IRIS: An Agentic Multi-Phase Framework for Automated Scientific Literature Review

Sergey Kolchenko [1] **Mahdi Zamanighomi** [1] **Amir Bayegan** [1]

Abstract

We present **IRIS** (**I**terative **R**efinement for **I**nformation **S**ynthesis), an agentic framework for automated scientific literature review. IRIS breaks the research process into three phases (iterative planning, parallel per-section research and writing, and final report compilation), orchestrated as a directed state graph. Two mechanisms are central to the design. First, *reflective research expansion*: an LLM-based critic repeatedly evaluates how well the accumulated evidence covers the target topic and generates new queries until a sufficiency threshold is met. Second, *critic-guided refinement*: when a section fails quality review, an LLM critic pinpoints what is missing (mechanistic detail, quantitative data, recent findings) and issues targeted follow-up searches rather than simply requesting a rewrite. Citations are tracked at the claim level, mapping every factual assertion back to specific source passages. On DeepResearch-Bench (50 PhD-level English research tasks across 22 domains), IRIS achieves a RACE overall score of 53.46, outperforming every commercial system reported in the benchmark, including Gemini 2.5 Pro Deep Research (48.88) and OpenAI Deep Research (46.98). IRIS wins all four RACE dimensions despite being restricted to PubMed and ArXiv, with no open-web retrieval.

1. Introduction

Over the past several years, LLMs have achieved strong performance on text generation, summarization, and question answering tasks (Brown et al., 2020; Touvron et al., 2023). Writing a rigorous scientific literature review, however, demands something qualitatively different: the author must plan what evidence to seek, gather it from disparate sources, judge its quality, and weave it into a coherent narrative with verifiable citations. None of these steps is trivial to automate, and their interaction makes the overall problem considerably harder and more time-consuming than any one step in isolation.

A growing body of work on "deep research" agents has begun to tackle this challenge, with Open Deep Research (LangChain, 2025) being the open-source system most relevant to ours. Yet existing systems have significant gaps. Importantly, single-pass generation tends to produce reports with uneven coverage and hallucinated claims. To address the challenge, RAG pipelines ground generation in retrieved documents but rarely iterate: they retrieve once and generate, with no mechanism to recognize that an important sub-topic was missed. Multi-agent architectures are more flexible in principle, but existing instantiations lack clear criteria for when to stop researching and how to recover when a section draft falls short.

This work is developed in the context of a broader drug-discovery platform at Cellarity that uses active learning to identify modulators of disease phenotypes from transcriptomic data (DeMeo et al., 2025). In that setting, literature review is a recurring bottleneck: scientists must continually synthesize evidence on disease biology, phenotypic endpoints, mechanisms of action, and prior compounds to inform both campaign design and interpretation of hits. IRIS automates this synthesis, producing structured reports with explicit source attribution that can be consumed directly by platform users or used as context for downstream agents.

We introduce IRIS, an agentic framework whose design is motivated by how a domain expert conducts a literature review: plan, search, assess coverage, search again if needed, draft, critique, and revise. IRIS builds on the graph-based "plan → retrieve → write" workflow of Open Deep Research (LangChain, 2025) and extends it with explicit iteration and reflection. Specifically, our key contributions are as follows:

1. **Diversified query decomposition**, in which a research topic or section description is expanded into multiple sub-queries spanning distinct intent categories (core-concept, mechanistic, recent-developments, comparative), avoiding the redundant near-duplicates that result from naively

[1]Cellarity, Somerville, MA, USA. Correspondence to: Amir Bayegan <CellarityPublications@cellarity.com, abayegan@cellarity.com>.

Accepted at the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026)

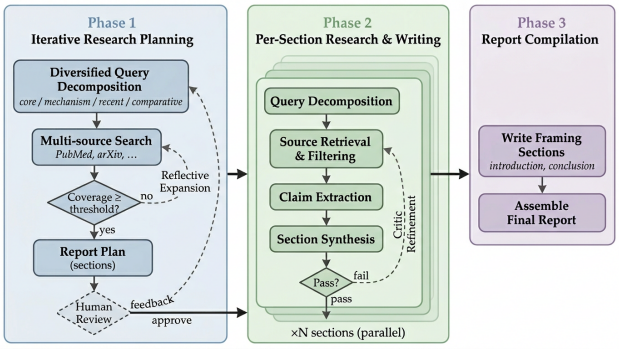

Figure 1. Three-phase architecture of IRIS. Arrows within phases denote iterative loops governed by LLM-based reflection and quality grading.

asking an LLM for multiple queries at once.

2. **Reflective research expansion**, a reflection loop in which an LLM critic estimates topic coverage and generates non-redundant queries until a sufficiency threshold is met.
3. **Critic-guided refinement**: rather than simply rejecting a weak section, an LLM critic diagnoses the type of missing information (mechanistic detail, quantitative evidence, recent findings) and prescribes targeted follow-up searches.
4. **An empirical evaluation** on DeepResearch-Bench showing that IRIS outperforms every reported commercial deep research agent on the RACE benchmark, with the largest gains on the insight and comprehensiveness dimensions.

IRIS additionally differs from Open Deep Research in two practical ways. First, it issues queries to multiple search APIs in parallel (e.g., PubMed and ArXiv at the same time) and merges duplicate results so each paper is processed once, whereas Open Deep Research queries one API at a time. Second, it exposes an optional human-in-the-loop plan checkpoint for interactive use.

2. Methods

The system is implemented as a directed state graph in LangGraph (LangChain, 2024) with asynchronous execution and typed state management. Figure 1 outlines the three phases.

2.1. Phase 1: Iterative Research Planning

In the planning phase, the LLM decomposes the research topic into a structured set of search queries spanning complementary intent categories: (i) core concepts to establish foundational context, (ii) mechanistic queries to interrogate underlying processes and causal relationships, (iii) recent developments to capture advances from the past several years, and (iv) comparative queries to identify systematic reviews and meta-analyses. These queries are executed against one or more search APIs (PubMed, ArXiv, Tavily, or DuckDuckGo), with parallel retrieval across sources when multiple APIs are configured.

After the initial search, a reflective expansion loop assesses whether the accumulated evidence sufficiently covers the report structure. If coverage falls below the target threshold, or if the reflector identifies critical gaps, it generates additional queries, each required to have substantially novel focus relative to prior searches. The loop runs for a bounded number of iterations.

Before Phase 2 begins, IRIS exposes an optional human-in-the-loop checkpoint: the proposed report plan is presented to the user, who can either approve it or supply free-text feedback that triggers plan regeneration. This checkpoint is disabled in our benchmark runs but is available for interactive use.

2.2. Phase 2: Per-Section Research and Writing

Once the research plan is finalized, Phase 2 conducts the actual literature research and writing for each section. The goal is to independently gather evidence for every section of the report, synthesize it into a draft, and iteratively refine the draft until it meets a quality threshold. All sections requiring primary research are processed in parallel via a shared subgraph. For each section, the system expands the section description into sub-queries using the same diversified intent categories as in Phase 1 (core concept, mechanistic, recent developments, comparative), treating query decomposition as a unified mechanism across phases rather than introducing separate heuristics.

Retrieved sources are filtered using a relevance threshold (rank-derived for sources such as PubMed and arXiv, and model-derived for web search APIs), minimum content length, and per-query source limits. When parallel queries to multiple APIs return the same URL (e.g., the same paper appearing in both PubMed and ArXiv), duplicates are merged so each unique source is processed once. Within each section's refinement loop, sources already consumed in earlier iterations are skipped to avoid redundant re-processing.

A source-extractor LLM assigns each document a usefulness score (1–5) and extracts claims with verbatim quotes and line numbers; documents scoring below 3 are discarded. All retained claims are stored with full provenance (URL, line numbers, verbatim text), enabling traceability from each generated statement to its supporting evidence.

A writer LLM synthesizes these claims into a section draft using an overlap-based aggregation heuristic: highly overlapping claims ($\geq$80%) are merged into a single statement, moderately overlapping claims (50–79%) are grouped within paragraphs, and low-overlap claims are presented

separately.

The draft is then evaluated by a grader LLM using weighted scoring across content alignment (40%), scientific quality (40%), and technical standards (20%). Passing requires at least 80% requirements coverage and no factual errors. Beyond scoring, the grader identifies missing elements (such as mechanistic detail, quantitative evidence, or recent clinical data) and generates targeted follow-up queries to address these gaps. This critic-guided refinement loop, analogous to peer review, runs for a bounded number of iterations, after which the highest-quality draft is selected.

2.3. Phase 3: Report Compilation

The final phase assembles the individual section drafts into a complete report. Sections that do not require primary research, such as the introduction and conclusion, are written at this stage using the completed research sections as context. The system then combines all sections in their planned order to produce the final report.

2.4. Model Specialization

The phases of the pipeline differ substantially in token volume and in the kind of reasoning required. Claim extraction and section writing dominate token usage, since each retrieved source must be read and the resulting draft can be long; assigning them a stronger, more expensive model would multiply cost without proportionally improving output. Grading, by contrast, runs on a much smaller token budget, but its decisions compound across the refinement loop, since a missed error propagates into the final report. We therefore use a fast, cost-efficient general-purpose model for the high-volume phases (planning, reflection, extraction, writing) and a stronger reasoning model for the grader. In our experiments these roles are filled by Claude Sonnet 4.6 and Claude Opus 4.6 (Anthropic, 2025b), respectively, but the underlying principle generalizes to any capable pair of models.

3. Experiments

3.1. Benchmark and Evaluation

We use DeepResearch-Bench (Du et al., 2025), which contains 100 PhD-level research tasks across 22 domains, each crafted by domain experts. Our evaluation uses the 50 English-language tasks, covering science, technology, finance, and several other fields.

DeepResearch-Bench evaluates reports using two metrics: RACE (Reference-based Adaptive Criteria-driven Evaluation), which scores report quality across four dimensions (comprehensiveness, insight, instruction following, and readability), and FACT, which audits citation factuality by retrieving each cited source and verifying whether it supports the associated claim. We report RACE only. FACT is designed for open-web retrieval and becomes uninformative in our setting. Its Effective Citations metric is an absolute count that favors web-crawling systems over curated-corpus approaches like IRIS. Additionally, its Citation Accuracy check for PubMed and arXiv largely reduces to re-reading abstracts already consumed by the agent during research. Because this asymmetry makes cross-system comparisons misleading, we leave retrieval-matched factuality evaluation to future work.

3.2. Configuration

IRIS uses Claude Sonnet 4.6 for planning, reflection, source extraction, and section writing, with Claude Opus 4.6 as the quality grader (Anthropic, 2025b). Search is limited to PubMed and ArXiv; to minimize the cost of the experiments, we did not include commercial web-search APIs such as Tavily though the framework supports them. We use three expansion iterations, a maximum search depth of four, and five queries per section. To contextualize these results, we compare against the published RACE scores reported in the DeepResearch-Bench paper (Du et al., 2025) for eight commercial systems spanning two categories: four LLMs augmented with a search tool (Claude-3.7-Sonnet w/Search (Anthropic, 2025a), Perplexity-Sonar-Reasoning-Pro (Perplexity AI, 2025), Gemini-2.5-Pro-Grounding (Google DeepMind, 2025), GPT-4.1 w/Search (OpenAI, 2025b)) and four dedicated deep research agents (Gemini 2.5 Pro Deep Research (Google DeepMind, 2025), OpenAI Deep Research (OpenAI, 2025a), Perplexity Deep Research (Perplexity AI, 2025), Grok Deeper Search (xAI, 2025)). All these systems retrieve from the open web.

3.3. Results

Table 1 groups systems into two categories. LLMs with search tools make one or a few retrieval calls during generation, without iterative critique. Deep research agents perform multi-step planning, retrieval, and synthesis; IRIS belongs to this category but is fully open in its architecture and restricted to academic search APIs. The results are noteworthy across several dimensions.

Competitive with commercial systems despite limited retrieval. IRIS scores 53.46 overall, placing it above Gemini 2.5 Pro Deep Research (+9.4%), OpenAI Deep Research (+13.8%), Perplexity (+26.5%), and Grok (+32.9%). This result holds despite a retrieval asymmetry: IRIS queries only PubMed and ArXiv, while all four commercial systems can draw on the open web.

Insight and comprehensiveness benefit most. The largest gains appear in insight (55.96, +15.4% over Gem-

Table 1. RACE evaluation results on DeepResearch-Bench (50 English tasks). Comp. = Comprehensiveness, Depth = Insight/Depth, Inst. = Instruction Following, Read. = Readability. Scores for commercial systems are taken from the DeepResearch-Bench paper (Du et al., 2025). Best value in each column is in **bold**.

System	Comp.	Depth	Inst.	Read.	**Overall**
LLMs with search tools					
Claude-3.7-Sonnet w/Search	38.99	37.66	45.77	41.46	40.67
Perplexity-Sonar-Reasoning-Pro	37.38	36.11	45.66	44.74	40.22
Gemini-2.5-Pro-Grounding	34.06	29.79	41.67	37.16	35.12
GPT-4.1 w/Search	29.42	25.38	42.33	40.77	33.46
Deep research agents					
Gemini 2.5 Pro Deep Research	48.53	48.50	49.18	49.44	48.88
OpenAI Deep Research	46.87	45.25	49.27	47.14	46.98
Perplexity Deep Research	40.69	39.39	46.40	44.28	42.25
Grok Deeper Search	37.97	35.37	46.30	44.05	40.24
IRIS (ours)	**53.71**	**55.96**	**50.77**	**51.72**	**53.46**

ini's 48.50) and comprehensiveness (53.71, +10.7% over Gemini's 48.53). This pattern is consistent with what the architecture is designed to do: the critic-guided refinement loop specifically targets missing analytical depth, so it is encouraging to see that reflected in the evaluation.

Readability is not sacrificed for depth. IRIS scores 50.77 on instruction following and 51.72 on readability, ahead of OpenAI Deep Research (49.27, 47.14) and Gemini (49.18, 49.44). The multi-phase design does not degrade coherence in pursuit of comprehensiveness.

Computational cost. Each task consumes roughly 569K tokens across 110 LLM calls and completes in about 21 minutes. Because sections are processed in parallel, report length does not translate linearly into wall-clock time.

4. Related Work

Agentic research frameworks build on RAG (Lewis et al., 2020), with Open Deep Research (LangChain, 2025) introducing a graph-based workflow that supports iterative search. IRIS extends this approach with reflective expansion and critic-guided refinement. GPT-Researcher (Elovic, 2023) follows a plan-and-execute paradigm but does not include a feedback loop after drafting. Commercial systems such as OpenAI and Gemini Deep Research also perform multi-step search and synthesis, though their implementations are not publicly documented.

Iterative self-critique methods such as Self-Refine (Madaan et al., 2023) and Reflexion (Shinn et al., 2023) demonstrate that LLMs can improve outputs through generated feedback. IRIS differs by grounding critique in a structured rubric and translating negative evaluations into targeted retrieval actions rather than rewriting text, focusing on resolving information gaps instead of surface-level prose issues.

Query decomposition is a common technique in retrieval systems, where multi-query RAG variants issue multiple retrieval calls per question. In contrast, our diversified decomposition organizes sub-queries around distinct intent categories rather than simple paraphrased versions of the original query.

5. Conclusion

IRIS demonstrates that a well-structured agentic pipeline (plan, search, critique, refine) can produce literature reviews competitive with commercial deep research products, even when restricted to academic search APIs such as PubMed and ArXiv. This competitiveness is driven primarily by gains in depth and comprehensiveness, where IRIS performs best due to its iterative refinement mechanisms.

The current system has limitations that also point toward its most promising directions for future work. The framework relies on carefully engineered prompts whose sensitivity to reformulation we have not yet studied, and real-time fact-checking during generation remains out of scope. The most impactful next step is extending retrieval beyond abstracts to full-text paywalled literature: most primary experimental findings and quantitative results appear only in the body of peer-reviewed papers, which IRIS currently does not reach, and full-text access would directly target the mechanistic-depth and quantitative-evidence dimensions where retrieval coverage most constrains report quality.

References

Anthropic. Claude 3.7 Sonnet and Claude Code. Anthropic Blog, February 2025a. URL https://www.anthropic.com/news/claude-3-7-sonnet. Accessed: April 2026.

Anthropic. Introducing Claude 4. Anthropic Blog, 2025b. URL https://www.anthropic.com/news/claude-4. Accessed: April 2026.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020. URL https://arxiv.org/abs/2005.14165.

DeMeo, B., Nesbitt, C., Miller, S. A., Burkhardt, D. B., Lipchina, I., Fu, D., Holderrieth, P., Kim, D., Kolchenko, S., Szalata, A., Gupta, I., Kerr, C., Pfefer, T., Rojas-Rodriguez, R., Kuppassani, S., Kruidenier, L., Doshi, P. B., Zamanighomi, M., Collins, J. J., Shalek, A. K., Theis, F. J., and Cortes, M. Active learning framework leveraging transcriptomics identifies modulators of disease phenotypes. *Science*, 390(6776), 2025. doi:

10.1126/science.adi8577. URL https://doi.org/10.1126/science.adi8577.

Du, M., Xu, B., Zhu, C., Wang, X., and Mao, Z. DeepResearch Bench: A comprehensive benchmark for deep research agents. *arXiv preprint arXiv:2506.11763*, 2025. URL https://arxiv.org/abs/2506.11763.

Elovic, A. GPT Researcher: Autonomous agent for comprehensive online research. GitHub repository, 2023. URL https://github.com/assafelovic/gpt-researcher.

Google DeepMind. Gemini 2.5: Our newest Gemini model with thinking. Google DeepMind Blog, March 2025. URL https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/. Accessed: April 2026.

LangChain. LangGraph: Building stateful, multi-actor applications with LLMs. GitHub repository, 2024. URL https://github.com/langchain-ai/langgraph.

LangChain. Open deep research. GitHub repository, 2025. URL https://github.com/langchain-ai/open_deep_research.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020. URL https://arxiv.org/abs/2005.11401.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., et al. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2303.17651.

OpenAI. Introducing deep research. OpenAI Blog, February 2025a. URL https://openai.com/index/introducing-deep-research/. Accessed: April 2026.

OpenAI. GPT-4.1. OpenAI Blog, 2025b. URL https://openai.com/index/gpt-4-1/. Accessed: April 2026.

Perplexity AI. Introducing Perplexity deep research. Perplexity Blog, February 2025. URL https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research. Accessed: April 2026.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2303.11366.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. URL https://arxiv.org/abs/2307.09288.

xAI. Grok 3 beta — the age of reasoning agents. xAI Blog, February 2025. URL https://x.ai/news/grok-3. Accessed: April 2026.