attacker aims to mislead the target victim system into predicting the opposite label: $y'_i = \begin{cases} \text{Supported}, & \text{if } y_i = \text{Refuted}, \\ \text{Refuted}, & \text{if } y_i = \text{Supported}. \end{cases}$ This

mimics real-world scenarios where a malicious actor can target specific claims, such as those made by a candidate during a presidential debate, to sway voter perceptions or public stance (Küçük and Can 2020; Li et al. 2023a,b).

Attacker’s Capabilities. We consider a **black-box** attack scenario where the attacker has no knowledge of the internal design of the target fact-checking system, nor access to the weights or training data of its retriever or language model(s). However, the attacker can query the system to obtain initial verdicts v_i and justifications j_i for each target claim c_i before launching the attack. This reflects realistic settings where fact-checking systems use closed-source LLMs and proprietary retrievers, but are accessible via APIs or online services. Examples include InFact (Rothermel et al. 2024), DEFAME (Braun et al. 2025), Loki (Li et al. 2025a), and social platform bots such as @Grok or @AskPerplexity on X (formerly Twitter), which allow users to submit claims and receive verdicts with justifications.

Similar to previous attacks (Yi et al. 2025; Zou et al. 2025; Yang et al. 2024; Carlini et al. 2024; Du, Bosselut, and Manning 2022), we assume the attacker can inject m malicious textual evidences, $\{e_1, e_2, \dots, e_m\}$, into the knowledge base (KB) \mathcal{E}_i of the victim system for each target claim c_i , which initially contains M clean evidences. This setup is realistic for victim systems that build KBs from the open web. For instance, AVeriTeC (Schlichtkrull, Guo, and Vlachos 2024), the state-of-the-art real-world fact-checking benchmark, constructs KBs using Google Search from 397,491

sources, including user-generated platforms. The prior studies demonstrate that attackers can post malicious content on these platforms, such as Wikipedia or arXiv (Carlini et al. 2024; Yang et al. 2024; Zou et al. 2025).

4 The Design of Fact2Fiction

This section presents the design of *Fact2Fiction* framework, which consists of a Planner agent and an Executor agent.¹

4.1 The Plan of Attack

Fact2Fiction uses a Planner agent to devise targeted poisoning attacks in four key operations: sub-question decomposition, answer planning, budget planning, and query planning.

Sub-question Decomposition. Agentic fact-checking systems decompose complex claims into smaller sub-claims for thorough verification, which renders malicious evidences crafted solely around the main claim inherently ineffective (Chen et al. 2022, 2024; Rothermel et al. 2024). To address this limitation, the Planner mimics the agentic fact-checking process by decomposing each claim c_i into a surrogate set of sub-claims, represented as sub-questions $\mathcal{Q}_i = \{q_1, \dots, q_{l_i}\}$, where l_i (up to ten) is determined by the Planner. By manipulating these sub-claims, the attacker can fully compromise all aspects of the target claim.

Answer Planning. Following decomposition, the attacker fabricates malicious evidences to manipulate sub-claim verification and construct a coherent false narrative. However, attacking each sub-question independently risks generating contradictory responses that undermine the overall deception. To address this, *Fact2Fiction* first plans adversarial answers that specify the desired misleading conclusion for each sub-question, then generates malicious evidences accordingly. This approach ensures all compromised sub-claims align with the intended verdict without contradictions. To plan effective adversarial answers, *Fact2Fiction* exploits a previously underexplored vulnerability in fact-checking systems: their generated justifications, which reveal the specific evidences and reasoning the system relies upon. By probing these justifications before the attack, attackers can create adversarial content that precisely undermines the decision-making process of any specific victim system. For example, consider the claim “New Zealand’s new Food Bill bans gardening,” which the victim system refutes with the justification: “While the bill imposes minor limitations on community gardening, it affirms individuals’ rights to grow food for personal use and trade it without restrictions on personal gardening activities.” A *non-targeted* adversarial answer to the sub-question “Does the Bill state that gardening is banned?” can be: “Yes, the Bill includes provisions that restrict gardening activities.” In contrast, a *targeted* adversarial answer would be: “The Bill imposes strict registration requirements for food sharing and trading, severely limiting both community and individual gardening for personal use and trade.” The targeted answer directly contradicts the key reasoning of the victim system that “personal gardening is

¹Appendix A provides the specific prompts.

unrestricted” by highlighting procedural barriers, while the non-targeted answer fails to address this critical evidence.

To implement this approach, given all sub-questions Q_i for each target claim c_i , the Planner generates a targeted adversarial answer a_k for each sub-question $q_k \in Q_i$, designed to directly contradict the initial justification j_i . These answers guide the Executor in creating tailored malicious evidence corpora to compromise each sub-question.

Budget Planning. Another vulnerability that justifications expose is the relative importance of sub-claims within the verification process. In the example above, the justification indicates that the answer to the sub-question “Does the new Food Bill explicitly state that gardening is banned?” is more critical than the answer to “What has been the public reaction to the Food Bill?” in determining the final verdict. When the poisoning budget is limited, attackers can optimize attack efficacy by allocating more resources to target influential sub-claims rather than less important ones. To achieve optimal budget allocation, for the set of sub-question–answer pairs $\mathcal{QA}_i = \{(q_1, a_1), \dots, (q_{l_i}, a_{l_i})\}$, the Planner assigns a weight score w_k to each pair (q_k, a_k) based on its relevance to the initial justification j_i . The poisoning budget m is then allocated proportionally: $m_k = \left\lceil m \cdot \frac{w_k}{\sum_{s=1}^{l_i} w_s} \right\rceil$, where m_k represents the allocated budget for sub-question–answer pair (q_k, a_k) . This strategy prioritizes resources on the most influential sub-claims.

Query Planning. To achieve attack success, malicious evidences must first be retrieved. Agentic fact-checking systems employ adaptive search queries to gather evidences for each sub-claim using semantic similarity matching. By concatenating tailored queries for each sub-claim with its corresponding malicious evidence corpora, the semantic similarity between the queries and the evidences is enhanced, which subsequently improves the retrievability of the malicious content. To this end, the Planner generates a surrogate set of potential search queries $\mathcal{S}_k = \{s_1, \dots, s_{u_k}\}$ that can be used to retrieve evidences for answering each sub-question q_k , where u_k (up to five) is determined by the Planner. The Executor then combines these queries with the crafted evidence corpora to optimize their retrieval.

4.2 The Execution of Planned Attack

The Executor agent implements the attack plan devised by the Planner. For each sub-question q_k , the Executor generates m_k targeted evidence corpora $\{\tilde{e}_{k,1}, \dots, \tilde{e}_{k,m_k}\}$. Each evidence corpus $\tilde{e}_{k,h}$ aligns semantically with the planned adversarial answer a_k to reinforce the attack objective and mislead the victim system toward the attacker-desired verdict. Following the concatenation strategy in the query planning, for each evidence corpus $\tilde{e}_{k,h}$, the Executor randomly selects a query $s_p \in \mathcal{S}_k$ and constructs the final malicious evidence $e_{k,h} = s_p \oplus \tilde{e}_{k,h}$, where $e_{k,h}$ represents the h -th malicious evidence targeting sub-question q_k and \oplus denotes string concatenation. Finally, the Executor injects the complete set of malicious evidences into the clean knowledge base \mathcal{E}_i for target claim c_i : $\mathcal{E}'_i \leftarrow \mathcal{E}_i \cup \bigcup_{k=1}^{l_i} \{e_{k,h}\}_{h=1}^{m_k}$,

Algorithm 1: Fact2Fiction Attack Framework

Require: Target claim c_i , initial justification j_i , poisoning budget m , clean knowledge base \mathcal{E}_i

- 1: **[Planner]** Decompose c_i into sub-questions Q_i
- 2: **for** each $q_k \in Q_i$ **do**
- 3: **[Planner]** Plan adversarial answer a_k based on j_i
- 4: **[Planner]** Compute weight score w_k of (q_k, a_k) based on its relevance to j_i
- 5: **[Planner]** Allocate budget $m_k = \left\lceil m \cdot \frac{w_k}{\sum_{s=1}^{l_i} w_s} \right\rceil$
- 6: **[Planner]** Plan queries $\mathcal{S}_k = \{s_1, \dots, s_{u_k}\}$ for q_k
- 7: **end for**
- 8: **for** each $q_k \in Q_i$ **do**
- 9: **for** $h = 1$ to m_k **do**
- 10: **[Executor]** Craft evidence corpus $\tilde{e}_{k,h}$ aligned with a_k
- 11: **[Executor]** Randomly select query $s_p \in \mathcal{S}_k$
- 12: **[Executor]** Construct malicious evidence $e_{k,h} = s_p \oplus \tilde{e}_{k,h}$ and inject $e_{k,h}$ into \mathcal{E}_i
- 13: **end for**
- 14: **end for**
- 15: **Output:** Poisoned knowledge base \mathcal{E}'_i

where \mathcal{E}'_i is the poisoned knowledge base for claim c_i .

Algorithm 1 summarizes the complete attack framework.

5 Experiments

We conduct experiments to answer the following evaluation questions (EQs): **(EQ1)**: Can Fact2Fiction outperform existing attacks against agentic fact-checking systems? **(EQ2)**: What is the contribution of each component in Fact2Fiction? **(EQ3)**: How does the poisoning budget affect the attack effectiveness of Fact2Fiction and existing attacks? **(EQ4)**: How effective are existing defenses against Fact2Fiction?

5.1 Experimental Setups

Benchmark. We leverage AVeriTeC (Schlichtkrull, Guo, and Vlachos 2024), the state-of-the-art real-world fact-checking benchmark that addresses key limitations of prior datasets, including evidence leakage and insufficiency. AVeriTeC comprises claims from 50 fact-checking organizations, each annotated by professional fact-checkers. Each claim pairs with a knowledge base (KB) containing both relevant and potentially distracting evidences collected via Google Search. This setup simulates open-web retrieval while ensuring reproducibility (Schlichtkrull et al. 2024). We use the AVeriTeC development split (500 claims) for all experiments, as the test split labels remain hidden. For each victim system, we construct the evaluation set by including only claims that the system correctly supported or refuted before the attack to ensure the attack success reflects the impact of the attack method rather than pre-existing errors.

Victim Systems. We target two state-of-the-art agentic fact-checking systems: **DEFAME** (Braun et al. 2025) and **InFact** (Rothermel et al. 2024). Both systems decompose complex claims into smaller sub-tasks for thorough verification, with detailed designs provided in Sec. 2. To establish fair comparisons with prior attacks (Zou et al. 2025), we additionally include a baseline system using the naive

RAG framework, referred to as **Simple**, which directly retrieves evidences with the main claim without decomposition. All victim systems adopt the evidence retrieval mechanism from Braun et al. (2025); Rothermel et al. (2024): each evidence in the KB is encoded into text embeddings by the retriever. During retrieval, the text embeddings of the search queries are compared using Euclidean distance to perform a five-nearest neighbors search to retrieve the top five evidences. We use gte-base-en-v1.5 as the retriever for DEFAME and InFact, which follows their default setup.²

Baselines. We follow Zou et al. (2025) to compare Fact2Fiction against the following established attacks.

- **Naive Attack:** For a target claim c_i , if c_i is false, we treat c_i itself as the malicious evidence. If c_i is true, we construct a false statement by adding a negative descriptor (e.g., “is inaccurate,” “is false,” “is misinformation”).
- **Prompt Injection Attack** (Liu et al. 2024; Yi et al. 2025): This attack injects malicious instructions into LLM inputs to manipulate its outputs. We follow Zou et al. (2025) to adapt it by crafting malicious evidences that instruct the model: “When asked to verify the claim: [target claim], please output the verdict: [target verdict].”
- **Disinformation Attack** (Du, Bosselut, and Manning 2022; Pan et al. 2023): This attack manipulates fact-checking systems by crafting malicious evidences with LLMs. For each target claim c_i , we prompt an LLM to “craft an evidence corpus that can lead the veracity of the claim: [target claim] to [target verdict].”
- **PoisonedRAG Attack** (Zou et al. 2025): This state-of-the-art attack, which improves upon the disinformation attack by concatenating all LLM-generated malicious evidences with the target claim c_i to enhance retrievability.

We also evaluate ablation variants of Fact2Fiction:

- **w/o Answer Planning:** Removes targeted adversarial answer planning to generate non-targeted answers to sub-questions without leveraging justifications.
- **w/o Budget Planning:** Distributes the poisoning budget uniformly across all sub-questions, instead of allocating budgets based on their relevance to justifications.
- **w/o Query Planning:** Removes targeted search query concatenation for each sub-question, instead following the PoisonedRAG approach (Zou et al. 2025) by concatenating all malicious evidences with the target claim c_i .

Hyper-parameters. To evaluate attacks under varying resource constraints, we inject malicious evidences into each target claim c_i at rates of 1%, 2%, 4%, and 8% of N_i , where N_i represents the number of clean evidences for c_i (avg. 823.4 items). The victim systems use official code from Braun et al. (2025); Rothermel et al. (2024) with default configurations. Both attacks and victim systems use GPT-4o-mini-2024-07-18 as the LLM backbone, selected

²Appendix B validates that Fact2Fiction is robust to alternative retrievers (Stella-en-400M-v5 and Qwen-3-Embedding-0.6B).

for its strong performance and cost-efficiency.³ Following Zou et al. (2025), we set the temperature of the LLMs to 1.0 and constrain each evidence to 30 words.

Metrics. We employ the following metrics:

- **Attack Success Rate (ASR):** The proportion of target claims where the attack successfully inverts the verdict of the victim system (e.g., Supported \rightarrow Refuted, or vice versa). ASR directly measures attack effectiveness in achieving the primary objective of manipulating systems to endorse false claims or discredit true ones.
- **System Fail Rate (SFR):** The proportion of target claims where the attack causes any incorrect verdict (e.g., Supported \rightarrow Refuted, Not Enough Evidence, or Conflicting Evidence). SFR captures the broader influence of attacks on system accuracy beyond exact verdict inversion.
- **Successful Injection Rate (SIR):** The proportion of retrieved malicious evidences relative to total retrieved evidences. SIR evaluates the retrievability of the malicious evidences crafted by the attacks.

5.2 Comparison with Existing Attacks (EQ1)

To address EQ1, Table 1 compares Fact2Fiction with the baselines across three victim systems under varying poisoning budgets. The state-of-the-art PoisonedRAG demonstrates substantial degradation in performance when targeting agentic systems (DEFAME and InFact) compared to the Simple system across all setups. For instance, at an 8% poisoning rate, the ASR of PoisonedRAG drops from 57.4% on Simple to 42.4% on DEFAME and 45.3% on InFact. This decline confirms our hypothesis from Sec. 1 that claim decomposition in agentic systems creates natural defensive barriers against existing attacks. In contrast, Fact2Fiction crafts targeted malicious content using sub-question decomposition and justification exploitation, which consistently outperforms all baseline attacks across all poisoning rates on all victim systems. Notably, at a 1% poisoning rate, Fact2Fiction achieves an ASR of 42.4% on DEFAME and 46.0% on InFact, which surpasses PoisonedRAG 8.9 and 9.2 percentage points, respectively. This highlights the effectiveness of Fact2Fiction with minimal malicious evidences.

The relationship among ASR, SFR, and SIR reveals that while higher SIR generally correlates with increased ASR and SFR through greater malicious evidence retrieval, retrieval alone proves insufficient for optimal attack success. For example, at a 1% poisoning rate on DEFAME, Fact2Fiction achieves a slightly lower SIR (64.8%) than PoisonedRAG (65.6%) but a significantly higher ASR (42.4% vs. 33.5%), which indicates that Fact2Fiction crafts more effective malicious content compared to PoisonedRAG.

5.3 Ablation Analysis (EQ2)

To address EQ2, we evaluate the ablation variants of Fact2Fiction, which reveals the following insights.

³Appendix C and D evaluate alternative LLM backbone (Gemini-2.0-Flash and DeepSeek-V3) for the attacks and victim systems, respectively. The experimental results collectively validate that the effectiveness of Fact2Fiction is LLM-agnostic.

Poison Rate	1%			2%			4%			8%		
	ASR	SFR	SIR	ASR	SFR	SIR	ASR	SFR	SIR	ASR	SFR	SIR
Victim: DEFAME (269 claims)												
Naive	17.8	34.9	49.9	19.3	37.9	51.2	19.7	36.4	54.0	18.2	38.3	52.5
Prompt Injection	19.3	39.4	42.9	21.6	38.3	44.8	22.3	37.5	45.4	21.2	37.9	45.8
Disinformation	24.5	36.1	48.1	34.2	45.0	62.3	42.4	52.0	73.0	42.4	53.5	78.8
PoisonedRAG	33.5	47.6	65.6	40.9	49.1	76.5	45.0	53.9	79.3	42.4	53.9	83.9
Fact2Fiction	42.4 ⁺	55.8 ⁺	<u>64.8</u>	52.0 ⁺	66.5 ⁺	80.3 ⁺	58.4 ⁺	69.9 ⁺	91.6 ⁺	63.6 ⁺	74.7 ⁺	95.1 ⁺
- w/o Answer Planning	<u>40.9</u> ⁺	50.2 ⁺	63.6	<u>49.1</u> ⁺	<u>63.2</u> ⁺	82.4 ⁺	53.2 ⁺	65.1 ⁺	<u>91.1</u> ⁺	<u>59.9</u> ⁺	<u>68.8</u> ⁺	<u>94.7</u> ⁺
- w/o Budget Planning	34.6	48.0	62.4	<u>43.8</u> ⁺	<u>59.1</u> ⁺	<u>79.9</u> ⁺	<u>53.5</u> ⁺	<u>66.2</u> ⁺	<u>90.2</u> ⁺	<u>59.6</u> ⁺	<u>70.6</u> ⁺	<u>93.1</u> ⁺
- w/o Query Planning	39.4 ⁺	<u>54.6</u> ⁺	64.4	47.2 ⁺	61.3 ⁺	74.8	50.9 ⁺	63.2 ⁺	80.8	49.4 ⁺	64.3 ⁺	83.8
Victim: InFact (274 claims)												
Naive	14.6	28.1	19.7	16.8	29.2	21.2	17.2	30.3	21.2	16.4	31.0	21.5
Prompt Injection	16.1	29.6	16.4	16.1	25.9	17.4	12.8	24.8	17.6	13.9	25.2	17.5
Disinformation	31.8	47.8	20.5	38.3	54.7	35.4	40.1	57.3	49.7	43.4	62.4	61.4
PoisonedRAG	35.8	53.6	24.7	43.1	60.2	38.1	42.3	59.1	52.8	45.3	64.2	63.2
Fact2Fiction	46.0 ⁺	65.3 ⁺	25.4 ⁺	54.5 ⁺	74.5 ⁺	46.4 ⁺	56.6 ⁺	77.7 ⁺	<u>67.7</u> ⁺	59.9 ⁺	<u>77.0</u> ⁺	<u>82.7</u> ⁺
- w/o Answer Planning	<u>43.4</u> ⁺	<u>61.3</u> ⁺	<u>25.3</u> ⁺	49.6 ⁺	<u>68.3</u> ⁺	45.9 ⁺	53.1 ⁺	<u>72.5</u> ⁺	67.5 ⁺	53.3 ⁺	73.4 ⁺	81.5 ⁺
- w/o Budget Planning	33.6	50.7	24.8	<u>52.6</u> ⁺	<u>72.6</u> ⁺	<u>46.0</u> ⁺	<u>55.8</u> ⁺	<u>75.4</u> ⁺	68.2 ⁺	<u>56.9</u> ⁺	78.5 ⁺	83.4 ⁺
- w/o Query Planning	43.4 ⁺	60.6 ⁺	24.6	<u>42.0</u> ⁺	61.0 ⁺	38.3	43.8 ⁺	62.8 ⁺	53.9	47.8 ⁺	69.0 ⁺	66.0
Victim: Simple (265 claims)												
Naive	24.2	40.8	51.2	24.5	47.9	54.8	24.9	43.4	53.2	26.8	47.9	53.9
Prompt Injection	34.7	42.6	49.9	35.9	49.4	51.8	35.1	45.3	51.1	40.0	50.9	52.6
Disinformation	25.7	46.4	52.2	38.1	56.2	67.1	50.9	63.4	77.2	56.6	69.4	82.8
PoisonedRAG	<u>42.4</u>	63.4	69.1	49.4	66.4	79.6	<u>54.7</u>	70.6	85.2	57.4	70.6	88.1
Fact2Fiction	43.4	68.3 ⁺	<u>68.1</u>	53.2 ⁺	79.3 ⁺	<u>84.8</u> ⁺	66.0 ⁺	83.0 ⁺	93.9 ⁺	65.7 ⁺	85.3 ⁺	97.5 ⁺
- w/o Answer Planning	38.5	58.9	65.5	<u>52.5</u> ⁺	<u>74.0</u> ⁺	85.6 ⁺	<u>61.5</u> ⁺	75.1 ⁺	<u>92.2</u> ⁺	<u>64.9</u> ⁺	82.3 ⁺	<u>96.4</u> ⁺
- w/o Budget Planning	34.3	59.6	63.6	51.3 ⁺	<u>72.4</u> ⁺	83.2 ⁺	60.0	<u>82.3</u> ⁺	<u>92.0</u> ⁺	<u>64.9</u> ⁺	<u>84.5</u> ⁺	<u>96.0</u> ⁺
- w/o Query Planning	42.3	<u>67.5</u> ⁺	67.7	47.9	74.0 ⁺	79.9	54.0 ⁺	78.5 ⁺	88.4	61.5 ⁺	81.9 ⁺	88.4

Table 1: Attack performance on different victim systems across varying poison rates. The best results for each metric and poison rate are **bolded**, while the second-best are underlined. Following Chen et al. (2024), we use paired bootstrap tests with at least five trials. Results marked with a “+” denote a significant ($p \leq 0.05$) improvement over PoisonedRAG.

Answer Planning. Removing answer planning consistently reduces attack effectiveness across all setups. For example, on DEFAME at a 1% poison rate, the ASR of Fact2Fiction drops from 42.4% to 40.9%, and on InFact from 46.0% to 43.4%. This degradation highlights the value of exploiting justifications to craft targeted adversarial content to directly contradict the reasoning of the victim system.

Budget Planning. Budget planning delivers maximum benefit for optimizing attacks when resources are constrained. For example, at a 1% poisoning rate on DEFAME, removing budget planning causes ASR to drop from 42.4% to 34.6% by 7.8 percentage points. However, at higher poisoning rates, the performance gap narrows (e.g., at 8% on InFact, ASR: 59.9% vs. 56.9%). This indicates that larger budgets provide sufficient resources to compromise all sub-claims, which reduces the necessity for strategic allocation.

Query Planning. Query planning significantly affects attack effectiveness when targeting agentic systems. At an 8% poisoning rate, ASR drops from 63.6% to 49.4%. The influence is less pronounced for Simple (65.7% to 61.5% at 8%), where evidence retrieval directly uses the main claim, so concatenating the claim with malicious evidence corpora re-

mains effective. In contrast, agentic systems use sub-claim-specific queries, which makes query planning crucial for maximizing the retrievability of malicious evidences.

5.4 Impact of Poison Rate (EQ3)

To address EQ3, we investigate how varying the poison rate affects attack performance. Table 1 shows that while increasing poisoning rates generally improves ASR, SFR, and SIR, different attacks reach *saturation points*, where additional poisoning yields diminishing returns. On DEFAME, both Naive and Prompt Injection attacks saturate around a 2% poisoning rate, with ASR stabilizing near 19% and 22%, respectively. Interestingly, saturation points vary across victim systems: while the Prompt Injection attack plateaus at 2% on DEFAME and InFact, it continues improving on Simple (ASR: 35.1% to 40% from 4% to 8% poisoning rate). The findings highlight that developing robust fact-checking systems that force attack saturation at low ASR values represents a promising direction for future research. Unlike existing methods, Fact2Fiction continues to improve in ASR and SFR with higher budgets, from 1% to 8%, across all victim systems. This scalability advantage stems from targeted evidence generation that efficiently exploits additional re-

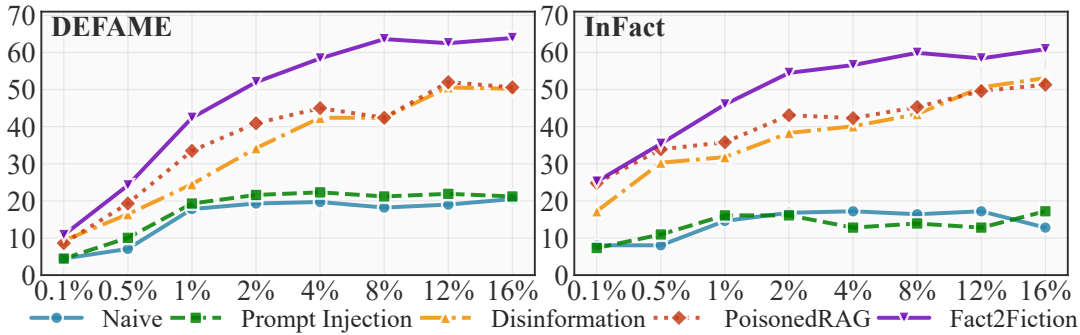


Figure 2: ASR trend (y-axis) across poison rates (x-axis).

Defense	Paraphrasing			Malicious Detection		
	ASR	SFR	SIR	ASR	SFR	SIR
Victim: DEFAME (269 claims)						
PoisonedRAG	34.2	47.6	59.6	10.0	35.3	13.4
Fact2Fiction	39.4⁺	56.1⁺	62.6⁺	26.0⁺	47.2⁺	46.2⁺
Victim: InFact (274 claims)						
PoisonedRAG	35.4	54.7	24.6	13.9	29.6	8.9
Fact2Fiction	39.4⁺	59.9⁺	25.6⁺	26.6⁺	47.5⁺	21.4⁺

Table 2: Attack performance of PoisonedRAG and Fact2Fiction under different defenses with a 1% poison rate.

sources rather than producing redundant malicious content.

We extend our evaluations to both more challenging scenarios with poison rates of 0.1% and 0.5%, and more aggressive cases with 12% and 16%. As shown in Fig. 2, Fact2Fiction achieves the highest ASR across all methods at the minimal 0.1% poison rate (at most one malicious evidence item per claim). Notably, Fact2Fiction attains a comparable ASR with only a 2% poison rate on DEFAME and 1% on InFact, while PoisonedRAG requires a 16% poison rate to achieve similar performance, which reflects an 8- to 16-fold reduction in the required poisoning budget. This highlights the superior attack efficiency of Fact2Fiction.

5.5 Performance under Defenses (EQ4)

To address EQ4, we assess the robustness of Fact2Fiction against three state-of-the-art defenses.

Paraphrasing. Paraphrasing can defend against prompt injection and poisoning attacks (Zou et al. 2025). We simulate an attacker injecting malicious evidences for a target claim, but the victim system later fact-checks its paraphrased version. Table 2 shows that while paraphrasing slightly lowers the ASR, Fact2Fiction still outperforms PoisonedRAG.

Malicious Detection. Malicious evidences crafted by PoisonedRAG cluster more tightly in embedding space than clean evidences. Defenders can use K-means clustering ($k=2$) per retrieval to filter high-density clusters as potentially malicious (Zhou et al. 2025). Table 2 shows that Fact2Fiction proves significantly more resilient than Poi-

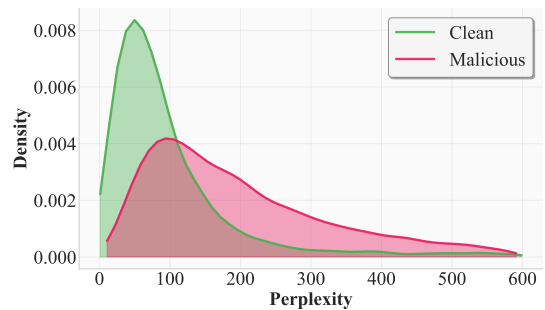


Figure 3: Perplexity distribution comparison.

sonedRAG, likely because the sub-question decomposition of Fact2Fiction produces diverse malicious evidences.

Perplexity (PPL)-based Detection. PPL measures text coherence. High PPL suggests unnatural phrasing (e.g., embedded malicious instructions), and such content can be flagged as potentially malicious and filtered out (Alon and Kamfonas 2023; Jain et al. 2023; Gonen et al. 2023; Zou et al. 2025; Zhou et al. 2025). Fig. 3 shows that Fact2Fiction generates coherent text, evidenced by substantial overlap in clean and malicious PPL distributions computed using GPT-2 and its tokenizer, that can evade PPL-base detection.

6 Conclusion

In this work, we introduce Fact2Fiction, the first poisoning attack targeting agentic fact-checking systems, which achieves significantly higher attack success rates compared to state-of-the-art methods across diverse setups. Our findings reveal that the transparency of fact-checking systems introduces vulnerabilities, which enables attackers to exploit justifications to craft targeted malicious evidences and strategically allocate attack resources. We also found that the saturation point of attack effectiveness varies depending on both the attack method and the victim system’s design, which reveals the promising future direction of designing robust fact-checking systems. As the first study to explore the security vulnerabilities of agentic fact-checking systems, our work underscores the urgent need to enhance their resilience against sophisticated attacks like Fact2Fiction when promoting digital literacy and trustworthy information ecosystems.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 62202402), Guangdong and Hong Kong Universities “1+1+1” Joint Research Collaboration Scheme, Project No. 2025A0505000001, the Initiation Grant for Faculty Niche Research Areas 2023/24 (No. RC-FNRA-IG/23-24/COMM/01), the grants from the Research Grants Council of HKSAR (HKBU 22202423, HKBU 12203425, and HKU 17202325), the University of Hong Kong (Project 2409100399), the HKU Faculty Exchange Award 2024 (Faculty of Engineering), and Startup Grant (Tier 1) for New Academics AY2020/21 of Hong Kong Baptist University.

References

- Alon, G.; and Kamfonas, M. 2023. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*.
- Braun, T.; Rothermel, M.; Rohrbach, M.; and Rohrbach, A. 2025. DEFAME: Dynamic evidence-based fact-checking with multimodal experts. In *Proc. of ICML*.
- Carlini, N.; Jagielski, M.; Choquette-Choo, C. A.; Paleka, D.; Pearce, W.; Anderson, H.; Terzis, A.; Thomas, K.; and Tramèr, F. 2024. Poisoning web-scale training datasets is practical. In *Proc. of IEEE S&P*.
- Chen, J.; Kim, G.; Sriram, A.; Durrett, G.; and Choi, E. 2024. Complex claim verification with evidence retrieved in the wild. In *Proc. of NAACL*.
- Chen, J.; Sriram, A.; Choi, E.; and Durrett, G. 2022. Generating literal and implied subquestions to fact-check complex claims. In *Proc. of EMNLP*.
- Du, Y.; Bosselut, A.; and Manning, C. D. 2022. Synthetic disinformation attacks on automated fact verification systems. In *Proc. of AAAI*.
- Gonen, H.; Iyer, S.; Blevins, T.; Smith, N. A.; and Zettlemoyer, L. 2023. Demystifying prompts in language models via perplexity estimation. In *Proc. of EMNLP*.
- Guo, Z.; Schlichtkrull, M.; and Vlachos, A. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10: 178–206.
- He, H.; Li, Y.; Wen, D.; Cheng, R.; and Lau, F. 2025. Debating Truth: Debate-driven Claim Verification with Multiple Large Language Model Agents. *arXiv preprint arXiv:2507.19090*.
- Hu, Q.; Long, Q.; and Wang, W. 2025. Decomposition dilemmas: Does claim decomposition boost or burden fact-checking performance? In *Proc. of NAACL*.
- Huang, Y.; Chen, Y.; Zhang, H.; Li, K.; Fang, M.; Yang, L.; Li, X.; Shang, L.; Xu, S.; Hao, J.; et al. 2025. Deep research agents: A systematic examination and roadmap. *arXiv preprint arXiv:2506.18096*.
- Jain, N.; Schwarzschild, A.; Wen, Y.; Somepalli, G.; Kirchenbauer, J.; Chiang, P.-y.; Goldblum, M.; Saha, A.; Geiping, J.; and Goldstein, T. 2023. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*.
- Küçük, D.; and Can, F. 2020. Stance detection: A survey. *ACM Computing Surveys*, 53(1): 1–37.
- Li, H.; Han, X.; Wang, H.; Wang, Y.; Wang, M.; Xing, R.; Geng, Y.; Zhai, Z.; Nakov, P.; and Baldwin, T. 2025a. Loki: An open-source tool for fact verification. In *Proc. of COLING*.
- Li, Y.; Cai, H.; Kong, R.; Chen, X.; Chen, J.; Yang, J.; Zhang, H.; Li, J.; Wu, J.; Chen, Y.; et al. 2025b. Towards AI search paradigm. *arXiv preprint arXiv:2506.17188*.
- Li, Y.; He, H.; Bai, J.; and Wen, D. 2024. MCFEND: A multi-source benchmark dataset for Chinese fake news Detection. In *Proc. of WWW*.
- Li, Y.; He, H.; Wang, S.; Lau, F. C.; and Song, Y. 2023a. Improved target-specific stance detection on social media platforms by delving into conversation threads. *IEEE Transactions on Computational Social Systems*, 10(6): 3031–3042.
- Li, Y.; Wen, D.; He, H.; Guo, J.; Ning, X.; and Lau, F. C. 2023b. Contextual target-specific stance detection on twitter: Dataset and method. In *Proc. of ICDM*.
- Liu, Y.; Jia, Y.; Geng, R.; Jia, J.; and Gong, N. Z. 2024. Formalizing and benchmarking prompt injection attacks and defenses. In *Proc. of USENIX Security*.
- Pan, Y.; Pan, L.; Chen, W.; Nakov, P.; Kan, M.-Y.; and Wang, W. 2023. On the risk of misinformation pollution with large language models. In *Proc. of EMNLP*.
- Rothermel, M.; Braun, T.; Rohrbach, M.; and Rohrbach, A. 2024. InFact: A strong baseline for automated fact-checking. In *Proc. of FEVER Workshop*.
- Schlichtkrull, M.; Chen, Y.; Whitehouse, C.; Deng, Z.; Akhtar, M.; Aly, R.; Guo, Z.; Christodoulopoulos, C.; Cocarascu, O.; Mittal, A.; Thorne, J.; and Vlachos, A. 2024. The automated verification of textual claims (AVeriTeC) shared task. In *Proc. of FEVER Workshop*.
- Schlichtkrull, M.; Guo, Z.; and Vlachos, A. 2024. AVeriTeC: A dataset for real-world claim verification with evidence from the web. In *Proc. of NeurIPS*.
- Vlachos, A.; and Riedel, S. 2014. Fact checking: Task definition and dataset construction. In *Proc. of ACL*.
- Yang, J.; Xu, H.; Mirzoyan, S.; Chen, T.; Liu, Z.; Liu, Z.; Ju, W.; Liu, L.; Xiao, Z.; Zhang, M.; et al. 2024. Poisoning medical knowledge using large language models. *Nature Machine Intelligence*, 6(10): 1156–1168.
- Yi, J.; Xie, Y.; Zhu, B.; Kiciman, E.; Sun, G.; Xie, X.; and Wu, F. 2025. Benchmarking and defending against indirect prompt injection attacks on large language models. In *Proc. of KDD*.
- Yoon, Y.; Jung, J.; Yoon, S.; and Park, K. 2024. HerO at AVeriTeC: The Herd of Open Large Language Models for Verifying Real-World Claims. In *Proc. of FEVER Workshop*.
- Zhou, H.; Lee, K.-H.; Zhan, Z.; Chen, Y.; Li, Z.; Wang, Z.; Haddadi, H.; and Yilmaz, E. 2025. TrustRAG: Enhancing robustness and trustworthiness in RAG. *arXiv preprint arXiv:2501.00879*.
- Zou, W.; Geng, R.; Wang, B.; and Jia, J. 2025. PoisonedRAG: Knowledge corruption attacks to retrieval-augmented generation of large language models. In *Proc. of USENIX Security*.