

# Claim Verification with Adversarial Reasoning and Planning

Kuan-Chieh Lo<sup>1</sup>, Valerie L. Shalin<sup>2</sup>, R. Kelly Garrett<sup>1</sup>, Srinivasan Parthasarathy<sup>1</sup>

<sup>1</sup>The Ohio State University, Columbus, OH USA

<sup>2</sup>Wright State University, Dayton, OH USA

{lo.311, garrett.258}@osu.edu, valerie.shalin@wright.edu, srini@cse.ohio-state.edu

## Abstract

The scale and speed of digital communication demands robust automated claim verification systems that handle complex, multi-hop reasoning. Existing approaches have critical limitations: single-agent systems exhibit confirmation bias, while conventional multi-agent frameworks rely on homogeneous agents that exhibit groupthink, limiting critical evaluation. We present **CARP** (Claim Verification with Adversarial Reasoning and Planning), a novel multi-agent claim verification framework that organizes heterogeneous agents powered by multiple different language models competing as support and refutation teams. This adversarial structure forces comprehensive evaluation from both perspectives while mitigating confirmation bias and groupthink. Our framework incorporates systematic claim decomposition, strategic verification planning, and multi-hop knowledge retrieval to handle complex reasoning tasks. We evaluate CARP on two claim verification datasets—HOVER and FEVEROUS—where it demonstrates significant improvements in verification accuracy compared to existing single-agent and homogeneous multi-agent approaches, particularly for complex claims requiring multi-hop reasoning and evidence synthesis. Ablation studies confirm that both adversarial evaluation model and multi-hop knowledge retrieval contribute substantially to performance, with benefits scaling with reasoning complexity.

## 1 Introduction

The integrity of publicly shared information is strained by the scale and speed of digital communication. New gatekeeping algorithms reward engagement over accuracy, outlet motivations (e.g., profit, political alignment, or audience retention) are opaque, and individual biases shape sharing. Misinformation spreads rapidly across platforms (Vosoughi, Roy, and Aral 2018), undermining democratic processes, eroding institutional trust, and jeopardizing public health (Gisondi et al. 2022; Caceres et al. 2022).

Despite unprecedented access to information, societies struggle with persistent misperceptions about critical social issues. These misperceptions carry significant implications for democratic governance, institutional legitimacy, and collective well-being. The sheer scale and speed of information circulation on digital platforms overwhelms manual verification, as the volume of content far exceeds what tradi-

tional fact-checking methods can address (Vosoughi, Roy, and Aral 2018).

Automated claim verification systems represent a promising strategy for mitigating the challenges of today’s noisy and uncertain information ecosystem. Researchers have made significant progress in training AI models to assess claims autonomously, yet these systems are not accurate enough to earn users’ trust. Effective claim verification requires decomposing complex statements, interpreting nuanced context, performing multi-hop reasoning across diverse sources, and managing the inherent ambiguity of real-world language (Guo, Schlichtkrull, and Vlachos 2022). Current single-agent AI systems face persistent challenges, including systematic biases that are amplified rather than corrected (Allan et al. 2025), domain-specific knowledge gaps between parametric knowledge and specialized task requirements (Lu, Lin, and Tsao 2024; Luo et al. 2023), and limitations in leveraging knowledge for complex reasoning tasks (Qin et al. 2023).

Recent fact-checking approaches attempt to overcome these limitations by assigning multiple agents, each with distinct roles, within the same language model framework. Multi-Agent Debate Refinement (MADR) leverages multiple agents with diverse roles using iterative refinement in generated explanations (Kim et al. 2024). Logical and Causal fact-checking (LoCal) employs multiple agents with logical and causal evaluation mechanisms to improve fact-checking consistency (Ma et al. 2025). Bilateral Defusing Verification (BiDeV) adds role-played LLMs to address claim vagueness and evidence redundancy (Liu et al. 2025). Yet key challenges remain unaddressed. Current systems show varying effectiveness in claim decomposition: MADR focuses primarily on explanation refinement while LoCal addresses logical consistency. Neither provides frameworks for handling both latent information extraction and complex relational reasoning simultaneously (Kim et al. 2024; Zhao et al. 2024). The lack of systematic methods for breaking down complex, multi-faceted claims into verifiable subclaims results in incomplete coverage of intricate factual assertions. Additionally, evidence selection remains a bottleneck, with systems like BiDeV requiring separate redundancy filtering mechanisms, highlighting integrated evidence-claim alignment as an open problem (Liu et al. 2025). Furthermore, most existing multi-agent systems em-

ploy multiple agents operating within the same underlying model (Erisken et al. 2025). Such homogeneous configurations can lead to biased results, as agents may reinforce each other’s flaws due to shared training and design (Sharma, Liao, and Xiao 2024).

We propose CARP, a comprehensive multi-agent adversarial fact-checking framework with innovations throughout the fact-checking pipeline: from claim analysis to evidence extraction to final fact verification. It begins with claim decomposition, changing complex claims into atomic subclaims and constructing claim-representation graphs to resolve co-references, capture entity relationships, and extract latent information missed by traditional methods (see Section 3.3). For evidence extraction, CARP uses multi-hop knowledge retrieval that iteratively expands evidence chains to improve coverage and quality (see Section 3.4). A strategic verification planner analyzes subclaim dependencies to generate optimized, phase-based strategies to ensure logical coherence (see Section 3.5). In the reasoning and fact verification stage, CARP treats the homogeneous agent problem by using heterogeneous agents powered by different language models and organized as adversarial teams. This prevents bias amplification and echo chamber effects, promoting balanced evaluation from both supporting and refuting perspectives (see Section 3.6). The contributions of this work are:

**1) Claim Analysis and Decomposition:** A novel two-stage claim processing systematically decomposes complex claims into verifiable atomic subclaims and constructs claim-representation graphs to resolve entity ambiguities, coreferences, and latent information extraction challenges to existing decomposition methods.

**2) Multi-Hop Evidence Extraction and Synthesis:** A retrieval system that iteratively expands evidence chains through beam search-based exploration, addressing evidence redundancy problems by systematically gathering, filtering, and synthesizing comprehensive evidence for complex multi-hop reasoning while maintaining relevance and reducing noise.

**3) Strategic Pipeline Coordination and Verification Planning:** A dependency-aware coordination module that orchestrates the entire fact-checking pipeline, analyzing subclaim relationships and generating optimized verification strategies with detailed task specifications, evidence requirements, and reasoning protocols for efficient and logically coherent fact-checking execution.

**4) Adversarial Heterogeneous Reasoning Architecture:** A team-based verification system uses agents with diverse underlying language models in competing support and refutation teams for reasoning and verification, addressing the limitation of homogeneous agents by ensuring comprehensive evidence evaluation, mitigating confirmation bias, echo chamber effects, and shared model biases.

## 2 Related Works

Fact-checking approaches can be categorized into single-agent systems, which manage the entire pipeline (claim detection, evidence retrieval, and claim verification), and

multi-agent systems, which distribute tasks across specialized components.

### 2.1 Single-Agent Approaches

*Traditional Approaches:* Early fact-checking systems employed feature engineering with traditional machine learning approaches (Hassan et al. 2017). Neural networks improvements range from LSTM (Hochreiter and Schmidhuber 1997) architectures to transformer (Vaswani et al. 2017)-based models for end-to-end fact-checking. Pre-trained language models like BERT (Devlin et al. 2019) and RoBERTa (Liu et al. 2019) improve evidence retrieval and claim verification (Kotonya and Toni 2020; Vedula and Parthasarathy 2021).

*Large Language Models:* Recent work has leveraged LLMs’ reasoning capabilities across the entire fact-checking pipeline, with systems like GPT-4 (Achiam et al. 2023) for end-to-end verification (Quelle and Bovet 2024). However, these approaches face fundamental limitations including confirmation bias (Echterhoff et al. 2024; O’Leary 2025) and unfaithful explanations (Kim et al. 2024).

*Program-Guided and Planning-Based Approaches:* To address reasoning limitations, structured frameworks have emerged. ProgramFC (Pan et al. 2023) presented a framework that generates reasoning programs to guide verification through specialized sub-tasks. PACAR (Zhao et al. 2024) employed dynamic planning through multiple modules with tailored actions for numerical reasoning and entity disambiguation.

### 2.2 Multi-Agent Fact-Checking Systems

*Multi-Agent Debate and Refinement:* MADR (Kim et al. 2024) leveraged multiple LLMs as agents with diverse roles in iterative refining processes to enhance explanation faithfulness. MAD (Liang et al. 2023) used multi-agent debate more generally, for collaborative problem-solving and to encourage divergent thinking through multi-agent debate.

*Logical and Causal Reasoning with Multi-Agents:* LoCal (Ma et al. 2025) proposed a framework based on multiple LLM-based agents. A decomposing agent breaks down complex claims into simpler sub-tasks, followed by specialized reasoning agents for fact verification and question answering tasks. LoCal also adds two evaluating agents that examine logical equivalence to the original claim and assess causal consistency using counterfactual challenges, providing confidence values and iteratively correcting logical and causal errors.

*Bilateral Processing and Complex Claim Handling:* BiDeV (Liu et al. 2025) integrates multiple role-played LLMs. BiDeV addresses two critical challenges: claim vagueness and evidence redundancy. A Vagueness Defusing module identifies latent information and resolves complex relations to simplify claims, while a Redundancy Defusing module eliminates redundant content. This bilateral approach represents a departure from traditional single-path verification by systematically addressing both input complexity and evidence noise.

*Adaptive Multi-Agent Evaluation and Verification Systems:* FACT-AUDIT (Lin et al. 2025) used adaptive multi-

agents for dynamic fact-checking of large language models, leveraging agents in experiential learning and complex reasoning using importance sampling and multi-agent collaboration. The DelphiAgent (Xiong et al. 2025) trustworthy multi-agent verification framework enhances decision making transparency and mitigates hallucinations when generating justifications.

### 2.3 Research Gaps and Limitations

Critical fact-checking limitations include: (1) *Perspective Diversity*: multi-agent frameworks use homogeneous agents, repeating rather than diversifying reasoning; (2) *Claim Processing*: decomposition methods lack systematic resolution of ambiguities and complex dependencies; (3) *Strategic Planning*: verification planning remains ad-hoc and uncoordinated; (4) *Evidence Integration*: systems struggle with multi-hop reasoning and source synthesis; and (5) *Coordination Mechanisms*: agents lack sophisticated collaborative protocols.

CARP addresses these gaps, offering a novel heterogeneous multi-agent architecture with advanced coordination mechanisms that leverage complementary strengths of diverse reasoning and specialized fact-checking capabilities.

## 3 Methodology

### 3.1 Problem Statement

Given a natural language claim  $C$  that may require complex inferential reasoning over heterogeneous information sources, the goal is to predict a binary verification label  $Y \in \{SUPPORT, REFUTE\}$  and generate an interpretable justification  $J$  that articulates the reasoning pathway. While real-world claims may vary in verifiability (due to insufficient evidence, uncertainty, or ambiguity), following previous works (Ma et al. 2025; Pan et al. 2023; Zhao et al. 2024), we adopt binary classification for tractability. Claims lacking conclusive support are classified conservatively as *REFUTE*.

### 3.2 Overview of CARP

To address the above limitations, we present CARP - a multi-agent planning and adversarial reasoning framework that leverages multiple LLM agents for comprehensive evidence exploration and reasoning via competing models and perspectives. CARP enhances response quality across agents via a *draft-then-revise* editing strategy during response generation (Section 3.7). The CARP framework has five components (Figure 1), as task-specific multi-LLM agents: (1) *Claim Decomposer*: Decomposes input claim  $C$  into atomic subclaims  $S = \{s_1, s_2, \dots, s_n\}$  and constructs a claim-representation graph  $G = (S, E, R)$  encoding structural dependencies between subclaims  $S$ , named entities  $E$ , and their relationships  $R$  to eliminate ambiguity and maintain contextual clarity; (2) *Knowledge Retrieval Engine*: Retrieves relevant evidence  $\{\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_n\}$  from external knowledge sources  $\mathcal{K}$  based on subclaim requirements; (3) *Verification Planner*: Analyzes subclaim dependencies and generates a verification strategy that determines evaluation

order and required actions for each subclaim; (4) *Adversarial Verification*: Employs two teams of agents that independently evaluate each subclaim, producing competing verdicts through structured cross-review and self-correction to mitigating individual agent biases and enhance reasoning rigor; and (5) *Moderator*: Integrates adversarial outputs to produce the final verdict and interpretable justifications, ensuring both transparency and auditability.

### 3.3 Claim Decomposition

The Claim Decomposer serves as the foundational component of CARP that transforms complex input claims into atomic subclaims. This component addresses interconnected assertions that require coordinated verification.

Given an input claim  $C$ , the decomposer first generates a preliminary set of subclaims  $S_{initial} = \{s_1, s_2, \dots, s_n\}$ , where each subclaim represents a minimal verifiable assertion. This follows the principle that complex claims can be factorized into simpler, independent verification tasks that preserve the claim’s semantic content.

However, previous studies have shown that naive decomposition suffers from several limitations that can impede downstream verification accuracy (Liu et al. 2025). Consider the following example:

**Input claim:** “This was the movie that the star of Respiro featured in with Tom Cruise. Bonnie Hunt costarred in this movie.”

**Initial subclaims:**

$s_1$ : “The star of Respiro featured in a movie with Tom Cruise”

$s_2$ : “Bonnie Hunt costarred in the same movie”

$s_3$ : “This movie exists”

This decomposition exhibits three fundamental problems: (1) vague pronoun references (“the same movie”, “this movie”), (2) unresolved coreferences where the same entity is referenced using different expressions across subclaims, and (3) missing context where subclaims lose critical background needed for verification. These lead to verification failures.

**Claim-Representation Graph.** To address these limitations, our decomposer constructs a Claim-Representation Graph (CRG)  $G = (S, E, R)$  that captures the structural dependencies between subclaims  $S$ , entities  $E$ , and their relationships  $R$ . Accordingly: Graph edges allow information to flow between subclaims in any direction, handling cases where subclaims depend on each other. When an entity is mentioned multiple times using different words (such as “Respiro”, “the same movie”, and “this movie”), all mentions connect to one shared entity node, so information from one subclaim can help identify the entity in another subclaim. Relationship edges between entities enable the system to find missing context by tracing connections through the graph.

The CRG construction process involves: (1) entity extraction and disambiguation, where named entities, pronouns, and implicit references are identified and linked to canonical representations; (2) relationship modeling, which captures semantic connections between entities both within and across subclaims; and (3) dependency analysis, identifying

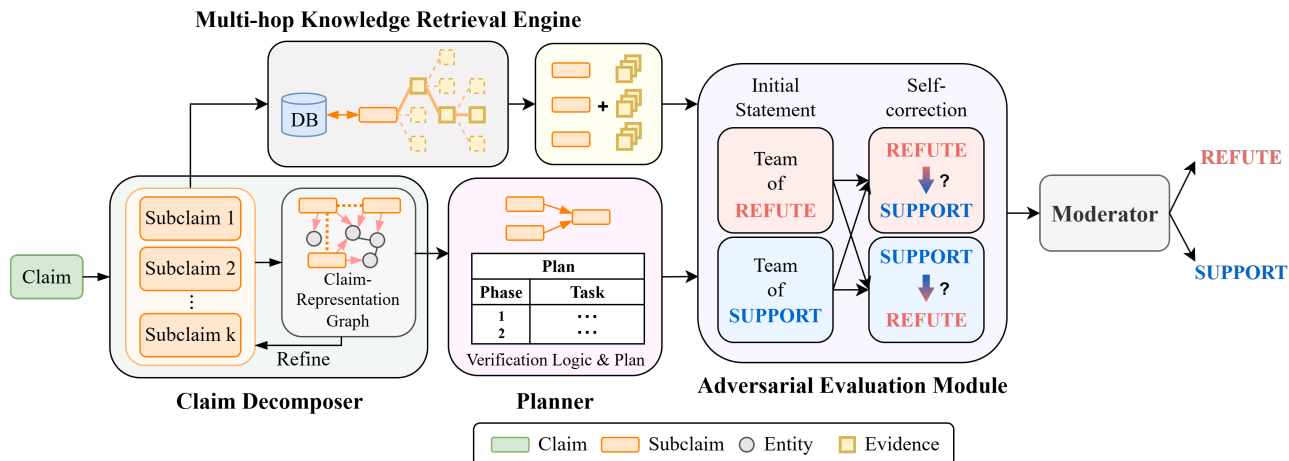


Figure 1: Overview of CARP framework architecture.

how subclaims depend on shared entities or contextual information. Using the CRG structure, the decomposer then refines the initial subclaims to resolve ambiguities and ensure independent verifiability. This refinement process transforms vague references into explicit entity mentions and incorporates necessary contextual information from related subclaims. Consider the same example:

**Refined subclaims:**

$s_1$ : “The star of Respiro featured in a movie with Tom Cruise”

$s_2$ : “Bonnie Hunt costarred in a movie with the star of Respiro and Tom Cruise”

$s_3$ : “There exists a movie featuring the star of Respiro, Tom Cruise, and Bonnie Hunt”

The example demonstrates that the refined subclaims eliminate pronoun ambiguities by explicitly referencing shared entities and provide sufficient context for independent verification. Each subclaim can be evaluated without requiring information from other subclaims, while the CRG maintains the structural relationships necessary for coherent reasoning.

**3.4 Knowledge Retrieval Engine**

Complex claims often require synthesizing information from multiple documents that cannot be identified through single-step keyword matching. We propose a multi-hop knowledge retrieval engine that employs beam search to expand document paths, gathering comprehensive evidence for complex queries.

We indexed the corpus into a vector database using dense embeddings from *NV-Embed-v2* (Lee et al. 2024), a state-of-the-art neural embedding model optimized for retrieval tasks. Documents are segmented into chunks for fine-grained retrieval. The multi-hop process retrieves evidence documents  $\{\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_n\}$  from knowledge sources  $\mathcal{K}$ , building reasoning paths iteratively starting with the original query  $q_0$ .

**Beam Search Strategy.** At each hop  $h$ , the system maintains  $k$  (beam width) active paths. Each path  $\mathcal{P}_h^{(i)} = \{\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_h\}$  represents a sequence of retrieved evidence

chunks. For each active path, we generate a contextualized query by combining the original query with accumulated evidence:  $q_h^{(i)} = \text{Combine}(q_0, \text{Context}(\mathcal{P}_h^{(i)}))$ , where *Combine* concatenates the original query with contextual information, and *Context*( $\mathcal{P}_h^{(i)}$ ) aggregates information from all evidence chunks in path  $i$ . This ensures that subsequent retrievals leverage previously gathered information, enabling the discovery of complementary evidence rather than redundant results.

We retrieve top- $m$  candidate chunks for each contextualized query, generating  $k \times m$  possible path extensions. These candidates are then ranked globally, and the top- $k$  scoring paths are selected for the next hop, preserving the most promising reasoning paths.

**Path Scoring.** We rank paths using a probabilistic framework. Let  $d_j$  denote the retrieval distance between the query and evidence chunk  $\mathcal{E}_j$  in the embedding space. The path log-likelihood is:

$$\log p(\mathcal{P}) = \sum_{j=1}^{|\mathcal{P}|} \log p(\mathcal{E}_j) = \sum_{j=1}^{|\mathcal{P}|} (\log \lambda - \lambda d_j), \quad (1)$$

where  $\lambda$  is a rate parameter controlling sensitivity to distance variations (we set  $\lambda = 1$  in our experiments). This formulation assigns higher likelihood to paths with consistently smaller retrieval distances, reflecting stronger relevance.

**Document Exclusion.** To promote diversity and prevent redundant retrieval, each path maintains an exclusion set tracking previously retrieved documents. We exclude all document chunks already present in the path. This encourages exploration across diverse sources while preventing circular retrieval patterns.

**3.5 Verification Planner**

Complex claims contain interdependent subclaims whose verification requires coordination of multiple actions. Consider the claim mentioned in Section 3.3. Without a verification planner, independent verification of subclaims leads

to inefficiencies. For example, each subclaim contains the unresolved reference "star of Respiro," which would be resolved three separate times instead of once. Additionally, each subclaim requires different verification strategies (e.g.,  $s_1$  needs factual lookup in filmographic sources, while  $s_3$  requires checking whether a movie satisfies all casting constraints.)

To address these inefficiencies, we employ a verification planner that analyzes the claim-representation graph (CRG) to construct a coordinated verification strategy. The planner receives the original claim, entities with their relationships, subclaims, and subclaim relationships as input. It produces a multi-phase verification plan that organizes subclaims into dependency-respecting phases, specifies verification actions and evidence requirements for each subclaim, and coordinates information retrieval to eliminate redundancy. We prompt the planner agent to generate a verification strategy in four stages: dependency analysis, phase construction, action specification, and resolution logic. We detail each stage below.

**Dependency Analysis.** The planner examines subclaim and entity relationships to identify verification constraints. It analyzes several dependency patterns: (1) *Logical dependencies* occur when the first subclaim affects whether the second makes sense. The planner ensures prerequisite subclaims are verified first; (2) *Temporal Dependencies* arise from sequential or causal relationships where claims about earlier events must be verified before later ones; (3) *Evidential Dependencies* occur when multiple subclaims can be verified with the same sources. The planner aggregates these to avoid repeatedly retrieving identical content; (4) *Independent Subclaims* have no connecting relationships and can be verified in parallel.

**Phase Construction.** Given dependency analysis, the planner organizes subclaims into sequential verification. Each phase includes a *phase id*, a *goal* describing what it aims to verify, and optional *dependency* indicating prerequisite phases that must be completed first. Subclaims within a phase are processed concurrently, while phases execute sequentially according to their dependencies.

**Action Specification.** For each phase, the planner specifies several verification details: (1) *Verification Actions* categorize the verification operation (e.g., check entity existence, check causal relationships, check numerical values, check event occurrence); (2) *Strategy* defines the specific method for verifying (e.g., factual lookup, logical reasoning, temporal validation); (3) *Evidence Type* identifies required evidence sources (e.g., academic studies, government databases, news reports, official documents); (4) *Query Hint* provides a search query to retrieve supporting evidence; (5) *Derived Questions (optional)* articulates the implicit question that must be answered for subclaims requiring inferential reasoning. For example, when verifying "The company achieved profitability quickly," the planner generates the derived question "What timeframe is considered quick for a company to become profitable?" to provide a temporal benchmark before checking when the company actually became profitable.

**Resolution Logic.** The planner defines how individual

subclaim verdicts combine for a final claim verdict. Here we consider binary verdicts where the claim is either *SUPPORT* or *REFUTE*. All subclaims must be verified for the overall claim to be *SUPPORTED*. This design extends to more complicated scenarios, such as claims where some but not all subclaims are verified. We describe the detailed technical specifications of the verification planner in Table 4 in the Appendix.

### 3.6 Adversarial Verification and the Moderator

Our adversarial verification module employs a structured team-based approach beyond conventional single-model debating frameworks. CARP consists of two specialized teams: the *Support Team* aims to find evidence and reasoning that validates the claim, while the *Refutation Team* seeks evidence and arguments that contradict it. Each team employs multiple agents powered by different language models (e.g., GPT, Claude, Llama, Qwen), ensuring knowledge diversity from models. This design forces comprehensive evaluation of both perspectives while leveraging diverse knowledge to mitigate confirmation bias and the risk of echo chamber effects that could emerge when all agents share the same training foundation.

The system receives retrieved documents from the knowledge retrieval engine along with the verification plan from the planner. Each team follows the phase-based verification strategy while maintaining their adversarial objectives throughout the process.

**Initial Statement Stage.** Agents from both teams independently evaluate each verification phase according to the planner's specifications. When retrieved documents lack sufficient information, due to knowledge gaps in the external database or outdated information, agents supplement their analysis using their parametric knowledge gained during pre-training. Support team agents identify evidence, logical connections, and contextual factors that validate the claim. Refutation team agents seek contradictory evidence, logical inconsistencies, or alternative interpretations that challenge validity.

The team-playing instructions emphasize thoroughness over conviction, directing agents to find the possible evidence for their assigned position. Thus, even when evidence appears to heavily favor one side, both perspectives receive comprehensive examination. Agents openly express when they cannot find sufficient evidence for their position, promoting intellectual honesty.

**Self-Correction Stage.** Both teams then engage in a structured review where they receive and critically examine all statements from the opposing team. This adversarial cross-examination phase identifies weaknesses, factual errors, or logical flaws in opposing arguments. Here, agents can reassess their original stance based on the opposing team's evidence and reasoning. They must provide explicit explanations for any stance changes along with updated confidence scores, creating a transparent record of how evidence evaluation evolves through adversarial interaction.

**Moderated Final Verdict.** A neutral moderator agent, separate from both adversarial teams, reviews the complete debating record to render a final verdict. It's decision incor-

porates both the strength of evidence and the quality of reasoning demonstrated throughout the adversarial process.

### 3.7 Draft-then-revise Strategy

Dividing writing into distinct stages—planning, drafting, and revising—reduces cognitive load and leads to higher-quality text (Flower and Hayes 1981). Accordingly, when querying agents’ responses, we employ a *draft-then-revise* strategy to enhance the completeness, coherence, and factual accuracy of generated responses.

In the draft phase, agents explore possible content and generate comprehensive initial outputs. In subsequent revision, agents refine the draft by improving logical structure, resolving inconsistencies, correcting errors, and ensuring adherence to task requirements.

We investigate two variants of our CARP framework: *single-LLM agent self-refinement* and *multi-LLM agent collaborative editing*. In the single-model configuration, where one language model serves as the agent, the agent first generates an initial response focusing on content completeness, then performs self-refinement (Madaan et al. 2023) by critiquing and revising its own work to address gaps, errors, and requirement misalignments.

The multi-model variant employs several LLM models to independently generate initial responses in parallel, followed by a designated primary model that synthesizes the best ideas and insights from all drafts to produce a unified response.

## 4 Experimental Setup

We conducted extensive evaluations to assess CARP performance across multiple datasets (detailed in Section 4.1) and compare its effectiveness against various baselines (detailed in Section 4.2).

Following previous works (Pan et al. 2023; Zhao et al. 2024; Ma et al. 2025), we evaluate CARP across three different, uniquely challenging knowledge accessibility settings. **(1) Gold evidence:** The system is supplied with a curated set of evidence documents  $\mathcal{E}_{\text{gold}}$  deemed sufficient for claim verification. The challenge lies in accurately interpreting this evidence and performing logical inference to reach a verdict. **(2) Open-book:** In this setting, agents are allowed to retrieve claim-related information from extensive external repositories  $\mathcal{K}$  (e.g., Wikipedia). This setup evaluates all aspects of the overall framework, including claim detection, evidence retrieval, and claim verification capabilities, which together constitute the core focus of our work. **(3) Closed-book:** This most constrained setting requires agents to verify claims using only their internalized knowledge ( $\mathcal{K} = \emptyset$ ), without external information access. Success in this setting relies entirely on the agents’ ability to reason over their pre-trained parametric knowledge.

### 4.1 Datasets

We examine CARP on two datasets that need complex multi-hop reasoning and evidence integration: *HOVER*<sup>1</sup> (Jiang

<sup>1</sup><https://hover-nlp.github.io/>

et al. 2020) and *FEVEROUS*<sup>2</sup> (Aly et al. 2021). We follow the conservative verification principle where claims without conclusive supporting evidence are classified as *REFUTE*.

**HOVER** is a multi-hop fact extraction and verification dataset where claims require evidence from 2 to 4 Wikipedia articles. In this dataset, the number of “hops” corresponds to the number of supporting documents needed for verification, with each hop representing a reasoning step that connects information across different articles. For our experiments, we use the development set containing 4,000 claims: 1,126 two-hop, 1,835 three-hop, and 1,039 four-hop claims.

**FEVEROUS** is the first large-scale fact-checking dataset incorporating both textual sentences and structured table data as evidence sources. Claims are manually constructed and verified against Wikipedia pages, with evidence drawn from sentences, table cells, or combinations thereof. Following (Pan et al. 2023), we focus on claims requiring exclusively sentence-based evidence, resulting in 2,962 claims from the development set (FEVEROUS-S). This subset enables evaluation of complex textual reasoning while maintaining consistency with HOVER’s evidence format. The knowledge corpus for HOVER consists of introductory sections from 5.2 million Wikipedia articles (October 2017 dump), while FEVEROUS uses 5.4 million full Wikipedia articles from the December 2020 dump.

### 4.2 Baseline Methods

To demonstrate the effectiveness of CARP, we compare it against representative baseline methods across four distinct categories:

**1. Pre-trained Models:** These methods utilize transformer-based pre-trained models for fact-checking without additional task-specific training. We include *Bert-FC* (Soleimani, Monz, and Worring 2020), which applies BERT (Devlin et al. 2019) for evidence retrieval and claim verification in a standard pipeline configuration, and *LisT5* (Jiang, Pradeep, and Lin 2021), which uses T5 (Raffel et al. 2020) for improved performance on complex reasoning tasks.

**2. Task-Specific Fine-tuned Models:** This category encompasses methods that fine-tune pre-trained models on fact-checking or natural language inference datasets. *RoBERTa-NLI* (Nie et al. 2019) fine-tunes RoBERTa on multiple Natural Language Inference (NLI) datasets to enhance reasoning. *DeBERTaV3-NLI* (He, Gao, and Chen 2021) leverages the advanced DeBERTaV3 architecture fine-tuned on both FEVER (Thorne et al. 2018) and NLI datasets. *MUTIVERS* (Wadden et al. 2021) represents models specifically trained on the FEVER dataset for fact extraction and verification tasks.

**3. LLM-based Approaches:** These methods directly employ LLMs with designed prompts for fact-checking. *Codex* (Chen et al. 2021) and *Flan-T5* (Chung et al. 2024) are two baselines that leverage LLMs in a few-shot manner (with examples in the prompt).

**4. Multi-agent-based Frameworks:** This category includes recent approaches that employ multiple agents for

<sup>2</sup><https://fever.ai/dataset/feverous.html>

fact-checking subtasks. *ProgramFC* (Pan et al. 2023) uses LLM-based agents to generate and execute reasoning programs step-by-step for claim verification. *PACAR* (Zhao et al. 2024) introduces specialized agents for claim decomposition, self-reflection, and planning within a centralized coordination framework. *LoCal* (Ma et al. 2025) employs multiple LLM-based agents to break down claims into subtasks using specialized reasoning agents, then employs dual evaluating agents to verify logical equivalence and causal consistency iteratively until both agents accept veracity.

### 4.3 Implementation Details

CARP uses both open-source and commercial language models to demonstrate the framework’s adaptability across different computational capabilities and resource constraints. For open-source configurations, we use *Llama-3.1-8B-Instruct*<sup>3</sup> and *Qwen2-7B-Instruct*<sup>4</sup>, as relatively lightweight yet capable models that are accessible to most research institutes and laboratories with limited computational budgets. For commercial model deployments, we utilize *GPT-4o*<sup>5</sup> and *Claude-3.5-Sonnet*<sup>6</sup>, which are among the most powerful commercial LLMs available, allowing us to evaluate the upper-bound performance capacity of our framework under optimal conditions.

Each component in CARP is implemented as a specialized multi-LLM agent with distinct functional responsibilities. We examine both *single-LLM deployments*, where all agents use a single underlying language model, and *multi-LLM ensembles*, where agents harness multiple diverse language models to enhance reasoning. In our multi-LLM experimental settings, each agent uses exactly two complementary models: *Llama-3.1-8B-Instruct* paired with *Qwen2-7B-Instruct* for cost-effective open-source deployments, and *GPT-4o* combined with *Claude-3.5-Sonnet* for high-performance commercial implementations.

The Knowledge Retrieval Engine employs *NV-Embed-v2* (Lee et al. 2024) for document indexing and embedding generation, with all vectors stored and managed through a Chroma vector database<sup>7</sup>. Our multi-hop retrieval strategy maintains a beam width of 3 to balance exploration breadth with computational efficiency, while imposing a maximum limit of 4 hops to prevent potentially infinite expansion during evidence chain construction. At each retrieval hop, the system selects the top-10 most relevant documents to ensure comprehensive coverage.

## 5 Experimental Results and Discussion

In our CARP experiments, we evaluate two distinct model configurations to demonstrate the framework’s flexibility and scalability: **single-LLM deployments** using either *Llama* (*Llama-3.1-8b*) or *Qwen* (*Qwen2-7b*) as the underlying language model, and **multi-LLM ensembles** combining *Llama* and *Qwen* or *GPT-4o* and *Claude-3.5*. The latter

configuration serves as the strong upper bound of our framework, representing the potential performance ceiling when powered by the most capable commercial models available. We present a case study in the Appendix (see Figure 3) that traces CARP’s step-by-step verification process, from initial claim decomposition through the final verdict, using *GPT-4o* and *Claude-3.5* as the underlying LLMs. In this section, our analysis focuses on understanding how CARP performs across varying reasoning complexities and evidence availability constraints. We organize our findings around the following research questions (RQs):

### RQ1: How does CARP perform when evidence is available (gold evidence and open-book settings)?

Table 1 presents the experimental results across both gold evidence and open-book settings on the HOVER and FEVEROUS-S datasets. Our analysis examines performance across three dimensions: reasoning complexity effects, underlying model capabilities, and deployment configurations (elaborated in findings 1-3).

**Finding 1:** *Performance degradation increases with reasoning complexity.* Across both settings, all methods show performance decline as reasoning complexity increases from 2-hop to 4-hop scenarios. In gold evidence settings, CARP configurations experience degradation ranging from 5.04 points (*GPT-4o+Claude-3.5*: 78.32 → 73.28) to 6.73 points (*Qwen2-7b*: 76.91 → 70.18), comparable to the strong baselines: *PACAR* (6.91-point decline) and *LoCal* (9.79-point drop). The pattern intensifies in open-book settings, where CARP shows 9.53 to 11.73 point declines across configurations, similar to *PACAR*’s 9.31-point drop but outperforming *LoCal*’s severe 11.12-point degradation. These results suggest claims needing longer reasoning chains are harder to verify. CARP demonstrates superior robustness as hop numbers increase, maintaining more stable performance than baseline methods.

**Finding 2:** *Open-source models achieve competitive performance under a well-designed framework.* Comparing model capabilities reveals that open-source LLMs deliver competitive results when supported by qualified evidence and effective framework design, though advanced commercial models maintain consistent advantages. In this analysis, we consider multi-LLM ensemble configurations: CARP (*Llama-3.1-8b+Qwen2-7b*) versus CARP (*GPT-4o+Claude-3.5*) for fair comparison. Commercial models outperform open-source configurations by modest margins ranging from 1.48 to 2.62 points across complexity levels in gold evidence settings and from 0.33 to 2.53 points in open-book settings. On FEVEROUS-S, the gaps are even smaller at only 1.16/1.83 points for gold/open-book settings, respectively. These results indicate that well-designed frameworks with qualified evidence can substantially narrow the performance gap between commercial and open-source models, enabling cost-effective deployment while maintaining competitive verification accuracy across diverse reasoning complexities.

**Finding 3:** *Multi-LLM ensembles outperform single-LLMs.* By comparing multi-LLM ensemble configuration CARP (*Llama-3.1-8b+Qwen2-7b*) with single-LLM deployments:

<sup>3</sup><https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct>

<sup>4</sup><https://huggingface.co/Qwen/Qwen2-7B-Instruct>

<sup>5</sup><https://openai.com/index/hello-gpt-4o/>

<sup>6</sup><https://www.anthropic.com/news/claude-3-5-sonnet>

<sup>7</sup><https://docs.trychroma.com/docs/overview/introduction>

Gold Evidence &amp; Open-book Settings)

Category	Model	HOVER (2-hop)		HOVER (3-hop)		HOVER (4-hop)		FEVEROUS-S	
		Gold	Open	Gold	Open	Gold	Open	Gold	Open
Pretrained	BERT-FC	53.40	50.86	50.90	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
Fine-tuned	RoBERTa-NLI	74.62	63.62	62.23	53.99	57.98	52.40	88.28	57.80
	DeBERTaV3-NLI	77.22	68.72	65.98	60.76	60.49	56.00	91.98	58.81
	MULTIVERS	68.86	60.17	59.87	52.55	55.67	51.86	86.03	56.61
LLM-based	Codex	70.63	65.07	66.46	56.63	63.49	57.27	89.77	62.58
	FLAN-T5	73.69	66.02	65.66	60.23	58.08	55.42	90.81	63.73
Multi-agent	ProgramFC (FLAN-T5)	75.65	70.30	66.48	63.43	66.75	57.74	92.69	68.06
	PACAR (GPT-3.5-Turbo)	76.86	73.13	70.10	64.07	69.95	63.82	<u>94.43</u>	<u>72.61</u>
	LoCal (GPT-3.5-Turbo)	<b>79.93</b>	72.71	73.26	64.11	70.14	61.59	91.09	68.22
Ours	CARP (Llama-3.1-8b)	76.53	73.62	72.87	64.78	70.09	61.24	92.78	68.54
	CARP (Qwen2-7b)	76.91	73.17	72.64	64.93	70.18	61.70	92.91	69.18
	CARP (GPT-4o)	<u>78.48</u>	<u>74.02</u>	<u>74.12</u>	<u>65.63</u>	<u>72.91</u>	<u>64.09</u>	<u>94.03</u>	<u>72.57</u>
	CARP (Claude-3.5)	77.97	73.73	73.83	65.70	71.56	62.97	93.55	72.31
	CARP (Llama-3.1-8b+Qwen2-7b)	76.73	<u>74.56</u>	73.20	<u>65.48</u>	70.66	62.83	93.34	71.32
	CARP (GPT-4o+Claude-3.5)	78.32	<b>74.89</b>	<b>74.68</b>	<b>66.52</b>	<b>73.28</b>	<b>65.36</b>	<b>94.50</b>	<b>73.15</b>

Table 1: Macro-F1 scores of CARP compared to baseline methods on the HOVER and FEVEROUS-S datasets. Evaluation is conducted under both gold evidence (with provided gold evidence) and open-book (with retrieved evidence) settings. Baseline methods are categorized into: pre-trained transformer models, fine-tuned models, LLM-based approaches, and multi-agent-based frameworks (detailed in *Baseline Methods* in the *Experimental Setup* section). The highest and second-highest scores in each column are highlighted in bold and underlined text, respectively. The underlying LLMs for verdict generation used in each framework configuration are shown in parentheses (all models listed are used without fine-tuning).

Closed-book Setting

Model	HOVER			FVRS-S
	2-hop	3-hop	4-hop	
Codex	55.57	53.42	45.59	57.85
FLAN-T5	48.27	52.11	51.13	55.16
ProgramFC	54.27	54.18	52.88	59.66
LoCal	57.51	53.60	53.89	59.47
CARP (Llama)	51.26	50.57	48.65	54.12
CARP (Qwen)	56.98	55.83	53.60	57.54
CARP (GPT)	<b>71.74</b>	<b>60.09</b>	<b>58.96</b>	<b>68.49</b>
CARP (Claude)	68.08	55.89	54.33	64.73
CARP (Llama+Qwen)	52.82	50.72	50.17	55.72
CARP (GPT+Claude)	<u>68.86</u>	<u>56.36</u>	<u>55.01</u>	<u>65.62</u>

Table 2: Macro-F1 scores of CARP compared to baseline methods under the closed-book setting. The FEVEROUS-S dataset is abbreviated as “FVRS-S”. Parentheses for CARP denote the underlying LLMs used in each configuration. “Llama” indicates Llama-3.1-8b; “Qwen” represents Qwen2-7b; “GPT” refers to GPT-4o, and “Claude” refers to Claude-3.5.

CARP (Llama-3.1-8b) and CARP (Qwen2-7b), we note that heterogeneous model combinations deliver superior performance compared to individual models. On HOVER, CARP (Llama+Qwen) outperforms its individual component models by 0.2 to 1.59 points. On FEVEROUS-S, the multi-LLM configuration achieves 0.43 to 2.78 point improvements over single models. We see similar trends with advanced commercial models: CARP (GPT-4o+Claude-3.5) versus CARP (GPT-4o) and CARP (Claude-3.5) suggests that model di-

versity provides complementary reasoning capabilities using the adversarial framework, with consistent ensemble benefits across all experimental conditions.

#### RQ2: How does CARP perform with missing evidence?

Table 2 presents the results under closed-book settings on the HOVER and FEVEROUS-S datasets. This isolates parametric knowledge contributions by assessing reasoning without external evidence, splitting framework design effects from the retrieval system.

**Finding 4:** *Current LLMs show limited effectiveness in verifying complex claims with only parametric knowledge.* The closed-book results reveal fundamental limitations in current LLMs’ ability to handle fact verification tasks using only parametric knowledge. The advanced commercial models (GPT and Claude) achieve only 58.96 and 54.33, respectively, on HOVER 4-hop scenarios, while small-scale open-source models (Llama-3.1-8b and Qwen2-7b) struggle more. The current LLMs appear to lack parametric knowledge and reasoning capabilities to verify complex, multi-faceted claims without external evidence.

**Finding 5:** *Knowledge conflicts in multi-LLM ensembles produce conservative verification behavior without external evidence.* Unlike open-book scenarios, where ensemble configurations generally outperform individual models, closed-book results show knowledge dilution. CARP (Llama+Qwen) performs between its individual components: underperforming Qwen while outperforming Llama. Error analysis shows more false negatives than false positives; the moderator appears to adopt conservative “REFUTED” decisions when models disagree without additional knowledge to resolve conflicts. A similar phenomenon

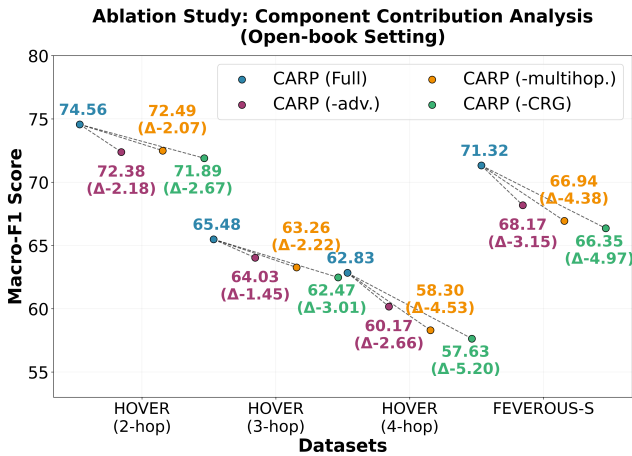


Figure 2: Ablation study results for CARP (Llama-3.1-8b+Qwen2-7b) in open-book settings show Macro-F1 scores for the full model (blue), without adversarial evaluation (purple), without multi-hop retrieval (orange), and without the Claim-Representation Graph (green). Dashed lines indicate performance drops ( $\Delta$ ) for each removed component.

occurs with advanced commercial models. However, conservative, cautious behavior may be practically preferable to overconfident false claims.

## 6 Ablation Study

To examine the individual impact of key components within the CARP framework under the open-book setting, we conduct ablation studies using the open-source *Llama-3.1-8b* and *Qwen2-7b* as the underlying models due to resource and cost constraints of commercial models. Figure 2 presents the results across different reasoning complexities and datasets.

### RQ3: How does the adversarial evaluation module contribute to CARP’s verification performance?

The “-adv.” configuration removes our adversarial evaluation mechanism. Models simply make decisions based on the verification plan and retrieved documents without multi-perspective reasoning. In this simplified approach, the system bypasses the adversarial debate process and relies on individual model decision-making rather than collaborative reasoning through competing teams.

**Finding 6:** *The adversarial evaluation module benefits scale with reasoning complexity.* Removing the adversarial evaluation module consistently drops performance across all experimental settings: 2.18/1.45/2.66 points on HOVER 2-hop/3-hop/4-hop, respectively, and 3.15 points on FEVEROUS-S. These results indicate that adversarial debate provides substantial benefits beyond computational overhead, particularly for complex reasoning scenarios where evidence interpretation and contradiction resolution are critical.

### RQ4: Is Multi-hop Knowledge Retrieval Effective?

The “-multihop.” configuration disables our beam search strategy of the knowledge retrieval engine, instead employ-

ing a simplified approach that returns the top- $k$  documents ( $k = 4$  in our experiments) for every query.

**Finding 7:** *Multi-hop retrieval becomes increasingly critical as reasoning complexity escalates.* Removing multi-hop capabilities results in performance degradation that amplifies with claim complexity: degradation increases from 2.07 points (HOVER 2-hop) to 4.53 points (HOVER 4-hop), with FEVEROUS-S experiencing a substantial 4.38-point drop. The progressive increase in performance degradation shows that iterative evidence expansion is essential with longer reasoning chains and more diverse information sources. Comprehensive retrieval design is crucial for effective verification, particularly given that current LLMs show limited effectiveness in verifying complex claims through memorized knowledge alone without high-quality external evidence (see also Finding 4).

### RQ5: Does Claim-Representation Graph (CRG) refinement improve subclaim quality for the verification?

The “-CRG” configuration removes the Claim-Representation Graph, bypassing subclaim refinement so that the planner and knowledge retrieval engine are fed directly by the initial, unrefined subclaims. This isolates the contribution of structured claim decomposition to downstream reasoning and retrieval.

**Finding 8:** *CRG subclaim refinement is increasingly vital as claim complexity grows.* Removing the Claim Representation Graph (CRG) consistently reduces performance across all settings, with decreases of 2.67, 3.01, and 5.20 points on HOVER’s 2-hop, 3-hop, and 4-hop tasks, respectively. Additionally, there is a 4.97 point drop on the FEVEROUS-S dataset. This performance decline is greater than the degradation observed when either the adversarial module or multi-hop retrieval is removed. Notably, the largest performance drops are seen on HOVER 4-hop ( $\Delta$ -5.20) and FEVEROUS-S ( $\Delta$ -4.97). This degradation increases significantly with the complexity of reasoning, indicating that longer reasoning chains depend more heavily on well-structured subclaims. As discussed in Section 3.3, poorly defined subclaims introduce structural ambiguity through vague pronouns, unresolved co-references, and missing context. This noise accumulates across every downstream component as complexity increases.

## 7 Computational Cost Analysis

Table 3 summarizes the estimated computational cost of CARP per claim across configurations and reasoning complexities, verified against 50 claims from the HOVER development set. Multi-LLM configurations (with two heterogeneous LLMs) incur approximately  $1.5\times$  more calls than single-LLM configurations due to the draft-then-revise strategy: multi-LLM agents run two parallel drafts plus one synthesis call per action, whereas single-LLM agents run one draft plus one self-revision (see Section 3.7). Differences in token usage across commercial and open-source configurations reflect model output verbosity; differences in latency reflect API throughput versus local GPU inference.

Model	Hop	Calls	Tokens	Latency
<b>Multi-LLM</b>				
CARP (GPT+Claude)	2-hop	~34	~140K	95–125
CARP (GPT+Claude)	3-hop	~46	~210K	120–150
CARP (GPT+Claude)	4-hop	~58	~255K	135–170
CARP (Llama+Qwen)	2-hop	~34	~115K	50–75
CARP (Llama+Qwen)	3-hop	~46	~175K	65–90
CARP (Llama+Qwen)	4-hop	~58	~220K	80–110
<b>Single-LLM</b>				
CARP (GPT)	3-hop	~29	~180K	85–115
CARP (Llama)	3-hop	~29	~90K	55–80

Table 3: Estimated computational cost per claim. “Calls” represent the total number of individual LLM inference calls per claim; “Tokens” indicate the cumulative token count across all calls; “Latency” is the processing time in seconds for a single claim. Commercial setups use *GPT-4o* and *Claude-3.5-Sonnet* APIs, while open-source configurations run on a single NVIDIA A6000 48GB GPU for each foundation model (*Llama-3.1-8B-Instruct* and *Qwen2-7B-Instruct*). Single-LLM results are reported at 3-hops only as a baseline for assessing the cost-benefit tradeoff of Multi-LLM configurations.

## 8 Concluding Remarks

**Contribution Summary:** We propose CARP, a novel multi-agent adversarial claim verification framework. Experiments using HOVER and FEVEROUS datasets demonstrate significant improvements over existing approaches across complexity levels and knowledge accessibility constraints. Adversarial evaluation and multi-hop retrieval significantly boost performance, with benefits that scale with reasoning complexity.

**Limitations:** The adversarial multi-agent architecture and multi-hop retrieval processes increase computational demands. Our evaluation has overlooked nuanced classifications like “partially supported” or “misleading context,” collapsing complex epistemic states into a binary verdict. This is an important gap, as real-world misinformation rarely falls cleanly into either category. Lastly, the FEVEROUS and HOVER datasets consist of factual claims in neutral language, omitting the emotionally charged multimodal content encountered in practice.

**Future work:** CARP establishes a novel automated fact-checking paradigm, supporting decision makers with multi-perspective analysis and robust detection of misleading content. In the future we explore avenues to mitigate these limitations - extending the label space to graduated verdicts (e.g., *partially supported*, *misleading context*, *refuted*), and investigating whether agent disagreement within our deliberative framework can be surfaced as calibrated confidence scores rather than resolved into a binary outcome; incorporating multimodal capabilities, developing adaptive and scalable coordination mechanisms; and integrating real-time information sources.

## Ethical Statement

Our work was conducted on publicly available data with no concerns for human subjects (IRB exempt). However, all automated fact-checking systems raise ethical considerations. These influence public discourse and information consumption, with potential unintended suppression of legitimate viewpoints. All LLMs also introduce the risk of bias associated with their training data. Final editorial decisions in using CARP should remain with qualified domain experts who can account for context, culture, and nuance beyond the systems’ capabilities.

## Acknowledgments

The authors acknowledge the Ohio Supercomputing Center for computational resources provided under project PAS2030, which supported the development of the CARP system.

## References

- Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Allan, K.; Azcona, J.; Sripada, S.; Leontidis, G.; Sutherland, C. A.; Phillips, L. H.; and Martin, D. 2025. Stereotypical bias amplification and reversal in an experimental model of human interaction with generative artificial intelligence. *Royal Society Open Science*, 12(4): 241472.
- Aly, R.; Guo, Z.; Schlichtkrull, M.; Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; Cocarascu, O.; and Mittal, A. 2021. Feverous: Fact extraction and verification over unstructured and structured information. *arXiv preprint arXiv:2106.05707*.
- Caceres, M. M. F.; Sosa, J. P.; Lawrence, J. A.; Sestacovschi, C.; Tidd-Johnson, A.; Rasool, M. H. U.; Gadamidi, V. K.; Ozair, S.; Pandav, K.; Cuevas-Lou, C.; et al. 2022. The impact of misinformation on the COVID-19 pandemic. *AIMS public health*, 9(2): 262.
- Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H. P. D. O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70): 1–53.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. 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)*, 4171–4186.
- Echterhoff, J.; Liu, Y.; Alessa, A.; McAuley, J.; and He, Z. 2024. Cognitive bias in decision-making with LLMs. *arXiv preprint arXiv:2403.00811*.

- Erisken, S.; Gothard, T.; Leitgab, M.; and Potham, R. 2025. MAEBE: Multi-Agent Emergent Behavior Framework. *arXiv preprint arXiv:2506.03053*.
- Flower, L.; and Hayes, J. R. 1981. A cognitive process theory of writing. *College Composition & Communication*, 32(4): 365–387.
- FORCE11. 2020. The FAIR Data principles. <https://force11.org/info/the-fair-data-principles/>.
- Geburu, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J. W.; Wallach, H.; Iii, H. D.; and Crawford, K. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92.
- Gisondi, M. A.; Barber, R.; Faust, J. S.; Raja, A.; Strehlow, M. C.; Westafer, L. M.; and Gottlieb, M. 2022. A deadly infodemic: social media and the power of COVID-19 misinformation.
- Guo, Z.; Schlichtkrull, M.; and Vlachos, A. 2022. A survey on automated fact-checking. *Transactions of the association for computational linguistics*, 10: 178–206.
- Hassan, N.; Arslan, F.; Li, C.; and Tremayne, M. 2017. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 1803–1812.
- He, P.; Gao, J.; and Chen, W. 2021. Deberv3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.
- Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.
- Jiang, K.; Pradeep, R.; and Lin, J. 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.
- Jiang, Y.; Bordia, S.; Zhong, Z.; Dognin, C.; Singh, M.; and Bansal, M. 2020. HoVer: A dataset for many-hop fact extraction and claim verification. *arXiv preprint arXiv:2011.03088*.
- Kim, K.; Lee, S.; Huang, K.-H.; Chan, H. P.; Li, M.; and Ji, H. 2024. Can llms produce faithful explanations for fact-checking? towards faithful explainable fact-checking via multi-agent debate. *arXiv preprint arXiv:2402.07401*.
- Kotonya, N.; and Toni, F. 2020. Explainable automated fact-checking: A survey. *arXiv preprint arXiv:2011.03870*.
- Lee, C.; Roy, R.; Xu, M.; Raiman, J.; Shoeybi, M.; Catanzaro, B.; and Ping, W. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.
- Liang, T.; He, Z.; Jiao, W.; Wang, X.; Wang, Y.; Wang, R.; Yang, Y.; Shi, S.; and Tu, Z. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.
- Lin, H.; Deng, Y.; Gu, Y.; Zhang, W.; Ma, J.; Ng, S.-K.; and Chua, T.-S. 2025. Fact-audit: An adaptive multi-agent framework for dynamic fact-checking evaluation of large language models. *arXiv preprint arXiv:2502.17924*.
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Liu, Y.; Sun, H.; Guo, W.; Xiao, X.; Mao, C.; Yu, Z.; and Yan, R. 2025. Bidev: Bilateral defusing verification for complex claim fact-checking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 541–549.
- Lu, R.-S.; Lin, C.-C.; and Tsao, H.-Y. 2024. Empowering large language models to leverage domain-specific knowledge in e-learning. *Applied Sciences*, 14(12): 5264.
- Luo, Z.; Xu, C.; Zhao, P.; Geng, X.; Tao, C.; Ma, J.; Lin, Q.; and Jiang, D. 2023. Augmented large language models with parametric knowledge guiding. *arXiv preprint arXiv:2305.04757*.
- Ma, J.; Hu, L.; Li, R.; and Fu, W. 2025. Local: Logical and causal fact-checking with llm-based multi-agents. In *Proceedings of the ACM on Web Conference 2025*, 1614–1625.
- Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegrefe, S.; Alon, U.; Dziri, N.; Prabhunoye, S.; Yang, Y.; et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36: 46534–46594.
- Nie, Y.; Williams, A.; Dinan, E.; Bansal, M.; Weston, J.; and Kiela, D. 2019. Adversarial NLI: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.
- O’Leary, D. E. 2025. Confirmation and specificity biases in large language models: An explorative study. *IEEE Intelligent Systems*, 40(1): 63–68.
- Pan, L.; Wu, X.; Lu, X.; Luu, A. T.; Wang, W. Y.; Kan, M.-Y.; and Nakov, P. 2023. Fact-checking complex claims with program-guided reasoning. *arXiv preprint arXiv:2305.12744*.
- Qin, C.; Zhang, A.; Zhang, Z.; Chen, J.; Yasunaga, M.; and Yang, D. 2023. Is ChatGPT a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.
- Quelle, D.; and Bovet, A. 2024. The perils and promises of fact-checking with large language models. *Frontiers in Artificial Intelligence*, 7: 1341697.
- Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140): 1–67.
- Sharma, N.; Liao, Q. V.; and Xiao, Z. 2024. Generative echo chamber? effect of llm-powered search systems on diverse information seeking. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 1–17.
- Soleimani, A.; Monz, C.; and Worring, M. 2020. Bert for evidence retrieval and claim verification. In *European Conference on Information Retrieval*, 359–366. Springer.

Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; and Mittal, A. 2018. FEVER: a large-scale dataset for fact extraction and VERification. *arXiv preprint arXiv:1803.05355*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Vedula, N.; and Parthasarathy, S. 2021. Face-keg: Fact checking explained using knowledge graphs. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 526–534.

Vosoughi, S.; Roy, D.; and Aral, S. 2018. The spread of true and false news online. *science*, 359(6380): 1146–1151.

Wadden, D.; Lo, K.; Wang, L. L.; Cohan, A.; Beltagy, I.; and Hajishirzi, H. 2021. MultiVerS: Improving scientific claim verification with weak supervision and full-document context. *arXiv preprint arXiv:2112.01640*.

Xiong, C.; Zheng, G.; Ma, X.; Li, C.; and Zeng, J. 2025. DelphiAgent: A trustworthy multi-agent verification framework for automated fact verification. *Information Processing & Management*, 62(6): 104241.

Zhao, X.; Wang, L.; Wang, Z.; Cheng, H.; Zhang, R.; and Wong, K.-F. 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.

## Paper Checklist

### 1. For most authors...

- (a) Would answering this research question advance science without violating social contracts, such as violating privacy norms, perpetuating unfair profiling, exacerbating the socio-economic divide, or implying disrespect to societies or cultures? **Yes**.
- (b) Do your main claims in the abstract and introduction accurately reflect the paper’s contributions and scope? **Yes**.
- (c) Do you clarify how the proposed methodological approach is appropriate for the claims made? **Yes, we justified each methodological component by connecting it to specific limitations in existing approaches in the Methodology section**.
- (d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? **No, our evaluation uses established benchmark datasets (HOVER and FEVEROUS) that do not contain population-specific demographic information**.
- (e) Did you describe the limitations of your work? **Yes, we discussed the potential limitations in the Concluding Remarks section**.
- (f) Did you discuss any potential negative societal impacts of your work? **No, we did not explicitly discuss potential negative societal impacts. However, in the**

*Concluding Remarks* section, we acknowledged limitations that have societal implications: our system’s computational demands may limit accessibility, and our evaluation on neutral factual claims omits the emotionally charged multimodal content typical of real-world misinformation.

- (g) Did you discuss any potential misuse of your work? **No, our limitations discussion focuses on technical constraints but does not elaborate on the potential misuse cases**.
  - (h) Did you describe steps taken to prevent or mitigate potential negative outcomes of the research, such as data and model documentation, data anonymization, responsible release, access control, and the reproducibility of findings? **Yes, for reproducibility and documentation, we used public benchmark datasets and provided comprehensive implementation details, including specific model versions, hyperparameters, and infrastructure choices. We also documented technical specifications and a detailed case study in the Appendix**.
  - (i) Have you read the ethics review guidelines and ensured that your paper conforms to them? **Yes**.
- ### 2. Additionally, if your study involves hypotheses testing...
- (a) Did you clearly state the assumptions underlying all theoretical results? **NA. Our study does not involve formal hypothesis testing**.
  - (b) Have you provided justifications for all theoretical results? **NA**
  - (c) Did you discuss competing hypotheses or theories that might challenge or complement your theoretical results? **NA**
  - (d) Have you considered alternative mechanisms or explanations that might account for the same outcomes observed in your study? **NA**
  - (e) Did you address potential biases or limitations in your theoretical framework? **NA**
  - (f) Have you related your theoretical results to the existing literature in social science? **NA**
  - (g) Did you discuss the implications of your theoretical results for policy, practice, or further research in the social science domain? **NA**
- ### 3. Additionally, if you are including theoretical proofs...
- (a) Did you state the full set of assumptions of all theoretical results? **NA. Our study does not include theoretical proofs**.
  - (b) Did you include complete proofs of all theoretical results? **NA**
- ### 4. Additionally, if you ran machine learning experiments...
- (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? **No, we used public benchmark datasets and provided comprehensive implementation details, but we did not release code**.

- (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? **Yes.**
  - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? **No, we reported single-run results without error bars or variance across multiple seeds.**
  - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? **No, we did not include compute resources, GPU specifications, or runtime costs.**
  - (e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? **Yes, we justified our evaluation approach across three knowledge accessibility settings (gold evidence, open-book, closed-book) on two benchmark datasets requiring complex multi-hop reasoning in the *Experimental Setup* section.**
  - (f) Do you discuss what is “the cost“ of misclassification and fault (in)tolerance? **No, we did not provide a comprehensive analysis of misclassification costs or fault tolerance mechanisms. However, the Finding 5 (in the *Experimental Results and Discussion* section) notes that our system exhibits conservative behavior with more false negatives than false positives in the closed-book settings, which may be practically preferable to overconfident false claims that could spread misinformation.**
5. Additionally, if you are using existing assets (e.g., code, data, models) or curating/releasing new assets, **without compromising anonymity...**
- (a) If your work uses existing assets, did you cite the creators? **Yes.**
  - (b) Did you mention the license of the assets? **No, we did not specify licenses for the datasets, models, or other assets used.**
  - (c) Did you include any new assets in the supplemental material or as a URL? **No, we did not release code, models, or new datasets.**
  - (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? **NA.** We use existing public benchmark datasets derived from Wikipedia. Consent considerations were addressed by the original dataset creators.
  - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? **No, the datasets consist of factual claims about public figures and events derived from Wikipedia, but we did not explicitly discuss personally identifiable information or offensive content.**
  - (f) If you are curating or releasing new datasets, did you discuss how you intend to make your datasets FAIR (see FORCE11 (2020))? **NA**
  - (g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset (see Gebru et al. (2021))? **NA**
6. Additionally, if you used crowdsourcing or conducted research with human subjects, **without compromising anonymity...**
- (a) Did you include the full text of instructions given to participants and screenshots? **NA.** Our study does not involve crowdsourcing or human subjects research.
  - (b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? **NA**
  - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? **NA**
  - (d) Did you discuss how data is stored, shared, and de-identified? **NA**

### A Verification Planner Specifications

Table 4 provides detailed technical specifications for the verification planner described in Section 3.5.

### B Case Study

Figure 3 presents a case study demonstrating CARP’s verification process for a complex claim.

Component	Description
<b>Dependency Analysis</b>	
Dependency Patterns	<p><b>Logical:</b> Identified from prerequisite relationships. One subclaim must be verified before another can be evaluated  <i>Example: Resolve an entity reference before verifying claims about that entity.</i></p> <p><b>Temporal:</b> Identified from causal relationships. Sequential events must be verified in chronological order  <i>Example: Verify a company's founding before checking its later acquisitions.</i></p> <p><b>Evidential:</b> Identified from same entity relationships. Subclaims sharing entities can reuse the same evidence retrieval  <i>Example: Multiple subclaims about one person's career share the same biographical database query.</i></p> <p><b>Independent:</b> No relationships connect the subclaims. Can be verified in parallel  <i>Example: Subclaims about different people with different sources require no coordination.</i></p>
<b>Phase Construction</b>	
Phase ID	Unique integer identifier (sequential: 1, 2, 3, ...)
Goal	Description of what the phase aims to verify
Dependency	(Optional) List of phase IDs specifying prerequisite phases that must complete first
Subclaims	List of subclaim specifications to verify in this phase (see Action Specification)
<b>Action Specification</b>	
Verification Actions	<p><b>Entity &amp; Existence:</b>  check_entity_property: Verify attributes or properties of an entity (e.g., nationality, occupation)  check_existence: Verify the existence of entities, objects, or phenomena  check_definition: Verify definitional or classificatory claims (e.g., "X is a type of Y")</p> <p><b>Relationships &amp; Affiliations:</b>  check_relationship: Verify relationships between entities (e.g., family relations, collaborations)  check_affiliation: Verify organizational memberships or institutional relationships  check_location: Verify location-based claims (e.g., geographic position, spatial relationships)</p> <p><b>Temporal &amp; Events:</b>  check_event_occurrence: Verify whether a specific event occurred  check_temporal_fact: Verify facts with temporal constraints (dates, durations, sequences)  check_sequence: Verify sequential or ordered relationships (e.g., chronological order)</p> <p><b>Analysis &amp; Reasoning:</b>  check_statistical_fact: Verify numerical or statistical claims (e.g., quantities, percentages)  check_causal_claim: Verify causal relationships between events or conditions  check_comparison: Verify comparative claims between entities (e.g., larger than)  check_quote_claim: Verify attributed statements, quotes, or declarations  infer_question_answering: Verify subclaims requiring implicit reasoning or inferred justification (requires <i>Derived Question</i> field)</p>
Strategy	<p><b>Factual lookup:</b> Query authoritative databases or reference sources for direct factual information</p> <p><b>Logical reasoning:</b> Apply reasoning to derive conclusions from factual information</p> <p><b>Temporal validation:</b> Verify chronological ordering, date ranges, or duration constraints</p> <p><b>Cross-referencing:</b> Compare and reconcile information across multiple independent sources</p> <p><b>Aggregation and counting:</b> Collect instances and verify quantitative claims</p> <p><b>Constraint checking:</b> Verify that all specified conditions are satisfied</p>
Evidence Type	Required evidence sources (e.g., academic studies, databases, news reports, filmographic records)
Query Hint	Natural-language search query for retrieving supporting evidence
Derived Question	<p>(Optional, for <i>infer_question_answering</i> action only): articulates the implicit question that must be answered</p> <p><i>Example:</i>  Subclaim: "The company achieved profitability quickly"  Derived Question: "What timeframe is considered quick for a company to become profitable?"</p>
<b>Resolution Logic</b>	
requires_all_supported	Boolean: In this work, we always set it to <i>TRUE</i> (all subclaims must be verified); the design allows for future extension to <i>FALSE</i> (partial support)
Explanation	Description of how subclaim verdicts combine and logical relationships (AND/OR)
Verdict States	<p><b>SUPPORT:</b> All required subclaims verified as true</p> <p><b>REFUTE:</b> Any required subclaim verified as false</p>

Table 4: Technical specifications of the verification planner

**Claim**

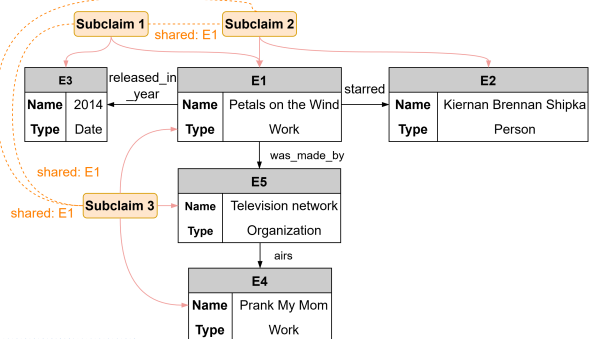
(Label: REFUTE)

Petals on the Wind is a 2014 movie sequel of a film. It starred the "Mad Men" actress Kiernan Brennan Shipka. It was made by the television network that airs Prank My Mom.

**Subclaims (initial)**

- 1. Petals on the Wind is a 2014 movie sequel.
- 2. The movie sequel starred Kiernan Brennan Shipka, an actress from 'Mad Men'.
- 3. It was made by the television network that airs 'Prank My Mom'.

**Claim-Representation Graph**



**Subclaims (refined)**

- 1. Petals on the Wind is a 2014 movie sequel.
- 2. **Petals on the Wind** starred Kiernan Brennan Shipka, an actress from 'Mad Men'.
- 3. **Petals on the Wind** was made by the television network that airs 'Prank My Mom'.

**Verification Plan**

**Phase 1**  
 Goal: Verify that 'Petals on the Wind' is a 2014 movie sequel.  
 Action: check\_temporal\_fact  
 Strategy: Verify the release year and sequel status of 'Petals on the Wind' through film databases and official announcements.

**Phase 2**  
 Goal: Verify that 'Petals on the Wind' starred Kiernan Brennan Shipka, an actress from 'Mad Men'.  
 Action: check\_relationship  
 Strategy: Check cast lists and actor profiles to confirm Kiernan Brennan Shipka's role in 'Petals on the Wind' and her involvement in 'Mad Men'.

**Phase 3**  
 Goal: Verify that 'Petals on the Wind' was made by the television network, which airs 'Prank My Mom'.  
 Action: check\_affiliation  
 Strategy: Confirm production details of 'Petals on the Wind' and verify the television network's airing of 'Prank My Mom'.

**Final Verdict Logic:**  
 The logical relationship between subclaims is conjunctive (AND), meaning all must be true. If any subclaim is refuted, it undermines the validity of the entire claim.

**Retrieved Documents**

**Title:** Flowers in the Attic  
**text:** ...Flowers in the Attic is a 1979 Gothic novel by V. C. Andrews. It is the first book in the Dollanganger Series, and was followed by "Petals on the Wind", "If There Be Thorns", "Seeds of Yesterday", and "Garden of Shadows". The novel is written in the first-person, from the point of view of Cathy Dollanganger. It was twice adapted into films in 1987 and 2014. ...  
**Title:** Heather Graham  
**Text:** ...In 2014, she starred in the second adaptation of the V. C. Andrews novel "Flowers in the Attic" on the Lifetime network. She played the character of Corinne Dollanganger; an evil mother who locks her four children in an attic in order to receive an inheritance from her dying father. Graham also appeared in two TV-movie sequels based on the V.C. Andrews' series: "Petals on the Wind" and "If There Be Thorns". ...

**Title:** Kiernan Shipka  
**Text:** ...Kiernan Shipka Kiernan Brennan Shipka (born November 10, 1999) is an American actress. She is known for playing Sally Draper on the AMC series "Mad Men." [...] As part of "Mad Men"'s ensemble cast, she won the Screen Actors Guild Award for Outstanding Performance by an Ensemble in a Drama Series in 2008 and 2009. Shipka has received praise for her performance on "Mad Men". [...] Shipka's credits after "Mad Men" include "Flowers in the Attic" as Cathy and Kat in the 2015 Oz Perkins horror film, ...

**Initial Statement**

**SUPPORT TEAM**

Following the verification plan phases: 1) 'Petals on the Wind' is confirmed as a 2014 movie sequel to 'Flowers in the Attic', as evidenced by Heather Graham's filmography which shows she appeared in both films. 2) While Kiernan Shipka is indeed known for playing Sally Draper in 'Mad Men' as confirmed by her biography, I cannot find definitive evidence that she starred in 'Petals on the Wind' - she appeared in 'Flowers in the Attic' as Cathy but not necessarily the sequel. 3) While the context confirms Lifetime as a major television network producing original movies and series, I cannot definitively confirm from the provided context that both 'Petals on the Wind' and 'Prank My Mom' aired on Lifetime. The evidence strongly supports parts of the claim but has significant gaps regarding Shipka's role in the sequel and the network's connection to both shows. (Confidence: 45)

The claim that 'Petals on the Wind' is a 2014 movie sequel is supported by evidence. 'Petals on the Wind' is indeed a sequel to the 2014 film adaptation of 'Flowers in the Attic'. This confirms subclaim 1. Subclaim 2 is also supported by evidence. Kiernan Brennan Shipka, known for her role in 'Flowers in the Attic' as Cathy Dollanganger, and reprised her role in its sequel, 'Petals on the Wind'. This confirms her involvement in both productions. **Subclaim 3 is supported by evidence that 'Petals on the Wind' was produced by Lifetime Television Network.** Lifetime is known for producing and airing TV movies, including adaptations of V.C. Andrews' novels. Additionally, Lifetime aired 'Prank My Mom,' confirming its involvement with both productions. Overall, each subclaim is well-supported by reliable sources, confirming the main claim. (Confidence: 90)

**Additional evidence:**  
**Title:** Lifetime (TV network)  
**Text:** Lifetime Television Network produced and aired both 'Flowers in the Attic' and its sequel 'Petals on the Wind.' Lifetime also aired 'Prank My Mom.'

**REFUTATION TEAM**

Based on the available evidence, I can strongly refute parts of this claim. First, while Petals on the Wind is indeed a 2014 movie sequel to Flowers in the Attic, it did NOT star Kiernan Shipka. The context shows that Kiernan Shipka appeared in the first film 'Flowers in the Attic' as Cathy, but not in 'Petals on the Wind.' This is a significant error in the claim. The evidence shows that Heather Graham appeared in both films as Corinne Dollanganger, but does not support Shipka's involvement in the sequel. While Shipka did appear in Mad Men as claimed, she was not in Petals on the Wind. Regarding the network claim, while the evidence confirms Lifetime produced these films (as shown by Heather Graham's involvement in the Lifetime network adaptations), there is insufficient evidence in the provided context to verify whether Lifetime airs 'Prank My Mom.' (Confidence: 75)

1. **"Petals on the Wind is a 2014 movie sequel."** This subclaim is accurate. 'Petals on the Wind' is indeed a sequel to 'Flowers in the Attic,' both of which are based on novels by V.C. Andrews. The film was released in 2014, confirming its status as a sequel and its release year. 2. **"Petals on the Wind starred Kiernan Brennan Shipka, an actress from 'Mad Men.'"** This subclaim is incorrect. Kiernan Shipka, known for her role as Sally Draper in 'Mad Men,' did not star in 'Petals on the Wind.' Instead, she starred in 'Flowers in the Attic,' the prequel to 'Petals on the Wind.' The role of Cathy Dollanganger in 'Petals on the Wind' was played by Rose McIver. 3. **"Petals on the Wind was made by Lifetime Television Network, the television network that airs 'Prank My Mom.'"** This subclaim is partially correct. 'Petals on the Wind' was indeed produced by Lifetime Television Network. However, there is no evidence that Lifetime aired 'Prank My Mom.' Overall, while some aspects of the claim are correct, significant parts are inaccurate or misleading, particularly regarding Kiernan Shipka's involvement and the airing network of 'Prank My Mom.' (Confidence: 70)

**Additional evidence:**  
**Title:** Lifetime (TV network)  
**Text:** Lifetime Television Network produced "Petals on the Wind."

**Self-correction**

**SUPPORT TEAM**

After reviewing all arguments, I maintain that Petals on the Wind (2014) is indeed a movie sequel, but must revise my position on Kiernan Shipka's involvement. The opposing team provided compelling evidence that Rose McIver, not Shipka, starred in Petals on the Wind, though Shipka was in the first film. While Lifetime did produce Petals on the Wind, there remains insufficient evidence about Prank My Mom's network affiliation. My teammate's argument claiming Shipka reprised her role appears to be incorrect based on available casting information. (Position changed: True)

After reviewing the opposing team's arguments and my teammate's input, I recognize that my initial support for the claim was partially incorrect. The opposing team correctly pointed out that Kiernan Shipka did not star in 'Petals on the Wind'; instead, she starred in 'Flowers in the Attic.' This is a significant error in the claim. Additionally, while Lifetime Television Network did produce 'Petals on the Wind,' there is insufficient evidence to confirm that Lifetime aired 'Prank My Mom.' Therefore, two of the three subclaims are not fully supported, undermining the overall claim. (Position changed: True)

**REFUTATION TEAM**

After reviewing all arguments, I maintain that key parts of the claim are incorrect. While 'Petals on the Wind' is indeed a 2014 movie sequel produced by Lifetime, Kiernan Shipka did not star in it. The evidence shows she was in 'Flowers in the Attic' but Rose McIver played the role in 'Petals on the Wind'. The opposing team member who claimed Shipka reprised her role provided incorrect information not supported by the sources. Additionally, while Lifetime produced the film, there remains insufficient evidence to verify whether Lifetime specifically airs 'Prank My Mom', as this detail is not supported in the provided sources. (Position changed: False)

After reviewing the arguments from both the opposing team and my teammate, I maintain that the claim is partially incorrect. The opposing team provided evidence that Kiernan Shipka starred in both 'Flowers in the Attic' and 'Petals on the Wind,' but this contradicts reliable sources indicating she only appeared in 'Flowers in the Attic.' The role of Cathy Dollanganger in 'Petals on the Wind' was played by Rose McIver, not Shipka. Additionally, while Lifetime did produce 'Petals on the Wind,' there is no conclusive evidence that Lifetime aired 'Prank My Mom.' Therefore, significant parts of the claim remain inaccurate. (Position changed: False)

**Moderator Final Verdict**

The claim consists of three subclaims: 1) 'Petals on the Wind' is a 2014 movie sequel, 2) it starred Kiernan Brennan Shipka, and 3) it was made by the network that airs 'Prank My Mom.' The SUPPORT team initially argued that all parts of the claim were correct, but later revised their position after acknowledging errors. The REFUTE team consistently provided evidence that Kiernan Shipka did not star in 'Petals on the Wind'; instead, Rose McIver played the role of Cathy Dollanganger in the sequel. This directly contradicts the second subclaim. Additionally, while Lifetime produced 'Petals on the Wind,' there is no conclusive evidence that Lifetime aired 'Prank My Mom.' The REFUTE team's arguments were stronger and more consistent, particularly regarding Shipka's involvement and the network details, leading to a refutation of the claim.  
**Final verdict: REFUTE**

Figure 3: This case study demonstrates CARP's entire claim verification process.