LoCal: Logical and Causal Fact-Checking with LLM based Multi-Agents

Anonymous Author(s)

ABSTRACT

With the development of social media, people are exposed to a vast amount of unverified information, making fact-checking particularly important. Existing fact-checking methods primarily encourage breaking down claims into more easily solvable sub-tasks, and deriving final answers through reasoning with external evidence. However, these models face logical issues regarding whether and how the sub-tasks can logically be combined to form the original claims, and encounter causal errors in the reasoning process due to insufficient evidence or hallucinations from LLMs. In addition, they often suffer from a lack of interpretability. In this paper, we propose **Lo**gical and **Ca**usa**l** fact-checking (LoCal), a novel fact-checking framework based on multiple LLM-based agents. The usage of multi-agent systems is due to their increasingly demonstrated ability to perform complex tasks in a manner similar to humans. LoCal primarily consists of a decomposing agent, multiple reasoning agents, and two evaluating agents. Specifically, the decomposing agent first utilizes the in-context learning ability of LLMs to break down complex claims into simpler sub-tasks, including fact verification tasks and question answering tasks. Afterwards, two types of reasoning agents are respectively utilized to retrieve external knowledge to address the fact verification tasks that require comparative analysis skills, and the question answering tasks that necessitate the ability of information extraction from evidence. We then combine the sub-tasks and their corresponding responses to generate a solution for evaluation. In order to enhance logical and causal consistency, two evaluating agents are respectively employed to examine whether the generated solution is logically equivalent to the original claim and determine whether the solution still holds when challenged by the counterfactual label. The evaluating agents provide confidence degrees for the solutions based on the evaluation results and iteratively correct the logical and causal errors in the reasoning process. We evaluate LoCal on two challenging datasets, and the results show that LoCal significantly outperforms all the baseline models across different settings of evidence availability. In addition, LoCal offers better interpretability by providing a structured solution along with detailed evaluating processes. We believe LoCal will provide valuable insights for future agent-based misinformation detection.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

CCS CONCEPTS

• **Information systems → Multimedia information systems**; • **Computing methodologies → Natural language processing**.

KEYWORDS

Fact-Checking, LLM-Based Agents, Logical and Causal Consistency, Interpretability, Confidence Evaluation

ACM Reference Format:

Anonymous Author(s). 2024. LoCal: Logical and Causal Fact-Checking with LLM based Multi-Agents. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

With the rise of social media platforms and online news media, information spreads at an incredibly fast pace. However, it also leads to the proliferation of fake news, rumors, and misleading content. Fact-checking helps identify and correct such misinformation, preventing it from spreading among the public and reducing its negative impact on society. As a result, fact-checking has become increasingly important for the web [1] [14, 20, 23, 30, 51, 58].

However, verifying real-world claims is often nontrivial. Early studies [29, 36, 49] have primarily focused on verifying a simple atomic claim that does not encompass the complex logic existing in real-world claims. More recent studies [3, 8, 39, 43, 64] have recognized the importance of addressing complex claims. As shown in Figure 1, these fact-checking models often concentrate on steps like claim decomposition, external retrieval, sub-question answering, and reasoning result integration. On one hand, the sub-tasks generated by claim decomposition need to reflect all the information in the original claim, and must be correctly integrated to arrive at the final answer, which we refer to as the logical process. On the other hand, the model needs to derive correct results for the sub-tasks based on retrieved evidence and ensure that the reasoning process leads to an exact answer rather than other possible answers, which we call the causal process. Existing fact-checking models are often fragile, prone to logical errors in claim decomposition and sub-task result integration. In addition, they also suffer from causal issues due to inadequate retrieval of evidence and hallucinations from LLMs, leading to incorrect responses of sub-tasks and resulting in inaccurate or non-unique final label (flipping the label is also acceptable).

In this work, to address the problems, we propose a logical and causal fact-checking method with multiple LLM-based agents, namely LoCal. Multi-agent systems decompose complex tasks into multiple simpler parts and assign them to different agents, which are well-suited for fact-checking tasks. The agents are all built upon

[1] **Relevance to the Web**: Fact-checking belongs to the Misinformation and Disinformation topic within the Responsible Web theme. This research aims to propose a new multi-agent framework to address fact-checking challenges on the Web.

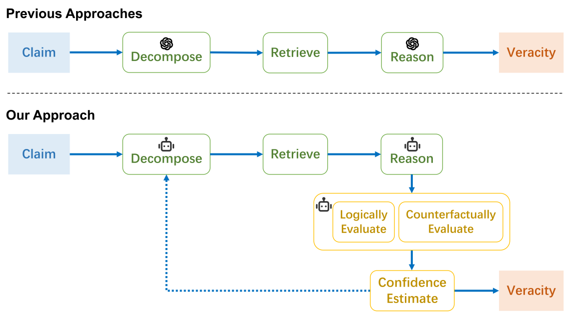

Figure 1: Comparison between our framework and previous approaches

large language models since LLMs are trained on vast amounts of data, making them valuable knowledge source for veracity prediction. More importantly, LLMs can leverage background knowledge as prior information, enabling them to understand and formalize various causal scenarios and accurately reason about unseen events. Unlike previous works [3, 8, 39, 43, 64], we innovatively introduce an evaluation phase in fact-checking to check logical and causal consistency. Specifically, as shown in Figure 2, three different types of agents are used in combination to address the fact-checking task in LoCal:

- A decomposing agent, responsible for using the in-context learning ability of LLMs to break down the input claim into multiple sub-tasks, including fact verification and question answering tasks.
- Two types of reasoning agents (fact verificating agents and question answering agents), respectively responsible for solving the fact verification tasks that require comparative analysis skills, and the question answering tasks that necessitate the capabilities of information extraction from retrieved evidences.
- Two evaluating agents (logically evaluating agent and counterfactually evaluating agent) which examine whether the solution is logically equivalent to the original claim and determine whether the solution still holds when challenged by the counterfactual label, thus determining whether to accept the answer or start a new iteration.

Moreover, LoCal provides better interpretability for the predicted veracity from two sides. First, LoCal provides a structured solution. LoCal connects the decomposed sub-tasks and the sequence of validations or answers for each sub-task into a solution that consists of multiple task-response pairs. The solution is structured and reviewable by humans. Second, LoCal offers a detailed description of the evaluating process. The logically evaluating agent provides an evaluation process determining whether the solution is equivalent to the original claim, while the counterfactually evaluating agent explains how contradictions arise.

We evaluate LoCal on two challenging datasets for verifying complex claims: FEVEROUS [2] and HOVER [22]. The results show that our method LoCal significantly outperforms state-of-the-art methods on both datasets by taking into account the logical and causal consistency in LLM-based multi-agent system. In addition, LoCal provides interpretability of fact-checking with structured solutions and detailed evaluating processes.

In summary, our main contributions are:

- We propose, for the first time, an LLM-based multi-agent system considering both logical and causal consistency for fact-checking. Our method provides structured solutions along with detailed evaluating processes to improve interpretability.
- We innovatively apply logical and causal evaluating agents in fact-checking, which examine whether the solutions are logically equivalent to the original claims and assess whether the solutions still hold when challenged by counterfactual labels, enhancing logical and causal consistency.
- Extensive experiments demonstrate that our fact-checking method significantly outperforms state-of-the-art methods across different settings of evidence availability, and provides interpretability with structured solutions and insightful evaluating processes.

2 RELATED WORK

2.1 Fact-Checking

With the explosive growth of information on the internet, fact-checking has been increasingly applied in detecting and correcting misinformation. Therefore, effective fact-checking methods have garnered significant attention. For a given claim, the goal of fact-checking is to find relevant evidence and then make a judgment on the veracity of the claim based on the evidence [17, 18, 57]. Earlier models [21, 34, 36, 49, 50, 52] primarily addressed simple atomic claims that could be verified using a single piece of evidence. However, complex real-world claims often require multi-evidence reasoning. To address the problem, recent fact-checking models [5, 29, 39, 43] have acknowledged the importance of handling complex claims. While most existing fact-checking models[5, 16, 24, 37, 59, 63, 65, 66] have achieved some promising results, they rely on large-scale human-annotated datasets, which are often costly to produce. To address the issue, recent research [31, 41, 61] has focused on fact-checking in zero-shot and few-shot scenarios. However, these methods achieve limited performance due to their restricted modeling capacity and lack of background knowledge.

2.2 LLMs for Fact-Checking

The rich knowledge and emergent reasoning capabilities of LLMs present new opportunities for fact-checking tasks. Recently, there have been numerous efforts exploring the use of LLMs for fact-checking tasks. For instance, Pan et al. [43] proposed a fact-checking framework that decomposes claims into a series of subtasks, leveraging LLMs' in-contextual learning capabilities to generate reasoning programs that guide the verification of claims. Chen et al. [7] employed both standard prompting ("No CoT" strategy) and zero-shot chain-of-thought (CoT) prompting strategies for fact-checking claims generated by both human and LLMs. Zhao et al. [64] introduced a framework incorporating a claim decomposer with self-reflection and an LLM-centric planning module, focusing on LLMs' applications in dynamic planning and action execution.

Recognizing that LLMs' inherent knowledge may be insufficient for detecting factual errors, some works have explored enhancing LLMs with external knowledge or tools. Specifically, Chern et al. [10] proposed a fact-checking framework integrating multiple tools (such as Google Search, Google Scholar, code interpreters, Python) for detecting factual errors in texts generated by LLMs. Cheung and Lam [11] combined knowledge retrieved from search engines with LLaMA's reasoning capabilities to predict the veracity of claims.

Some recent works [10, 64] have explored the use of agents to address fact-checking tasks, but the performance of these models is limited due to logical and causal errors encountered during the fact-checking process. Unlike previous efforts, our proposed LoCal employs an LLM-based multi-agent framework that includes a decompositing agent, multiple reasoning agents, and two evaluating agents tailored to ensure logical and causal consistency. In addition, our model improves the interpretability of fact-checking by providing structured solutions along with a detailed description of the evaluation process.

2.3 Interpretability of Fact Checking

In the face of complex real-world claims, methods that simply assign labels to claims directly from black-box models often lack convincing interpretability [18, 53, 56]. Many models have endeavored to enhance the interpretability of fact-checking. Some works [13, 35, 45, 46, 54, 62] emphasize evidence using attention weights, but this type of explanation is not easily understandable to humans. Some studies [1, 15] generate logical justifications using knowledge graphs, which are not flexible. Many works [4, 25, 28] generate summaries of comments provided by expert fact-checkers, but annotating such datasets is costly and difficult to scale. Some studies [6, 26, 32, 44] enrich context from source documents to assist task-specific response generation, but the free-form natural language is not concise. In contrast to previous works, our proposed LoCal divides fact-checking into several steps, and provide a structured solution along with detailed description of evaluating processes, thereby enhancing the model's interpretability.

3 OUR PROPOSED MODEL

3.1 Problem Formulation

Given a claim S, possibly accompanied by a piece of gold evidence e_{gold} according to the work mode configuration, a fact-checking model needs to return a Boolean output Y, indicating the veracity of the given claim S.

3.2 LoCal

We aim to address complex claims that require multi-hop reasoning and multiple pieces of evidence. We propose a **Lo**gical and **Ca**usal fact-checking method (LoCal) based on multiple LLM agents. LoCal uses a decomposing agent to break down complex claims into simpler sub-tasks, including fact verification and question answering tasks. The two types of tasks are assigned to specialized reasoning agents for resolution. The sub-tasks and their corresponding answers are combined to form a solution. Next, in order to enhance logical and causal consistency, two evaluating agents (logically evaluating agent and counterfactually evaluating agent) are respectively used to check whether the solution is logically equivalent to the original claim and whether the solution remains valid when challenged by the counterfactual label. Based on the evaluation results, we determine whether to iteratively repeat the processes of decomposing, reasoning and evaluating. We draw a conclusion and estimate its confidence until both evaluating agents accept the solution, or the maximum number of iterations is reached.

3.2.1 Decomposing Agent. In most scenarios, a claim is complex, which means it can be composed of many atomic sub-tasks. With the purpose of verifying the veracity of a complex claim, we could decompose and atomize it first. To achieve this goal, we introduce an LLM-based agent here to act as a decomposing agent. The decomposing agent can generate a sub-task sequence, denoted as $S = \{s_1, s_2, \ldots, s_k\}$, and a function f for deduction in a later stage. Notably, a sub-task s_i can not only take the form of an exact atomic claim (fact verification task), but also a one-hop question (question answering task). Specifically, we define $s_i = (t_i, I_i, a_i)$, where t_i specifies the type of sub-task (such as a one-hop question or an atomic claim), I_i is a natural language sentence describing the sub-task s_i, a_i is the variable that stores the result of the sub-task s_i. Since subsequent steps often depend on the results of previous steps, we allow the parameter I_i to reference the variables $a_1, \ldots, a_{i-1}$ from prior steps. The deduction function f denotes a logical expression concerning $a_1, a_2, \ldots, a_k$, used to derive the veracity label of claim S, that is, $f(a_1, a_2, \ldots, a_k) \in \{TRUE, FALSE\}$.

3.2.2 Evidence Retrieval. For each sub-task, relevant evidence is necessary for further deduction. If the work mode configuration is gold, which means a piece of gold evidence is provided at the beginning and no other knowledge source can be accessed, we will directly use it as the relevant evidence for each sub-task. In other case, if the configuration is open with no gold evidence provided, we will leverage a retrieval toolkit R to collect information for each sub-task s_i from the Internet. The toolkit R will return a batch of relevant evidences, which can be denoted as $R(s_i) = \{e_j\}, j = 1, 2, \ldots, |R(s_i)|$. In general, the collection of relevant evidences E_i for the sub-task s_i can be represented as

$$E_i = \begin{cases} \{e_{gold}\} & \text{if work mode is gold.} \\ R(s_i) = \{e_j\} & \text{if work mode is open.} \end{cases}$$

3.2.3 Reasoning Agents. We introduce two types of LLM-based reasoning agents respectively for fact verification tasks and question answering tasks. For each sub-task $s_i = (t_i, I_i, a_i)$ and its relevant evidences E_i, a suitable agent will be selected to process them and generate a corresponding answer. More precisely, when a sub-task s_i is an exact claim, a fact verificating agent will be used to determine whether it is true or not, returning a Boolean answer a_i accordingly. If a sub-task is one-hop-question, a question answering agent will look up the given evidence to extract a direct answer a_i to that question. During this process, each sub-task s_i will be paired with an answer a_i. Finally, we can get the final Boolean answer that represents the veracity of the original claim $v = f(a_1, a_2, \ldots, a_k)$, where f is the deduction function generated in decomposing stage.

3.2.4 Evaluating Agents. In previous approaches, there are no further steps to verify the correctness of the predicted veracity in the reasoning stage. They focus on improvements in the intermediate steps including decomposition, retrieval, and so on to improve the

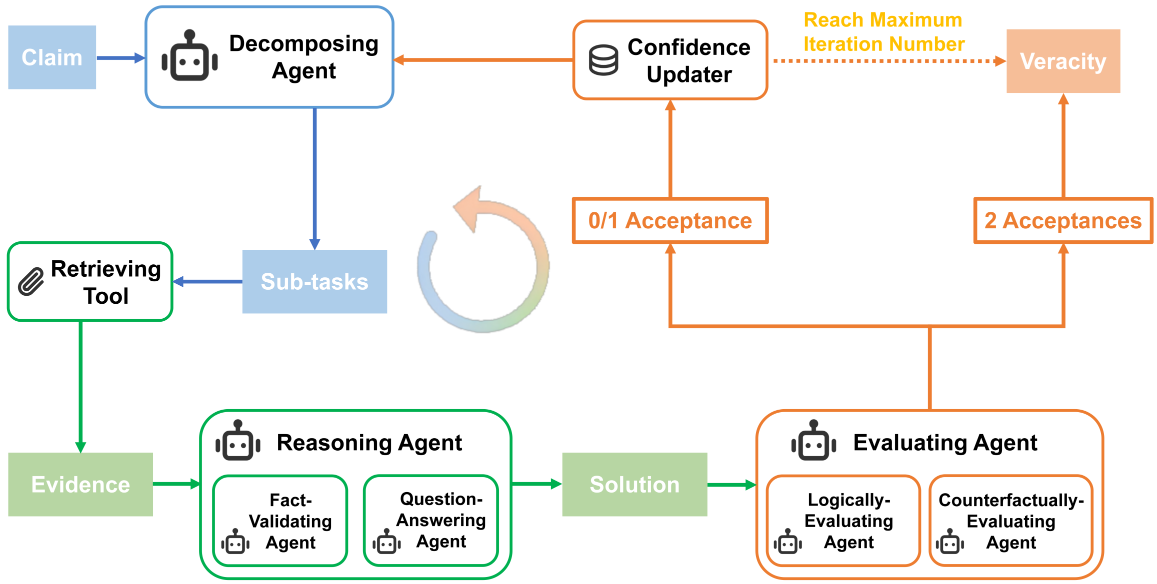

Figure 2: The overall framework of LoCal.

performance of fact-checking. However, these methods are incapable of avoiding the logical and causal errors that arise during the process.

To address this issue, we introduce a novel stage, named the evaluating stage for logical and causal checking. Specifically, we aim at identifying and correcting logical and causal errors that are undetectable in the initial reasoning process. Before the evaluating stage, a summary description of the previous reasoning process is required. We sequentially concatenate each sub-task s_i and its corresponding response a_i, replacing any references with their pointing objects, to obtain a summary description, denoted as d. For convenience, we may also use the term "solution" to refer to combination of d and v. In the evaluating stage, two different LLM-based agents, namely logically evaluating agent and counterfactually evaluating agent, will be respectively used to verify the predicted veracity from logical and causal perspectives.

Logically Evaluating Agent. In order to verify whether the veracity v of the original claim is correct in terms of logic, we introduce a logically evaluating agent that takes the summary description d (not including the predicted veracity v) as input, performs reasoning based on it, and outputs a new veracity v' that aligns with the logic in d. We denote the function of the logically evaluating agent as $L(d)$, whose input is the summary description d and output is a Boolean variable v' representing the new veracity.

Counterfactually Evaluating Agent. Unlike logical evaluation, counterfactual evaluation verifies the veracity v of the original claim from a causal perspective by assuming an opposite veracity prediction and trying to find conflicts. We use another counterfactually evaluating agent for causal evaluation. In this process, the former prediction v will be negated first, used as the assuming predicted veracity for the summary description d. After that, both d and $\neg v$ will be fed into the counterfactually evaluating agent for conflict detection. The counterfactually evaluating agent will be asked to return a Boolean answer, denoted as $C(d, \neg v)$, indicating whether there are conflicts or not.

3.2.5 Confidence Updater. Confidence Updater is designed to estimate the confidence of the predicted veracity of the original claim v based on the results of logical and causal evaluation. For convenience, we denote two Boolean variables, p_l and p_c, which represent whether the logically evaluating agent and counterfactually evaluating agent accept the predicted veracity v, respectively. Formally, for logically evaluating agent, we consider it as a state of acceptance when the new prediction is consistent with the original one, i.e., $p_l = \neg(L(d) \text{ XOR } v)$. For counterfactually evaluating agent, we consider it as a state of acceptance when any conflict is detected after the label flipping, i.e., $p_c = C(d, \neg v)$.

Next, we infer the final veracity and its confidence degree based on p_l and p_c. Firstly, we define two veracity counters, $\#POS$ and $\#NEG$, respectively representing the number of times that $True$ or $False$ is considered as the more credible veracity during the iterative process. Then, we will follow the below workflow based on the results of the logical and counterfactual evaluations.

- **2 Acceptances:** If both p_l and p_c are true, which means both logical and causal evaluating agents accept the predicted veracity, we consider the prediction is trustworthy, and take it as the final veracity with confidence degree as 1. The workflow finishes.
- **1 Acceptance:** If only one evaluating agent accepts the predicted veracity, we hold the view that the predicted veracity

has a certain level of confidence, but it is not entirely reliable. Therefore, we increment the veracity counter associated with the veracity by one, and return to the decomposing step for the next iteration.

- **0 Acceptance:** If both evaluating agents reject, we would prefer the opposite veracity result. Therefore, we increment the veracity counter associated with the opposite of the predicted veracity, and proceed to the decompose stage for the next iteration.

Before starting the workflow, we will set a maximum iteration number. If a 2-Acceptance result has never been achieved within maximum iterations, we will determine the final veracity associated with the larger value between $\#POS$ and $\#NEG$, whose confidence degree will be considered as $\max(\#POS, \#NEG)/(\#POS + \#NEG)$.

4 EXPERIMENTAL SETUP

4.1 Datasets

In our experiments, we evaluate our LoCal model using two different datasets: HOVER [22] and FEVEROUS [2]. These two datasets encompass various domains and complexities, providing a robust benchmark for fact-checking tasks. HOVER is a dataset that requires hierarchical fact extraction and multi-hop reasoning across different sections of Wikipedia articles. Based on the number of "hops" needed when doing fact-checking, we can divide HOVER into 1126 two-hop claims, 1835 three-hop claims and 1039 four-hop claims. FEVEROUS is another dataset, containing claims annotated with evidence from both unstructured text and structured data, such as tables from Wikipedia. To better compare model performances, we process the FEVEROUS dataset in the same manner as described in the previous method [43], focusing on 2962 claims that use sentence evidence only.

4.2 Baselines

To better demonstrate the advancements of LoCal, we compare it against 11 baselines, which can be categorized into four groups based on their approaches: (1) Pre-trained models, (2) Fine-tuned models, (3) LLM-based methods, (4) LLM Agent-based methods.

(1) **Pre-trained models.** These approaches leverage pre-trained models based on Transformer to do fact-checking tasks. The following two baselines are of this kind. Bert-FC [55]: It introduces pre-trained BERT into evidence retrieval and claim verification. LisT5 [21]: This approach leverages T5 for fact-checking.

(2) **FC/NLI fine-tuned models.** These methods use pre-trained models which are further fine-tuned on other fact-checking datasets or natural language inference (NLI) datasets. Three of our baselines belong to this category. RoBERTa-NLI [38]: This method chooses RoBERTa as the base model and fine-tunes it on four NLI datasets. DeBERTaV3-NLI [19]: The base model of this method is DeBERTaV3, and it is fine-tuned using FEVER and four NLI datasets. MUTIVERS [59]: A method, fine-tuned on FEVER, predicts fact-checking labels and identifies rationales in a multitasking manner.

(3) **LLM-based methods.** This kind of methods leverage LLMs directly with designed prompts to generate the answer. Codex [9] and Flan-T5 [12] are two baselines that leverage LLMs in a few-shot manner, which means a few in-context examples are provided for LLMs to learn the task. We take ChatGPT [64] as a zero-shot baseline, which means only prompting the LLM to find evidence and predict the veracity without examples.

(4) **LLM Agent-based methods.** These methods also include LLMs, not using them directly, but regarding them as agents for specific sub-tasks in the pipeline. ProgramFC (n) [43]: It is a model that leverages LLM-based agents to generate reasoning programs for each claim, which are then used to execute fact-checking step by step. The variable (n) represents the number of times Program FC is executed repeatedly. PACAR [64]: A model that introduces LLM-based agents for decomposing, self-reflection and an LLM-centric planning module in the fact-checking tasks.

4.3 Implementation Details

For a fair comparison, we follow ProgramFC [43], using Flan-T5 [12] as the base for reasoning agents, which is an improved T5 model [48] that has achieved state-of-the-art zero-shot/few-shot performance on many QA benchmarks. Due to the discontinuation of Codex used in ProgramFC, we follow PACAR [64], using gpt-3.5-turbo as the base for the decompositing and evaluating agent. For the sake of fair comparison and cost considerations, we do not use GPT-4, which may perform better in the tasks. In the claim decomposition step, we use few-shot learning, limiting the decomposing agent to access only 20 samples from HOVER or FEVEROUS. We conduct experiments in the gold evidence (abbr. gold) and open book settings (abbr. open). For the open book setting, we build an index for the knowledge base accompanying the HOVER and FEVEROUS datasets and use the Pyserini toolkit [33] as the retrieval tool. For each sub-task, we use the top 5 paragraphs provided by the retrieval tool as supporting evidence. In the sub-task answering and evaluation steps, we let the reasoning agents and evaluating agents operate in a zero-shot manner to reduce the model's burden. The maximum number of iterations is set to 3. We use the macro-F1 score to evaluate the fact-checking performance of all models, following the approach of Pan et al. [43].

5 EXPERIMENTAL RESULTS

5.1 Main Results

Table 1 presents a comprehensive comparison of our proposed LoCal model with various state-of-the-art baselines across all settings. As we can see, LoCal achieves the best performance in 7 out of 8 evaluations, and outperforms all baseline models on average, demonstrating its effectiveness. LoCal surpasses the best baseline models ProgramFC and PACAR in both the gold evidence and open book settings. It is particularly evident on HOVER, where LoCal's average performance improves by 2.14% and 2.31% compared to the best baseline models in both settings. We believe that LoCal enhances the performance of fact-checking through improvements in logical and causal consistency.

We also find that the multi-agent models outperform LLM-based models. This highlights the superiority of the multi-agent framework, due to its demonstrated ability to perform complex tasks in

Table 1: Main Results: Macro-F1 scores of LoCal and baselines on HOVER and FEVEROUS datasets. The best and second-best results in each column are indicated with bold and underlined text, respectively. * denotes that the improvement of the metric over the baseline methods is statistically significant (p< 0.01).

	Models	Gold-Evidence					Open-Book				
		HOVER 2-hop	HOVER 3-hop	HOVER 4-hop	FEVEROUS	AVERAGE	HOVER 2-hop	HOVER 3-hop	HOVER 4-hop	FEVEROUS	AVERAGE
I	**BERT-FC**	53.40	50.90	50.86	74.71	57.47	50.68	49.86	48.57	51.67	50.20
I	**LisT5**	56.15	53.76	51.67	77.88	59.87	52.56	51.89	50.46	54.15	52.27
II	**RoBERTa-NLI**	74.62	62.23	57.98	88.28	70.78	63.62	53.99	52.40	57.80	56.95
II	**DeBERTaV3-NLI**	<u>77.22</u>	65.98	60.49	91.98	73.92	68.72	60.76	56.00	58.81	61.07
II	**MUTIVERS**	68.86	59.87	55.67	86.03	67.61	60.17	52.55	51.86	56.61	55.30
III	**Codex**	70.63	66.46	63.49	89.77	72.59	65.07	56.63	57.27	62.58	60.39
III	**Flan-T5**	73.69	65.66	58.08	90.81	72.06	69.02	60.23	55.42	63.73	62.10
III	**ChatGPT**	71.42	64.87	63.65	83.49	70.86	66.94	60.56	58.73	55.72	60.49
IV	**ProgramFC(n=1)**	74.10	66.13	65.69	91.77	74.42	69.36	60.63	<u>59.16</u>	67.80	64.24
IV	**ProgramFC(n=5)**	75.65	68.48	66.75	<u>92.69</u>	75.89	<u>70.30</u>	<u>63.43</u>	57.74	<u>68.06</u>	<u>64.88</u>
IV	**PACAR**	76.86	<u>70.10</u>	<u>69.95</u>	**94.43**	<u>77.84</u>	-	-	-	-	-
IV	**Our LoCal**	**79.93***	**73.26***	**70.14***	91.09	**78.61***	**72.71***	**64.11***	**61.59***	**68.22***	**66.66***

a manner similar to humans. Compared to the strong LLM-based baselines Flan-T5 and ChatGPT, LoCal shows an average improvement of 6.55% and 7.75% in the gold evidence setting and 4.56% and 6.17% in the open book setting. Through the decomposing agent, reasoning agents, and evaluating agents in multi-agent collaboration, LoCal has achieved significantly superior performance compared to LLMs.

Fact-checking in the open book setting is challenging because it requires retrieving external evidence, which often includes many irrelevant items. LoCal outperforms the best baseline on the 2-hop, 3-hop, and 4-hop claims of HOVER and the FEVEROUS dataset by 2.41%, 0.68%, 3.85%, and 0.16%, with an average improvement of 1.78%. These results highlight the model's ability to deal with insufficient and irrelevant evidence, through its logical and causal evaluations. Additionally, the strong baseline model ProgramFC (n=5) requires a full execution of 5 iterations, imposing a significant burden on LLMs. However, our model only requires a maximum of 3 iterations to reach the final veracity and performs iterative fact-checking only when the evaluating agents do not accept the predicted veracity.

The HOVER dataset is more challenging in the gold evidence setting compared to FEVEROUS because FEVEROUS's gold evidence is more direct and obvious, making it easier to obtain the final answers. Our LoCal model improves by 3.07%, 3.16%, and 0.19% on the 2-hop, 3-hop, and 4-hop claims of HOVER, with an average improvement of 2.14%. These results indicate that our model can better handle claims containing complex gold evidence with iterative logical and causal evaluation. For the simple FEVEROUS dataset, the performance of state-of-the-art baselines already exceeds 90% (indicating that it is a simple task), and our LoCal achieves a comparable performance exceeding 90%. For complex tasks with more errors in claim decomposition and reasoning, LoCal benefits from enhancing logical and causal consistency with the corresponding evaluating agents. However, for simpler tasks, they might "overcorrect", leading to some correct answers being modified incorrectly. Overall, our model achieves the highest average performance on both HOVER and FEVEROUS and is better suited for complex tasks.

5.2 Ablation Experiments

To further evaluate the effectiveness of the mechanisms proposed by the LoCal model, we conduct various ablation experiments in the open book setting. For convenience, we refer to the evaluating agents as Eval, the logically evaluating agent as L-Eval, and the counterfactually evaluating agent as C-Eval.

As shown in Table 2, we test the performance of LoCal removing the Eval, removing the L-Eval, and removing the C-Eval. The experimental results on different hop levels (2-hop, 3-hop, 4-hop) of the HOVER dataset and the FEVEROUS dataset show that removing both evaluating agents results in a 1.13% performance drop. When we remove the C-Eval, the results show an average average drop of 0.74%. When we remove L-Eval, the results show an average drop of 0.35%. This demonstrates the effectiveness of both the logical and counterfactual evaluation. Additionally, we observe a more significant drop of 1.77% on 4-hop claims when removing the Eval, compared to smaller drops on 2-hop and 3-hop claims, indicating that ensuring logical and causal consistency is more effective for complex tasks. Note that, each ablation experiment reaches the final answer after executing up to 3 iterations, which is consistent with the non-ablation experimental settings. Therefore, the performance improvement does not stem from an increase in the number of execution iterations but from enhanced logical and causal consistency.

5.3 Analysis of Corrected Errors by LoCal

To evaluate the improvements of LoCal in logical and causal consistency, we selected 100 samples from the HOVER and FEVEROUS datasets, with 50 from each. These samples are initially assigned incorrect labels but are ultimately corrected by our proposed logical and causal evaluation. Based on the stages of error occurrence, we analyzed 4 types of errors that could be corrected:

Table 2: Ablation results of LoCal

Models	HOVER			FEVEROUS	AVG
	2-hop	3-hop	4-hop		
LoCal	**72.71**	**64.11**	**61.59**	**68.22**	**66.66**
w/o Eval	71.66	63.07	59.82	67.56	65.53
w/o C-Eval	72.13	63.47	60.04	68.04	65.92
w/o L-Eval	72.38	63.57	61.49	67.81	66.31

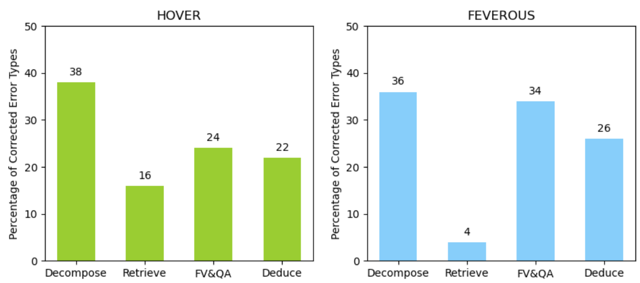

Figure 3: The percentage of the four corrected error types in the samples that were initially assigned incorrect labels but were corrected by our proposed logical and causal evaluation.

- **Decomposing Errors:** Errors made by the decomposing agent in breaking down the claim, including incorrect or missing references, and incorrect decomposition logic.
- **Retrieval Errors:** The decomposition is correct, but the retrieval tool fails to retrieve sufficient relevant evidence for the sub-tasks, leading to untrustworthy or missing answers.
- **FV&QA Errors:** Relevant evidence is retrieved, but the fact verification and question answering agents provide incorrect or contradictory answers to the sub-tasks.
- **Deducing Errors:** The answers to the sub-tasks are correct, but the deduction function makes errors when integrating sub-tasks to form the predicted veracity, such as confusion over logical operators like "and", "or", "both", and "not both".

We present the corrected analysis results in Figure 3. It shows that corrected decomposition errors are the most frequently corrected type, accounting for 38% on HOVER and 36% on FEVEROUS, which indicates that LoCal can logically identify unreasonable sub-task decomposition. The corrected FV&QA errors and Deducing errors come second, which means LoCal can causally identify the incorrect or contradictory information in sub-tasks and overall deduction. Additionally, we observe that LoCal could also correct some retrieval errors by iterative logical and causal checking.

5.4 Results in Closed Book Setting

We also evaluated the performance of various models in the closed book setting, where models rely solely on their internal parametric knowledge without any external evidence, including gold and retrieved evidence. We divided the baseline models into three groups:

The first group directly uses LLMs like Codex, FLAN-T5, and InstructGPT [40]. The second group applies different prompting methods to InstructGPT. Self-Ask: Guide the LLM to autonomously generate and answer relevant questions [47]. Chain-of-Thought (CoT): Prompt with demonstrations [60]. Zero-Shot Chain-of-Thought (ZS-CoT): With the prompt "let's think step by step" [27]. The third group consists of models that have achieved claim decomposition, including ProgramFC, and QACheck [42] which is a reasoning model that uses a claim verifier and a question generator.

The results in Table 3 show that most models achieve a macro-F1 score between 50% and 60%, indicating that current methods struggle with fact-checking tasks without retrieving external knowledge. Our LoCal model achieves the best performance on the 2-hop, 4-hop, and average scores. For the 3-hop HOVER dataset and the FEVEROUS dataset, QACheck and CoT achieve the best performance. All three top-performing models employed step-by-step reasoning, which underscores the importance of task decomposition for fact-checking. Additionally, on the 4-hop HOVER dataset, LoCal outperforms ProgramFC by 1.01%, indicating LoCal achieved more accurate results for complex claims due to the assurance of logical and causal consistency.

Table 3: Results in the Closed-Book setting

Models	HOVER			FEVEROUS	AVG
	2-hop	3-hop	4-hop		
Codex	55.57	53.42	45.59	57.85	53.11
Flan-T5	48.27	52.11	51.13	55.16	51.67
InstructGPT	56.51	51.75	49.68	60.13	54.52
Self-Ask	51.54	51.47	52.45	56.83	53.07
CoT	57.20	53.66	51.83	**61.05**	55.94
ZS-CoT	50.30	52.30	51.58	54.78	52.24
ProgramFC	54.27	54.18	52.88	59.66	55.25
QACheck	55.67	**54.67**	52.35	59.47	55.54
ours	**57.51**	53.60	**53.89**	59.47	**56.12**

5.5 Case Study

To illustrate our approach, we conduct a case study presented in Table 4. In iteration 1, we find that the reasoning agents provide correct answers based on retrieved evidence. However, sub-task 1 is wrong with the added "not" compared to the original claim, causing the deduction function incorrect. Given that both sub-task 1 and sub-task 3 were true, the model concluded the original claim was true. Fortunately, our LoCal will iteratively conduct logical and causal consistency checking, finally achieving the right veracity. Specifically, LoCal can:

- use the logically evaluating agent (L-Eval) to identify the correct logical structure and obtain the correct answer, "False" (*Reject*);
- use the counterfactually evaluating agent (C-Eval) to support the counterfactual answer and find no contradiction when flipping the label, meaning "False" label also aligns with the solution (*Reject*);
- increase the veracity counter of "False" by 1 (Both evaluating agents reject, 0 Acceptance), start a new iteration, and finally conclude that the veracity is "False".

One advantage of LoCal is that it enhances the *interpretability* of fact-checking compared to end-to-end models by breaking down fact-checking into several clear execution steps, which can aid

Table 4: Case Study: The green highlight indicates correct information, while the red highlight indicates untrustworthy or misleading information. The underlined text represents correct evaluating results, implying the correct veracity.

Claim: The novelist Paul Thomas Mann was American, not the author of Snow Falling on Cedars. **Right Veracity:** False

▹ **Iteration 1**

Decomposing Agent: sub-task1 = Paul Thomas Mann was not American.

Evidence1: Paul Thomas Mann was a German novelist, short story writer, social critic, philanthropist, essayist, …

Fact Verificating Agent: Answer1 = True

sub-task2 = Who was the author of Snow Falling on Cedars?

Evidence2: Snow Falling on Cedars is a film … based on David Guterson's award-winning novel of the same title. …

Question Answering Agent: Answer2 = David Guterson

sub-task3 = Paul Thomas Mann was not David Guterson.

Evidence3: Paul Thomas Mann was a German novelist … David Guterson (born May 4, 1956) is an American novelist …

Fact Verificating Agent: Answer3 = True

Predicted veracity = Answer1 and Answer3 = True

L-Eval: Based on the solution provided, it is clear that Paul Thomas Mann was not American and was not the author of Snow Falling on Cedars. Therefore, the claim that Paul Thomas Mann was American and not the author of Snow Falling on Cedars is False. The solution and the original claim are not equivalent.

C-Eval: The predicted veracity "False" indicates that the claim stating Paul Thomas Mann was American and not the author of Snow Falling on Cedars is incorrect. The possible solution confirms that Paul Thomas Mann was not American. The possible solution also confirms that Paul Thomas Mann was not the author of Snow Falling on Cedars. The predicted veracity correctly concludes that the claim is false based on the information provided in the possible solution. So there is no contradiction in the solution.

Zero Acceptance

▹ **Iteration 2**

…

Final Veracity: False

human understanding and debugging. As shown in Table 4, LoCal connects the sub-tasks with their answers, forming a structured solution composed of multiple task-answer pairs, which facilitates human review. Additionally, LoCal provides a detailed description of the evaluating process, where the L-Eval determines whether the solution is equivalent to the original claim, and the C-Eval explains how contradictions arise.

5.6 Analysis of Required Iterations

To evaluate the efficiency of LoCal, we statistically analyzed the number of required execution iterations on the HOVER (2-hop, 3-hop, 4-hop) and FEVEROUS datasets. As depicted in Figure 4, most execution iterations are 1, indicating that LoCal can quickly provide conclusions for most claims. For more complex claims, multiple iterations are necessary. Approximately one-third of the claims require further iterations of fact-checking to reach a conclusion.

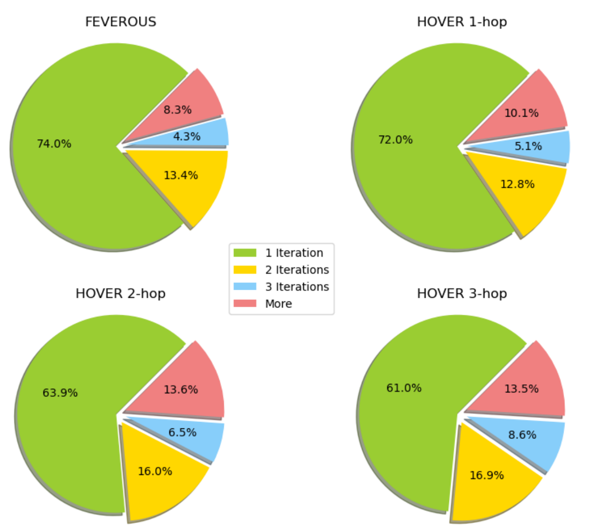

Figure 4: The proportion of required iteration numbers to obtain the final result

6 CONCLUSION

In summary, we propose LoCal, a novel logical and causal fact-checking method with LLM-based multi-agents. LoCal primarily consists of a decomposing agent, multiple reasoning agents, and two evaluating agents. Specifically, the decomposing agent breaks down complex claims into multiple simpler sub-tasks, the reasoning agents use the retrieved evidence to handle fact verification and question answering sub-tasks separately, and two evaluating agents (logically evaluating agent and counterfactually evaluating agent) check whether the solution is logically and causally valid. By iteratively performing the steps, we obtain a more accurate fact veracity result due to logical and causal insurance. Additionally, our LoCal provides a better interpretability with the structured solution along with a detailed evaluating process. The results from two challenging datasets demonstrate the effectiveness of LoCal. In future work, we would explore adapting LoCal to more real-world scenarios, such as fake news detection and multi-modal fact-checking.

REFERENCES

[1] Naser Ahmadi, Joohyung Lee, Paolo Papotti, and Mohammed Saeed. 2019. Explainable fact checking with probabilistic answer set programming. *arXiv preprint arXiv:1906.09198* (2019).

[2] Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. Feverous: Fact extraction and verification over unstructured and structured information. *arXiv preprint arXiv:2106.05707* (2021).

[3] Rami Aly and Andreas Vlachos. 2022. Natural Logic-guided Autoregressive Multi-hop Document Retrieval for Fact Verification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 6123–6135.

[4] Pepa Atanasova. 2024. Generating fact checking explanations. In *Accountable and Explainable Methods for Complex Reasoning over Text*. Springer, 83–103.

[5] Giorgio Barnabò, Federico Siciliano, Carlos Castillo, Stefano Leonardi, Preslav Nakov, Giovanni Da San Martino, and Fabrizio Silvestri. 2023. Deep active learning for misinformation detection using geometric deep learning. *Online Social Networks and Media* 33 (2023), 100244.

[6] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*. PMLR, 2206–2240.

[7] Canyu Chen, Haoran Wang, Matthew Shapiro, Yunyu Xiao, Fei Wang, and Kai Shu. 2022. Combating health misinformation in social media: Characterization, detection, intervention, and open issues. *arXiv preprint arXiv:2211.05289* (2022).

[8] Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. 2022. Generating Literal and Implied Subquestions to Fact-check Complex Claims. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 3495–3516.

[9] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374* (2021).

[10] I Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, et al. 2023. FacTool: Factuality Detection in Generative AI–A Tool Augmented Framework for Multi-Task and Multi-Domain Scenarios. *arXiv preprint arXiv:2307.13528* (2023).

[11] Tsun-Hin Cheung and Kin-Man Lam. 2023. Factllama: Optimizing instruction-following language models with external knowledge for automated fact-checking. In *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 846–853.

[12] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research* 25, 70 (2024), 1–53.

[13] Limeng Cui, Kai Shu, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. defend: A system for explainable fake news detection. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 2961–2964.

[14] Yi R Fung, Kung-Hsiang Huang, Preslav Nakov, and Heng Ji. 2022. The battlefront of combating misinformation and coping with media bias. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4790–4791.

[15] Mohamed H Gad-Elrab, Daria Stepanova, Jacopo Urbani, and Gerhard Weikum. 2019. Exfakt: A framework for explaining facts over knowledge graphs and text. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 87–95.

[16] Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*. PMLR, 10764–10799.

[17] Max Glockner, Yufang Hou, and Iryna Gurevych. 2022. Missing Counter-Evidence Renders NLP Fact-Checking Unrealistic for Misinformation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 5916–5936.

[18] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics* 10 (2022), 178–206.

[19] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543* (2021).

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

[21] 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)*. 402–410.

[22] 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. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 3441–3460.

[23] Yiqiao Jin, Yeon-Chang Lee, Kartik Sharma, Meng Ye, Karan Sikka, Ajay Divakaran, and Srijan Kumar. 2023. Predicting information pathways across online communities. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1044–1056.

[24] Yiqiao Jin, Xiting Wang, Ruichao Yang, Yizhou Sun, Wei Wang, Hao Liao, and Xing Xie. 2022. Towards fine-grained reasoning for fake news detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 5746–5754.

[25] Shailza Jolly, Pepa Atanasova, and Isabelle Augenstein. 2022. Generating fluent fact checking explanations with unsupervised post-editing. *Information* 13, 10 (2022), 500.

[26] Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024* (2022).

[27] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35 (2022), 22199–22213.

[28] Neema Kotonya and Francesca Toni. 2020. Explainable Automated Fact-Checking for Public Health Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 7740–7754.

[29] Amrith Krishna, Sebastian Riedel, and Andreas Vlachos. 2022. Proofver: Natural logic theorem proving for fact verification. *Transactions of the Association for Computational Linguistics* 10 (2022), 1013–1030.

[30] David MJ Lazer, Matthew A Baum, Yochai Benkler, Adam J Berinsky, Kelly M Greenhill, Filippo Menczer, Miriam J Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, et al. 2018. The science of fake news. *Science* 359, 6380 (2018), 1094–1096.

[31] Nayeon Lee, Yejin Bang, Andrea Madotto, and Pascale Fung. 2021. Towards Few-shot Fact-Checking via Perplexity. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1971–1981.

[32] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[33] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2356–2362.

[34] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained Fact Verification with Kernel Graph Attention Network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 7342–7351.

[35] Yi-Ju Lu and Cheng-Te Li. 2020. GCAN: Graph-aware Co-Attention Networks for Explainable Fake News Detection on Social Media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 505–514.

[36] 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. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4300–4312.

[37] Van-Hoang Nguyen, Kazunari Sugiyama, Preslav Nakov, and Min-Yen Kan. 2020. Fang: Leveraging social context for fake news detection using graph representation. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1165–1174.

[38] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A New Benchmark for Natural Language Understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4885–4901.

[39] Nedjma Ousidhoum, Zhangdie Yuan, and Andreas Vlachos. 2022. Varifocal Question Generation for Fact-checking. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 2532–2544.

[40] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[41] Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2021. Zero-shot Fact Verification by Claim Generation. *Proceedings of ACL 2021* (2021).

[42] Liangming Pan, Xinyuan Lu, Min-Yen Kan, and Preslav Nakov. 2023. QACheck: A Demonstration System for Question-Guided Multi-Hop Fact-Checking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 264–273.

[43] 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* (2023).

[44] Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813* (2023).

[45] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2017. Where the truth lies: Explaining the credibility of emerging claims on the web and social media. In *Proceedings of the 26th international conference on world wide web companion*. 1003–1012.

[46] Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. Declare: Debunking fake news and false claims using evidence-aware deep learning. *arXiv preprint arXiv:1809.06416* (2018).

[47] 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*. 5687–5711.

[48] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* 21, 140 (2020), 1–67.

[49] Sudha Rao and Hal Daumé III. 2019. Answer-based Adversarial Training for Generating Clarification Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 143–155.

[50] Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. COVID-Fact: Fact Extraction and Verification of Real-World Claims on COVID-19 Pandemic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2116–2129.

[51] Dietram A Scheufele, Andrew J Hoffman, Liz Neeley, and Czerne M Reid. 2021. Misinformation about science in the public sphere. *Proceedings of the National Academy of Sciences* 118, 15 (2021), e2104068118.

[52] Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get Your Vitamin C! Robust Fact Verification with Contrastive Evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 624–643.

[53] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-Augmented Black-Box Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 8364–8377.

[54] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 395–405.

[55] 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*. Springer, 359–366.

[56] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663* (2021).

[57] James Thorne and Andreas Vlachos. 2018. Automated Fact Checking: Task Formulations, Methods and Future Directions. In *Proceedings of the 27th International Conference on Computational Linguistics*. 3346–3359.

[58] Lin Tian, Xiuzhen Zhang, and Jey Han Lau. 2023. Metatroll: Few-shot detection of state-sponsored trolls with transformer adapters. In *Proceedings of the ACM Web Conference 2023*. 1743–1753.

[59] David Wadden, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. 2022. MultiVerS: Improving scientific claim verification with weak supervision and full-document context. In *Findings of the Association for Computational Linguistics: NAACL 2022*. 61–76.

[60] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[61] Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Lu Wang. 2022. Generating Scientific Claims for Zero-Shot Scientific Fact Checking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2448–2460.

[62] Fan Yang, Shiva K Pentyala, Sina Mohseni, Mengnan Du, Hao Yuan, Rhema Linder, Eric D Ragan, Shuiwang Ji, and Xia Hu. 2019. Xfake: Explainable fake news detector with visualizations. In *The world wide web conference*. 3600–3604.

[63] Ruichao Yang, Xiting Wang, Yiqiao Jin, Chaozhuo Li, Jianxun Lian, and Xing Xie. 2022. Reinforcement subgraph reasoning for fake news detection. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2253–2262.

[64] 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)*. 12564–12573.

[65] Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning Over Semantic-Level Graph for Fact Checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 6170–6180.

[66] Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: Graph-based Evidence Aggregating and Reasoning for Fact Verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 892–901.

A ALGORITHM

The detailed structure of our LoCal is presented in Algorithm 1.

Algorithm 1 LoCal

procedure ONE_ITERATION($S, iter, max, \#POS, \#NEG$)
  $\mathcal{S} \leftarrow [s_1, \ldots, s_k] = [\text{DECOMPOSE}(S)]$
  $\mathcal{E} \leftarrow [E_1, \ldots, E_k] = [\text{RETRIEVE}(s_i) \text{ for } s_i \text{ in } \mathcal{S}]$
  $\mathcal{A} \leftarrow [a_1, \ldots, a_k] = [\text{REASON}(s_i, E_i) \text{ for } s_i, E_i \text{ in } \mathcal{S}, \mathcal{E}]$
  $v \leftarrow \text{DEDUCE}(\mathcal{A})$
  $d \leftarrow \text{SUMMARIZE}(\mathcal{S}, \mathcal{A})$
  $p_l \leftarrow \neg(\text{LOGICAL\_EVAL}(d) \text{ XOR } v)$
  $p_c \leftarrow \text{COUNTERFACTUAL\_EVAL}(d, \neg v)$
  if p_l is `True` **and** p_c is `True` **then**
    return v
  else if p_l is `True` **or** p_c is `True` **then**
    CONFIDENCE($v, \#POS, \#NEG$)
  else
    CONFIDENCE($\neg v, \#POS, \#NEG$)
  end if
  if $iter >= max$ **then**
    if $\#POS > \#NEG$ **then**
      return `True`
    else
      return `False`
    end if
  else
    return ONE_ITERATION($S, iter + 1, max, \#POS, \#NEG$)
  end if
end procedure

In Algorithm 1, *iter* represents the current iteration number, *max* represents the maximum number of iterations, and the rest of the notations are consistent with those used in the main text.

B ADVANTAGES COMPARED TO COT

CoT prompts guide LLMs to think step by step, essentially decomposing claims. However, CoT lacks logic and causal evaluation. To further assess the effectiveness of LoCal, we compared its performance with ChatGPT using CoT in the gold evidence setting.

As shown in Table 5, LLM with CoT still performs poorly on fact-checking tasks, showing the inherent limitations of LLMs, such as hallucination issues and limited reasoning capabilities, which hinder their ability to effectively address fact-checking tasks. In contrast, our proposed LoCal model, based on an LLM-powered multi-agent system, achieves superior performance by ensuring logical and causal consistency.

Table 5: Results of LoCal and CoT in the Gold-Evidence setting

Models	HOVER			FEVEROUS	AVG
	2-hop	3-hop	4-hop		
ChatGPT	71.42	64.87	63.65	83.49	70.86
ChatGPT-CoT	72.85	65.61	64.08	84.22	71.69
ours	**79.93**	**73.26**	**70.41**	**91.09**	**78.67**

C A COMPLETE CASE STUDY

To provide a detailed demonstration of our approach, we conducted a comprehensive case study in Table 6, which is an expanded description of Table 4.

The first iteration is the same as in Table 4. Since both L-Eval and C-Eval do not accept the predicted veracity, the veracity counter associated with "False" is incremented by 1, leading to the second iteration. In the second iteration, all components provided trustworthy information, and the right veracity was obtained. This case study shows how LoCal enhances logical and causal consistency to improve the accuracy of fact-checking, while providing structured solutions along with detailed evaluating processes.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009

Table 6: A Complete Case Study: The green highlight indicates correct information, while the red highlight indicates untrustworthy or misleading information. The underlined text represents correct evaluating results, implying the correct veracity.

Claim: The novelist Paul Thomas Mann was American, not the author of Snow Falling on Cedars. **Right Veracity:** False

▹ **Iteration 1**

Decomposing Agent: sub-task1 = Paul Thomas Mann was not American.

Evidence1: Paul Thomas Mann was a German novelist, short story writer, social critic, philanthropist, essayist, …

Fact Verificating Agent: Answer1 = True

sub-task2 = Who was the author of Snow Falling on Cedars?

Evidence2: Snow Falling on Cedars is a film … based on David Guterson's award-winning novel of the same title. …

Question Answering Agent: Answer2 = David Guterson

sub-task3 = Paul Thomas Mann was not David Guterson.

Evidence3: Paul Thomas Mann was a German novelist … David Guterson (born May 4, 1956) is an American novelist …

Fact Verificating Agent: Answer3 = True

Predicted veracity = Answer1 and Answer3 = True

L-Eval: Based on the solution provided, it is clear that Paul Thomas Mann was not American and was not the author of Snow Falling on Cedars. Therefore, the claim that Paul Thomas Mann was American and not the author of Snow Falling on Cedars is False. The solution and the original claim are not equivalent.

C-Eval: The predicted veracity "False" indicates that the claim stating Paul Thomas Mann was American and not the author of Snow Falling on Cedars is incorrect. The possible solution confirms that Paul Thomas Mann was not American. The possible solution also confirms that Paul Thomas Mann was not the author of Snow Falling on Cedars. The predicted veracity correctly concludes that the claim is false based on the information provided in the possible solution. So there is no contradiction in the solution.

Zero Acceptance

▹ **Iteration 2**

Decomposing Agent: sub-task1 = Paul Thomas Mann was American.

Evidence1: Paul Thomas Mann was a German novelist, short story writer, social critic, philanthropist, essayist, …

Fact Verificating Agent: Answer1 = False

sub-task2 = Paul Thomas Mann was not the author of Snow Falling on Cedars.

Evidence2: David Guterson (born May 4, 1956) is an American novelist, … He is best known as the author of the book "Snow Falling on Cedars".

Fact Verificating Agent: Answer2 = True

Predicted veracity = Answer1 and Answer2 = False

L-Eval: The evidence provided clearly states that Paul Thomas Mann was not American and was not the author of Snow Falling on Cedars. Therefore, the claim that Paul Thomas Mann was American and not the author of Snow Falling on Cedars is False based on the evidence provided. The solution and the original claim are equivalent.

C-Eval: The predicted veracity "True" means that Paul Thomas Mann was indeed American and not the author of Snow Falling on Cedars. The possible solution states that it is false that Paul Thomas Mann was American. The possible solution also states that it is true that Paul Thomas Mann was not the author of Snow Falling on Cedars. The predicted veracity contradicts the possible solution by stating that Paul Thomas Mann was American.

Two Acceptance

Final Veracity: False