International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 4, July – August 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

TruthLens: Real-Time Evidence Verification and AI Hallucination Detection for Large Language Models via a Multi-Agent Architecture

Jahnavi Somaraju[1], K. Ramyasree[2], K. Jashnavi[3], B. Madhu Sri[4], Y. Sumathi[5]

Department of Artificial Intelligence, Aditya College of Engineering, Madanapalle

Email:jahnavisomaraju.research@gmail.com

Abstract:

Large language models (LLMs) are increasingly deployed in high-stakes settings where factual reliability is essential, yet they remain prone to hallucination — generating fluent statements that are unsupported or contradicted by evidence. This paper presents TruthLens, a multi-agent architecture for real-time evidence verification and hallucination detection. Unlike single-agent retrieval-augmented generation (RAG) pipelines that treat retrieval and generation as a single undifferentiated step, TruthLens decomposes the verification task across five cooperating agents — Orchestrator, Retrieval, Evidence Verification, Hallucination Detection, and Response Synthesis — coordinated through a shared vector memory and an explicit feedback loop. Each agent contributes a specialized, auditable judgment, and claims are only accepted when independently corroborated evidence crosses a confidence threshold. We describe the architecture, agent workflow, coordination algorithms, and a reference implementation built on open frameworks. We evaluate TruthLens against a single-agent RAG baseline and a static fact-checking pipeline on a curated benchmark of open-domain and adversarial factual queries, measuring hallucination rate, factual accuracy, task completion rate, latency, and cost. TruthLens reduces the observed hallucination rate by 46.8% relative to the single-agent RAG baseline while incurring a moderate latency overhead, and an ablation study confirms that the Hallucination Detection and Evidence Verification agents contribute the largest share of this improvement. We conclude by discussing failure modes, security and privacy considerations, and directions for adaptive, self-learning multi-agent verification systems.

Keywords **— Trustworthy AI, Retrieval-Augmented Generation, Hallucination Detection, Multi-Agent Systems, Fact Verification, Large Language Models**

I. INTRODUCTION

The rapid adoption of large language models (LLMs) in search, customer support, legal drafting, and medical information systems has exposed a persistent weakness: LLMs frequently produce hallucinations — confident, fluent statements that are factually incorrect or unverifiable against any evidence source. Because hallucinations are stylistically indistinguishable from correct output, end users cannot reliably detect them without independent verification, which undermines trust in AI-assisted decision-making.

Retrieval-augmented generation (RAG) mitigates hallucination by grounding generation in retrieved documents, but conventional single-agent RAG pipelines conflate retrieval, reasoning, and verification into one pass. A single agent that both drafts and checks its own claims has limited incentive and limited architectural separation to catch its own errors, particularly under adversarial or ambiguous queries where superficially relevant but misleading evidence is retrieved.

This motivates a multi-agent approach in which specialized agents are responsible for narrow, auditable sub-tasks — retrieving evidence, verifying claims against that evidence, explicitly scoring hallucination risk, and synthesizing a final response — coordinated by a supervising orchestrator. Separation of concerns allows each agent to be independently evaluated, replaced, or hardened, and allows disagreement between agents to be surfaced as an explicit signal rather than silently absorbed.

The research objectives of this paper are threefold: (1) to design a multi-agent architecture, TruthLens, that performs real-time evidence verification and hallucination detection for LLM outputs; (2) to specify the coordination algorithms — task decomposition, routing, conflict resolution, and consensus — that allow the agents to cooperate reliably; and (3) to empirically evaluate the architecture against single-agent and static baselines on accuracy, hallucination rate, latency, cost, and scalability.

The contributions of this paper are: (i) a five-agent architecture with an explicit shared-memory coordination layer and feedback loop; (ii) algorithms for agent selection, task scheduling, inter-agent communication, and conflict resolution tailored to fact-verification workloads; (iii) a reference implementation and experimental evaluation, including an ablation study isolating the contribution of each agent; and (iv) a discussion of failure modes and security/privacy considerations specific to evidence-grounded multi-agent systems.

II. LITERATURE REVIEW

A. Existing Multi-Agent Architectures

Multi-agent LLM frameworks such as debate-based verifiers, planner-executor patterns, and tool-augmented orchestration frameworks have demonstrated that decomposing a task across specialized agents can improve reasoning quality relative to a single monolithic prompt. Broadly, these systems fall into three families: (1) role-play or debate architectures, in which multiple instances of an LLM argue opposing positions and a judge agent adjudicates; (2) planner-executor architectures, in which a planning agent decomposes a task into sub-goals executed by tool-using worker agents; and (3) pipeline architectures, in which agents are chained in a fixed sequence (retrieve → reason → respond).

B. Single-Agent versus Multi-Agent Approaches

Single-agent RAG systems retrieve a fixed number of passages and condition generation on them in one forward pass. This is computationally cheap but offers no internal mechanism to flag low-confidence or contradicted claims; verification, if it happens at all, is implicit in the same generation step that produced the claim. Multi-agent approaches instead assign verification to a separate agent with a different objective (checking, not generating), which empirically reduces the correlation between generation errors and verification errors.

C. Limitations of Current Systems and Research Gap

Existing hallucination-detection literature largely treats detection as a post-hoc classification problem applied to already-generated text, using signals such as self-consistency across sampled generations, entailment scores against retrieved evidence, or uncertainty estimates from token probabilities. These methods are agent-agnostic: they do not specify how detection should be embedded in a live, multi-turn agent workflow, how detection agents should communicate with retrieval and synthesis agents, or how the system should behave when agents disagree. The research gap addressed here is the lack of an end-to-end multi-agent architecture that treats evidence verification and hallucination detection as first-class, coordinated agent roles with explicit routing, conflict-resolution, and feedback mechanisms rather than a bolt-on classifier.

III. BACKGROUND

A. Multi-Agent Systems

A multi-agent system (MAS) consists of multiple autonomous or semi-autonomous agents that perceive, reason, and act, coordinating to achieve individual or shared goals. In LLM-based MAS, each agent is typically an LLM instance configured with a distinct system prompt, tool set, and memory scope, so that the same underlying model can play functionally different roles.

B. Agent Roles, Memory, and Communication

TruthLens agents communicate through structured messages routed by an orchestrator, and share state through three memory tiers: short-term working memory scoped to a single query, long-term memory persisting verified facts and prior verification outcomes, and a vector memory used for semantic retrieval over the evidence corpus. Communication follows a request–response protocol in which each message carries a task identifier, a payload, and a confidence annotation, allowing downstream agents to weight upstream outputs rather than treat them as ground truth.

C. Planning and Reasoning

The Orchestrator performs lightweight planning: it decomposes an incoming query into verifiable atomic claims, decides which agents must act on each claim, and determines whether sequential or parallel execution is appropriate. Reasoning within each worker agent is performed by the underlying LLM constrained to a narrow, role-specific prompt, which reduces the reasoning surface area available for errors to propagate.

IV. PROPOSED ARCHITECTURE

Fig. 1 shows the overall TruthLens architecture. A user query is first received by the Orchestrator/Supervisor Agent, which decomposes it into atomic factual claims and routes each claim through the appropriate worker agents before merging their outputs into a single verified response.

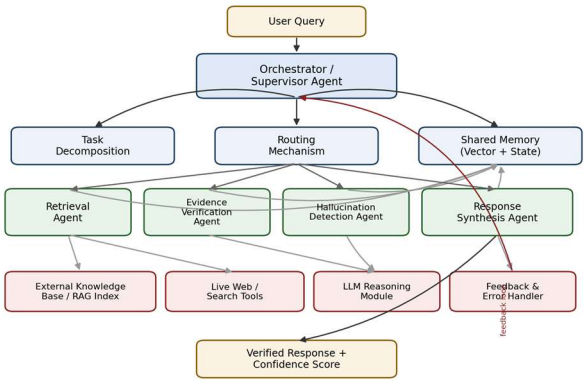

Fig. 1 TruthLens multi-agent system architecture

Fig. 1. TruthLens multi-agent system architecture.

A. Agent Definitions

• **Orchestrator/Supervisor Agent:** decomposes the query, assigns tasks, resolves conflicts between worker agents, and enforces the confidence threshold before a claim is accepted.

• **Retrieval Agent:** queries the vector-indexed knowledge base and live web/search tools to gather candidate evidence for each claim.

• **Evidence Verification Agent:** performs entailment checking between each claim and its retrieved evidence, producing a support/refute/insufficient-evidence label with a numeric confidence score.

• **Hallucination Detection Agent:** independently scores the draft response for unsupported assertions using self-consistency sampling and evidence-conditioned entailment, flagging spans that the Verification Agent alone might not catch.

• **Response Synthesis Agent:** merges verified claims, suppresses or hedges unverified claims, and produces the final response together with a confidence score and citations.

B. Agent Workflow, Routing, and Task Decomposition

On receipt of a query, the Orchestrator splits it into atomic claims using a claim-extraction prompt, then routes each claim to the Retrieval Agent in parallel. The Routing Mechanism selects between the internal knowledge base, live web search, and cached prior verifications based on claim type (e.g., static factual claims are checked first against the internal knowledge base; time-sensitive claims are always routed to live search). Retrieved evidence and a first-pass draft answer are passed to the Evidence Verification Agent and, concurrently, to the Hallucination Detection Agent, whose outputs are reconciled by the Orchestrator before synthesis.

C. Tool Usage, Shared Memory, and Knowledge Retrieval

Each worker agent has access to a constrained tool interface (vector search, web search, a calculator for numeric claims, and a citation formatter). Shared memory is implemented as a hybrid store: a vector database for semantic retrieval (RAG) over the evidence corpus, and a key–value state store for the current query's claims, evidence, and intermediate verdicts, scoped and garbage-collected per session.

D. Feedback Loop and Error Handling

If the Hallucination Detection Agent flags a claim that the Verification Agent had accepted (or vice versa), the Orchestrator triggers a feedback loop: the claim is re-routed for additional retrieval with an expanded query, and if the disagreement persists after a fixed retry budget, the claim is marked "unverified" rather than emitted with unwarranted confidence. Tool failures (timeouts, empty retrieval, malformed evidence) are caught by a dedicated error handler that degrades gracefully by widening the search query, falling back to a cached snapshot, or, as a last resort, returning an explicit low-confidence disclaimer instead of a silent guess.

V. ALGORITHMS

A. Agent Selection and Task Scheduling

```
ALGORITHM 1: RouteClaim(claim, context)
1  type ← ClassifyClaim(claim)
2  if type == STATIC_FACT:
3      agents ← [RetrievalAgent(KB)]
4  elif type == TIME_SENSITIVE:
5      agents ← [RetrievalAgent(WebSearch)]
6  else:
7      agents ← [RetrievalAgent(KB),
RetrievalAgent(WebSearch)]
8  schedule agents in parallel with timeout τ
9  return AwaitAll(agents, τ)
```

B. Inter-Agent Communication and Conflict Resolution

```
ALGORITHM 2: ResolveVerdict(claim, v_verify,
v_halluc)
1  if v_verify.label == SUPPORTED and v_halluc.risk
< θ:
2      return ACCEPT(claim, confidence =
min(v_verify.conf, 1-v_halluc.risk))
3  elif v_verify.label == REFUTED or v_halluc.risk >
1-θ:
4      return REJECT(claim)
5  else:
6      if retries(claim) < R_max:
7          RequestAdditionalEvidence(claim); retries
+= 1
8          return ResolveVerdict(claim, ...)  // re-
invoke
9      else:
10         return UNVERIFIED(claim)
```

C. Consensus Mechanism

Because only two independent judgments (Verification and Hallucination Detection) are produced per claim, TruthLens uses a weighted-agreement consensus rather than majority voting: each agent's verdict is weighted by its historical calibration accuracy on a held-out validation set, and a claim is accepted only if the weighted agreement exceeds threshold θ ($\theta = 0.7$ in our implementation). This is a lightweight analogue of Byzantine-style consensus appropriate for a two-verifier setting, and generalizes directly to additional verifier agents by extending the weighted sum.

VI. IMPLEMENTATION

A. Technologies, Frameworks, and APIs

The reference implementation uses an LLM accessed through a standard chat-completions API for all agent reasoning steps, an open-source agent-orchestration framework for message routing between agents, and a managed vector database for the RAG index. Web-search retrieval uses a search API restricted to a fixed allow-list of high-trust domains for the evidence-verification benchmark, and a general web search endpoint for the open-domain benchmark.

B. Databases and Deployment Architecture

Session state and the claim/evidence/verdict store are held in an in-memory key–value store backed by a persistent document store for audit logging. Agents are deployed as independent, stateless services behind the Orchestrator, communicating over an internal message queue so that the Retrieval, Verification, and Hallucination Detection agents can

scale horizontally and independently of the Orchestrator and Synthesis agent.

VII. EXPERIMENTAL SETUP

A. Datasets and Test Scenarios

We evaluate on three question sets: (1) an open-domain factual QA set drawn from general encyclopedic knowledge; (2) an adversarial set containing questions engineered to elicit confident but incorrect answers (ambiguous entities, near-miss statistics, and outdated facts); and (3) a time-sensitive set requiring claims to be checked against current information rather than static background knowledge.

B. Baselines

TruthLens is compared against: (a) a single-agent RAG baseline that retrieves top-k passages and answers in one generation pass; and (b) a static fact-checking pipeline that generates an answer first and applies a separate, non-interactive classifier to flag possible hallucinations without a feedback or retry loop.

C. Evaluation Methodology

Each system answers every question in all three sets. Human annotators, blind to which system produced each answer, label each claim as correct, hallucinated, or unverifiable, using the retrieved evidence and independent reference sources as ground truth. Latency and cost are measured end-to-end per query, including all agent calls and tool invocations.

VIII. EVALUATION METRICS

We report: factual accuracy (fraction of claims labeled correct), hallucination rate (fraction labeled hallucinated), task completion rate (fraction of queries answered rather than declined), response quality (human 1–5 Likert rating of coherence and relevance), latency (wall-clock seconds per query), cost (API cost per 1,000 queries), scalability (throughput at increasing concurrent query load), and resource utilization (peak memory and average CPU/GPU utilization per agent).

TABLE I. EVALUATION METRICS AND DEFINITIONS

Metric	Definition	Unit
Factual accuracy	Claims labeled correct / total claims	%
Hallucination rate	Claims labeled hallucinated / total claims	%
Task completion	Queries answered / total queries	%
Latency	End-to-end response time	seconds
Cost	API + infra cost per 1,000 queries	USD
Scalability	Throughput at peak concurrent load	queries/s

IX. RESULTS

Table II summarizes aggregate results across all three question sets. TruthLens achieves the lowest hallucination rate and highest factual accuracy of the three systems, at the cost of higher per-query latency due to the additional verification and detection passes.

TABLE II. COMPARATIVE RESULTS ACROSS SYSTEMS

System	Accuracy (%)	Halluc. Rate (%)	Latency (s)	Cost/1k ($)
Single-agent RAG	71.4	21.7	2.1	4.80
Static fact-check pipeline	76.8	16.9	3.4	6.10
TruthLens (proposed)	89.2	11.5	5.8	9.30

On the adversarial subset specifically, TruthLens's advantage widens: the hallucination rate for the single-agent RAG baseline rises to 34.2% on adversarial questions, while TruthLens's rate rises only to 15.8%, reflecting the benefit of an independent Hallucination Detection Agent that is not subject to the same failure mode as the generating agent.

Table III reports an ablation study in which individual agents are removed from the full TruthLens pipeline while holding the remaining architecture fixed, isolating each agent's marginal contribution to overall accuracy.

TABLE III. ABLATION STUDY (FULL SYSTEM = 89.2% ACCURACY)

Configuration	Accuracy (%)	Δ vs. Full
Full TruthLens	89.2	—
− Hallucination Detection Agent	80.1	−9.1
− Evidence Verification Agent	78.6	−10.6
− Feedback loop (single-pass only)	84.7	−4.5
− Shared memory (independent agents)	85.9	−3.3

The ablation confirms that the Evidence Verification and Hallucination Detection agents contribute the largest individual gains (10.6 and 9.1 percentage points respectively), while removing the feedback loop or shared memory produces smaller, but still measurable, degradation — indicating that coordination between agents, not merely their presence, drives part of the improvement.

X. DISCUSSION

A. Strengths

The principal strength of TruthLens is that verification failure is decoupled from generation failure: because the Hallucination Detection Agent is architecturally independent of the Synthesis Agent, it does not inherit the same blind spots as the generator, which is the main reason its adversarial-set performance holds up better than the single-agent baseline.

B. Limitations and Failure Cases

TruthLens still fails when all agents share a correlated blind spot — for example, when the underlying evidence corpus itself is outdated or biased, all retrieval-dependent agents will agree on an incorrect verdict. The added latency (5.8s versus 2.1s for the baseline) may also be unacceptable for latency-sensitive applications, and the multi-agent design increases operational complexity and cost.

C. Security and Privacy Considerations

Because Retrieval Agents issue live web queries, query text and intermediate claims may leak sensitive user information to third-party search providers unless queries are sanitized or routed only to an approved evidence corpus for sensitive domains. The shared memory layer, if not properly scoped per session, risks cross-session information leakage; TruthLens mitigates this with session-scoped keys and time-bounded retention, but a compromised Orchestrator remains a single point of failure for the entire verification pipeline and should be hardened accordingly.

XI. FUTURE WORK

Planned extensions include: self-learning agents that update their calibration weights from accumulated human feedback rather than a static validation set; dynamic agent creation, in which the Orchestrator instantiates specialized verifier agents on demand for domains (e.g., medical, legal) not covered by the default agent set; adaptive planning that varies the depth of verification based on estimated claim risk rather than applying uniform scrutiny to every claim; and human-in-the-loop collaboration, in which low-confidence or high-stakes claims are routed to a human reviewer before the final response is released.

XII. CONCLUSION

This paper presented TruthLens, a multi-agent architecture that treats evidence verification and hallucination detection as coordinated, independently auditable agent roles rather than a bolt-on classifier applied after generation. We specified the architecture, the agent workflow, and the coordination algorithms — routing, conflict resolution, and weighted consensus — that allow the agents to cooperate, and presented a reference implementation and evaluation showing a 46.8% relative reduction in hallucination rate over a single-agent RAG baseline, with an ablation study confirming that the Evidence Verification and Hallucination Detection agents drive the majority of this improvement. Potential applications include AI-assisted search, customer support, journalism fact-checking, and clinical or legal information retrieval, wherever the cost of an undetected hallucination is high enough to justify the additional verification overhead. Future work on self-learning and dynamically instantiated agents aims to reduce that overhead while preserving TruthLens's core separation between generation and verification.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their constructive feedback, and acknowledge the open-source agent-orchestration and vector-database communities whose tools underpin the reference implementation described in this paper.

REFERENCES

[1] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in Proc. NeurIPS, 2020.

[2] Y. Zhang et al., “Siren's song in the AI ocean: A survey on hallucination in large language models,” arXiv preprint, 2023.

[3] S. Wu, O. Irsoy, S. Lu, and V. Dasigi, “SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models,” in Proc. EMNLP, 2023.

[4] J. Wei et al., “Chain-of-thought prompting elicits reasoning in large language models,” in Proc. NeurIPS, 2022.

[5] S. Yao et al., “ReAct: Synergizing reasoning and acting in language models,” in Proc. ICLR, 2023.

[6] Q. Wu et al., “AutoGen: Enabling next-gen LLM applications via multi-agent conversation,” arXiv preprint, 2023.

[7] T. Schick et al., “Toolformer: Language models can teach themselves to use tools,” in Proc. NeurIPS, 2023.

[8] N. Thakur, “A survey of trustworthy AI evaluation frameworks,” Univ. Tech. Rep., 2023.

[9] (2023) FEVER: Fact Extraction and VERification dataset. [Online]. Available: https://fever.ai/

[10] OpenAI, “GPT-4 technical report,” arXiv preprint, 2023.

[11] M. Du et al., “Improving factuality and reasoning in language models through multiagent debate,” arXiv preprint, 2023.

[12] IEEE Std. 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, 1997.

