A Multi-Agent RAG Architecture for Citation-Grounded Scientific Literature Synthesis

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large language models can synthesize scientific text but frequently hallucinate citations and misattribute claims. We argue that review papers are an ideal domain for AI augmentation precisely because hallucinations are detectable: the ground truth exists in published literature, enabling systematic verification. We present a multi-agent architecture that exploits this property, built on Claude Code as an orchestrating agent. The system uses specialized skills for retrieval (Zotero-integrated search, RAG-based corpus querying) and spawns independent subagents for writing and verification. We demonstrate this architecture through a case study on pharmaceutical lyophilization, synthesizing 91 papers into a review manuscript. We describe our design rationale, implementation, and lessons learned, contributing architectural principles for verifiable AI-assisted scientific writing.

1 Introduction

Large language models demonstrate remarkable capabilities in scientific text generation, yet they remain fundamentally unreliable for scholarly communication. The core problem is hallucination: models fabricate citations, misattribute findings to sources that do not support them, and generate plausible but unverifiable claims (Huang et al., 2023). Where every claim must be traceable to evidence, these failures are not minor inconveniences—they undermine the epistemic foundation of scientific writing. Recent systems such as The AI Scientist (Lu et al., 2024), Coscientist (Boiko et al., 2023), and ChemCrow (Bran et al., 2023) have demonstrated LLM capabilities in experimental design and tool use, but the challenge of generating verifiable scientific prose with accurate citations remains largely unsolved.

We argue that review papers occupy a unique position in this landscape: they are simultaneously valuable targets for AI augmentation and tractable problems for verification. Unlike original research—which requires experiments, novel data, and genuine discovery—review papers are fundamentally text-based information synthesis. The "ground truth" exists in published literature: every claim in a review should be traceable to a cited source, and every attribution can be checked against the original paper. This property transforms the hallucination problem from an open detection challenge ("Is this claim true?") into a constrained verification task ("Does this source support this claim?")—the latter being mechanically solvable given access to the source documents.

This observation suggests a design principle: rather than attempting to detect hallucinations post hoc, systems for AI-assisted scientific writing should prevent them structurally. If the model can only cite claims that exist in a curated corpus, and if an independent verification agent checks every citation against its source, then hallucination becomes architecturally constrained. The system cannot fabricate a citation because citations are drawn from an indexed corpus; it cannot misattribute a finding because a separate agent verifies each attribution. The question shifts from "Did the model hallucinate?" to "Is the corpus complete and is the verification thorough?"—questions tractable for human oversight.

We present a multi-agent architecture that implements these principles, built on Claude Code as an orchestrating agent. The system separates retrieval, writing, and verification into independent components: specialized skills handle literature search (Zotero-integrated querying) and retrieval-augmented generation (semantic search over extracted claims), while independent subagents perform synthesis (drafting publication-ready prose) and adversarial verification (checking claims against sources). Human researchers intervene at two critical junctures: curating the corpus that defines the system's epistemic boundary, and reviewing the verified output before publication.

We demonstrate this architecture through a case study on pharmaceutical lyophilization, synthesizing 91 papers into a review manuscript. Our contribution is not a benchmark or empirical evaluation but an architectural argument: the structure of review papers—synthesis from verifiable sources—makes them amenable to AI augmentation in ways that original research is not, and multi-agent separation with grounded generation provides a principled approach to citation-verified scientific writing.

2 Related work

2.1 Retrieval-augmented generation for scientific writing

Retrieval-augmented generation (RAG) addresses the fundamental limitation that language model knowledge is frozen at training time and prone to fabrication (Lewis et al., 2020). By conditioning generation on retrieved documents, RAG systems can ground claims in external knowledge. Scientific writing, however, presents distinct challenges: standard chunking strategies destroy the semantic structure that makes claims citable, and generic similarity search produces topically related passages rather than specific supporting evidence.

Our approach addresses these limitations through claim-level corpus construction. Rather than chunking documents arbitrarily, we extract discrete claims with rich metadata: source section, page number, verbatim quote, and evidence type. This structure enables precise retrieval and provides provenance for verification. The RAG corpus serves as an epistemic boundary: the system can only cite claims that exist in the corpus with verifiable attribution.

2.2 Multi-agent LLM systems

Tool-augmented language models can perform actions beyond text generation—searching, executing code, and invoking APIs (Yao et al., 2023). Scientific applications have shown particular promise: The AI Scientist (Lu et al., 2024) automates ideation, experimentation, and paper writing; Coscientist (Boiko et al., 2023) integrates LLMs with robotic laboratory equipment; ChemCrow (Bran et al., 2023) augments LLMs with chemistry-specific tools. Multi-agent frameworks such as AutoGen (Wu et al., 2023) and ChatDev (Qian et al., 2023) distribute tasks across specialized agents, enabling adversarial dynamics where agents critique each other's output; multi-agent debate has been shown to improve factual accuracy (Du et al., 2023).

Our architecture employs multi-agent separation specifically to isolate writing from verification. The writer and reviewer subagents share no state, preventing the failure mode where a verifier rationalizes errors it participated in creating. This adversarial structure mimics peer review: an independent agent evaluates output it did not produce.

2.3 Citation verification and hallucination detection

Hallucination—generating plausible but unsupported content—remains a persistent challenge for language models (Huang et al., 2023). Post-hoc verification systems like FActScore (Min et al., 2023) decompose generated text into atomic facts and verify each against knowledge sources, while attribution-focused approaches, evaluated by benchmarks such as ALCE (Gao et al., 2023), prompt models to generate text with inline citations. Source-grounded generation constrains outputs to content derivable from provided sources, but scaling this to literature synthesis—where the corpus exceeds context limits—remains challenging.

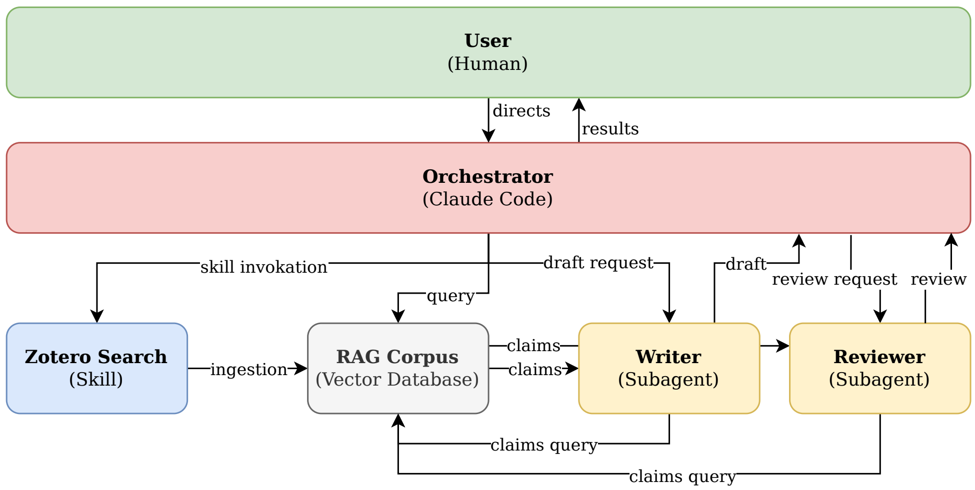

Figure 1: System architecture. The human directs the orchestrator (Claude Code), which delegates to specialized components: skills for retrieval (Zotero Search), a vector database for grounded claims (RAG Corpus), and independent subagents for writing and verification. Arrows indicate data flow: the orchestrator invokes skills and spawns subagents; claims flow from the corpus to both writer and reviewer; the draft passes from writer to reviewer for adversarial verification. The orchestrator coordinates but does not generate content directly.

Our approach combines these strategies architecturally. The RAG corpus provides source-grounded generation at scale: claims are drawn from an indexed corpus rather than model memory. The adversarial reviewer performs verification against a closed corpus where ground truth is mechanically accessible. This converts hallucination detection from an open problem ("Is this claim true?") to a tractable closed task ("Does this claim appear in the corpus with this attribution?").

3 Design rationale

The fundamental challenge in AI-assisted scientific writing is hallucination: language models fabricate citations and misattribute findings. Rather than detecting hallucinations post hoc, our architecture prevents them structurally through four design principles: separation of concerns, grounded generation, adversarial verification, and human-in-the-loop oversight.

3.1 Separation of concerns

The system decomposes literature synthesis into three independent functions: retrieval, writing, and verification (Figure 1). Each function is assigned to a dedicated component with a single responsibility: an orchestrating agent coordinates the workflow but does not generate content directly, delegating retrieval to specialized skills, drafting to a writer subagent, and fact-checking to a reviewer subagent.

This separation enables accountability. When an error appears, its source can be traced: missing citations indicate retrieval failures, misattributed claims indicate writing failures, and undetected errors indicate verification failures.

3.2 Grounded generation

The RAG corpus serves as an epistemic boundary: the writing agent can only cite claims that exist in the corpus. When a claim is absent, the agent inserts a placeholder (e.g., `[Citation needed: industrial-scale validation]`) rather than fabricating an assertion. Each claim in the corpus traces to a specific paper, section, page number, and verbatim quote—metadata that enables mechanical verification.

This design converts hallucination prevention into corpus curation: the question "Did the model hallucinate?" becomes "Is this claim in the corpus?"—answerable by querying a finite, indexed collection. The corollary is that output quality is bounded by corpus quality. Domain experts define the system's epistemic scope through paper selection, limiting autonomy but also limiting error.

3.3 Adversarial verification

The writer and reviewer subagents operate as independent adversaries, sharing no state and communicating only through the draft text. The reviewer sees the output without access to the writer's reasoning, retrieval queries, or intermediate steps—a separation that prevents the failure mode where a verifier rationalizes errors it participated in creating.

The adversarial dynamic mimics peer review: an independent agent evaluates work it did not produce, checking each claim against its source verbatim quote. Incorrect citations are flagged, numerical discrepancies are caught, and unsupported generalizations are identified. Undetected errors represent verification failures, creating accountability that incentivizes accuracy.

3.4 Human-in-the-loop

Human expertise operates at two junctures: corpus curation and final review. During corpus curation, domain experts select papers, determining what claims the system can make. During final review, researchers evaluate the verified draft before publication.

This arrangement reflects a division of labor: AI accelerates the mechanical aspects of synthesis—searching, retrieving, and drafting—while humans provide domain judgment, scientific assessment, and accountability. Automated verification can detect misquotations and numerical discrepancies but cannot assess whether a cited study was well-designed or whether the synthesis draws appropriate conclusions. By constraining AI to a supporting role, the architecture captures efficiency benefits while preserving scientific integrity.

4 System architecture

Our system uses Claude Code as an orchestrating agent that coordinates *skills* (reusable prompt templates with tool access) and *subagents* (autonomous agents for complex tasks). This separation ensures the orchestrator delegates without generating content directly.

4.1 Skills for retrieval

Zotero search. This skill queries a curated Zotero library rather than the open web, constraining searches to domain-expert-selected papers. The key innovation is a code execution pattern: instead of returning results directly to the LLM context (risking overflow), Python code executes in a sandbox, processing hundreds of items and returning only top-ranked results.

A single query triggers parallel search strategies: semantic search (vector similarity), keyword search (title/author/year and full-text modes), and tag-based search. Each strategy fetches up to 50 items, yielding 250+ candidates. Results are deduplicated by item key and ranked by query term frequency in title (highest weight), abstract frequency, tag matches, and recency (2020+ bonus). Only the top 20 results return to the orchestrator; post-hoc filtering supports item type, date range, and tag constraints.

This addresses three limitations of raw Zotero MCP: context overflow (250+ items fetched, 20 returned), single-strategy limitation (automated multi-strategy search), and lack of ranking (relevance-scored results).

RAG paper writer. This skill implements retrieval-augmented generation over extracted claims rather than arbitrary document chunks, following a four-stage pipeline.

Stage 1: PDF processing. Docling extracts text while preserving structure—each segment is tagged with section (normalized to canonical names) and page number.

Stage 2: Claim extraction. An LLM extracts discrete, verifiable claims as atomic standalone statements. Each claim includes `text` (rewritten claim), `verbatim` (exact supporting quote), `paper_key`,

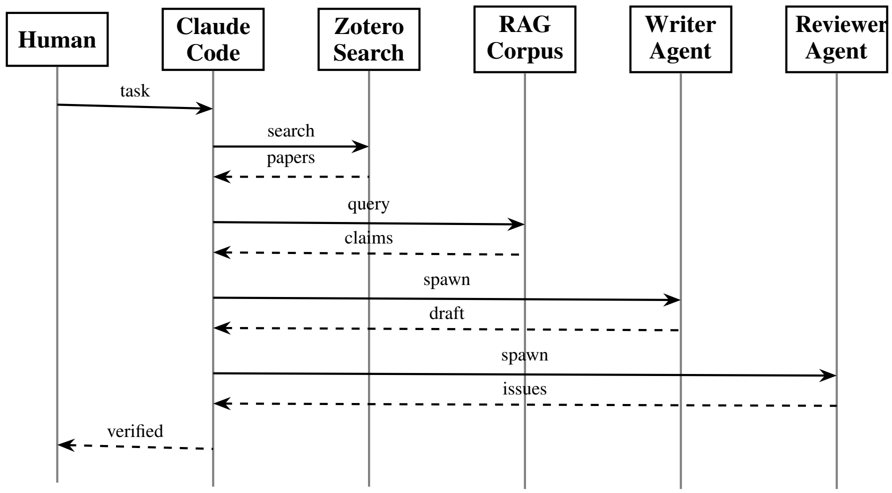

Figure 2: Sequence diagram showing data flow for a single writing task. The orchestrator (Claude Code) coordinates skills (Zotero Search, RAG Corpus) and spawns independent subagents (Writer, Reviewer). Solid arrows indicate requests; dashed arrows indicate responses. Human intervention occurs at task initiation and final review.

`authors`, `year`, `doi`, `section`, `page`, `claim_type` (finding/method/background/limitation), and `evidence_type` (experimental/computational/review/theoretical). This 4×4 schema enables fine-grained retrieval filtering. Extraction uses structured output via tool use or JSON mode, yielding 40–100 claims per paper.

Stage 3: Embedding and storage. Claims are embedded using Voyage AI's `voyage-3` model (1024 dimensions with asymmetric query/document embeddings) and stored in ChromaDB with HNSW indexing. The corpus configuration locks the embedding model at creation, preventing the silent failure of mixed embeddings.

Stage 4: Query. Queries match against the corpus via cosine similarity with metadata filtering (claim type, evidence type, year range, paper keys). Results include similarity scores and full metadata, with verbatim quotes enabling verification.

Traceability. Every claim traces to a specific paper, section, page, and verbatim quote—the foundation for verification. The writing agent cannot cite nonexistent claims, and the reviewer can verify any claim against its source.

4.2 Subagents for synthesis and verification

Scientific manuscript writer. This subagent synthesizes retrieved claims into publication-ready prose under explicit constraints: every factual claim must be cited (or marked `[Citation needed: X]`); IMRaD structure with appropriate voice conventions; technical rigor in definitions, units, and statistics. Output is LaTeX with `\citep{}` commands matching Zotero keys.

Science reviewer. This subagent performs adversarial verification in four phases: (1) line-by-line evaluation of each claim's support and clarity, (2) classification as VERIFIED, UNVERIFIED, UNFOUNDED, or VAGUE, (3) logic checking for argument flow and non-sequiturs, and (4) writing quality assessment. The key constraint is that quantitative claims must match source documents exactly.

Adversarial separation. Writer and reviewer share no state: the reviewer sees only output text without knowledge of the writer's reasoning, mimicking peer review through independent evaluation of work not produced by the evaluator.

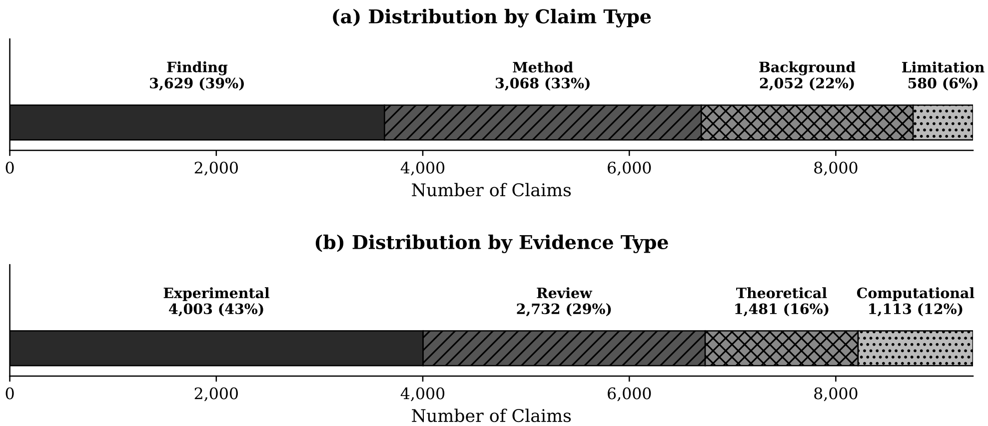

Figure 3: Distribution of 9,329 claims extracted from 64 papers in the lyophilization corpus. Stacked bars show proportions by claim type (top) and evidence type (bottom). Findings and experimental evidence dominate, reflecting the domain's empirical focus.

4.3 Data flow

The orchestrator coordinates four phases: (1) **Search**—invoke Zotero skill to identify relevant papers; (2) **Ingest**—process papers through the RAG pipeline; (3) **Retrieve and write**—query corpus and pass results to manuscript writer; (4) **Verify**—pass draft to science reviewer for source checking. Human researchers intervene at corpus curation and final review, positioning AI as an accelerant while preserving human judgment.

4.4 Implementation

The system stack comprises Claude Code (Opus 4.5) for orchestration, Docling 2.x for PDF processing, Gemini Flash via OpenRouter for claim extraction ($0.10/paper), Voyage AI voyage-3 for embeddings, ChromaDB for vector storage, Zotero with local API, and LaTeX with Biber. The corpus and configuration are version-controlled for reproducibility.

5 Case study: lyophilization digital twins

We validated the architecture by using it to write a review paper on digital twins for pharmaceutical lyophilization—computational models that simulate the freeze-drying process used to stabilize vaccines, biologics, and other temperature-sensitive drugs. This section describes the corpus, presents four illustrative cases, and summarizes aggregate statistics.

5.1 Domain and corpus

Lyophilization was selected for three reasons: the domain is technically complex, involving heat and mass transfer, phase transitions, and formulation science—a stress test for accurate synthesis; claims are often quantitative (temperatures, times, pressures), enabling objective verification; and one author has domain expertise, providing ground truth for evaluating system output.

The corpus comprises 64 papers from a curated Zotero collection, processed through the RAG pipeline described in Section 4. Claim extraction yielded 9,329 discrete claims with full metadata; Figure 3 shows the distribution across claim types and evidence types.

The following cases illustrate system behavior across a range of scenarios, from straightforward success to those requiring human intervention.

5.2 Case 1: Grounded retrieval succeeds

When querying for manometric temperature measurement (MTM), the retrieval skill returned claims from Tang et al. (2005), including *"MTM measurements were used to select the optimum shelf temperature, to determine drying end points, and to evaluate residual moisture content in real-time."* The manuscript writer synthesized this into *"Manometric temperature measurement represents a versatile process analytical technology capable of real-time monitoring across multiple freeze-drying parameters."* The science reviewer verified all claims against source verbatim quotes: precise retrieval, grounded synthesis, successful verification.

This case represents the intended workflow: the synthesis preserved the source's scope (three specific applications) while improving readability, and numerical values from other retrieved claims—such as "two-thirds of total primary drying time"—appeared verbatim in the output. No human intervention was required; the system produced publication-ready prose with verifiable citations.

5.3 Case 2: Reviewer catches misattribution

The manuscript writer produced *"the formulation must remain in a solid state during primary drying,"* which the science reviewer flagged as UNVERIFIED. The source verbatim stated *"After freezing, the formulation should be in solid state"* (Tang & Pikal, 2004). The writer had substituted "during primary drying" for "after freezing"—plausible, since both involve low temperatures, but the source describes the post-freezing state before sublimation begins. The corrected text read *"After freezing, the formulation must remain in a solid state."* The reviewer caught this phase-timing error by mechanical comparison against stored quotes.

This error is instructive because it would likely survive human review. Both phrases sound correct, and both involve temperature constraints in lyophilization; a reviewer without the source open might accept the paraphrase. The adversarial reviewer, mechanically comparing against verbatim quotes with no knowledge of the writer's intent, caught the discrepancy before it could propagate.

5.4 Case 3: Corpus gap handled correctly

When asked to address industrial-scale model validation, the manuscript writer produced *"systematic validation of these computational frameworks at manufacturing scale remains limited in the published literature* [Citation needed: industrial-scale validation]." The corpus contained laboratory and pilot-scale studies but lacked manufacturing-scale data; rather than fabricating a citation, the system marked its epistemic boundary. The human researcher then decides whether to expand the corpus, acknowledge the gap, or remove the claim.

The placeholder reflects a genuine gap in the published literature: most lyophilization modeling papers report results from research-scale equipment. This is the epistemic boundary working as designed—the system's knowledge is bounded by the corpus, and gaps become visible rather than papered over with hallucinated citations. Corpus curation is thus iterative, not one-time; the system reveals what it lacks.

5.5 Case 4: Human expertise overrides system output

The writer produced *"Disaccharide cryoprotectants are essential for successful mRNA-LNP lyophilization... These findings establish disaccharides as reliable stabilizers."* The reviewer verified both citations, each tracing accurately to its source. But the domain expert recognized an overgeneralization: Muramatsu et al. used 10% sucrose *with 10% maltodextrin* in a specific formulation; Zhao et al. used different ratios in a distinct composition. The synthesis implied "disaccharides" were the key variable when the *complete formulation* determines stability. The corrected text read *"Disaccharide-based formulations have shown promise, though optimal concentrations remain formulation-specific."* This error—overgeneralizing formulation-specific findings—passes citation verification but fails scientific reasoning, requiring domain expertise to detect.

This case reveals the verification ceiling: the system can confirm that sources say what the synthesis claims, but it cannot assess whether combining those sources produces valid scientific reasoning. A reader of the original paragraph might conclude that any disaccharide ensures successful lyophilization—a dangerous oversimplification. Catching this required knowing that formulation sci-

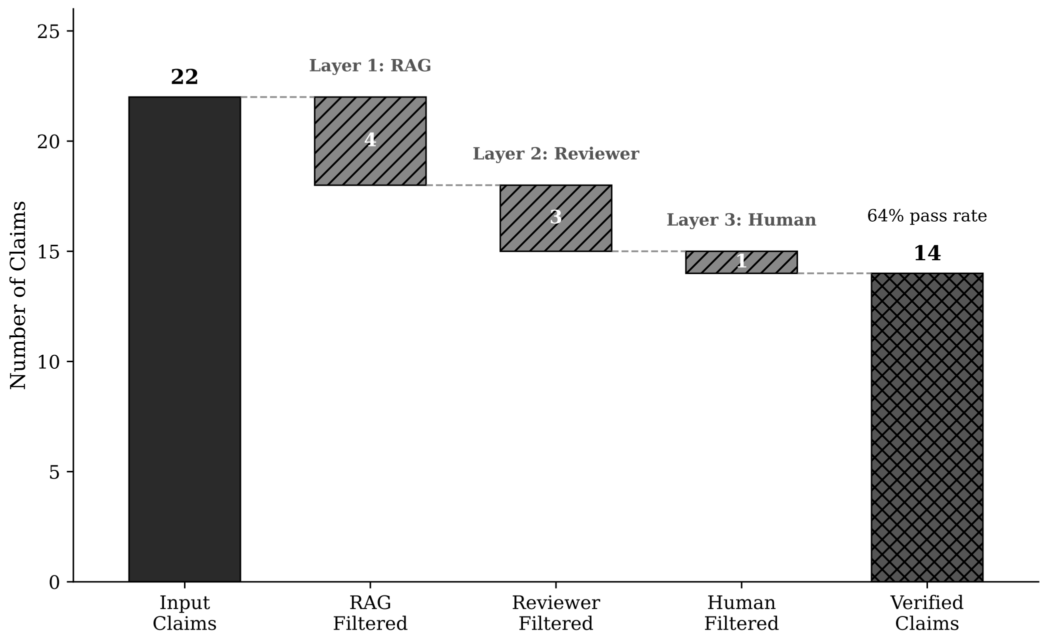

Figure 4: Waterfall chart of layered verification. From 22 synthesized claims, Layer 1 (RAG) flagged 4 corpus gaps, Layer 2 (adversarial reviewer) caught 3 misattributions, and Layer 3 (human expert) caught 1 timing error. 14 claims (64%) passed all verification layers.

ence is holistic, with excipient concentration, co-excipients, lipid composition, and process parameters all interacting. No amount of citation checking substitutes for this domain knowledge.

5.6 Observations

To quantify system performance beyond individual cases, we conducted a controlled test: the pipeline processed 22 synthesized claims drawn from across the review manuscript, tracking each claim through all verification layers. Figure 4 summarizes the results. The architecture caught 8 errors across three layers: RAG flagged 4 corpus gaps (Layer 1), the adversarial reviewer caught 3 misattributions (Layer 2), and human review identified 1 reasoning error (Layer 3). The remaining 14 claims passed all verification layers.

The key finding is not that 64% of claims passed but that 36% would have been published errors without this architecture. Each verification layer catches errors the others miss: RAG constraints prevent hallucinated citations entirely, the adversarial reviewer detects subtle misattributions through mechanical comparison against verbatim quotes, and human expertise catches overgeneralizations that pass citation verification but fail scientific reasoning. Without layered verification, these 8 erroneous claims would have appeared in the final manuscript.

The four cases presented above illustrate these layers qualitatively. Case 1 represents the 14 verified claims, where precise retrieval and faithful synthesis produce output that passes all checks. Case 2 exemplifies the 3 misattributions caught by Layer 2: the reviewer's mechanical comparison against verbatim quotes detected phase-timing errors that human review might miss. Case 3 demonstrates Layer 1 in action, with the system marking corpus gaps rather than hallucinating citations, accounting for 4 flagged claims. Case 4 represents the 1 claim caught only at Layer 3, where overgeneralizing formulation-specific findings passes citation verification but fails scientific reasoning, requiring domain expertise.

6 Discussion

Review papers are tractable for AI augmentation because they synthesize claims from verifiable sources, converting hallucination detection into citation verification. The system cannot cite papers not in the corpus or attribute claims without verifiable quotes—architectural constraints that make certain failure modes impossible. The architecture has clear limitations: corpus dependency bounds output quality, citation verification cannot assess scientific reasoning, and multi-agent orchestration is slower than single-model generation.

This architecture differs fundamentally from autonomous science systems such as The AI Scientist (Lu et al., 2024) and Coscientist (Boiko et al., 2023), which target discovery—generating hypotheses, running experiments, producing novel findings. We target synthesis, where ground truth exists in published sources and verification can be mechanized. The architecture applies when three conditions hold: the task is synthesis rather than discovery, a finite corpus can be curated, and domain experts are available for curation and final review. When these conditions hold, layered verification—retrieval constraints, adversarial review, human expertise—catches errors at different levels.

7 Conclusion

Review papers are uniquely tractable for AI augmentation because hallucinations are detectable against published sources. Our architecture implements layered verification: RAG prevents fabrication, adversarial review catches misattribution, and human expertise catches invalid synthesis. The system does not eliminate human judgment but concentrates it where it matters, automating the mechanical verification that humans routinely skip.

For researchers considering AI-assisted writing, the practical division of labor is this: AI handles mechanical tasks—searching, retrieving, drafting, checking citations against sources—while humans handle scientific tasks—selecting what to review, curating the corpus, assessing whether synthesis draws valid conclusions. This division exploits what each does well: AI scales verification that humans skip, and humans provide judgment that AI cannot. The result is not autonomous scientific writing but augmented scientific writing, where architectural constraints convert an open problem (detecting hallucination) into a tractable one (verifying citations).

References

Boiko, Daniil A. et al. (Dec. 21, 2023). "Autonomous Chemical Research with Large Language Models". In: *Nature* 624.7992, pp. 570–578. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-023-06792-0. URL: https://www.nature.com/articles/s41586-023-06792-0 (visited on 12/29/2025).

Bran, Andres M. et al. (Oct. 2, 2023). *ChemCrow: Augmenting Large-Language Models with Chemistry Tools*. DOI: 10.48550/arXiv.2304.05376. arXiv: 2304.05376 [physics]. URL: http://arxiv.org/abs/2304.05376 (visited on 12/29/2025). Pre-published.

Du, Yilun et al. (May 23, 2023). *Improving Factuality and Reasoning in Language Models through Multiagent Debate*. arXiv: 2305.14325. URL: https://arxiv.org/abs/2305.14325.

Gao, Tianyu et al. (2023). "Enabling Large Language Models to Generate Text with Citations". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. URL: https://arxiv.org/abs/2305.14627.

Huang, Lei et al. (Nov. 9, 2023). *A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions*. arXiv: 2311.05232. URL: https://arxiv.org/abs/2311.05232.

Lewis, Patrick et al. (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *Advances in Neural Information Processing Systems*. Vol. 33, pp. 9459–9474. URL: https://arxiv.org/abs/2005.11401.

Lu, Chris et al. (Sept. 1, 2024). *The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery*. DOI: 10.48550/arXiv.2408.06292. arXiv: 2408.06292 [cs]. URL: http://arxiv.org/abs/2408.06292 (visited on 12/29/2025). Pre-published.

Min, Sewon et al. (2023). "FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation". In: *Proceedings of the 2023 Conference on Empirical Methods in

Natural Language Processing. Association for Computational Linguistics, pp. 12076–12100. URL: https://arxiv.org/abs/2305.14251.
Qian, Chen et al. (July 16, 2023). *ChatDev: Communicative Agents for Software Development*. arXiv: 2307.07924. URL: https://arxiv.org/abs/2307.07924.
Wu, Qingyun et al. (Aug. 16, 2023). *AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation*. arXiv: 2308.08155. URL: https://arxiv.org/abs/2308.08155.
Yao, Shunyu et al. (2023). "ReAct: Synergizing Reasoning and Acting in Language Models". In: *International Conference on Learning Representations*. URL: https://arxiv.org/abs/2210.03629.

AI Co-Scientist Challenge Korea Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction claim a multi-agent architecture for citation-grounded scientific writing, which is fully described in Sections 3–4 and demonstrated in Section 5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 discusses limitations including corpus dependency, verification ceiling (cannot assess scientific reasoning), and efficiency trade-offs of multi-agent orchestration.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper presents an architectural design and case study, not theoretical results requiring proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 4 describes the full system architecture, implementation stack (Section 4.4), and the case study (Section 5) provides corpus details and verification methodology.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While AI Co-Scientist Challenge Korea does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The RAG corpus contains extracted claims from copyrighted papers and cannot be publicly released. The architecture and methodology are fully described for reproduction with different corpora.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Section 4.4 specifies the implementation stack including models (Claude Opus 4.5, Gemini Flash), embedding model (Voyage AI voyage-3), and vector database (ChromaDB). Section 5.1 describes corpus size (64 papers, 9,329 claims).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [N/A]

Justification: This is a design paper with a qualitative case study. The verification results (Section 5.5) report counts of errors caught by each layer, not statistical experiments requiring significance tests.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The system uses cloud API services (Claude, Gemini Flash, Voyage AI) rather than local compute. Per-paper processing cost ($0.10) is noted in Section 4.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://nips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available academic papers for corpus construction and does not involve human subjects, deception, or potential for harm.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

Justification: The paper focuses on architectural design for scientific writing assistance. The positive impact (accelerating literature synthesis) is implicit; potential misuse (generating misleading scientific content) is mitigated by the verification architecture but not explicitly discussed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper describes an architecture using existing commercial APIs (Claude, Gemini) and does not release new models or scraped datasets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All referenced systems (Claude, Gemini, Voyage AI, ChromaDB, Docling, Zotero) are credited in Section 4.4. Academic papers in the corpus are properly cited.

    Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new datasets or models. It describes an architecture that others can implement using the documented components.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human subjects research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: The paper does not involve human subjects research requiring IRB approval.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.