VeriChain: Reliable Fact-Checking via Multi-Agent Collaboration and Reasoning Refinement

Anonymous ACL submission

Abstract

Fact-checking with large language models (LLMs) plays a critical role in combating misinformation. However, current LLM-based fact-checking approaches struggle with reliable reasoning. On the one hand, LLMs may produce seemingly persuasive multi-step explanations that are logically inconsistent with the provided evidence. On the other hand, when the given evidence is incomplete, LLMs tend to incorrectly interpret the absence of supporting evidence as refutation of the claim, an error known as argument from ignorance. Both of these behaviors lead to unreliable reasoning and fact-checking results. To address these challenges, we propose VeriChain, a novel framework that leverages multi-agent collaboration for fact-checking. The core idea of VeriChain is to decompose fact-checking into a collaborative multi-agent process, in which the reasoning conclusions are evaluated by a Verifier Agent and progressively refined through a dynamic verification loop. The Verifier checks the logical consistency of each inference step using First-Order Logic (FOL), identifying contradictions with the given evidence and instances of insufficient support. Based on these assessments, VeriChain iteratively refines the reasoning and retrieves additional knowledge as needed, thereby producing more accurate fact-checking results. Extensive experiments demonstrate the effectiveness of VeriChain.

1 Introduction

Fact-checking with large language models (LLMs) has become increasingly essential in the fight against misinformation, particularly in an era where the volume and velocity of online information far exceed the capacity of human fact-checkers. Traditional fact-checking pipelines, which extract representations of claims and evidence and then compare them in the feature space for verification, struggle to keep up with rapidly evolving complex claims. By harnessing the strengths of LLMs in interpreting or decomposing claims, integrating external evidence, and producing step-by-step reasoning, these approaches offer significant advantages over conventional methods (Pan et al., 2023; Zhang and Gao, 2023; Wang and Shu, 2023).

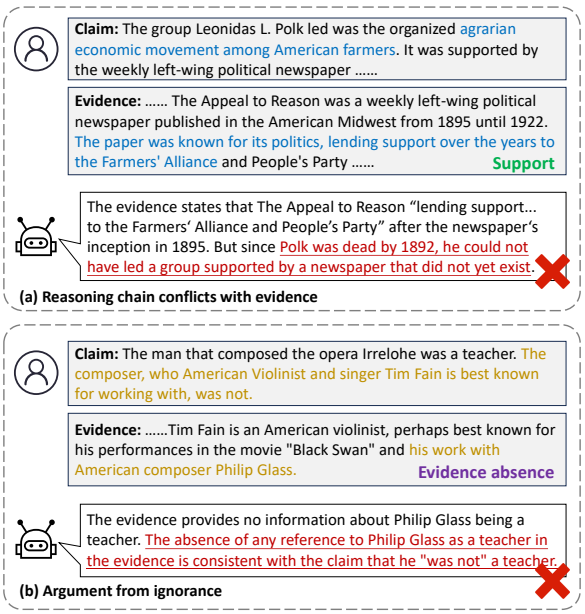

Figure 1: In LLM-based fact-checking, hallucinations often manifest as (a) reasoning chains inconsistent with the evidence, and (b) incorrect denial of claims due to insufficient supporting evidence, a type of error known as the argument from ignorance.

Despite recent advancements, current LLM-based fact-checking methods still face challenges, particularly those arising from LLM hallucinated reasoning and insufficient retrieved evidence. As illustrated in Figure 1(a), LLMs often generate lengthy reasoning chains that appear coherent and convincing at first glance. However, some of these intermediate steps are not fully supported by the given evidence, resulting in reasoning that is inconsistent with the available facts. This issue is partly due to LLMs' pre-training and post-training objectives, which encourage the generation

of plausible-looking but potentially unsupported reasoning chains. In addition, as shown in Figure 1(b), when the retrieved evidence is incomplete or partially missing, LLMs frequently misinterpret the absence of supporting information as evidence against the claim, a type of error known as the argument from ignorance. This can lead the model to incorrectly reject claims when sufficient evidence is unavailable, a situation that commonly arises in single-step evidence retrieval. These limitations, including hallucinated reasoning and misinterpretation due to evidence insufficiency, pose challenges for high-stakes fact-checking tasks, where precise, evidence-grounded reasoning is critical. Consequently, there is a need for methods that can systematically evaluate and iteratively refine LLM reasoning to ensure reliable and robust fact-checking.

To address these challenges, we introduce VeriChain, a multi-agent framework for reliable fact-checking that iteratively refines the verification results through dynamic verification loops. Specifically, VeriChain organizes the fact-checking process into a set of specialized agents, each responsible for a distinct task, including claim analysis, evidence retrieval, and reasoning. Among them, a dedicated Verifier Agent plays a central role by evaluating the reasoning chain produced by the LLM agent based on principles of FOL. By leveraging FOL, the Verifier Agent formalizes each inference step as a logical predicate and evaluates its truth value, thereby assessing the consistency of the reasoning chain with the retrieved evidence and identifying cases of insufficient evidential support. This FOL-based formalization allows the agent to systematically detect contradictions and knowledge gaps that may not be readily apparent in natural language reasoning. When contradictions or insufficient evidence are detected, VeriChain triggers a dynamic verification loop that iteratively invokes the relevant agents to refine the reasoning and acquire additional evidence, continuously updating the reasoning chain until the inconsistencies and knowledge gaps are resolved or a maximum number of iterations is reached. Extensive experiments on real-world, complex fact-checking benchmarks demonstrate that VeriChain consistently outperforms baseline methods. Our contributions are summarized as follows:

- We propose VeriChain, a multi-agent framework designed to mitigate hallucinations in LLM-based fact-checking, which manifest as inconsistencies between reasoning chains and the available evidence, as well as erroneous conclusions arising from insufficient evidential support.
- We introduce a dedicated Verifier Agent grounded in FOL and a dynamic verification loop within VeriChain, where the Verifier assesses reasoning chains and the dynamic loop refines inference based on the assessment, thereby enabling reliable fact-checking.
- Extensive experiments on real-world, complex fact-checking benchmarks demonstrate that VeriChain consistently outperforms existing baseline methods, providing more accurate and reliable fact-checking results.

2 Related Work

With the explosive growth of online information, fact-checking has become increasingly crucial for detecting misinformation. Earlier methods (Wang, 2017; Thorne et al., 2018; Augenstein et al., 2019; Jiang et al., 2020) primarily focused on simple atomic claims that could be verified using a single piece of evidence. However, claims in real-world scenarios are often multifaceted, and claim verification remains a knowledge-intensive task. To address this problem, many fact-checking methods (Schuster et al., 2021; Jiang et al., 2021; Majumder et al., 2021; Saakyan et al., 2021) have acknowledged the importance of external knowledge and reasoning in verifying complex claims (Sundriyal et al., 2023; Li et al., 2023; Chen et al., 2023; Yang et al., 2023; Jiang et al., 2024).

LLM-based fact-checking. As large language models continue to advance in reasoning capabilities (Wang et al., 2022; Wei et al., 2022; Sun et al., 2024), recent studies have explored LLM-based approaches to fact-checking. For example, some studies prompt LLMs to perform fact-checking through iterative questioning (Press et al., 2023) or program-guided reasoning (Pan et al., 2023). Nevertheless, researchers have found that LLMs face challenges such as hallucinations (Ji et al., 2023; Du et al., 2023; Feng et al., 2023; Luo et al., 2024) and unreliable reasoning, which undermine the performance of fact-checking. In this work, to mitigate hallucinations in LLM-based fact-checking, we introduce a multi-agent collaborative framework. A set of agents analyze claims, retrieves evidence, and performs reasoning, while a Verifier Agent evaluates

the reasoning process. Based on the Verifier's evaluation, VeriChain invokes the appropriate agents to either refine the reasoning or gather additional evidence, iterating this process until a satisfactory conclusion is reached.

3 VeriChain

VeriChain (as shown in Figure 2) comprises four specialized agents: (1) the **Analyser** interprets the claim; (2) the **Collector** retrieves relevant evidence; (3) the **Reasoner** performs multi-step reasoning; and (4) the **Verifier** evaluates the reasoning through FOL and triggers a dynamic verification loop.

3.1 Task Formulation

The complex fact-checking task focuses on verifying a claim based on the relevant evidence. Specifically, given a claim C and evidence E, a fact-checking model M aims to predict the veracity label Y using the information provided by E:

$$Y = M(C, E), \quad Y \in \{\text{support}, \text{refute}\}$$

3.2 Motivation for the Overall Framework

As discussed above, current LLM-based fact-checking methods face challenges. These approaches rely on LLMs to generate reasoning chains that seem plausible but may contain hallucinations or unsupported statements, thereby undermining the reliability of the verification. These challenges can be traced back to the pretraining and finetuning paradigms of LLMs. Formally, let an LLM be a parameterized model $\mathcal{M}_\theta$ that, given a text prompt x, outputs a sequence of tokens $y = (y_1, y_2, \ldots, y_T)$. During pretraining, the model is optimized to maximize the likelihood of the next token over a large corpus $\mathcal{D}_{\text{pre}}$. This encourages the model to produce fluent, coherent sequences, but does not explicitly constrain factual correctness. During fine-tuning or instruction-tuning, the model is further trained on a smaller dataset $\mathcal{D}_{\text{ft}}$ to follow instructions or generate detailed reasoning:

$$\theta^* = \arg\max_\theta \sum_{t=1}^{T} \log P_\theta(y_t \mid y_{<t}, x). \quad (1)$$

where y may include multi-step reasoning chains. To encourage long, elaborate outputs, reward functions or human feedback $R(y)$ may be incorporated, e.g., in reinforcement learning from human feedback (RLHF):

$$\theta^* = \arg\max_\theta \mathbb{E}_{y \sim P_\theta(\cdot|x)}[R(y)]. \quad (2)$$

The combination of maximizing likelihood over long sequences and optimizing for higher rewards on detailed reasoning chains induces a bias toward producing extended outputs. Consequently, the model may fabricate or hallucinate facts to maintain coherence and satisfy the reward, which explains the observed hallucination problem in multi-step reasoning tasks.

To address these issues, our VeriChain framework focuses on verifying the reasoning chains generated by LLMs, ensuring reliable and evidence-grounded fact-checking. First-order logic provides a principled framework for this purpose, enabling the formal representation and evaluation of each reasoning step. Specifically, the Verifier assesses each reasoning chain and determines whether it is reliable, inconsistent, or insufficient. Based on this detection, VeriChain employs a dynamic verification loop to invoke the appropriate agents to iteratively refine the reasoning chain or supplement missing evidence, progressively improving the correctness of the reasoning for fact-checking.

3.3 Preliminary Fact-Checking

Analyser Given a fact-checking claim $c \in C$, the Analyzer module first decomposes the original claim into a set of finer-grained sub-claims. For each sub-claim, Analyzer integrates relevant background knowledge, thereby producing an enriched and structured representation c_A that captures both the decomposition and contextual information.

Collector The Collector subsequently retrieves evidence relevant to the enriched representation c_A by issuing queries to the Bing search engine. For each retrieved web page, the Collector extracts and concatenates the text snippets corresponding to the sentences highlighted or selected by Bing, effectively simulating a human collector performing the search query, pressing Enter, and collecting the displayed content. The aggregated and structured set of evidence is denoted as e_C, which will serve as the input for subsequent reasoning stages.

Reasoner Based on the aggregated evidence e_C and the original claim c, the Reasoner generates a reasoning chain R that outlines the logical steps connecting the evidence to the claim. Based on this reasoning chain, the Reasoner then predicts a

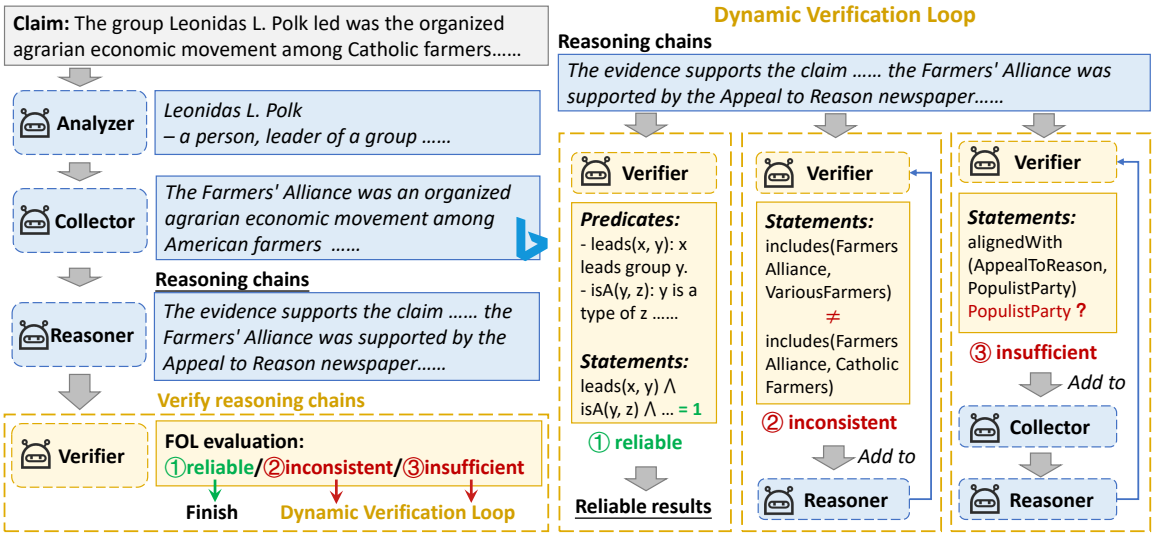

Figure 2: The framework of VeriChain.

veracity label $\hat{y}$, capturing whether the claim is supported or refuted. Formally, this reasoning process can be expressed as: $(R, \hat{y}) = \text{Reasoner}(c, e_C)$.

3.4 Fact-Checking with Justification

First-Order Logic First-Order Logic is a formal system used to represent statements about objects, their properties, and relationships. In FOL, formulas are constructed from predicates that describe properties or relations of objects. Each predicate can be evaluated as True or False. For example, a predicate IsScientist(p) may represent "p is a scientist". Given a person Alice, IsScientist(Alice) is True if evidence confirms she is a scientist, and False otherwise. Predicates can be combined to form more complex statements, such as "If Alice is a scientist, then she works in a lab". FOL also allows quantifiers like "for all" or "there exists" to reason about multiple objects, e.g., "There exists a person who is a scientist working in this lab." In addition to evaluating individual predicates, FOL allows logical operations such as conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$) to compute the truth of complex statements. In particular, conjunction ($\wedge$) represents the logical "and" operator: a formula $P \wedge Q$ is True only if both predicates P and Q evaluate to True given the evidence, and False if at least one of them is False. In the VeriChain framework, FOL is used to evaluate the reasoning chains generated by the Reasoner. Each step of a reasoning chain is converted into a predicate, and its truth value is determined based on the available evidence. This formal evaluation allows VeriChain to detect inconsistencies between reasoning and evidence or cases of insufficient supporting evidence, and to guide the iterative refinement of both reasoning and evidence retrieval.

Verifier The Verifier assesses the reasoning chain R against the collected evidence e_C using FOL. Both the reasoning chain and the evidence are transformed into sets of FOL predicates: $[f_{R_1}, \ldots, f_{R_m}]$ for the reasoning steps and $[f_{e_1}, \ldots, f_{e_k}]$ for the evidence. Each predicate represents a specific fact or logical relation. The evaluation result falls into three categories:

- **Reliable:** All reasoning predicates are satisfied, and the model's predicted label matches the ground truth. This indicates that the reasoning chain is logically consistent and fully supported by the evidence:

$$\hat{y} = y, \quad \bigwedge_{i=1}^{m} f_{R_i} = 1.$$

- **Inconsistent:** At least one reasoning predicate contradicts the available evidence. This implies that there exists an error or conflict in the reasoning process, where a reasoning step is not supported by the evidence:

$$\exists i \in \{1, \ldots, m\},\ j \in \{1, \ldots, k\}$$
$$\text{s.t. } f_{R_i} \neq f_{e_j}.$$

- **Insufficient:** Some evidence predicates are not defined. This indicates that the available evidence is not sufficient to support the conclusions of the reasoning chain, and additional

information may be required:

$$\exists\, j \in \{1, \ldots, k\} \text{ s.t. } f_{e_j} \text{ is undefined.}$$

Dynamic Verification Loop VeriChain iteratively refines reasoning chains or retrieves supplementary knowledge by leveraging feedback from the Verifier, repeating this process until a reliable outcome is obtained or a predefined maximum number of iterations is reached. If a reasoning chain is evaluated as reliable, it is accepted and returned as the final verification outcome. When the inconsistency is detected, the Reasoner updates the reasoning chain according to the Verifier's feedback, and the revised chain is subsequently re-evaluated. In scenarios where the available evidence is insufficient to determine the truth of certain predicates, the Collector retrieves additional information specifically targeted at these undefined evidence predicates. This enables the Reasoner to construct a new, more complete reasoning chain for subsequent verification. The iterative procedure guarantees that each reasoning chain is either confirmed as logically reliable or progressively refined, thereby enhancing the accuracy of the claim verification.

4 Experiments

4.1 Experimental Setup

Datasets. Two widely used and challenging datasets are adopted to evaluate the fact-checking performance of both the baselines and our VeriChain: (i) Hover (Jiang et al., 2020) and (ii) FEVEROUS-S (Pan et al., 2023). Both datasets require verifying each claim using multiple pieces of evidence and performing multi-step reasoning.

Baselines. To demonstrate the effectiveness of our approach, we compare VeriChain against four categories of baselines: (i) Pre-trained methods: BERT-FC (Soleimani et al., 2020) and LisT5 (Jiang et al., 2021). (ii) Fine-tuned methods: RoBERTa-NLI (Nie et al., 2019), DeBERTaV3-NLI (He et al., 2021), and MULTIVERS (Wadden et al., 2021). (iii) LLM-ICL methods: FLAN-T5 (Chung et al., 2022) and GPT-3.5-turbo. (iv) LLM-reasoning methods: Chain-of-Thought (Wei et al., 2022), Self-Ask (Press et al., 2023), PACAR (Zhao et al., 2024), Hiss (Zhang and Gao, 2023), FOLK (Wang and Shu, 2023), ProgramFC (Pan et al., 2023), FactcheckGPT (Wang et al., 2024) and BiDeV.

Evaluation Metrics. Following previous works, we adopt Macro-F1 as the evaluation metric to better address the class imbalance between support and refute samples, as the metric assigns equal importance to each class.

Implementation Details. In our method, we use GPT-3.5-turbo as the backbone model for all agents, accessed through the OpenAI API. The temperature of the model is set to 0 to ensure deterministic and reproducible outputs across all experiments. We use the Bing search engine as the retriever invoked by our Collector. Unlike dense retrievers, which require creating a candidate index or depending on a predefined set of sources, the Bing search engine offers extensive knowledge coverage and provides up-to-date information without these limitations. For our experiments, unless otherwise specified, the dynamic verification loop in VeriChain is configured with a fixed maximum of 3 iterations.

4.2 Overall Performance

Table 1 presents the performance of different models on the Hover and Feverous-S datasets under two settings: annotated evidence as gold-setting and retrieved evidence as open-setting. VeriChain consistently achieves the best results across all datasets, showing a clear advantage over all baseline models. For example, on Feverous-S, VeriChain achieves 94.52%(Gold) and 72.34% (Open), demonstrating its capability on large-scale, multi-hop reasoning tasks. Traditional baseline models, such as BERT-FC and LisT5, perform significantly worse under both settings. Their performance drops notably on multi-hop tasks (Hover(hop-3/hop-4)), highlighting their limitations in handling complex reasoning and long-chain information integration. NLI-based models (e.g., RoBERTa-NLI, DeBERTaV3-NLI) achieve higher results on Gold settings but still underperform on Open settings, indicating limited generalization to open-domain scenarios. Large language models and reasoning-enhanced approaches (e.g., GPT-3.5-turbo, FLAN-T5, Chain-of-Thought, PACAR) show improved performance, especially on multi-hop tasks, suggesting that integrating multiple reasoning mechanisms and external knowledge effectively enhances model performance. Overall, VeriChain's strong performance demonstrates its robustness and reasoning capability in open-domain, multi-hop tasks, highlighting the importance of mitigating LLM hallucinations for complex fact-checking.

	Hover(hop-2)		Hover(hop-3)		Hover(hop-4)		Feverous-S	
	Gold	**Open**	Gold	**Open**	**Gold**	**Open**	**Gold**	**Open**
BERT-FC	53.41	50.68	50.91	49.86	50.86	48.57	74.71	51.67
LisT5	56.15	52.56	53.76	51.89	51.67	50.46	77.88	54.15
RoBERTa-NLI	74.62	63.62	62.23	53.99	57.98	52.41	88.28	57.81
DeBERTaV3-NLI (2021)	77.22	68.72	65.98	60.76	60.49	56.01	91.98	58.81
MULTIVERS	68.86	60.17	59.87	52.55	55.67	51.86	86.03	56.61
GPT-3.5-turbo	70.63	65.07	66.46	56.63	63.49	57.27	89.77	62.58
FLAN-T5	73.69	69.02	65.66	60.23	58.08	55.42	90.81	63.73
Chain-of-Thought	74.32	70.22	65.54	58.86	60.58	57.63	90.08	64.04
Self-Ask	60.56	54.23	56.77	48.87	55.76	51.76	80.89	61.16
PACAR	76.86	70.88	70.10	63.28	69.95	58.97	94.43	65.86
HiSS	73.06	66.25	65.14	58.56	64.67	57.64	89.26	65.99
FOLK	73.24	67.29	65.84	58.61	64.73	58.79	89.52	66.89
ProgramFC	74.59	69.89	66.75	61.21	65.00	58.21	91.23	67.22
Factcheck-GPT	74.88	70.25	66.32	60.11	66.62	59.25	91.39	67.24
BiDeV	77.59	73.44	69.91	63.62	70.63	60.41	92.39	69.01
VeriChain	**80.24**	**75.84**	**73.32**	**68.38**	**71.94**	**64.08**	**94.52**	**72.34**

Table 1: Macro-F1 scores of VeriChain and baselines on Hover and Feverous-s under both gold and open settings. Bold numbers indicate significant improvements ($p < 0.05$) based on 10 rounds of bootstrapping sampling.

4.3 Ablation Study

To further investigate the contribution of each component in VeriChain, we conduct an ablation study by removing the Analyzer, Collector, Reasoner, and Verifier agents. Table 3 summarizes the performance on both Gold and Open settings across the Hover and Feverous-S datasets. As shown, removing any module leads to a drop in performance, highlighting the importance of each component. Specifically, removing the Verifier causes the most significant decrease, dropping the Gold scores from 80.24% to 69.62% on Hover(hop-2) and from 94.52% to 82.53% on Feverous-S. This demonstrates that the verification mechanism is critical for ensuring reliable reasoning and accurate results. The Reasoner also plays a crucial role, with its removal reducing performance. This indicates that the reasoning module is essential for multi-hop inference and complex task understanding. The Analyzer and Collector contribute moderately but consistently across all datasets, suggesting its role in input analysis and contextual feature extraction. The Collector's removal results in slightly larger drops, reflecting its importance in aggregating relevant information for subsequent reasoning. A similar trend is observed under the Open setting. The Verifier and Reasoner are again the most critical components, while the Analyzer and Collector provide incremental but consistent gains. Overall, this ablation study confirms that the performance improvements of VeriChain are the result of a synergistic combination of all four agents, with the Verifier and Reasoner being particularly vital for robust reasoning in both Gold and Open scenarios.

4.4 Analysis

Analysis of Three Verification Situations Figure 4 shows the distribution of the Verifier's assessment categories in the first evaluation across the Hover and Feverous-S datasets. Several clear trends emerge from the results. First, the proportion of reliable outputs remains the largest category, typically ranging from 70% to 80%, confirming that LLMs can produce mostly trustworthy reasoning but still leave substantial room for improvement. Second, among the two error types, insufficient responses consistently outnumber inconsistent ones, indicating that LLMs are more prone to under-supported reasoning rather than producing logically contradictory conclusions. Third, the Open setting exhibits a higher fraction of insufficient responses compared to the Gold setting, reflecting the additional challenge posed by open-domain retrieval, where evidence is more likely to be incomplete. Finally, as task difficulty increases, from 2-hop to 4-hop in Hover, the proportion of reliable outputs gradually decreases. This indicates that increasing task complexity raises the likelihood of LLMs

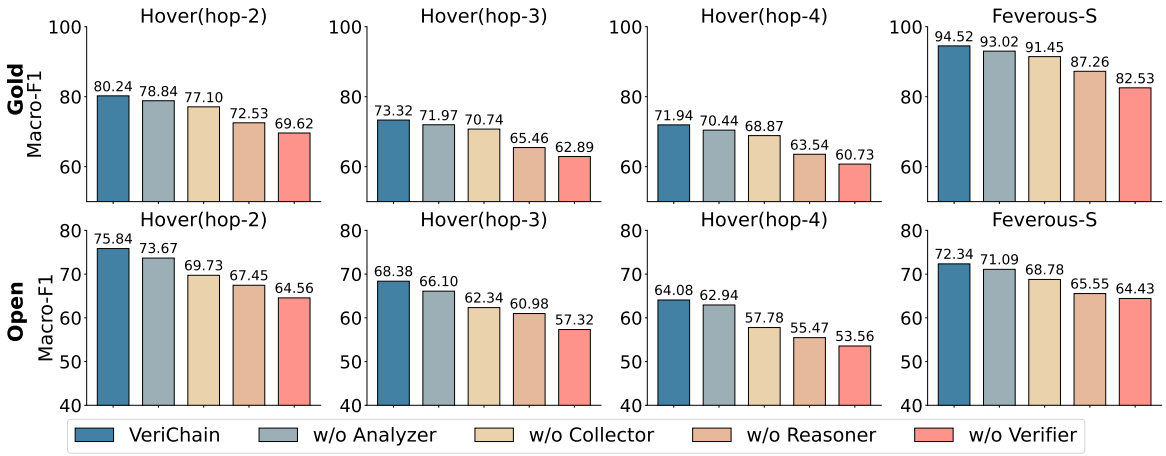

Figure 3: Results of ablation study. We removed the Analyzer, Collector, Reasoner, and Verifier from VeriChain, respectively. Notably, removing the Verifier also disables the dynamic verification loop, since the loop is inherently driven by the Verifier's assessments.

producing unreliable inferences or encountering knowledge gaps. Overall, the results highlight two challenges in LLM-based fact-checking, evidence insufficiency and reasoning inconsistency, both of which become more pronounced in open-domain or higher-hop settings. VeriChain effectively alleviates these issues by enforcing logic-grounded verification and iterative refinement, leading to more reliable fact-checking outcomes.

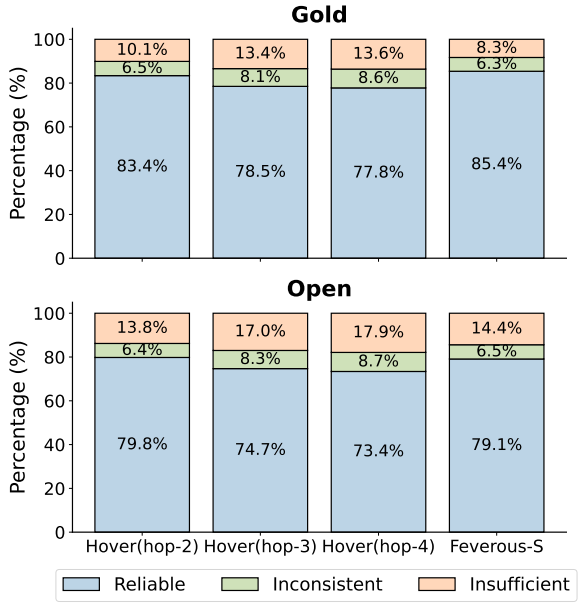

Figure 4: Distribution of the Verifier's evaluation outcomes (reliable, insufficient, and inconsistent) on reasoning chains produced in a single verification iteration across Hover and Feverous-S.

Generalizability Across Different Agents To evaluate the generalizability of the VeriChain framework, we test it with three different underlying agent models: GPT-3.5, 4o-mini, and Claude-3.7. Figure 5 shows the performance under both Gold and Open settings. The results indicate that VeriChain consistently achieves strong performance regardless of the agent backbone. In the Gold setting, all three agents yield comparable scores across datasets. Similarly, in the Open setting, performance remains robust across agents. These findings demonstrate that VeriChain is highly compatible with different LLM-based agents, effectively maintaining reasoning and verification capabilities across diverse model architectures. The consistent performance observed across multiple backbone architectures underscores the robustness of the framework and highlights its potential for broad applicability and generalizability across different model architectures.

Analysis of Maximum Iterations Figure 6 illustrates how different maximum iteration settings in the dynamic verification loop affect VeriChain's performance. Performance generally improves as the number of iterations increases from 1 to 3 across all datasets. However, further increasing the iteration count to 4 does not yield additional gains and, in some cases, results in a slight decrease (e.g., Hover(hop-4) drops from 71.94% to 71.88%), suggesting that excessive iterations may introduce redundancy or minor noise in reasoning. These results indicate that 3 iterations strike the best balance between reasoning depth and efficiency, achieving optimal performance across multi-hop and complex tasks. Overall, these results demonstrate that the dynamic verification loop enhances VeriChain's

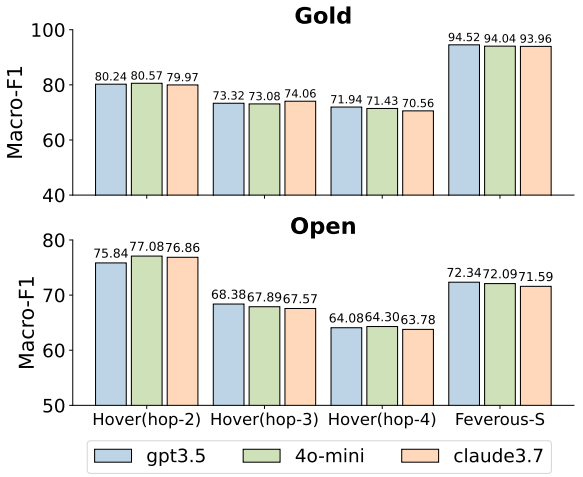

Figure 5: Performance of the VeriChain framework when instantiated with three underlying agent models (GPT-3.5, 4o-mini, and Claude-3.7) under both Gold and Open settings. The results demonstrate VeriChain's robustness and consistent gains across heterogeneous agent backbones.

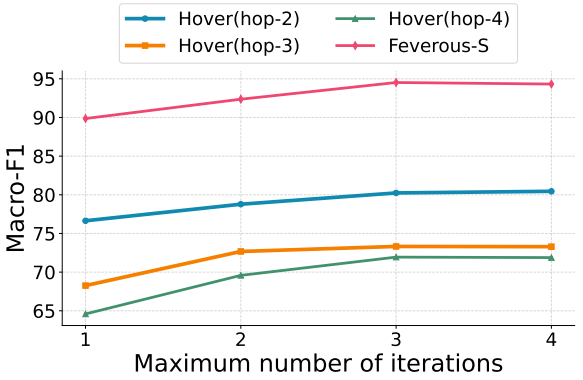

Figure 6: Impact of different maximum iteration settings in VeriChain's Dynamic Verification Loop.

performance, while highlighting the importance of selecting an appropriate number of iterations to maximize benefits.

Impact of Knowledge Sources To assess the influence of different external knowledge sources for fact-checking, we compare performance using Bing and the MS MARCO Passage Corpus (Izacard et al., 2021) under the Open setting (shown in Figure 7). The results show that leveraging Bing as a knowledge source consistently yields higher performance across all datasets compared to MS MARCO. Bing, as a large-scale, up-to-date web search engine, provides richer and more relevant information, which helps the model generate more accurate and complete responses. In contrast, MS MARCO, although structured and curated, is comparatively limited in coverage, resulting in lower performance on complex multi-hop tasks. Overall, this analysis underscores the importance of selecting high-coverage, reliable knowledge sources to enhance fact-checking in open-domain scenarios.

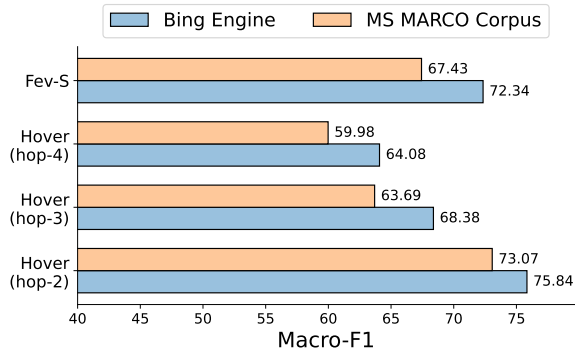

Figure 7: Comparison of VeriChain's performance under the Open setting when using different external knowledge sources.

5 Conclusion

In this work, we present VeriChain, a multi-agent fact-checking framework designed to address challenges in LLM-based fact-checking. While LLMs excel at decomposing claims and synthesizing information, their tendency to generate unsupported intermediate inferences and to conflate missing evidence with negative conclusions fundamentally limits their reliability in high-stakes verification scenarios. VeriChain mitigates these issues by introducing a dedicated Verifier Agent that evaluates the reasoning chains under a first-order logic formulation. By grounding each inference step in explicit logical predicates, the Verifier Agent can systematically detect contradictions with the given evidence as well as unsupported assumptions, failure modes that are often obscured in unconstrained natural language reasoning. Based on the Verifier's assessment, VeriChain further employs a dynamic verification loop to iteratively refine the reasoning process. By selectively invoking retrieval or reasoning agents to acquire additional knowledge or revise the inference steps, VeriChain enforces evidence-grounded fact-checking throughout the pipeline. Extensive experiments on challenging real-world benchmarks demonstrate that VeriChain yields substantial and consistent improvements over existing LLM-based fact-checking methods. In the future, we will explore the use of our framework in other domain tasks that focus on reasoning.

Limitations

At present, VeriChain primarily focuses on enabling reliable reasoning in LLM-based fact-checking. In future work, we will explore extending the framework to other knowledge-intensive reasoning tasks.

References

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. Multifc: A real-world multi-domain dataset for evidence-based fact checking of claims. *arXiv preprint arXiv:1909.03242*.

Mengyang Chen, Lingwei Wei, Han Cao, Wei Zhou, and Songlin Hu. 2023. Can large language models understand content and propagation for misinformation detection: An empirical study. *arXiv preprint arXiv:2311.12699*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2022. Scaling instruction-finetuned language models. *arXiv e-prints*, pages arXiv–2210.

Li Du, Yequan Wang, Xingrun Xing, Yiqun Ya, Xiang Li, Xin Jiang, and Xuezhi Fang. 2023. Quantifying and attributing the hallucination of large language models via association analysis. *arXiv preprint arXiv:2309.05217*.

Chao Feng, Xinyu Zhang, and Zichu Fei. 2023. Knowledge solver: Teaching llms to search for domain knowledge from knowledge graphs. *arXiv preprint arXiv:2309.03118*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Bohan Jiang, Zhen Tan, Ayushi Nirmal, and Huan Liu. 2024. Disinformation detection: An evolving challenge in the age of llms. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, pages 427–435. SIAM.

Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 2021. Exploring listwise evidence reasoning with t5 for fact verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 402–410.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. Hover: A dataset for many-hop fact extraction and claim verification. *arXiv preprint arXiv:2011.03088*.

Zizhong Li, Haopeng Zhang, and Jiawei Zhang. 2023. A revisit of fake news dataset with augmented fact-checking by chatgpt. *arXiv preprint arXiv:2312.11870*.

Ruilin Luo, Tianle Gu, Haoling Li, Junzhe Li, Zicheng Lin, Jiayi Li, and Yujiu Yang. 2024. Chain of history: Learning and forecasting with llms for temporal knowledge graph completion. *arXiv preprint arXiv:2401.06072*.

Bodhisattwa Prasad Majumder, Sudha Rao, Michel Galley, and Julian McAuley. 2021. Ask what's missing and what's useful: Improving clarification question generation using global knowledge. *arXiv preprint arXiv:2104.06828*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan, and Preslav Nakov. 2023. Fact-checking complex claims with program-guided reasoning. *arXiv preprint arXiv:2305.12744*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. Covid-fact: Fact extraction and verification of real-world claims on covid-19 pandemic. *arXiv preprint arXiv:2106.03794*.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin c! robust fact verification with contrastive evidence. *arXiv preprint arXiv:2103.08541*.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. Bert for evidence retrieval and claim verification. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42*, pages 359–366. Springer.

Hongda Sun, Yuxuan Liu, Chengwei Wu, Haiyu Yan, Cheng Tai, Xin Gao, Shuo Shang, and Rui Yan. 2024. Harnessing multi-role capabilities of large language

models for open-domain question answering. In *Proceedings of the ACM Web Conference 2024*, pages 4372–4382.

Megha Sundriyal, Tanmoy Chakraborty, and Preslav Nakov. 2023. From chaos to clarity: Claim normalization to empower fact-checking. *arXiv preprint arXiv:2310.14338*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

David Wadden, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. 2021. Multivers: Improving scientific claim verification with weak supervision and full-document context. *arXiv preprint arXiv:2112.01640*.

Haoran Wang and Kai Shu. 2023. Explainable claim verification via knowledge-grounded reasoning with large language models. *arXiv preprint arXiv:2310.05253*.

William Yang Wang. 2017. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, and 1 others. 2024. Factcheck-bench: Fine-grained evaluation benchmark for automatic fact-checkers. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14199–14230.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Chang Yang, Peng Zhang, Wenbo Qiao, Hui Gao, and Jiaming Zhao. 2023. Rumor detection on social media with crowd intelligence and chatgpt-assisted networks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5705–5717.

Xuan Zhang and Wei Gao. 2023. Towards llm-based fact verification on news claims with a hierarchical step-by-step prompting method. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1011.

Xiaoyan Zhao, Lingzhi Wang, Zhanghao Wang, Hong Cheng, Rui Zhang, and Kam-Fai Wong. 2024. Pacar: Automated fact-checking with planning and customized action reasoning using large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12564–12573.