LARGE KNOWLEDGE MODEL: A KNOWLEDGE FOUNDATION FOR AGENTIC SCIENCE AT SCALE

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic science envisions many autonomous agents investigating concurrently while building on a shared, evolving body of scientific knowledge. This requires a knowledge foundation that supports high-concurrency access, preserves traceable and reusable reasoning, and grows incrementally. We propose the **Large Knowledge Model (LKM)**, a growing, agent-native knowledge foundation that provides a general representation of scientific knowledge across disciplines. LKM organizes the scientific literature into reasoning graphs, with claims as the core nodes and associated reasoning chains that make explicit how premises and evidence support conclusions. These source-grounded objects are persistent and addressable; cross-paper links organize them into aligned question, workflow, and evidence views. Newly extracted papers extend the foundation incrementally while preserving existing object identities. Building on this foundation, we develop an **agent-native, reasoning-aware scientific retrieval system** that retrieves claims together with their reasoning chains and sources, enabling agents to inspect and reuse the evidence underlying scientific conclusions. Across benchmarks, agents using LKM retrieve more evidence, cite more faithfully, and answer scientific questions more accurately: LKM nearly doubles the known supporting and contradicting evidence retrieved on SciFact-Open (818 versus 443 claim–paper pairs), reasoning graphs raise citation F1 on ScholarQABench by more than 5 points over the same retrieved papers, and LKM retrieval improves a fixed answering model by 9.3, 4.2, and 14.7 points over no retrieval on ChemBench, PubMedQA, and SciBench. LKM lays the foundation for a scientific ecosystem in which AI scientists not only recall accumulated knowledge but also extend it, returning new questions, workflows, and evidence to a memory that every subsequent investigation can build on.

1 INTRODUCTION

Agentic science promises a research paradigm in which many autonomous agents pursue investigations concurrently and build on accumulated scientific knowledge. Agents can already search the literature and synthesize evidence (He et al., 2025; Skarlinski et al., 2024; Asai et al., 2026); sustained investigation further requires them to establish what previous work has shown, how its conclusions were reached, and when they apply. As investigations multiply, agents need a shared, evolving scientific record that they can inspect and reuse across tasks and disciplines.

The existing record was written for human readers. A paper's argument, from its premises and evidence to its conclusions and the conditions under which they hold, is spread across prose, equations, figures, and references, and document or passage retrieval leaves agents to reconstruct it for every query (Lewis et al., 2020; Asai et al., 2026). The central challenge is a knowledge representation that spans disciplines with diverse terminology, methods, and forms of evidence, supports agent access and reasoning, and remains faithful to the source literature. Scientific arguments offer a common basis: claims can be expressed as propositions connected to the premises, evidence, and reasoning that support them under stated conditions. Reasoning graphs built from these relations make scientific knowledge accessible to agents while retaining the context and provenance needed to interpret it.

We present the *Large Knowledge Model* (LKM), a growing, agent-native knowledge foundation for science, built through four operations (Figure 3). *Extraction* turns each paper into a source-grounded reasoning graph whose questions, claims, and reasoning chains are addressable objects linked by

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

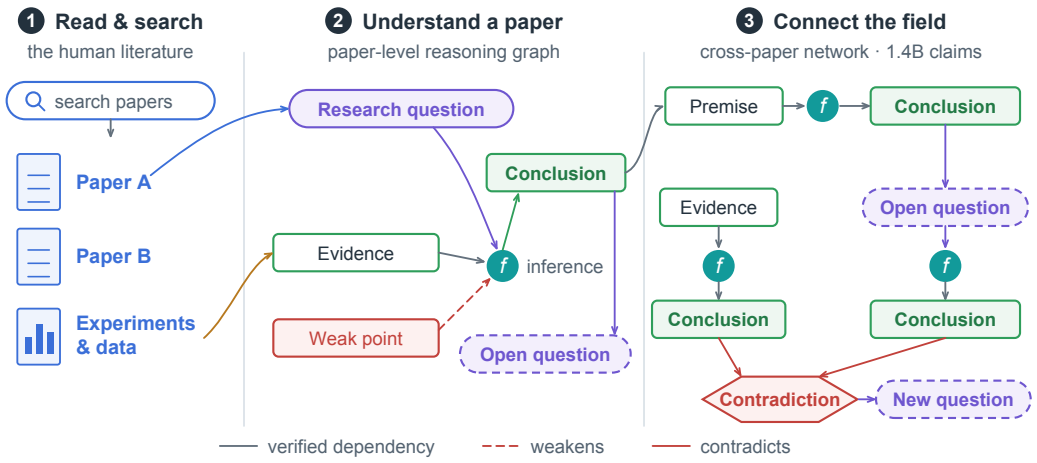

Figure 1: **From reading papers to new questions.** An agent searches and reads the literature (left). LKM extracts each paper into a reasoning graph in which the research question, evidence, and weak points enter an inference that reaches a conclusion and leaves open questions (center). Across papers, conclusions become premises of other inferences, open questions are taken up by later work, and contradictions between conclusions raise new questions (right).

typed dependencies. *Organization* joins recurring objects across papers and arranges them into a *Scientific Reasoning Landscape* with three aligned views: a Question Landscape of research problems and open directions, a Workflow Landscape of reusable research procedures, and an Evidence Landscape relating conclusions to their support, disagreement, and conditions. *Retrieval* lets an agent find a question, claim, or procedure and follow it to its premises, reasoning, and source. *Updates* are incremental: each newly extracted paper extends LKM without rewriting existing object identities. Because every agent reaches the same persistent objects, what one investigation retrieves, another can inspect, compare, and extend. Figure 1 traces this path from reading and searching papers to the contradictions and open questions that surface across a field.

LKM is deployed today. Its reasoning graphs span the major scientific disciplines and hold about 1.4 billion vectorized claims; it organizes nearly one million cross-paper question families, and answers requests from 50 concurrent clients with a median latency of 740 ms. Agents access it through an open API, a command-line interface, and agent skills, and researchers through web interfaces. Agents that build on LKM do better scientific work. LKM nearly doubles the known supporting and contradicting evidence retrieved for scientific claims on SciFact-Open (818 versus 443 claim–paper pairs within a 50-paper budget). On ScholarQABench, giving the writer the reasoning graphs of the same retrieved papers raises citation F1 by 5.3 and 5.1 points. With the answering model fixed, LKM retrieval improves accuracy over no retrieval by 9.3, 4.2, and 14.7 points on ChemBench, PubMedQA, and SciBench.

This foundation points toward a new mode of scientific progress. Open questions in the Question Landscape can seed new investigations; workflow families supply tested procedures to adapt; and the Evidence Landscape is designed to expose which observation would resolve a disagreement. The next step closes the loop: agents will return their own observations, failures, and conclusions to LKM as new reasoning objects, so that they accumulate knowledge collectively rather than rediscovering it in isolation.

Our contributions are:

1. **An agent-native representation of scientific knowledge** that turns papers into source-grounded reasoning graphs with addressable questions, claims, and multi-step reasoning (Section 3).

2. **A knowledge foundation at the scale of the literature**, organized into a Scientific Reasoning Landscape of questions, workflows, and evidence, and updated incrementally to about 1.4 billion claims (Section 4).

3. **A reasoning-aware retrieval system and evidence that agents benefit from it**: agents using LKM retrieve more evidence, cite more faithfully, and answer scientific questions more accurately (Section 5).

2 Related Work

Scientific retrieval and synthesis. Dense retrieval and retrieval-augmented generation connect language models to large corpora (Karpukhin et al., 2020; Lewis et al., 2020). Scientific systems build on this foundation: OpenScholar combines scientific retrieval with citation-backed synthesis (Asai et al., 2026), PaperQA2 uses language agents to search, synthesize, and resolve conflicting findings (Skarlinski et al., 2024), and PaSa and PaSaMaster perform iterative agentic paper search (He et al., 2025; Du et al., 2026). These systems retrieve documents or passages at query time. LKM supplies them with a persistent knowledge foundation in which the reasoning of each paper is already explicit, so retrieval returns scientific objects that can be expanded, compared, and cited.

Structured representations of science. Scholarly corpora and knowledge graphs organize bibliographic links, entities, and contributions (Lo et al., 2020; Jaradeh et al., 2019; Luan et al., 2018; Jain et al., 2020). Argument-centered representations capture how knowledge is established: argumentative zoning labels rhetorical roles (Teufel & Moens, 2002), nanopublications and micropublications attach assertions to provenance, evidence, and challenges (Groth et al., 2010; Clark et al., 2014), and *Linking Scholarly Contents* builds intra- and inter-paper argumentation graphs (Song et al., 2022). *The Discovery Engine* envisions navigable knowledge landscapes for agent-oriented discovery (Baulin et al., 2025). LKM realizes argument-level structure across the literature of the major scientific disciplines, organizes it around research questions and multi-step reasoning, and aligns it into question, workflow, and evidence landscapes that agents can query.

Graph-augmented retrieval. GraphRAG extracts entity–relation graphs and summarizes their communities for query-focused synthesis (Edge et al., 2024); LightRAG combines entity graphs with keyword and vector retrieval (Guo et al., 2025). Their graphs are organized around entities, relation descriptions, and topical communities. LKM makes scientific roles first-class instead: questions, premises, reasoning steps, conclusions, and open directions, connected by directed inferential relations such as `premise_of` and `concludes`. This representation exposes how a finding is established, under which conditions it holds, and how later work reuses or challenges it.

3 Extracting Reasoning Graphs from Papers

The unit of LKM is not a document or a passage but a scientific reasoning object. We represent each paper p as a typed graph with inference factors, $\mathcal{G}_p = (\mathcal{V}_p, \mathcal{F}_p, \mathcal{E}_p)$, where $\mathcal{V}_p$ contains the paper's scientific objects, $\mathcal{F}_p$ contains inference factors that map an ordered set of premises to a conclusion, and $\mathcal{E}_p$ contains typed relations among objects and factors. An agent can enter this graph through a question, a claim, or a procedure and follow the dependencies relevant to its task.

3.1 Reasoning Objects

Table 1 lists the object types. A *question* states what an investigation seeks to resolve; a *claim* is a proposition that serves as a premise or conclusion; a *reasoning chain* is an ordered sequence of steps $\sigma = (s_1, \ldots, s_T)$ explaining how a factor's premises yield its conclusion. *Settings* record the experimental or domain conditions under which claims hold, and *highlights* and *weak points* record where an argument is strongest and where it most needs scrutiny. Relations include `addresses` (question to claim), `premise_of` (premise to factor), `concludes` (factor to conclusion), `subproblem_of`, `highlight_of`, and `weakpoint_of`. Every object keeps its source references, so a claim retrieved from LKM always leads back to the paper that established it.

3.2 Construction

LKM builds each graph in stages from the paper's normalized full text. It first extracts the paper-level problem, its conclusions and their subproblems, and the open questions the paper leaves unresolved.

Table 1: **Reasoning objects in a paper-level graph.**

Object	Role in LKM
Question	Research problem, subproblem, or open question that frames the paper's claims
Claim	Decontextualized proposition that appears as a premise or conclusion
Inference factor	Maps an ordered premise set to a conclusion and records the method used
Reasoning chain	Ordered steps attached to a factor, explaining how the conclusion is reached
Setting	Experimental or domain conditions under which claims and chains apply
Highlight / weak point	Strongest supporting element and load-bearing vulnerability of an argument

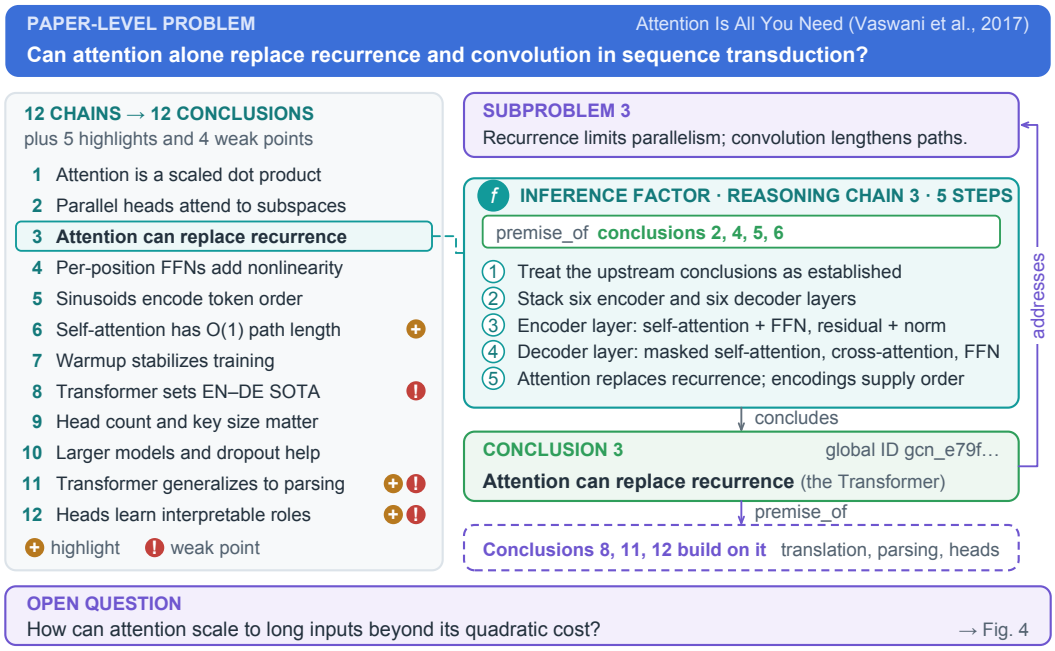

Figure 2: **Anatomy of an extracted paper.** The reasoning graph LKM stores for *Attention Is All You Need* (Vaswani et al., 2017). Left: its twelve chains, each named by its conclusion, with highlights (+) and weak points (!). Right: chain 3 expanded; conclusions 2, 4, 5, and 6 are its premises, and five steps conclude the Transformer, which addresses its subproblem and supports conclusions 8, 11, and 12.

For each conclusion, it reconstructs the reasoning chain and identifies the premises and evidence that carry the argument, together with load-bearing weak points and supporting highlights. It then rewrites conclusions and premises as self-contained propositions that remain meaningful outside their original paragraph, preserving their identifiers and source references. After structural validation, these objects form the paper's reasoning graph. Extraction is performed by DeepSeek-V4-Flash.

We audited extraction fidelity on 100 randomly sampled papers comprising 3,904 extracted objects. An LLM judge compared each object with the full text, and domain experts reviewed every flagged case; the confirmed hallucination rate is 0.97%. Most confirmed errors are semantic: a correct number attached to the wrong object, a miscount, a reversed direction, or an unsupported attribution to an experiment.

Figure 2 shows the graph extracted from *Attention Is All You Need*. The paper yields twelve reasoning chains, each reaching one conclusion that addresses its own subproblem, together with five highlights and four weak points. Scaled dot-product attention is the premise of multi-head attention, and multi-head attention, the feed-forward network, positional encoding, and the complexity analysis enter the inference that concludes the Transformer architecture, which in turn is a premise of the translation, parsing, and attention-head results. The inference expands into five ordered steps; a weak point marks that the translation results come from single runs without confidence intervals, and a

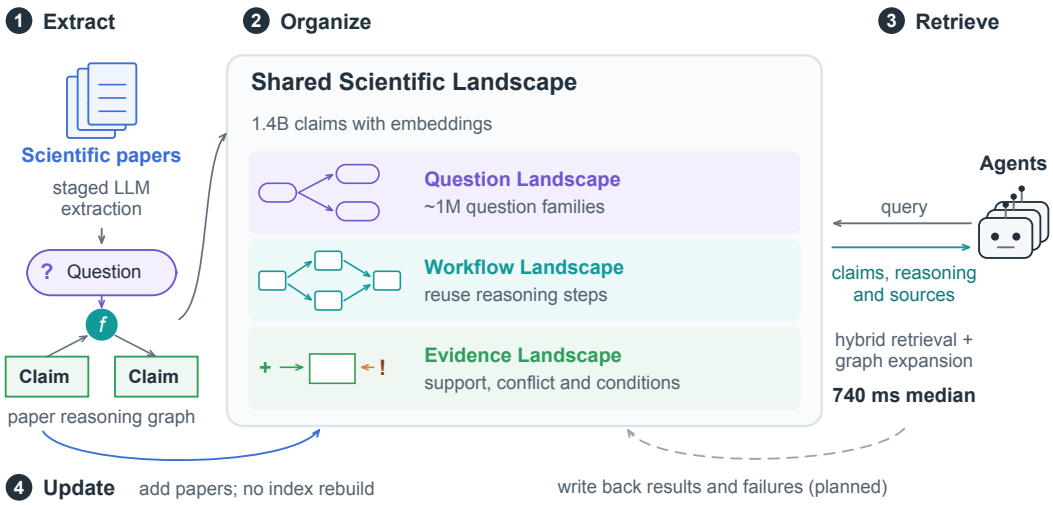

Figure 3: **LKM as a knowledge foundation for agentic science.** Papers are (1) extracted into reasoning graphs and (2) organized into Question, Workflow, and Evidence Landscapes. Agents (3) retrieve claims together with their reasoning and sources, and the foundation is (4) updated incrementally with each new paper. The dashed path marks the planned write-back of results produced by agents.

highlight marks that the architecture transfers to parsing. The quadratic cost of self-attention is stored as an open question, which is where the cross-paper traversal of Section 4.2 begins.

3.3 Identity and Retrieval

LKM keeps each paper's own objects while giving recurring objects a shared identity across the literature. A public claim's identity is determined by its object type, its decontextualized proposition text x_v, and its parameters in canonical order, and is materialized as a SHA-256 hash, the global ID shown in Figure 2. Objects with the same identity attach to one global representative that retains all of its source-paper occurrences; factors are integrated by matching their premises, conclusion, and type. Semantic similarity never merges identities. It forms a separate layer of neighborhoods and clusters used for discovery and for the cross-paper views of Section 4.

For retrieval, each public claim is embedded as $\mathbf{z}_v = \mathrm{Embed}(x_v) \in \mathbb{R}^{512}$ with Qwen3-Embedding-8B (Zhang et al., 2025), using its Matryoshka support to request 512-dimensional vectors directly; reasoning chains are embedded from their ordered step text. Claim vectors let agents find *what* was established, and chain vectors let them find *how* it was established. Hybrid retrieval combines these vectors with BM25 over titles and full text. Because retrieved identifiers resolve to graph objects, a single request can move from a semantic match to the claim's premises, reasoning steps, settings, and source paper.

4 Organizing Knowledge at Scale

Paper-level graphs expose the reasoning within individual investigations. Organization connects that reasoning across the literature, so that an agent can see what a field has asked, how it has investigated those questions, and what the resulting evidence establishes. Figure 3 summarizes the resulting foundation: papers are extracted, organized into three aligned landscapes, retrieved by agents, and continually updated as new papers arrive.

4.1 Scale and Incremental Updates

The deployed system contains reasoning graphs spanning all of arXiv, bioRxiv and ChemRxiv preprints, and journals in mathematics, physics, chemistry, astronomy, earth and life sciences,

engineering, information science, and environmental science. They hold about 1.4 billion vectorized public claims, complemented by abstract representations for a broader set of papers. Local and global records, canonical bindings, embeddings, and clusters are persisted in a cloud-native analytical database built on ClickHouse, and the vector index is served by 16 machines with 32 CPU cores and 256 GB of memory each. Under 50 concurrent clients sustained for 10 minutes, the production search endpoint answers with a median end-to-end latency of 740 ms and a 95th-percentile latency of 980 ms, fast enough to sit inside an agent's reasoning loop.

LKM is updated one paper at a time. When a paper has been extracted, each public object is resolved by its exact identity: it joins an existing global representative if one exists and creates a new one otherwise. Existing bindings are never rewritten, and only the objects introduced or matched by the new paper enter the embedding update. The cost of adding a paper is therefore proportional to its own content rather than to the size of the corpus, and a newly published result becomes retrievable without rebuilding the index.

4.2 The Scientific Reasoning Landscape

Over the corpus $\mathcal{P}$ of extracted papers, we define the *Scientific Reasoning Landscape* as $\mathcal{L} = (\mathcal{Q}, \mathcal{W}, \mathcal{B}, \mathcal{R})$, where $\mathcal{Q}$ is a set of cross-paper question families, $\mathcal{W}$ a set of workflow families, $\mathcal{B}$ a set of claim-centered evidence analyses, and $\mathcal{R}$ the relations connecting them. All three views are built from the same paper-level objects, so an inquiry can move from a question to the procedures used to study it and on to the evidence behind their conclusions (Figure 3, center).

Question Landscape. The Question Landscape organizes the literature by the problems that motivate it. Question families are formed by clustering the research questions that papers address together with the open questions they leave unresolved; an LLM then summarizes each family's shared question, its main variants, and the papers that investigate it, with every summary linked to its member objects. On an arXiv snapshot of 34.4 million questions, this produces 988,894 families that cover 99.5% of the papers, ranging from focused recurring problems to broad research themes (Section A.3). An open question thus becomes an entry point to everything the literature has tried.

Workflow Landscape. The Workflow Landscape organizes recurring ways of producing scientific knowledge (Belhajjame et al., 2015; Kulkarni et al., 2018). A workflow family clusters reasoning chains that follow a common procedure and summarizes its inputs and outputs, reasoning skeleton, task variants, and technique slots, the alternative techniques that can fill each step. Each family also records whether it is a dry, wet, or mixed investigation and how its results are verified. For an agent, a workflow family is reusable planning material: a sequence of scientific purposes, the choices available at each step, and source examples of how others instantiated it.

Evidence Landscape. The Evidence Landscape turns claim retrieval into evidence analysis (Wadden et al., 2020). Given a question, the Evidence Engine issues complementary queries, retrieves claims, expands their reasoning chains, and organizes the result as a source-linked evidence graph. Evidence is separated into supporting, opposing, conditional, insufficient, and methodologically divergent items, each tied to its premises, settings, and source. Contradiction groups record which claims are in tension and which conditions explain the disagreement, and research-gap records identify the observation that would resolve it.

Figure 4 shows how an agent traverses the landscape. It starts from the open question stored with the Transformer paper, how attention can scale beyond its quadratic cost, and the Question Landscape divides it into three subquestions answered by different papers (Wang et al., 2020; Dao et al., 2022; Dao, 2023). Linformer asks whether a low-rank projection can approximate attention; FlashAttention asks whether exact attention can cut memory traffic; FlashAttention-2, which builds on it, asks how to raise GPU utilization for long inputs. Each subquestion leads to a method in the Workflow Landscape and to the evidence behind its claims: the speedups and accuracy each paper reports, a proven IO lower bound for FlashAttention, and weak points such as Linformer's speed being measured on a single GPU type.

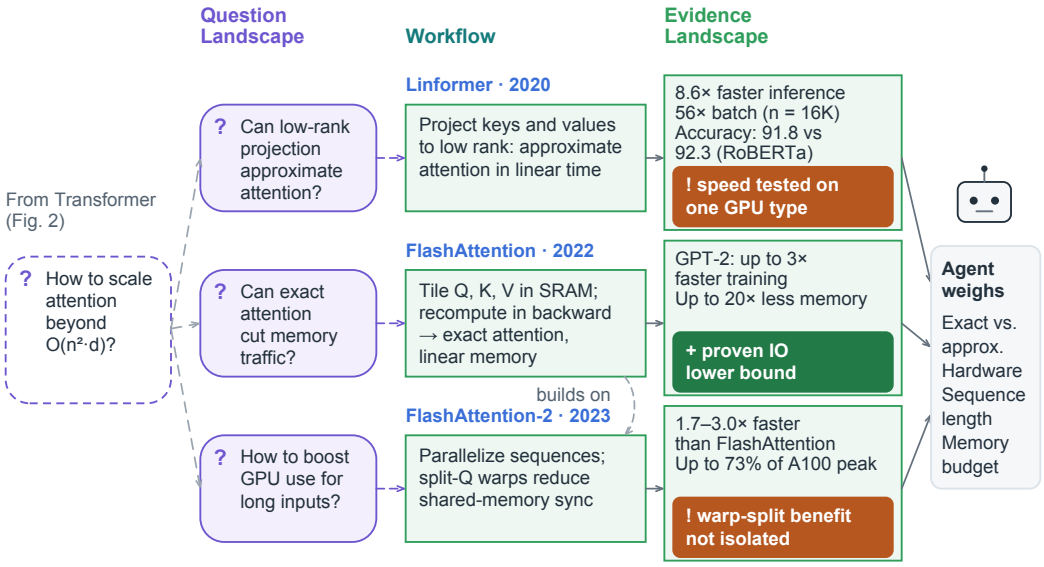

Figure 4: **An agent traversing the reasoning landscape.** Starting from the open question left by the Transformer paper (Figure 2), LKM surfaces three routes to efficient attention. Each route connects a subquestion to a method and to its evidence, with highlights (+) and weak points (!) attached, so the agent can choose an approach by exactness, hardware, and sequence length.

4.3 Interfaces for Agents

Agents reach LKM through an OpenAPI service, a command-line interface, and agent skills that expose the same operations as tool calls. The core operations are hybrid search over claims, questions, and abstracts; search over complete reasoning chains; retrieval of a paper's full reasoning graph; tracing why a claim holds through its premises and steps; and PDF extraction for papers not yet in LKM. Researchers use the same objects through web interfaces (Section A).

5 Experiments

We test whether agents do better scientific work when they recall reasoning rather than documents. The experiments follow three capabilities: discovering the evidence that bears on a scientific claim (Section 5.1), writing answers whose citations support what they state (Section 5.2), and finding papers and answering knowledge-intensive questions (Section 5.3). All experiments query the deployed LKM service.

5.1 Evidence Discovery

Judging a scientific claim requires finding the papers that support or contradict it. We evaluate this with SciFact-Open, an open-domain claim-verification benchmark (Wadden et al., 2022a; 2020), using its 279 claims and all 460 human-annotated positive claim–paper pairs. To account for relevant papers outside the benchmark's original annotation pool, we pool candidates from the released retriever and LKM and admit a new positive pair only when a DeepSeek judge (`deepseek-v4.1-flash`; SUPPORT and CONTRADICT thresholds 0.80 and 0.90) and MultiVerS (Wadden et al., 2022b) agree; human annotations always take precedence. The resulting reference set contains 1,125 known-positive pairs. Each system retrieves 50 papers per claim.

Within the same 50-paper budget, LKM retrieves 818 known-positive pairs, compared with 443 for the released retriever (Table 2). The claim-paired recall gain is 33.3 points (95% bootstrap interval 24.6–42.1). The gain holds for both directions of evidence: SUPPORT recall rises from 34.4% to 75.8%, and CONTRADICT recall from 47.4% to 67.8%, so an agent recalling from LKM sees more of the evidence that could overturn a claim, not only the evidence that confirms it. Retrieval breadth

Table 2: **Evidence discovery on SciFact-Open at 50 papers per claim.** monoT5 reranks LKM's candidates without changing the retrieved set.

Metric	Released	LKM	LKM+monoT5
Known-positive pairs retrieved	443	**818**	**818**
Evidence recall	39.38%	**72.71%**	**72.71%**
SUPPORT recall	34.35%	**75.80%**	**75.80%**
CONTRADICT recall	47.36%	**67.82%**	**67.82%**
Claim hit rate	**89.73%**	85.27%	85.27%
MRR@50	**0.7174**	0.5364	0.5707

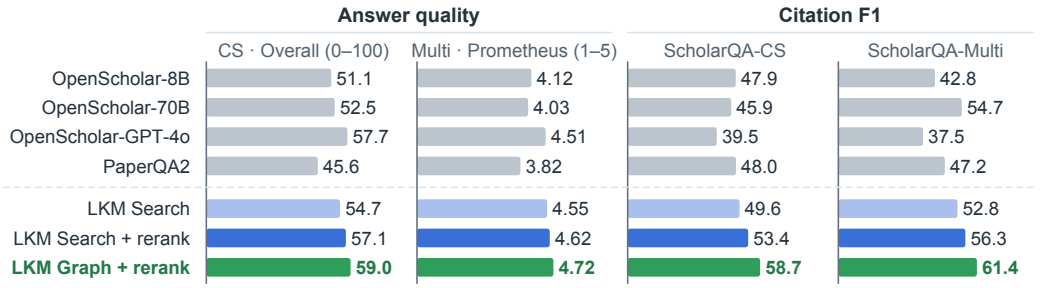

Figure 5: **Scientific synthesis on ScholarQABench.** Answer quality (left; GPT-4o rubric Overall on ScholarQA-CS, Prometheus on ScholarQA-Multi) and citation F1 (right) for three LKM configurations and published results of OpenScholar and PaperQA2 (gray). LKM Graph + rerank differs from LKM Search + rerank only in giving the writer the reasoning graphs of the top papers.

and ranking play complementary roles: LKM finds 540 known positives in its first 20 ranks and 278 more in ranks 21–50, and a downstream reranker such as monoT5 moves 81 of those later positives into the top 20, adapting the enlarged evidence pool to an agent's attention budget.

5.2 Attributable Scientific Synthesis

Long-form scientific answers are only as trustworthy as their citations. We evaluate synthesis and attribution on ScholarQA-CS (100 questions) and ScholarQA-Multi (108 questions) from ScholarQABench (Asai et al., 2024) with an OpenScholar-style pipeline: retrieval, reranking, drafting from the top 10 papers, self-feedback, and citation repair, with GPT-4o as the writer. We compare three LKM configurations: LKM Search, LKM Search with reranking, and LKM Graph with reranking, which shares its candidates, reranking, and 10-paper writing budget with LKM Search + rerank and additionally gives the writer the reasoning graphs of the top three papers. Answer quality follows the benchmark's native protocols, the GPT-4o rubric Overall score on ScholarQA-CS and the Prometheus score for organization, relevance, and coverage on ScholarQA-Multi; citation precision, recall, and F1 are computed with the benchmark's AttrScore evaluator. Published results of OpenScholar, PaperQA2 (Skarlinski et al., 2024), and SciRAG (Ding et al., 2025) serve as references.

LKM Graph obtains the highest answer quality (59.0 on ScholarQA-CS, 4.72 on ScholarQA-Multi) and citation F1 (58.7 and 61.4) among our configurations on both subsets (Figure 5). These scores also exceed the published results, which were obtained with different corpora, retrievers, and writers. The matched comparison isolates the contribution of reasoning: with the same candidates, reranking, and writing budget, adding reasoning graphs raises citation F1 by 5.3 points on ScholarQA-CS and 5.1 points on ScholarQA-Multi, and improves precision and recall together (Table 5). Reranking alone accounts for a smaller share of the gain, and LKM Graph also exceeds the reranked commercial search API baseline. Knowing how each paper reached its conclusions lets the writer both cite more accurately and support more of what it states.

Table 3: **Paper search and question answering.** Left: NDCG@k on PaSaMaster. Right: accuracy (%) on ChemBench, PubMedQA, and SciBench with GPT-5.4 fixed as the answering model.

Search API	@5	@10	@20
Commercial API	21.44	19.77	19.28
LKM Semantic	21.17	20.43	20.54
LKM Hybrid	**22.19**	**21.00**	**20.93**

Retrieval	ChemBench	PubMedQA	SciBench
None	64.78	77.20	76.21
Web search	70.88	81.00	86.90
arXiv	71.14	76.60	87.07
Commercial API	72.82	81.20	88.62
LKM	**74.08**	**81.40**	**90.90**

5.3 Paper Search and Knowledge-intensive Question Answering

We evaluate direct paper retrieval on the 244 queries of PaSaMaster (Du et al., 2026), comparing LKM's semantic and hybrid retrieval with a production scientific literature search API by NDCG@k. For question answering, GPT-5.4 is fixed as the answering model on ChemBench (n=2,785) (Mirza et al., 2025), PubMedQA (n=500) (Jin et al., 2019), and SciBench (n=580) (Wang et al., 2024), and we vary only the retrieval source.

LKM Hybrid achieves the highest NDCG at all three depths (Table 3), showing that reasoning objects also serve as entry points for finding papers. In question answering, LKM retrieval improves the fixed model by 9.3, 4.2, and 14.7 points over no retrieval and gives the highest accuracy on all three benchmarks, although its lead over the commercial API on PubMedQA is a single question and these point estimates carry no paired intervals. The largest gain appears on SciBench, whose problems require applying established scientific relations, consistent with LKM storing such relations as explicit claims and reasoning chains. Together with the preceding results, this traces a consistent picture: recalling reasoning lets agents find more of the relevant evidence, attribute their statements more faithfully, and reason more accurately.

6 Conclusion and Outlook

LKM rebuilds the scientific literature as a knowledge foundation for agentic science. It extracts the literature across disciplines into source-grounded reasoning graphs holding about 1.4 billion claims, organizes them into a Scientific Reasoning Landscape of questions, workflows, and evidence, and is updated with every newly published paper. Agents that build on it discover nearly twice as much known evidence for scientific claims, cite their sources more faithfully, and answer knowledge-intensive questions more accurately.

Limitations. Extraction by language models is imperfect: our audit covers 100 papers and measures hallucination (0.97%) but not omission, and extraction errors propagate into bindings, clusters, and retrieved context. Evidence carried mainly by figures, tables, or equations may be captured incompletely, and exact-identity binding leaves paraphrased propositions as separate identities linked only by semantic neighborhoods. The landscape views and the Evidence Engine's categorization are not yet directly evaluated; Section 5.1 measures evidence retrieval, not categorization. Our comparisons cover retrieval baselines and a matched graph ablation rather than controlled runs of other synthesis or graph-augmented retrieval systems.

Outlook. Science advances only as fast as its results accumulate. LKM moves the unit of accumulation from the paper to the reasoning object: because questions, workflows, and evidence share common objects, a technique proven on one problem can surface for another, even in fields that name it differently, extending literature-based discovery (Swanson, 1986) from findings to reasoning. The decisive step is to close the loop. When agents return their observations, failures, and conclusions with provenance and verification, negative results that rarely reach print are kept, open questions sharpen, and each investigation starts from what all others have learned. The literature then becomes not an archive that agents read, but a shared, self-correcting body of knowledge that they write together.

Reproducibility statement

We support reproducibility by documenting the system construction and each evaluation stage at the level needed to recreate the reported comparisons. Section 5 specifies the benchmark datasets and sample sizes, retrieval depths and budgets, baselines, metrics, fixed answering model, and the model and decision thresholds used to construct the pooled SciFact-Open reference set. Section 3 describes graph construction, canonical identifiers, embedding generation, and retrieval, while Appendix A reports the clustering parameters and the public interfaces through which LKM outputs can be inspected. We use cited public benchmarks and preserve their original human annotations. Because LKM is a continuously updated production system, exact retrieval results depend on the evaluated index snapshot and query time; fixed offline snapshots are identified where applicable, and live-service results should be interpreted with this source of variation in mind.

AI use statement

In this work, we used generative AI tools for writing assistance and language polishing, retrieval and discovery of related work, assistance with research ideation and experiment execution, code and figure preparation, and synthetic dataset generation. Large language models are also components of the system and its evaluation as described in the paper, including reasoning-graph extraction (Section 3) and the answering models, writers, and judges of the benchmarks (Section 5). The authors reviewed all AI-assisted text, code, figures, and results, and take responsibility for the final content of this paper, including its text, claims, and research artifacts.

References

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'Arcy, David Wadden, Matt Latzke, Minyang Tian, Pan Ji, Shengyan Liu, Hao Tong, Bohao Wu, Yanyu Xiong, Luke Zettlemoyer, Graham Neubig, Dan Weld, Doug Downey, Wen-tau Yih, Pang Wei Koh, and Hannaneh Hajishirzi. Openscholar: Synthesizing scientific literature with retrieval-augmented lms. *arXiv preprint arXiv:2411.14199*, 2024.

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, et al. Synthesizing scientific literature with retrieval-augmented language models. *Nature*, 650:857–863, 2026. doi: 10.1038/s41586-025-10072-4.

Vladimir Baulin, Austin Cook, Daniel Friedman, Janna Lumiruusu, Andrew Pashea, Shagor Rahman, and Benedikt Waldeck. The discovery engine: A framework for AI-driven synthesis and navigation of scientific knowledge landscapes. *arXiv preprint arXiv:2505.17500*, 2025. doi: 10.48550/arXiv.2505.17500.

Khalid Belhajjame, Jun Zhao, Daniel Garijo, Matthew Gamble, et al. Using a suite of ontologies for preserving workflow-centric research objects. *Journal of Web Semantics*, 32:16–42, 2015. doi: 10.1016/j.websem.2015.01.003.

Tim Clark, Paolo N. Ciccarese, and Carole A. Goble. Micropublications: a semantic model for claims, evidence, arguments and annotations in biomedical communications. *Journal of Biomedical Semantics*, 5:28, 2014. doi: 10.1186/2041-1480-5-28.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. doi: 10.48550/arXiv.2307.08691.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *arXiv preprint arXiv:2205.14135*, 2022. doi: 10.48550/arXiv.2205.14135.

Hang Ding, Yilun Zhao, Tiansheng Hu, Manasi Patwardhan, and Arman Cohan. Scirag: Adaptive, citation-aware, and outline-guided retrieval and synthesis for scientific literature. *arXiv preprint arXiv:2511.14362*, 2025.

Yuwen Du, Tian Jin, Jing Kang, Xianghe Pang, Jingyi Chai, Tingjia Miao, Fenyi Liu, WenHao Wang, Sikai Yao, Yuzhi Zhang, et al. Towards self-evolving agentic literature retrieval. *arXiv preprint arXiv:2605.14306*, 2026. URL https://arxiv.org/abs/2605.14306.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024. URL https://arxiv.org/abs/2404.16130.

Paul Groth, Andrew Gibson, and Jan Velterop. The anatomy of a nanopublication. *Information Services & Use*, 30:51–56, 2010. doi: 10.3233/ISU-2010-0613.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. In *Findings of EMNLP*, pp. 10746–10761, 2025. doi: 10.18653/v1/2025.findings-emnlp.568.

Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, and Weinan E. PaSa: An LLM agent for comprehensive academic paper search. In *Proceedings of ACL*, pp. 11663–11679, 2025. doi: 10.18653/v1/2025.acl-long.572.

Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. SciREX: A challenge dataset for document-level information extraction. In *Proceedings of ACL*, pp. 7506–7516, 2020. doi: 10.18653/v1/2020.acl-main.670.

Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gabor Kismihok, Markus Stocker, and Soren Auer. Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge. *arXiv preprint arXiv:1901.10816*, 2019. URL https://arxiv.org/abs/1901.10816.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of EMNLP-IJCNLP*, pp. 2567–2577, 2019. doi: 10.18653/v1/D19-1259.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*, pp. 6769–6781, 2020. doi: 10.18653/v1/2020.emnlp-main.550.

Chaitanya Kulkarni, Wei Xu, Alan Ritter, and Raghu Machiraju. An annotated corpus for machine reading of instructions in wet lab protocols. In *Proceedings of NAACL-HLT*, pp. 97–106, 2018. doi: 10.18653/v1/N18-2016.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2005.11401.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. S2ORC: The semantic scholar open research corpus. In *Proceedings of ACL*, pp. 4969–4983, 2020. doi: 10.18653/v1/2020.acl-main.447.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of EMNLP*, pp. 3219–3232, 2018. doi: 10.18653/v1/D18-1360.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martino Rios-Garcia, et al. A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists. *Nature Chemistry*, 17:1027–1034, 2025. doi: 10.1038/s41557-025-01815-x.

Michael D. Skarlinski, Sam Cox, Jon M. Laurent, James D. Braza, Michaela Hinks, Michael J. Hammerling, Manvitha Ponnapati, Samuel G. Rodriques, and Andrew D. White. Language agents achieve superhuman synthesis of scientific knowledge. *arXiv preprint arXiv:2409.13740*, 2024. URL https://arxiv.org/abs/2409.13740.

Ningyuan Song, Hanghang Cheng, Huimin Zhou, and Xiaoguang Wang. Linking scholarly contents: The design and construction of an argumentation graph. *Knowledge Organization*, 49(4):213–235, 2022. doi: 10.5771/0943-7444-2022-4-213.

Don R. Swanson. Undiscovered public knowledge. *The Library Quarterly*, 56(2):103–118, 1986. doi: 10.1086/601720.

Simone Teufel and Marc Moens. Summarizing scientific articles: Experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445, 2002. doi: 10.1162/089120102762671936.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. doi: 10.48550/arXiv.1706.03762.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *Proceedings of EMNLP*, pp. 7534–7550, 2020. doi: 10.18653/v1/2020.emnlp-main.609.

David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Iz Beltagy, Lucy Lu Wang, and Hannaneh Hajishirzi. Scifact-open: Towards open-domain scientific claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 4719–4734. Association for Computational Linguistics, 2022a. doi: 10.18653/v1/2022.findings-emnlp.347.

David Wadden, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. Multivers: Improving scientific claim verification with weak supervision and full-document context. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 61–76. Association for Computational Linguistics, 2022b. doi: 10.18653/v1/2022.findings-naacl.6.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. doi: 10.48550/arXiv.2006.04768.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. SciBench: Evaluating college-level scientific problem-solving abilities of large language models. In *Proceedings of ICML*, volume 235 of *Proceedings of Machine Learning Research*, pp. 50622–50649, 2024. URL `https://proceedings.mlr.press/v235/wang24z.html`.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025. doi: 10.48550/arXiv.2506.05176.

A LKM Interfaces and Agent Integration

A.1 LKM Website

The public LKM site (URL withheld for anonymous review) exposes four complementary components. **Knowledge Explorer** searches claims and questions, opens reasoning chains, and shows the source-paper graph. **Evidence Engine** retrieves evidence that supports or challenges a research question and produces a cited synthesis. **Science Landscape** organizes common-question clusters by discipline. **Workflows** presents reproducible workflow families derived from reasoning chains. All four operate on source-grounded objects and preserve the link to the originating paper.

A.2 Integration Modes

Agents can access LKM in three ways. The **OpenAPI** provides direct HTTP access and is appropriate for production services and custom orchestration. A **command-line client** offers shell-level access for research scripts, batch jobs, and reproducible command-line workflows. **Agent skills** expose the same operations as tool calls inside an agent runtime, handling request construction and response decoding. These modes share the same identifiers and semantics; the choice is an integration concern rather than a different knowledge source.

A.3 QUESTION-FAMILY CONSTRUCTION

The clustering input contains 34,413,217 question nodes from arXiv papers, represented in the shared 512-dimensional embedding space. We construct an exact mutual k-nearest-neighbor graph over all nodes with $k = 40$, retaining edges whose cosine similarity is at least 0.78; connected components with at least three nodes form dense cores. Cores whose centroid cosine similarity is at least 0.84 are merged into candidate family centers. Each question node is assigned to at most one center by highest cosine similarity and is accepted when its similarity reaches the center's adaptive threshold, capped at 0.65; otherwise it remains unassigned. Families larger than 2,000 nodes are recursively reclustered, raising the core similarity threshold by 0.03 per level for at most six levels. A final family contains at least four question nodes from at least three papers. An LLM summarizes each family from its member questions and papers, producing the shared research question and its principal variants; the summary remains linked to the member objects.

Restricting to families with at least five nodes yields 988,894 families with 31,910,266 member nodes (92.73% node coverage), drawn from 2,882,306 of 2,896,083 input papers (99.52% paper coverage). Family size has mean 32.27, median 18, 90th percentile 62, 99th percentile 243, and maximum 3,273 nodes; the number of distinct papers per family has mean 20.20, median 10, 90th percentile 42, 99th percentile 166, and maximum 3,245.

A.4 LKM OPENAPI

The public API (base URL withheld for anonymous review) exposes the documented endpoints summarized below.

Table 4: LKM OpenAPI endpoints and key parameters. `/papers/parse*` denotes the family of asynchronous endpoints under this prefix that submit a PDF, poll the parse task, and return the extracted graph.

Interface	Method/path	Purpose	Key parameters
Knowledge search	`POST /search`	Retrieve abstracts, claims, questions, or chains	`query`, `keywords`, `scopes`, `retrieval_mode`, `filters`, `offset/limit`
Reasoning search	`POST /reasoning/search`	Retrieve similar complete reasoning chains	Natural-language query; retrieval mode; paper/date filters; pagination
Paper graph	`POST /papers/graph`	Return the complete graph extracted from a paper	`paper_id`, `package_id`, DOI, or title
Claim reasoning	`GET /claims/{id}/reasoning`	Trace why a known conclusion claim holds	Global conclusion ID (`gcn_...`)
Batch hydration	`POST /variables/batch`	Retrieve details for multiple nodes	List of global node IDs
Feedback	`POST /feedback`	Report a data or service issue	Feedback type, message, optional node/paper ID
PDF extraction	`POST/GET /papers/parse*`	Submit, poll, and retrieve PDF extraction	PDF payload; parse task ID
Citations	`POST /papers/citations`	Retrieve references and citing papers	Paper identifier and query options

For `/search`, valid scopes are `abstract`, `claim`, `premise`, `conclusion`, `question`, `problem`, `open_question`, `subproblem`, and `reasoning_chain`. `hybrid` combines semantic and lexical retrieval; `semantic` favors speed; `lexical` targets exact terms. A successful response has `code=0`; hits are in `data.variables`, paper metadata in `data.papers`, and pagination is indicated by `has_more`. Scores are ranking signals, not scientific confidence.

The complete schemas, generated examples, authentication details, and error codes are maintained in a public API reference, whose link is withheld for anonymous review.

B Additional Results

Table 5 reports citation precision, recall, and F1 on ScholarQABench for all configurations behind Figure 5, together with retrieval baselines evaluated in the same pipeline and published systems for which ScholarQABench citation F1 is available. Without retrieval, answers carry no citations and all three scores are zero.

Table 5: **Citation quality on ScholarQABench** (AttrScore; precision, recall, and F1 in %). The first two groups use the same OpenScholar-style pipeline with GPT-4o as the writer; published systems are listed with their reported F1. OpenScholar values are taken from the preprint that released ScholarQABench (Asai et al., 2024); the system is otherwise cited in its journal version (Asai et al., 2026).

	ScholarQA-CS			**ScholarQA-Multi**		
System	P	R	F1	P	R	F1
arXiv search	33.6	35.4	34.5	35.5	35.7	35.6
Web search	24.5	29.0	26.6	29.9	35.6	32.5
Commercial API	49.6	55.4	52.3	54.3	59.5	56.8
Commercial API + rerank	51.2	56.0	53.5	55.7	60.2	57.9
LKM Search	48.5	50.7	49.6	51.8	53.8	52.8
LKM Search + rerank	52.2	54.6	53.4	55.0	57.7	56.3
LKM Graph + rerank	**57.2**	**60.3**	**58.7**	**59.8**	**63.1**	**61.4**
OpenScholar-8B (Asai et al., 2024)	–	–	47.9	–	–	42.8
OpenScholar-70B (Asai et al., 2024)	–	–	45.9	–	–	54.7
OpenScholar-GPT-4o (Asai et al., 2024)	–	–	39.5	–	–	37.5
PaperQA2 (Skarlinski et al., 2024)	–	–	48.0	–	–	47.2
SciRAG-GPT-4o (Ding et al., 2025)	–	–	34.0	–	–	37.8