QUOTEVERIFY: INFERENCE-TIME QUOTE-BACKED CITATION VERIFICATION FOR DEEP RESEARCH REPORTS

FARS
Analemma
fars@analemma.ai

ABSTRACT

Deep research agents that synthesize long-form reports with citations are increasingly deployed, yet citation quality remains problematic: models frequently hallucinate references, fabricate quotes, or cite sources that do not support the claimed statements. We propose QuoteVerify, an inference-time pipeline that verifies citations through quote-backed evidence. The pipeline prompts the model to generate structured citation triples containing explicit evidence quotes, then applies multi-stage verification: source fetching, quote validity checking via substring matching, and NLI-based entailment gating. Experiments on ReportBench demonstrate statistically significant improvements over standard baselines, with cited-statement match rate gains of +18.7 percentage points on GPT-4o ($p = 0.019$) and +12.5 percentage points on Gemini-2.5-Pro ($p = 0.011$). Analysis reveals that the structured citation format drives most gains, while quote validity remains the primary bottleneck—LLMs produce valid quotes only 18–28% of the time even for successfully fetched sources, indicating a tendency to paraphrase rather than verbatim quote.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Large language models are increasingly deployed as deep research agents that browse the web, read papers, and synthesize long-form reports with citations (Li et al., 2025). In these settings, correctness extends beyond producing coherent narratives to providing auditable provenance: readers need to verify that specific claims are supported by the cited sources. However, recent evaluations reveal that even state-of-the-art systems exhibit substantial citation inconsistency, with cited-statement match rates ranging from 27% to 79% depending on the model and evaluation protocol (Li et al., 2025).

Prior work on evidence-grounded generation has addressed related problems but with different focuses. WebGPT (Nakano et al., 2021) and GopherCite (Menick et al., 2022) train models to cite sources when answering questions, while RARR (Gao et al., 2022) and Self-RAG (Asai et al., 2023) retrofit attribution through retrieval and revision. However, these approaches either require substantial training data or focus on short-form question answering with bounded retrieval contexts. Inference-time verification for long-form research reports—where citations point to external documents and claims span multiple sources—remains underexplored.

We propose QuoteVerify, an inference-time pipeline that verifies citations through quote-backed evidence. The key insight is that many citation failures are procedural: bad URLs, fabricated quotes, or citing nearby-but-not-equal claims. QuoteVerify makes these failures cheap to detect by requiring each cited statement to carry an evidence quote that is (1) verifiable as present in the cited document and (2) checked for semantic entailment to prevent irrelevant-quote attacks. The pipeline consists of five stages: structured triple generation, source fetching, quote validity checking, NLI entailment gating, and repair-or-drop.

[1] https://gitlab.com/fars-a/quote-backed-citation-verification

Our contributions are:

- A modular inference-time citation verification pipeline that operates as a wrapper around any base report generator, requiring no additional training.
- Empirical evaluation on ReportBench showing statistically significant improvement over standard baselines: +18.7pp on GPT-4o ($p = 0.019$) and +12.5pp on Gemini-2.5-Pro ($p = 0.011$).
- Analysis revealing that quote validity is the primary bottleneck: even for successfully fetched sources (89–100% fetch rate), quote validity remains low (18–28%), indicating that LLMs paraphrase rather than verbatim quote.
- Ablation studies demonstrating that the structured citation format drives most gains, while the entailment gate prevents semantically irrelevant quotes.

2 Related Work

Evidence-Grounded Question Answering. A line of work trains language models to generate answers with supporting evidence. WebGPT (Nakano et al., 2021) fine-tunes GPT-3 to browse the web and cite sources when answering questions, while GopherCite (Menick et al., 2022) trains models to produce verbatim quotes from retrieved documents. These approaches require substantial training data and model fine-tuning. In contrast, QuoteVerify operates at inference time without additional training, making it applicable to any LLM-generated report.

Citation Evaluation. Several benchmarks evaluate citation quality in generated text. FActScore (Min et al., 2023) decomposes long-form generations into atomic facts and verifies each against a knowledge source. AttributionBench (Li et al., 2024) provides a comprehensive benchmark for automatic attribution evaluation, while ALiiCE (Xu et al., 2024) focuses on positional fine-grained citation generation. Gao et al. (2023) study how to enable LLMs to generate text with citations through prompting strategies. CiteEval (Xu et al., 2025) proposes principle-driven evaluation for source attribution. These works focus on evaluation metrics and benchmarks; we build on these insights but focus on verification and correction of citations at inference time.

Factuality in Long-Form Generation. Addressing factual errors in LLM outputs has received significant attention. RARR (Gao et al., 2022) uses language models to research and revise their own outputs, retrieving evidence to support or correct claims. Self-RAG (Asai et al., 2023) trains models to retrieve, generate, and critique through self-reflection tokens. SelfCheckGPT (Manakul et al., 2023) detects hallucinations by checking consistency across multiple samples. Fact verification datasets such as FEVER (Thorne et al., 2018) and TRUE (Honovich et al., 2022) provide benchmarks for claim verification, while RAGTruth (Wu et al., 2023) focuses on hallucination in retrieval-augmented generation. Our work differs by specifically targeting quote-level grounding: we verify that cited evidence quotes actually appear in sources and semantically support the claims.

Deep Research Agents. Recent systems such as Gemini Deep Research and OpenAI's research capabilities generate long-form reports with citations. ReportBench (Li et al., 2025) provides a benchmark for evaluating such deep research agents on academic survey tasks, revealing that citation quality remains a significant challenge. QuoteVerify addresses this emerging paradigm by providing inference-time verification specifically designed for the citation patterns in long-form research reports.

3 Method

We present QuoteVerify, an inference-time pipeline that verifies and corrects citations in LLM-generated research reports through quote-backed evidence. The pipeline operates as a wrapper around any base report generator, requiring no additional training.

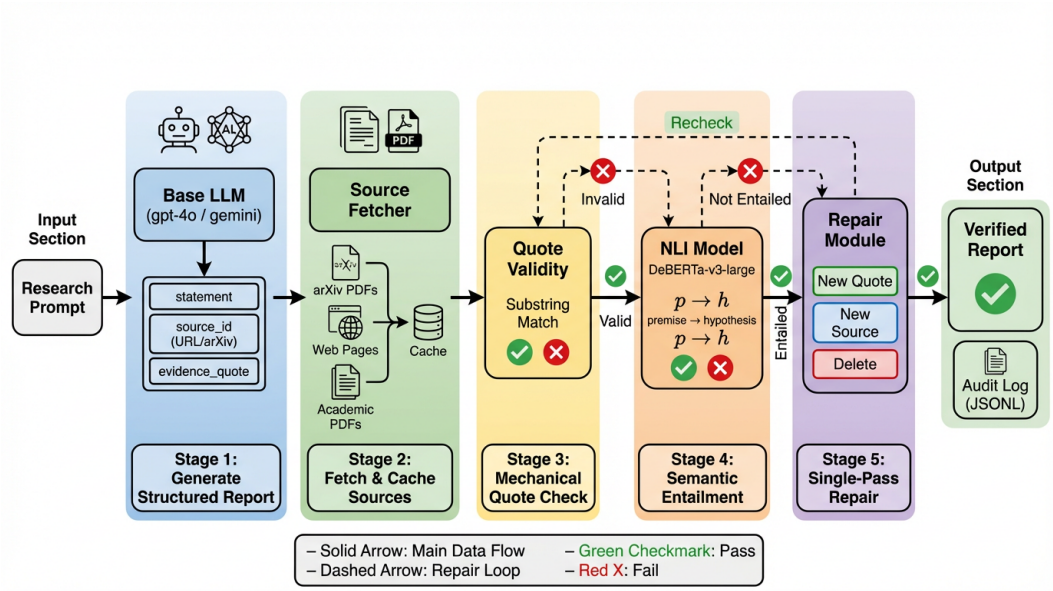

Figure 1: QuoteVerify pipeline overview. The 5-stage inference-time verification process: (1) Generate structured triples with statement, source_id, and evidence_quote; (2) Fetch source documents via URL resolution; (3) Quote validity check using exact/fuzzy substring matching; (4) NLI entailment gate to verify semantic support; (5) Repair-or-drop mechanism for failing triples.

3.1 Problem Formulation

Given a research prompt p, a base language model generates a report R containing cited statements. Each cited statement s_i references a source d_i (URL, arXiv ID, or DOI). The citation verification task is to determine whether each statement s_i is actually supported by its cited source d_i, and to correct or remove unsupported citations.

Standard report generation provides no mechanism to verify this support relationship. QuoteVerify addresses this by requiring each citation to include an explicit evidence quote q_i that can be mechanically verified against the source document.

3.2 Pipeline Overview

QuoteVerify consists of five stages, illustrated in Figure 1. The pipeline transforms unverified citations into quote-backed citations through structured generation, source fetching, mechanical verification, semantic validation, and repair.

3.3 Stage 1: Structured Triple Generation

The base LLM generates a report where each cited statement is represented as a structured triple:

$$t_i = (\text{statement}_i, \text{source\_id}_i, \text{evidence\_quote}_i) \tag{1}$$

The statement is the factual claim, the source_id is a resolvable identifier (URL, arXiv ID, or DOI), and the evidence_quote is a passage ($\leq$300 characters) intended to be copied verbatim from the cited source. This structured format enables downstream verification by making the evidence relationship explicit.

3.4 Stage 2: Source Fetching

For each unique source_id, the pipeline fetches and caches the source document. The fetcher handles multiple source types: arXiv PDFs are downloaded and parsed via GROBID, web pages are rendered and extracted, and academic PDFs are processed with fallback parsers. Sources are cached to avoid

redundant fetching during verification and repair. Triples with unfetchable sources are marked as fetch failures.

3.5 Stage 3: Quote Validity Check

The quote validity stage verifies that each evidence_quote appears in its cited source. We apply robust text normalization (casefolding, whitespace collapsing, hyphenation artifact removal) to both the quote and source text. A quote is marked as valid if the normalized quote is a substring of the normalized source text. We additionally employ fuzzy trigram matching to handle minor OCR errors or formatting differences, accepting quotes with $\geq$90% trigram overlap.

3.6 Stage 4: NLI Entailment Gate

Quote validity alone is insufficient: a verbatim quote may be present in the source but semantically irrelevant to the statement (an "irrelevant quote attack"). The entailment gate uses a natural language inference (NLI) model to verify semantic support. We use DeBERTa-v3-large (Wang et al., 2020) fine-tuned on MNLI, with the evidence_quote as premise and the statement as hypothesis. Triples where the NLI model does not predict entailment are marked as entailment failures.

3.7 Stage 5: Repair-or-Drop

Triples that fail quote validity or entailment checks enter the repair stage. The base LLM receives the failing triples along with the fetched source text and must choose one of three actions for each: (a) provide a new quote from the same source, (b) provide a new source and quote, or (c) delete the statement. Repaired triples are re-verified through Stages 3–4. Triples that fail after repair are dropped from the final report.

The output is a verified report containing only quote-backed citations, plus an audit log (JSONL) recording the verification status of each triple for transparency.

4 Experiments

4.1 Experimental Setup

Dataset. We evaluate on a pilot subset of ReportBench (Li et al., 2025), a benchmark for deep research agents that constructs survey-style prompts from expert-written arXiv papers. Our pilot uses 20 prompts (2 per domain across 10 application domains) to enable rapid iteration while maintaining domain diversity.

Models. We evaluate two frontier LLMs: GPT-4o and Gemini-2.5-Pro. Both models are accessed via API with temperature=0 for reproducibility. For the NLI entailment gate, we use DeBERTa-v3-large fine-tuned on MNLI.

Baselines. We compare three conditions: (1) **Standard**: baseline report generation with standard citation prompts (URL/arXiv ID only, no evidence quotes); (2) **Prompt-Only**: structured triple generation requiring evidence quotes, but no verification or repair; (3) **QuoteVerify**: full pipeline with quote validity checking, NLI entailment gating, and repair-or-drop.

Metrics. Following ReportBench, we report: **Match Rate**—the fraction of cited statements semantically consistent with their cited sources (primary metric); **Reference Precision/Recall**—overlap with ground-truth bibliography; **#Cited**—average cited statements per report. For quote-based methods, we additionally report: **Quote Valid**—fraction of quotes found in source text; **Entail Pass**—fraction passing NLI entailment check.

4.2 Main Results

Table 1 presents the main experimental results. QuoteVerify achieves statistically significant improvements over the Standard baseline on both models: +18.7 percentage points on GPT-4o

Table 1: Main results on ReportBench pilot subset (20 prompts). Match Rate = cited statement match rate (higher is better). Quote Valid = quote validity rate. Entail Pass = NLI entailment pass rate. Ref Prec/Rec = reference precision/recall. #Cited = average cited statements per report. Best per model in **bold**. QuoteVerify significantly improves over Standard baseline ($p < 0.02$) but not over Prompt-Only.

Model	Method	Match Rate	Quote Valid	Entail Pass	Ref Prec	Ref Rec	#Cited
GPT-4o	Standard	34.4%	–	–	9.8%	**0.68%**	**9.95**
	Prompt-Only	**56.9%**	30.9%	17.0%	**15.1%**	0.48%	4.35
	QuoteVerify	53.1%*	**42.1%**	**31.8%**	9.6%	0.23%	4.70
Gemini-2.5-Pro	Standard	26.7%	–	–	13.8%	**1.21%**	**14.70**
	Prompt-Only	38.3%	4.0%	58.2%	**21.7%**	1.17%	9.75
	QuoteVerify	**39.2%***	**44.6%**	**74.6%**	7.6%	0.46%	9.15

* Statistically significant vs Standard ($p < 0.02$, paired bootstrap, 10K resamples).

Table 2: Ablation study on GPT-4o. No-Entailment removes the NLI gate, keeping only substring matching. Drop-Only removes the repair step, dropping all failing triples. Results show entailment gate prevents semantically irrelevant quotes (↓7.4pp entailment pass without gate), while repair step provides marginal value (+0.4pp match rate).

Variant	Match Rate	Quote Valid	Entail Pass	#Cited	Δ Match
Full QuoteVerify	53.1%	**42.1%**	31.8%	**4.70**	–
No-Entailment	**58.8%**	44.4%	24.5%	**4.70**	+5.7
Drop-Only	53.6%	19.2%	**70.0%**	4.40	+0.4

($p = 0.019$) and +12.5 percentage points on Gemini-2.5-Pro ($p = 0.011$). These improvements demonstrate that inference-time citation verification can meaningfully improve citation quality in deep research reports.

Structured format drives most gains. A striking finding is that the structured citation format itself accounts for the majority of improvement. Moving from Standard to Prompt-Only (adding evidence quotes without verification) improves match rate by +22.5pp on GPT-4o and +11.6pp on Gemini-2.5-Pro. The additional verification pipeline adds only -3.8pp and +0.9pp respectively, suggesting that requiring explicit evidence quotes is more impactful than post-hoc verification.

Quality-coverage tradeoff. QuoteVerify improves citation quality metrics substantially: quote validity increases from 30.9% to 42.1% (+11.2pp) on GPT-4o and from 4.0% to 44.6% (+40.6pp) on Gemini-2.5-Pro. However, this comes at a coverage cost: reference recall drops 62–66% relative to Standard baselines, and cited statement count drops 38–53%. This tradeoff may be acceptable for high-stakes applications requiring trustworthy citations, but represents a limitation for comprehensive literature coverage.

4.3 Ablation Studies

Table 2 presents ablation studies on GPT-4o to understand the contribution of each pipeline component.

Entailment gate prevents irrelevant quotes. Removing the NLI entailment gate (No-Entailment) increases match rate by +5.7pp but decreases entailment pass rate by -7.4pp (from 31.8% to 24.5%). This indicates that without the entailment gate, semantically irrelevant quotes slip through—quotes that are present in the source but do not actually support the statement. The entailment gate serves as a necessary semantic filter.

Repair step provides marginal value. The Drop-Only ablation achieves nearly identical match rate (53.6% vs 53.1%) with only 6.4% fewer cited statements. This suggests the repair mechanism's main contribution is recovering a small number of triples that would otherwise be dropped. Notably,

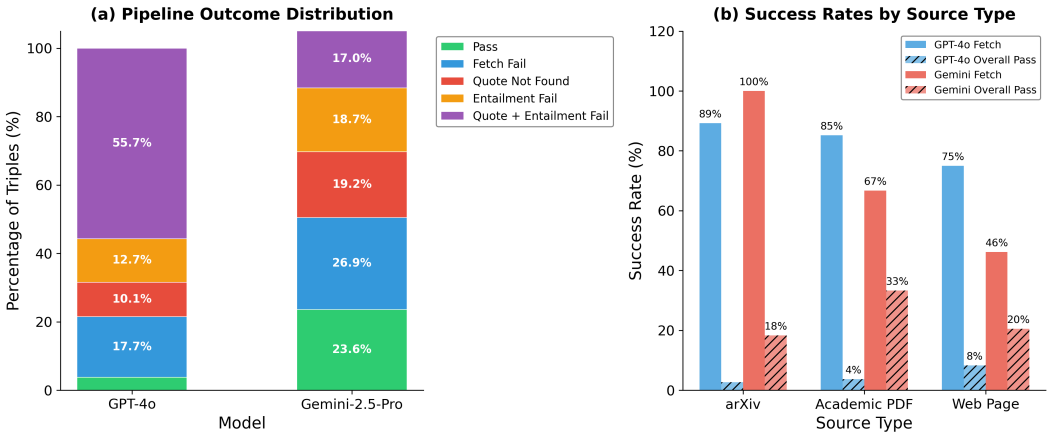

Figure 2: Pipeline outcome analysis. (a) Distribution of pipeline outcomes by model, showing that GPT-4o has concentrated failures in Quote+Entailment (55.7%) while Gemini-2.5-Pro has more distributed failures. (b) Success rates by source type, revealing that high fetch rates (75–100%) do not translate to high overall pass rates (3–33%) due to quote validity bottleneck.

Drop-Only achieves the highest entailment pass rate (70.0%) because only high-quality triples that pass both quote validity and entailment checks survive.

4.4 Error Analysis

Figure 2 analyzes pipeline failure modes to identify the primary bottleneck.

Quote validity is the primary bottleneck. Despite high source fetch success rates (89–100% for arXiv papers), overall pass rates remain low (2.7–18.3%). The gap between fetch success and overall pass reveals that quote validity—not source availability—is the fundamental limitation. Even when sources are successfully retrieved, LLMs produce quotes that do not appear verbatim in the source text. This indicates that current LLMs paraphrase rather than copy exact passages, limiting the effectiveness of quote-based verification.

Failure modes differ by model. GPT-4o shows concentrated failures in the Quote+Entailment category (55.7% of triples), indicating that most generated quotes both fail to appear in sources and fail semantic entailment. Gemini-2.5-Pro shows more distributed failures across categories, with higher fetch failures (26.9%) but better quote validity when sources are available. This suggests model-specific optimization opportunities for improving quote generation fidelity.

5 Conclusion

We presented QuoteVerify, an inference-time pipeline for citation verification in deep research reports. By requiring structured citation triples with explicit evidence quotes and applying multi-stage verification through quote validity checking and NLI entailment gating, QuoteVerify achieves statistically significant improvements over standard baselines: +18.7pp on GPT-4o ($p = 0.019$) and +12.5pp on Gemini-2.5-Pro ($p = 0.011$). Our analysis reveals that the structured citation format drives most gains, while quote validity remains the primary bottleneck—even for successfully fetched sources, LLMs produce valid quotes only 18–28% of the time.

These findings suggest two directions for future work. First, improving quote generation fidelity through retrieval-augmented prompting or fine-tuning could address the validity bottleneck. Second, relaxed matching strategies that tolerate minor paraphrasing while preserving semantic fidelity may better balance precision and coverage. Scaling evaluation to the full ReportBench dataset will further validate these approaches.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *ArXiv*, abs/2310.11511, 2023.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, N. Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. Rarr: Researching and revising what language models say, using language models. pp. 16477–16508, 2022.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. pp. 6465–6488, 2023.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. True: Re-evaluating factual consistency evaluation. *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, 2022.

Minghao Li, Ying Zeng, Zhihao Cheng, Cong Ma, and Kai Jia. Reportbench: Evaluating deep research agents via academic survey tasks, 2025. URL `https://arxiv.org/abs/2508.15804`.

Yifei Li, Xiang Yue, Zeyi Liao, and Huan Sun. Attributionbench: How hard is automatic attribution evaluation? pp. 14919–14935, 2024.

Potsawee Manakul, Adian Liusie, and M. Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *ArXiv*, abs/2303.08896, 2023.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, G. Irving, and Nat McAleese. Teaching language models to support answers with verified quotes. *ArXiv*, abs/2203.11147, 2022.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, M. Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *ArXiv*, abs/2305.14251, 2023.

Reiichiro Nakano, Jacob Hilton, S. Balaji, Jeff Wu, Ouyang Long, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, W. Saunders, Xu Jiang, K. Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback. *ArXiv*, abs/2112.09332, 2021.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. pp. 809–819, 2018.

Alex Wang, Kyunghyun Cho, and M. Lewis. Asking and answering questions to evaluate the factual consistency of summaries. *ArXiv*, abs/2004.04228, 2020.

Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Cheng Niu, Randy Zhong, Juntong Song, and Tong Zhang. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. pp. 10862–10878, 2023.

Yilong Xu, Jinhua Gao, Xiaoming Yu, Baolong Bi, Huawei Shen, and Xueqi Cheng. Aliice: Evaluating positional fine-grained citation generation. pp. 545–561, 2024.

Yumo Xu, Peng Qi, Jifan Chen, Kunlun Liu, Rujun Han, Lan Liu, Bonan Min, Vittorio Castelli, Arshit Gupta, and Zhiguo Wang. Citeeval: Principle-driven citation evaluation for source attribution. *ArXiv*, abs/2506.01829, 2025.