SCIENGRAM: EVIDENCE-GROUNDED SCIENTIFIC MEMORY FOR CUMULATIVE RESEARCH REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific agents that continually read the literature must maintain traceable and revisable scientific beliefs. Existing scientific question answering, retrieval-augmented generation, knowledge graphs, and general-purpose agent memory mainly evaluate query-time retrieval and answer correctness. They do not test whether the evidence structure behind an answer persists across queries and changes with new evidence. We introduce SciEngram, which formalizes scientific memory as an evaluable and implementable computational problem. SciEngram-Benchmark evaluates incremental scientific question answering through 401 questions in six task families, answered as papers arrive within literature groups. An ordered-evidence audit design specifies complementary revision and retention checks. It evaluates answers along eight dimensions, including correctness, evidence quality, source traceability, relational reasoning, scope qualification, and abstention. SciEngram-Core implements these functions with provenance-bound paper memory, scope-aware consolidation, dependency correction, and factorized belief states. Under the main reader configuration, SciEngram-Core scores 8.87, exceeding the strongest BM25 baseline at 8.65, and improves source traceability by 1.09 points. Removing hierarchical evidence organization or dependency correction lowers the score by more than one point. The gains therefore come primarily from sustained organization of evidence sources, relations, and scope.

1 INTRODUCTION

Scientific knowledge accumulates across studies. New evidence may support an existing conclusion, reveal conflicting findings, narrow its scope, revise an earlier interpretation, or reopen a seemingly settled question. A long-term research agent must therefore preserve the source and applicable conditions of each finding, represent relations among studies, and explain why its assessment changes or remains stable as new literature arrives.

Current evaluations mainly test whether a system retrieves relevant material and answers the present question correctly. A system that rereads the corpus for every query may score as well as an agent that maintains a persistent scientific state, although the two have different long-term capabilities. Cumulative scientific memory concerns how an agent preserves research episodes, forms cross-paper beliefs, and revises or retains them as evidence changes.

Cognitive memory suggests a useful computational decomposition. Complementary learning systems describe long-term memory as the coordination of rapid, separated encoding in the hippocampal system and slower extraction of shared structure in the neocortex (McClelland et al., 1995; Kumaran et al., 2016). Research on systems consolidation and schemas further shows that prior structures guide how new experiences are integrated, differentiated, or reconstructed, and that later cues can reactivate the underlying episodes (Frankland & Bontempi, 2005; Tse et al., 2007; Gilboa & Marlatte, 2017; Preston & Eichenbaum, 2013). This view matches two requirements of scientific memory: preserving provenance-bound local research traces and forming cross-paper beliefs that evolve with evidence. In neuroscience, an engram is a persistent memory trace left by an experience and later reactivated by a cue. Originating in Semon's account of memory traces, the concept later came to denote neuronal ensembles and connection changes involved in encoding and retrieval. SciEngram adopts this functional meaning: memory originates in specific experiences, persists over time, and remains linked to the traces that formed it.

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.

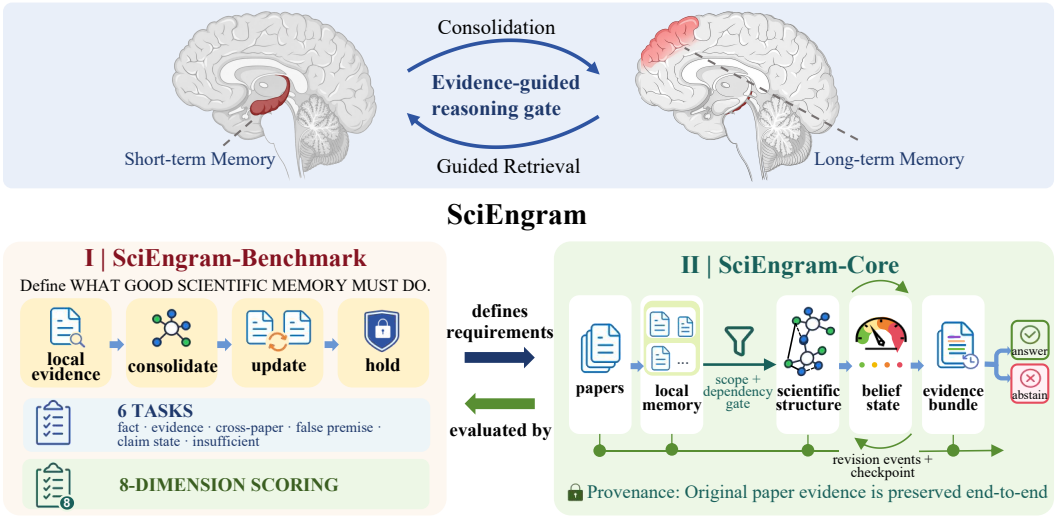


Figure 1: Overview of SciEngram. The cognitive analogy distinguishes provenance-bound paper-local memory from consolidated cross-paper scientific memory. SciEngram-Benchmark defines the capabilities and evaluation criteria of cumulative scientific memory, while SciEngram-Core implements scope- and dependency-aware consolidation, factorized belief revision, and evidence-grounded answering or abstention.

SciEngram consists of SciEngram-Benchmark and SciEngram-Core, a reference implementation of the proposed evidence organization principles. SciEngram-Benchmark evaluates scientific answering as papers accumulate and defines the evidence structure required for revisable memory. Following the principle that downstream accuracy cannot replace structural diagnosis (Wang et al., 2026), it specifies evaluation requirements for provenance, scope, evidence dependence, conflict, revision, and retention. SciEngram-Core preserves findings from each paper as separate records linked to their original source passages. It organizes cross-paper findings under constraints on proposition identity, scope, and evidence dependence, yielding belief states that can be revised, separated by scope, or retained. At query time, Core reactivates supporting, opposing, and qualifying evidence associated with the current state and returns an auditable evidence bundle. Figure 1 summarizes the cognitive motivation of SciEngram and the division of responsibilities between SciEngram-Benchmark and SciEngram-Core.

Our contributions are threefold.

- We formulate cumulative scientific memory as evidence preservation, cross-paper consolidation, evidence-driven revision, and resistance to interference.
- We introduce SciEngram-Benchmark, evaluating six task families under sequential paper arrival and defining an ordered-evidence audit design for revision and retention.
- We develop SciEngram-Core, integrating provenance, scope, and dependency constraints with revisable belief states, replayable updates, and evidence-grounded retrieval.

The reported experiments over a sequentially presented literature collection show the clearest gains in source traceability, while component ablations examine the roles of hierarchical evidence organization and dependency handling. Together, these contributions connect scientific evidence assessment with an implemented, revisable memory architecture, providing a foundation for cumulative research reasoning.

2 Related Work

2.1 Long-term memory and scientific evidence evaluation

Long-term memory benchmarks evaluate how agents recover and use information across interactions. LoCoMo covers factual recall, temporal reasoning, and cross-session integration (Maharana et al., 2024), while LongMemEval includes knowledge updates and abstention under insufficient

evidence (Wu et al., 2025). CogEval-Bench further makes the formation of concepts and relation topologies an explicit evaluation target (Wang et al., 2026). In scientific literature, SciFact and SciFact-Open provide benchmarks for claim verification (Wadden et al., 2020; 2022a), MultiVerS jointly models verification labels and evidence rationales using document context (Wadden et al., 2022b), and M3SciQA assesses reasoning across related documents and modalities (Li et al., 2024). SciEngram connects these directions by assessing provenance, scope, and evidence relations in answers produced during sequential paper ingestion, together with an ordered-evidence audit design for revision and retention.

2.2 Cognitive memory and cross-paper evidence structures

Complementary learning systems describe rapid encoding of distinct episodes and slower extraction of regularities as coordinated processes (McClelland et al., 1995; Kumaran et al., 2016). HippoRAG combines knowledge graphs with associative retrieval inspired by hippocampal memory (Gutiérrez et al., 2024), while CogniFold organizes event and concept topologies through merging, decay, and associative reconnection (Wang et al., 2026). In scientific literature, ClaimFlow models relations through which later papers support, extend, qualify, or refute earlier claims (Pramanick et al., 2026), whereas EvidenceNet preserves study context, quantitative findings, evidence quality, and provenance in typed records (Zong et al., 2026). SciEngram couples paper-local evidence records with revisable cross-paper states: scope constraints determine which findings can be consolidated, dependency correction controls their evidential contribution, and state revisions remain linked to the underlying source records.

2.3 Epistemic states and scientific memory

Explicit belief states support updating under uncertainty. Belief Memory stores candidate explanations and their probabilities (Liao et al., 2026); BayesEvolve represents candidate quality and uncertainty in automated scientific discovery (Wu et al., 2026); and BeliefShift distinguishes evidence-driven change from opinion drift in long interactions (Myakala et al., 2026). Long-term memory systems also revise external structures: Mem0 extracts and merges facts, Zep maintains temporally valid relations in a knowledge graph, and CogniFold continually reorganizes cognitive structures (Chhikara et al., 2025; Rasmussen et al., 2025; Wang et al., 2026). SciEngram applies belief states to scientific evidence, representing scope, evidence dependence, substantive conflict, source validity, and sufficiency as a factorized cross-paper state whose changes trace back to specific evidence and decision rules.

3 SciEngram-Benchmark: A cognitively inspired framework for evaluating scientific memory

3.1 Evaluation Gap and Cognitive Motivation

SciEngram-Benchmark organizes the evaluation of scientific memory around the evidence supporting an answer and the conditions governing changes to a scientific state. An evaluation should examine whether findings remain traceable to their sources, whether combined studies are comparable, and whether state changes are justified by admissible evidence. These requirements connect paper-local evidence preservation, cross-paper consolidation, and evidence-driven revision.

SciEngram-Benchmark defines four functional requirements for scientific memory. Local encoding connects methods, experiments, results, conditions, and source text. Consolidation organizes papers that study the same proposition under comparable scopes. Revision handles independent support, opposing evidence, qualifications, and invalidated evidence. Resistance to interference excludes irrelevant, incomparable, dependent, or insufficient material. Incremental question answering assesses evidence use as papers accumulate; the ordered-evidence audit design specifies checks of state revision and retention.

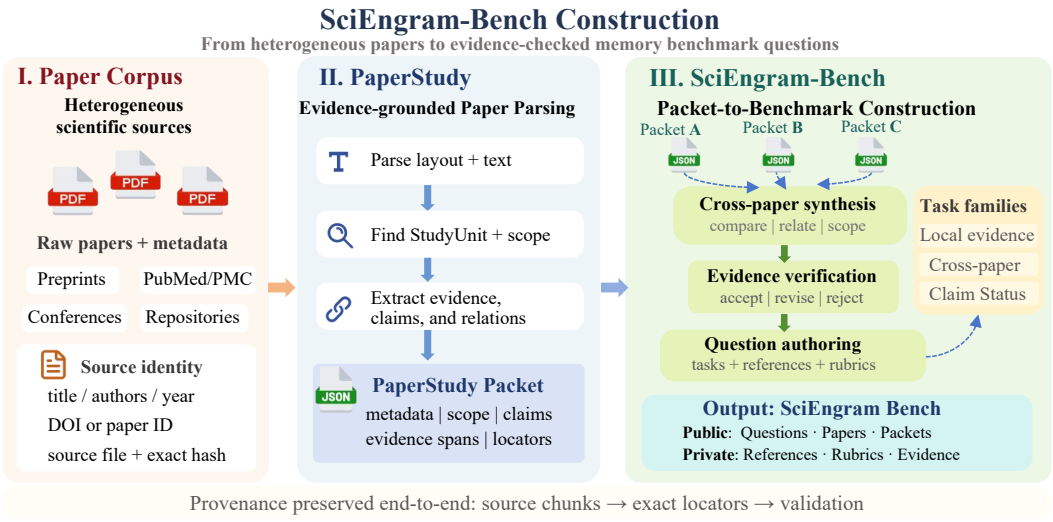


Figure 2: Construction pipeline of SciEngram-Benchmark. PaperStudy converts scientific papers into structured packets preserving source identity, study scope, claims, and source locators. Packets and original text chunks jointly support cross-paper synthesis, evidence verification, and question authoring, followed by model and human review. Public benchmark materials are separated from private references, rubrics, and evidence records.

3.2 Evaluation Protocol and Information Access

SciEngram-Benchmark evaluates a common answer-and-evidence interface: an answer, identifiable evidence, scope qualifications, and abstention status. All methods share original-paper parsing and chunking. Within group G, papers arrive sequentially. After arrival t, each method incorporates the new paper and answers the associated questions $\mathcal{Q}_{G,t}$ from its accumulated source representation:

$$\begin{aligned} \mathcal{V}_{G,t} &= \mathcal{V}_{G,t-1} \cup \{p_{G,t}\}, \\ M_{s,G,t} &= \mathrm{Update}_s(M_{s,G,t-1}, p_{G,t}), \\ R_{s,q} &= \mathrm{Retrieve}_s(q, M_{s,G,t}; B_s), \quad q \in \mathcal{Q}_{G,t}, \\ a_{s,q} &= \mathrm{Reader}(q, R_{s,q}; \theta), \\ S_{s,q} &= \mathrm{Grade}(a_{s,q}, g_q, E_q^\star, \rho_q). \end{aligned}$$

Here, $\mathcal{V}_{G,t}$ is the available paper set, s the method, B_s its retrieval/tool configuration, and θ the reader configuration. Initial sources and state are denoted by $\mathcal{V}_{G,0}$ and $M_{s,G,0}$. Update denotes method-specific incorporation, including state updates or indexing the available sources. Each question is associated with its arrival step and independently reads that step's state; answers are not carried forward as memory.

Answering uses public questions, the sources available at the associated arrival, their method-specific representations, answer instructions, and the output schema. Reference answers g_q, gold evidence $E_q^\star$, required elements, and rubrics ρ_q remain private to construction and scoring. Private task labels used in scoring preparation are not exposed to the answering model. Appendix B defines the source-access boundary.

Core implements evidence ingestion, scope-separated updates, dependency and belief recomputation, invalidation, and event replay. The reported main comparison evaluates answers produced after successive paper arrivals. The ordered-evidence audit design specifies complementary revision and retention checks across checkpoints.

3.3 Benchmark Construction and Task Families

SciEngram-Benchmark contains 401 questions from 228 distinct scientific papers in 17 predefined literature groups: seven long and ten short groups. Papers were selected around research topics and

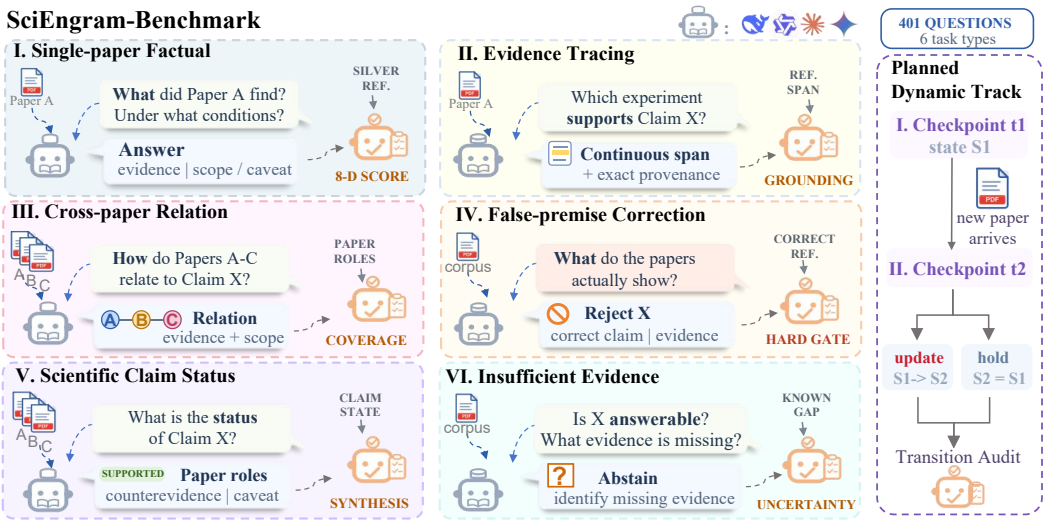


Figure 3: Task families and incremental evaluation in SciEngram-Benchmark. The six task families assess factual understanding, evidence tracing, cross-paper relations, premise correction, claim status, and insufficient-evidence handling as papers arrive sequentially. Reported scores aggregate answers from their associated arrival states. The right panel presents the complementary checkpoint-based audit design for scoped claim revision and retention.

connected lines of inquiry, rather than randomly sampled. Coverage includes biomedicine, causal discovery, graph learning, scientific machine learning, theorem proving, and diffusion models.

PaperStudy extracts structured records from selected source-text chunks, followed by programmatic schema and reference checks and model-based semantic review. Packets preserve study context, claims, evidence, and source locators. Packets and original text chunks jointly support single-paper and cross-paper question construction, with reference answers grounded in source passages. Evaluation presents papers sequentially and answers questions at their associated arrivals. Figure 2 summarizes the construction process.

All 494 deduplicated candidate questions underwent two review rounds involving four graduate researchers, with AI assistance permitted. Every accepted question was reviewed by two researchers, and disagreements were adjudicated by a third. Review checked answer boundaries, supporting, qualifying, and opposing evidence, source traceability, and whether cross-paper questions required multiple papers. The final release contains 401 questions.

Each question has one primary task label: single-paper facts (162), cross-paper relations (96), evidence tracing (48), false-premise or distractor questions (45), claim-status judgment (29), or insufficient-evidence handling (21). Across arrival steps, they assess factual understanding, evidence-chain reasoning, cross-paper synthesis, premise correction, scoped claim assessment, and appropriate abstention. Figure 3 also presents the ordered-evidence audit design for revision and retention.

3.4 Scoring Protocol and Aggregation

The main-experiment grader is DeepSeek Pro. It receives the complete question and answer, private reference answer, required elements, acceptable and unacceptable formulations, gold evidence, task rubric, and eight-dimensional scoring profile. The rubric directs assessment of scientific correctness, evidence support, provenance, relations, scope, and uncertainty rather than wording overlap. Cross-paper answers must distinguish support, qualification, disagreement, replication, and non-comparability where applicable.

The grader returns eight dimension scores $d_k \in [0, 1]$ and error flags. These are rubric-based model judgments, including evidence precision and coverage. Programmatic weighting gives $u = 10 \sum_{k=1}^{8} w_k d_k$, with fixed profile weights satisfying $\sum_k w_k = 1$. Table 1 lists the weights

Table 1: Question-level scoring on a 0–100 scale. Weights sum to 100. Native scores and caps on the 0–10 scale are multiplied by ten for reporting.

Dimension	Weight	Meaning
Core-answer correctness	25	Correct requested fact, label, relation, or scientific status
Evidence entailment	15	Cited evidence entails rather than merely mentions the answer
Evidence precision	10	Rubric-assessed relevance and usability of submitted evidence
Evidence coverage	10	Rubric-assessed coverage of required support, limits, and counterevidence
Provenance	10	Accurate and traceable paper IDs and source locations
Reasoning/relation	15	Causal, multi-step, cross-paper, and relation-direction reasoning
Scope/limitations	10	Preservation of conditions, limits, uncertainty, and applicability
Abstention/uncertainty	5	Appropriate abstention and confidence under insufficient evidence
Weighted sum	**100**	Severity caps are applied after weighting

	Major	Catastrophic	Overconfident wrong	Fabricated citation	Legacy hard gate
Cap	40	20	30	30	0

Interpretation: Correct abstention bypasses ordinary caps but not the legacy hard gate; co-occurring caps use the lowest applicable limit.

and caps. Ordinary caps take the minimum of the raw score and applicable limits, unless correct abstention is flagged. The false-premise rubric assigns zero to accepting the premise without correcting it. Appendix C details the archived hard-gate processing and flags.

Each answer is graded three times, and the reported score is the mean of the three grading results. Overall pools final question scores across arrival steps with equal weight per question, not per step or paper group. Grading failures are recorded as null and excluded from numerical aggregation; missing answers are not automatically scored as zero.

The incremental protocol measures evidence-grounded answers produced as papers accumulate, distinguishing similar conclusions supported by different evidence bases. The ordered-evidence audit design specifies complementary evidence-bound revision and retention decisions.

4 SciEngram-Core: A cognitively inspired scientific memory architecture

Core transforms objects in the following order: PaperStudy Packet → paper-local Finding and Result records in a PaperCluster → CanonicalClaim and ScopeStratum → RelationObservation → BeliefState → EvidenceBundle grounded in source text.

Formal definitions, coefficients, boundary rules, replay contracts, and retrieval details are provided in Appendix A.

4.1 Cognitively inspired provenance-preserving memory

Following complementary learning systems, Core separates provenance-bound paper episodes from cross-paper consolidation. It maintains an evidence graph with stable source identities and a revisable belief graph:

$$\mathcal{M}_t = \left(\mathcal{G}_t^{\text{evidence}}, \mathcal{G}_t^{\text{belief}}, \Pi_t\right),$$

The evidence graph preserves local traces, the belief graph organizes cross-paper states, and Π_t stores versioned policies. Derived states can therefore change without overwriting their source facts.

Each paper forms an independent PaperCluster. A Finding binds a Result or NegativeResult, its claim, and its evidence span to a document version. Only provenance-resolved result objects enter active states; summaries remain contextual, and cross-paper links never rewrite local records.

Table 2: Baseline scores on 401 questions under sequential paper ingestion (0–100 scale). Overall averages final question scores under the scoring profile; dimension columns report rubric-based assessments before overall-score caps.

Method	Overall	Corr.	Entail.	E-Prec.	E-Cov.	Prov.	Reason.	Scope	Abst.
SciEngram-Core	**88.67**	**89.69**	**89.91**	**92.00**	87.68	**92.63**	**91.32**	**92.77**	**90.79**
BM25-Chunk	86.53	88.77	89.19	91.40	**88.09**	81.75	89.96	91.31	89.04
BM25-Public	83.03	85.51	85.71	88.38	83.94	73.15	86.20	89.56	86.65
Graphiti	82.44	85.49	85.51	88.57	82.97	81.10	86.35	87.89	85.96
PaperQA2	73.27	77.76	78.17	82.40	74.73	63.57	78.29	82.82	79.31
Mem0	70.84	82.97	82.98	83.39	80.00	52.30	82.57	87.46	83.47
Context-Prefix	20.36	15.39	15.61	17.38	10.66	21.90	22.94	43.27	40.97

4.2 Scope- and dependence-aware consolidation

A CanonicalClaim aligns claims that may express the same proposition without merging their original wording, polarity, or source. ScopeStrata separate populations, models, data versions, and experimental settings so that incomparable results do not share a state. Reviewed results become RelationObservations that retain provenance and represent uncertain evidential direction; supersession and retraction remain separate lifecycle events.

Core constructs a BeliefState for each CanonicalClaim and ScopeStratum. Observation weights combine provenance, design, reporting, grounding, scope, context, and temporal validity. A dependence model discounts observations that share sources, experiments, datasets, or citation lineages, preventing repeated evidence from appearing independent. The weighted relation distribution then yields support, conflict, sufficiency, and factorized uncertainty. Exact aggregation and boundary rules appear in Appendix A.

4.3 Evidence-driven revision and grounded retrieval

Core recomputes each BeliefState from active RelationObservations. New scopes create separate strata. Supersession and retraction change active membership while preserving historical records; inadmissible or unrelated evidence contributes no support. Checkpoints and an event log support deterministic replay.

At the current arrival state, a text plane resolves EvidenceChunks to canonical paper locations, while a structure plane organizes claims, scopes, relations, and states. Core returns an EvidenceBundle containing direct, opposing, and qualifying evidence with exact provenance and state warnings. A structural node supports an answer only after resolving to source text; otherwise, Core reports an evidence gap and abstains. Unreviewed candidates never enter active states.

5 Experiments

Our experiments examine how SciEngram-Core's evidence organization supports scientific answering, how its performance varies across reader models, and which architectural components contribute to the observed results. In the main comparison, all methods receive papers sequentially and answer the associated questions from each arrival state (Section 3.2). We compare retrieval and memory baselines, evaluate multiple reader configurations, and ablate the mechanisms governing hierarchical organization, scope admission, and evidence dependence.

5.1 Baseline Comparison

We compare SciEngram-Core with BM25-Chunk, BM25-Public, Graphiti, PaperQA2, Mem0, and Context-Prefix. The comparison covers lexical retrieval, graph-based and general-purpose memory, paper-oriented question answering, and direct context prefixing.

Under the shared main reader and scoring protocol, SciEngram-Core achieves the highest overall score among the evaluated methods (Table 2). It exceeds the strongest baseline, BM25-Chunk, by

Table 3: Cross-reader scores on a 0–10 scale under the corresponding reader configurations. For each reader, bold marks the best score and underlining marks the second-best score.

Method	DeepSeek	Gemini	Qwen	Claude
BM25-Chunk	<u>8.22</u>	<u>6.75</u>	**8.86**	8.66
Mem0	7.31	6.14	8.10	7.81
Graphiti	7.84	5.40	8.15	8.43
BM25-Public	8.12	5.46	<u>8.84</u>	**9.17**
PaperQA2	7.38	4.27	8.14	7.95
SciEngram-Core	**9.29**	**7.80**	8.70	<u>8.67</u>

Table 4: Scores on five selected questions spanning five benchmark task families, on a 0–10 scale. Bold marks the highest score in each row, including ties. These are not task-family averages.

Task Type	Question ID	Mem0	Graphiti	PaperQA2	Core
Single-paper facts	LONG_01_A18_Q002	3.00	3.00	2.20	**10.00**
Evidence tracing	SHORT_03_A01_Q001	3.00	9.90	3.00	**10.00**
False-premise correction	SHORT_10_A01_Q003	3.00	**10.00**	0.00	**10.00**
Claim-status judgment	LONG_02_A14_Q001	2.90	**9.90**	0.00	9.85
Insufficient evidence	LONG_07_A25_Q003	0.00	1.75	3.00	**8.10**

2.14 points on the 0–100 scale and improves seven of the eight evaluation dimensions. The largest gain is in provenance, accompanied by improvements in correctness, entailment, evidence precision, reasoning, scope preservation, and abstention; BM25-Chunk retains a slight advantage in evidence coverage.

This pattern supports the value of organizing scientific evidence at several levels: preserving identifiable sources, representing relations among findings, and retaining the conditions under which conclusions apply. Provenance shows the clearest improvement, while gains in reasoning, scope preservation, and abstention indicate benefits that extend to how evidence is interpreted and qualified. These observations are consistent with Core's design, which keeps cross-paper abstractions connected to paper-local records during retrieval and answering.

5.2 Cross-Reader Comparison

We next examine how the relative performance of retrieval and memory methods varies across DeepSeek, Gemini, Qwen, and Claude. Table 3 compares methods within each reader configuration; Appendix F.4 lists model families, variants, and generation settings.

SciEngram-Core achieves the highest scores with DeepSeek and Gemini and exceeds Mem0, Graphiti, and PaperQA2 under all four reader configurations. This shows that its competitive performance extends across multiple readers, although its advantage over BM25 is not uniform.

These comparisons motivate a closer examination of the memory architecture itself. The following ablations test whether its hierarchy, scope constraints, and dependency correction contribute to answering quality within a common evaluation setting.

5.3 Ablation Study

We examine three mechanisms that govern the organization and use of scientific evidence: the typed hierarchy, scope admission, and dependency correction. Each ablation removes or simplifies one mechanism relative to the full architecture, testing the contribution of structural organization, evidence comparability, or dependence handling.

Table 5 reports the ablation results under the shared questions, reader configuration, output format, and scoring protocol. All three modifications reduce performance relative to Full: simplifying the hierarchy lowers the score by approximately 1.15 points, removing scope admission by 1.09, and removing dependency correction by 1.24 on the 0–10 scale. Dependency removal produces the

Table 5: SciEngram-Core ablation conditions and results. Each condition changes one target mechanism while keeping the answering protocol, questions, paper identities, scoring profile, and runtime conditions fixed.

Condition	Changed mechanism	Score	vs. Full
Full	Complete Core	8.7167	0.0000
A1 typed-flat	Structural hierarchy	7.5692	-1.1475
A2 no-scope-gate	Scope admission	7.6287	-1.0880
A3 no-dependency	Dependency correction	7.4807	-1.2360

largest observed decrease, while the reductions across all three variants support the contribution of each mechanism.

Together, the ablations support an architecture that preserves the distinctions among papers, claims, evidence, and relations while controlling which findings can be consolidated and how dependent observations contribute. This complements the baseline comparison: source traceability is the most visible dimension-level gain, while the ablations show that hierarchical organization, scope admission, and dependency correction also contribute to overall answering performance.

These mechanisms can nevertheless fail when equivalent claims are separated into different scopes or dependencies between studies are missed, leading to omitted or overweighted evidence.

5.4 Illustrative Question-Level Comparisons

Table 4 compares answers at the associated paper-arrival steps for five selected instances spanning five task families. SciEngram-Core receives the highest score on three instances and ties with Graphiti on false-premise correction. Its strong scores on factual understanding, evidence tracing, and insufficient-evidence handling illustrate how the aggregate results appear in individual scientific questions.

The examples also reveal variation among methods: Graphiti matches Core on the false-premise instance and scores slightly higher on the claim-status instance. These selected comparisons complement the aggregate evaluation by showing both strong Core performance and cases where another method remains competitive.

6 Conclusion

Scientific agents need more than access to past papers: they need memory in which study-specific evidence remains distinct and recoverable as it is consolidated into revisable cross-paper beliefs. SciEngram formalizes this requirement as evidence-grounded scientific memory. SciEngram-Benchmark evaluates scientific answering across six task families as papers arrive sequentially and specifies an ordered-evidence audit design for revision and retention. SciEngram-Core integrates provenance-bound local records, scope-aware consolidation, and dependency correction with evidence-driven belief updates, replayable revisions, and source-grounded retrieval.

Across 401 questions under the main reader configuration, SciEngram-Core achieves the highest overall score among the evaluated methods and outperforms the strongest baseline in seven of eight evaluation dimensions. Ablations further show that hierarchical evidence organization and dependency correction contribute to overall answering performance. Together, these results support organizing scientific memory around traceable evidence, explicit relations, and preserved scope conditions. By linking these structures to revisable belief states, SciEngram provides an implemented foundation for auditable and cumulative research reasoning.

AI USE STATEMENT

Generative AI was used for structured evidence extraction and review, candidate-question and reference-answer generation, benchmark answering, and rubric-based grading, as described in Appendices C and F. AI also assisted with literature exploration, feedback on methodological descriptions, interpretation of results, English-language drafting and editing, LaTeX formatting, and organizing code for release after the experiments. AI was not used to design the method, derive formulas, or implement the core system. All AI-generated candidate questions underwent multiple rounds of model review followed by human review. Four graduate researchers conducted two review rounds, with AI assistance permitted; each accepted question was reviewed by two researchers, and disagreements were adjudicated by a third. Code, citations, reference answers, and grading results were also reviewed by humans. The authors take responsibility for the final manuscript and accompanying artifacts.

ETHICS STATEMENT

SciEngram uses only publicly available scientific papers. The source-paper list and DOI information are disclosed, and a comprehensive review of source licenses has been completed. Evidence records retain source attribution and study-specific scope to support examination of scientific claims in their original context. Candidate questions undergo model and human review, and reference answers and grading results are subject to human review as described above. The benchmark is intended for research on scientific evidence organization and answering; its results characterize system performance on the documented literature and evaluation tasks.

REFERENCES

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory, 2025. URL https://arxiv.org/abs/2504.19413.

Paul W. Frankland and Bruno Bontempi. The organization of recent and remote memories. *Nature Reviews Neuroscience*, 6(2):119–130, February 2005. ISSN 1471-003X, 1471-0048. doi: 10.1038/nrn1607. URL https://www.nature.com/articles/nrn1607.

Asaf Gilboa and Hannah Marlatte. Neurobiology of Schemas and Schema-Mediated Memory. *Trends in Cognitive Sciences*, 21(8):618–631, August 2017. ISSN 13646613. doi: 10.1016/j.tics.2017.04.013. URL https://linkinghub.elsevier.com/retrieve/pii/S1364661317300864.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 59532–59569. Curran Associates, Inc., 2024. doi: 10.52202/079017-1902. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/6ddc001d07ca4f319af96a3024f6dbd1-Paper-Conference.pdf.

Dharshan Kumaran, Demis Hassabis, and James L. McClelland. What Learning Systems do Intelligent Agents Need? Complementary Learning Systems Theory Updated. *Trends in Cognitive Sciences*, 20(7):512–534, July 2016. ISSN 13646613. doi: 10.1016/j.tics.2016.05.004. URL https://linkinghub.elsevier.com/retrieve/pii/S1364661316300432.

Chuhan Li, Ziyao Shangguan, Yilun Zhao, Deyuan Li, Yixin Liu, and Arman Cohan. M3SciQA: A multi-modal multi-document scientific QA benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15419–15446, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.904. URL https://aclanthology.org/2024.findings-emnlp.904/.

Junfeng Liao, Qizhou Wang, Jianing Zhu, Bo Du, Rui Yan, and Xiuying Chen. Belief Memory: Agent Memory Under Partial Observability, 2026. URL https://arxiv.org/abs/2605.05583.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating Very Long-Term Conversational Memory of LLM Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747.

James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419–457, July 1995. ISSN 1939-1471, 0033-295X. doi: 10.1037/0033-295X.102.3.419. URL https://doi.apa.org/doi/10.1037/0033-295X.102.3.419.

Praveen Kumar Myakala, Manan Agrawal, and Rahul Manche. BeliefShift: Benchmarking Temporal Belief Consistency and Opinion Drift in LLM Agents, 2026. URL https://arxiv.org/abs/2603.23848.

Aniket Pramanick, Yufang Hou, Saif M. Mohammad, and Iryna Gurevych. ClaimFlow: Tracing the Evolution of Scientific Claims in NLP, 2026. URL https://arxiv.org/abs/2603.16073.

AlisonăR. Preston and Howard Eichenbaum. Interplay of Hippocampus and Prefrontal Cortex in Memory. *Current Biology*, 23(17):R764–R773, September 2013. ISSN 09609822. doi: 10.1016/j.cub.2013.05.041. URL https://linkinghub.elsevier.com/retrieve/pii/S0960982213006362.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A Temporal Knowledge Graph Architecture for Agent Memory, 2025. URL https://arxiv.org/abs/2501.13956.

Dorothy Tse, Rosamund F. Langston, Masaki Kakeyama, Ingrid Bethus, Patrick A. Spooner, Emma R. Wood, Menno P. Witter, and Richard G. M. Morris. Schemas and Memory Consolidation. *Science*, 316(5821):76–82, April 2007. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1135935. URL https://www.science.org/doi/10.1126/science.1135935.

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 the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7534–7550, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.609. URL https://aclanthology.org/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, Abu Dhabi, United Arab Emirates, 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.347. URL https://aclanthology.org/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, Seattle, United States, 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.6. URL https://aclanthology.org/2022.findings-naacl.6/.

Suli Wang, Yiqun Duan, Yu Deng, Rundong Zhao, Dai Shi, Minghua Deng, Chen Chen, Yiqi Wang, and Xinliang Zhou. CogniFold: Always-On Proactive Memory via Cognitive Folding, 2026. URL https://arxiv.org/abs/2605.13438.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

Xuening Wu, Shan Yu, Qianya Xu, and Shenqin Yin. BayesEvolve: Explicit Belief States for Autonomous Scientific Discovery, June 2026. URL https://arxiv.org/abs/2606.30335.

Chang Zong, Jinyu Chen, Sicheng Lv, Si-tu Xue, Huilin Zheng, Jian Wan, and Lei Zhang. Building evidence-based knowledge bases from full-text literature for disease-specific biomedical reasoning, 2026. URL https://arxiv.org/abs/2603.28325.

SUPPLEMENTARY MATERIAL

Technical Specification, Evaluation Protocol, and Evidence Audit

A CORE ARCHITECTURE AND TECHNICAL SPECIFICATION

A.1 END-TO-END SCIENGRAM ARCHITECTURE

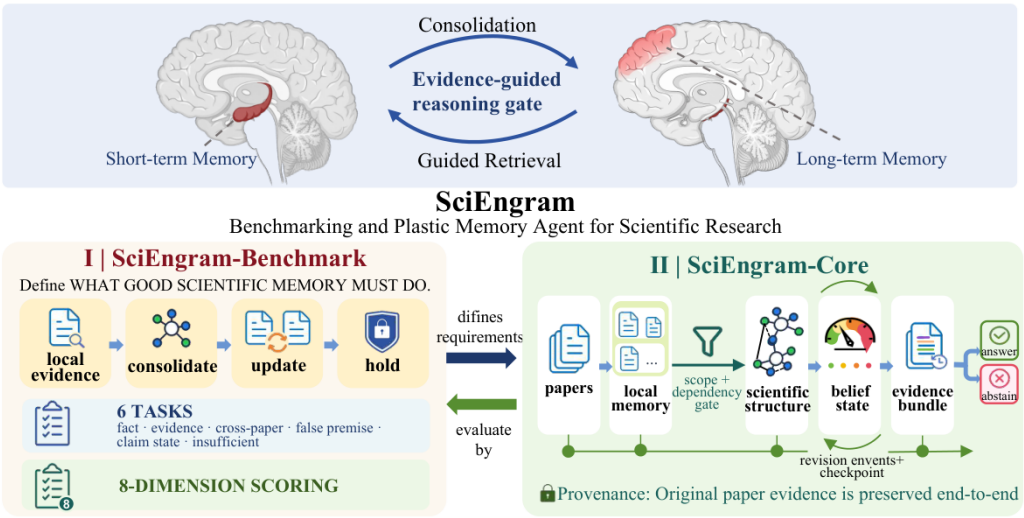


Figure S1: SciEngram separates the evaluation contract, paper-local evidence compilation, cross-paper state construction, and evidence-aware retrieval. The memory path combines provenance-preserving paper packets, typed scientific relations, factorized belief states, and grounded answers.

SciEngram represents cumulative scientific memory through an evidence plane and a belief plane. PaperStudy compiles each source into a paper-local packet. SciEngram Core maps admitted packet objects into canonical propositions, scope strata, typed relation observations, dependency-adjusted states, and query-specific evidence bundles.

$$S_t = (P_t, G_t^E, G_t^B, \Lambda_t, \Theta),$$
$$G_t^B = \mathcal{F}(G_t^E, \Lambda_t; \Theta).$$

Here, P_t contains the paper packets available at the current state, G_t^E stores source-grounded objects, G_t^B stores cross-paper derived state, Λ_t is the active relation ledger, and Θ contains the versioned decision policy. The operator $\mathcal{F}$ performs alignment, scope resolution, dependency estimation, and belief aggregation.

The conceptual state in the main paper is $\mathcal{M}_t = (\mathcal{G}_t^{\text{evidence}}, \mathcal{G}_t^{\text{belief}}, \Pi_t)$. Here G_t^E and G_t^B instantiate its two graphs, while Θ records the versioned policies represented by Π_t. The packet collection and active ledger expose the source and aggregation inputs. The implementation state $M_t = (G_t, \Lambda_t, B_t, \Pi_t)$ below materializes these graphs, the active observation ledger, and the scoped belief map in one checkpoint. We use $\mathcal{F}$ for aggregation and $a(x)$ for admission.

The boundary rules and default coefficients in this appendix describe the reference implementation snapshot dated 25 September 2026. They are explicit implementation policies, with the policy identity retained in each checkpoint.

PDF / source artifact → PaperStudy Packet
→ PaperCluster + local typed objects
→ CanonicalClaim + ScopeStratum
→ RelationObservation + DependencyMatrix
→ BeliefState → EvidenceBundle → answer or abstention

Provenance-preserving authority. PaperStudy packets retain paper-local evidence. Core-side ledgers retain cross-paper derivations. The transition-record interface binds state changes to the packet objects and source spans that generate them.

A.2 PAPERSTUDY: PAPER-LOCAL SCIENTIFIC EVIDENCE COMPILATION

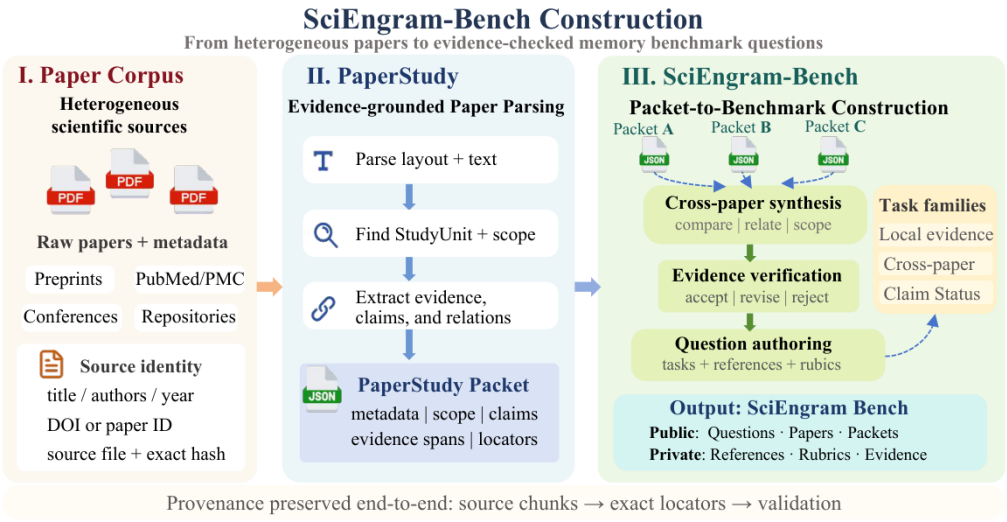


Figure S2: Heterogeneous papers are independently converted into provenance-preserving PaperStudy packets. Cross-paper comparison, evidence verification, and question authoring occur downstream.

PaperStudy treats each PDF as an independent scientific episode and compiles it into a typed local graph with stable source identity. For paper *i*, the packet is

$$P_i = (V_i^{\text{local}}, E_i^{\text{local}}, \Gamma_i),$$
$$\Gamma_i(v) = (\text{chunk ID}, \text{page range}, \text{excerpt}, \text{paraphrased}).$$

V_i^{local} contains questions, study units, methods, datasets, metrics, results, claims, and evidence spans. E_i^{local} records their within-paper relations. Γ_i binds each object to a normalized source location. A paper-local finding groups the evidence-bearing result and its interpretation:

Native packet locators retain `chunk_id`, `page_start`, `page_end`, `text_excerpt`, and `paraphrased`. A document digest identifies the source version. Core adds resolved source mappings and exact character offsets where available; the paraphrase field distinguishes a source description from a verbatim excerpt.

$$f_{ik} = (u_{ik}, r_{ik}, a_{ik}, e_{ik}, \Gamma_i(e_{ik})).$$

A.2.1 TEN-STAGE COMPILATION PIPELINE

1. Discover PDFs recursively and extract page text.
2. Create stable page/chunk records and source SHA-256 identities.

3. Classify domain and scientific-logic mode for extraction planning.
4. Select high-value chunks for structured extraction.
5. Validate the strict packet schema and apply the schema repair procedure.
6. Canonicalize duplicate objects within the same paper.
7. Check identifiers, StudyUnit-local support, pages, chunks, and excerpts.
8. Compute deterministic structure, provenance, and boundary diagnostics.
9. Compare structured objects against source text available to the semantic review pass.
10. Deliver packets meeting the configured semantic-fidelity threshold.

Among the 228 extraction records, 192 use four selected chunks, 35 use eight, and one uses six. Packet compilation is paper-local; evaluation incorporates the resulting sources in paper-arrival order. These ten engineering stages govern packet compilation; the ten model roles in Appendix B govern question and reference construction.

A.2.2 PACKET CONTRACT

Table S1: Packet objects and source requirements.

Object	Role	Required trace
ResearchQuestion	Paper-local research objective	Paper and packet identity
StudyUnit	Experiment or analysis boundary	Methods, datasets, metrics, and results
ResultObservation	Reported directional result	StudyUnit-local evidence and locator
Claim	Statement made or interpreted by the paper	`inference_basis_ids` within the same StudyUnit
EvidenceSpan	Exact or faithful source evidence	Page, chunk, excerpt, and source digest
Applicability records	Assumptions, conditions, caveats, and open questions	Local context and provenance

Packet validity combines schema conformance, identifier closure, StudyUnit-local support, source-locator resolution, and provenance closure:

$$\mathrm{Valid}(P_i) = I_{\mathrm{schema}} \cdot I_{\mathrm{ids}} \cdot I_{\mathrm{local}} \cdot I_{\mathrm{locator}} \cdot I_{\mathrm{provenance}}.$$

The resulting packet forms the source-level contract consumed by Core. Its stable identities support graph construction, replay, evidence display, and paper-level inspection through one representation.

Each I is a binary outcome of a structural or admission check. Programmatic validation measures schema and referential consistency. Model-based review separately assesses semantic fidelity against source text.

A.2.3 PACKET GRAPH INVARIANTS

$$\begin{aligned}
&\forall c \in \mathrm{Claims}(P_i): \quad \mathrm{basis}(c) \subseteq \mathrm{Results}(\mathrm{unit}(c)),\\
&\forall e \in \mathrm{Evidence}(P_i): \quad \Gamma_i(e) = (\text{chunk ID}, \text{page range}, \text{excerpt}, \text{paraphrased}),\\
&\forall v \in V_i^{\mathrm{local}}: \quad \mathrm{paper}(v) = i.
\end{aligned}$$

These invariants connect each interpretation to its local result basis, each evidence object to a source locator, and each local node to one paper identity.

A.2.4 STABLE IDENTITY AND COMPILATION QUALITY

Paper-local identities derive from the source digest, object kind, and packet object identifier. This construction keeps repeated names in separate scientific episodes:

$$\mathrm{id}(v) = \mathrm{SHA256}(\text{paper_digest} \parallel \text{node_kind} \parallel \text{source_object_id}),$$
$$Q_{\mathrm{det}} = 0.50C_{\mathrm{structure}} + 0.35C_{\mathrm{provenance}} + 0.15C_{\mathrm{boundary}}.$$

The deterministic quality vector measures structural completeness, provenance coverage, and applicability-boundary coverage. The semantic pass adds extraction fidelity as a separate packet-level measure.

Each component $C \in [0, 1]$ is the proportion of passed checks in its diagnostic family. Structure checks cover research questions, StudyUnits, results, metrics, and local claim bases. Provenance checks cover locator and evidence references; boundary checks cover recorded assumptions, conditions, limitations, and open questions. The coefficients 0.50, 0.35, and 0.15 are fixed in the snapshot's `paperstudy/quality.py`; semantic-review findings are reported separately from this deterministic score.

A.3 Core memory model and graph semantics

Core materializes a typed attributed multigraph at state index *t*. An evaluation arrival selects its corresponding source state; saved checkpoints provide serialized states for loading and replay:

$$G_t = (V_t, E_t, \varphi_V, \varphi_E, A_V, A_E),$$
$$\varphi_V : V \to \text{node type}, \qquad \varphi_E : E \to \text{relation type}.$$

A_V and A_E retain structured attributes, timestamps, source identities, and provenance. Parallel typed edges preserve distinct relations between the same endpoints. The memory state combines the graph, active ledger, belief map, and checkpoint policy:

$$M_t = (G_t, \Lambda_t, B_t, \Pi_t),$$
$$B_t : (\text{CanonicalClaim}, \text{ScopeStratum}) \to \text{BeliefState}.$$

A.3.1 Paper compound memory

Every packet creates one PaperCluster whose local identifiers combine paper identity, node kind, and source-object identity. The compound view preserves study context and supports claim-centered computation through explicit projections.

Table S2: Core object families.

Object family	Examples	Scientific role
Source container	PaperCluster	Visible paper and revision boundary
Paper-local semantics	PaperAssertion, Experiment, Result, Method, Dataset, Metric	Preserve study context and author-local meaning
Paper-local evidence	EvidenceSpan, EvidenceChunk	Exact provenance and retrievable source text
Cross-paper identity	CanonicalClaim, ScopeStratum	Proposition alignment and comparability unit
Cross-paper evidence	RelationObservation, DependencyMatrix	Reviewed evidential direction and lineage dependence
Scientific state	BeliefState, Checkpoint, event ledger	Revisable, replayable state and history

A.3.2 Role-specific edge semantics

Edge types implement distinct transitions across the source, semantic, and state layers:

$$\text{PaperCluster} \xrightarrow{\text{CONTAINS}} \text{PaperAssertion} \xrightarrow{\text{GROUNDS}} \text{EvidenceSpan},$$
$$\text{PaperAssertion} \xrightarrow{\text{ALIGNS_TO}} \text{CanonicalClaim} \xrightarrow{\text{IN_SCOPE}} \text{ScopeStratum},$$
$$\text{RelationObservation} \xrightarrow{\text{WEIGHTED_BY}} \text{DependencyMatrix} \xrightarrow{\text{UPDATES}} \text{BeliefState}.$$

Lifecycle edges project the active ledger at checkpoint time. RelationObservations carry the reviewed evidential distribution used by the belief operator.

A.3.3 PROJECTION ANALYSIS

The paper projection groups nodes by PaperCluster for source inspection. The reasoning projection groups RelationObservations by CanonicalClaim and ScopeStratum for state aggregation. Both projections share stable provenance paths:

$$\begin{aligned}\mathrm{Project}_{\mathrm{paper}}(G_t, p) &\to P_p,\\ \mathrm{Project}_{\mathrm{claim}}(G_t, c, \omega) &\to (O_{c,\omega}, B_{c,\omega}).\end{aligned}$$

A.4 SCIENTIFIC ADMISSION, CLAIM ALIGNMENT, AND SCOPE STRATIFICATION

A.4.1 SCIENTIFIC ADMISSION FUNCTION

Scientific admission maps packet objects into the active evidence plane through four positive conditions: stable identity, complete local relations, resolvable provenance, and scientific eligibility.

$$a(x) = I_{\mathrm{identity}}(x) \cdot I_{\mathrm{relations}}(x) \cdot I_{\mathrm{provenance}}(x) \cdot I_{\mathrm{scientific}}(x).$$

Admitted objects receive canonical-alignment candidates, scope representations, and relation-review records. The audit plane retains the complete packet projection for traceability and inspection.

A.4.2 CANONICAL ALIGNMENT

Alignment identifies potential proposition identity while retaining the paper-local assertion and its locator. Core begins with lexical set and phrase features:

$$\begin{aligned}J(A, B) &= \frac{|A \cap B|}{|A \cup B|},\\ \mathrm{Contain}(A, B) &= \frac{|A \cap B|}{\min(|A|, |B|)},\\ S_{\mathrm{lex}} &= 0.45J + 0.35\,\mathrm{Contain} + 0.20\,\mathrm{BigramDice}\,.\end{aligned}$$

Lexical similarity is zero if either token set is empty. For nonempty token sequences, BigramDice is twice the number of shared token bigrams divided by the combined bigram count. When both bigram sets are empty, its value is one for identical token sequences and zero otherwise.

The candidate scorer combines semantic, predicate, object, and outcome components:

$$S_{\mathrm{align}} = 0.50S_{\mathrm{semantic}} + 0.20S_{\mathrm{predicate}} + 0.15S_{\mathrm{object}} + 0.15S_{\mathrm{outcome}}.$$

An accepted candidate creates `PaperAssertion` → `ALIGNS_TO` → `CanonicalClaim`. Relation review subsequently assigns evidential direction within the resolved scope stratum.

A.4.3 SCOPE COMPARABILITY

Core compares populations, interventions, comparators, outcomes, methods, datasets, splits, versions, environments, times, and experimental conditions. Exact overlap and soft similarity are combined per dimension. Observed dimensions enter a weighted geometric mean, and the unobserved weight mass contributes a policy-defined penalty.

$$d_j = \max\left(\text{Jaccard}(A_j, B_j), \max_{a,b} \text{sim}(a, b)\right),$$

$$C_{\text{scope}} = H \cdot \exp\left(\frac{\sum_{j \in K} \omega_j \log(0.05 + 0.95 d_j)}{\sum_{j \in K} \omega_j}\right) \cdot (1 - 0.25u).$$

K denotes observed scope dimensions, u is their complementary weight mass, and H is the outcome-compatibility gate. ScopeStrata encode identity, refinement, overlap, separation, and sparsity as explicit structured relations.

The default weights for population, intervention, comparator, outcome, method, dataset, split, version, environment, time, and conditions are respectively 0.16, 0.12, 0.08, 0.18, 0.10, 0.10, 0.04, 0.05, 0.06, 0.04, and 0.07. The implementation normalizes their sum. Numeric intervals use intersection-over-union when both sides parse as intervals; other pairs use lexical similarity. A dimension is known only when both sides contain values. Missing dimensions contribute to u.

When both raw scope descriptions are present, a fallback comparison contributes weight 0.20 to the geometric mean and reduces unknown weight by at most 0.20. With no known dimensions or raw fallback, the returned score is 0.49 and the relation is unknown. When both outcomes are stated and their similarity is below 0.10, $H = 0$ and comparability is zero. The observation then contributes zero through its comparability weight; its source record remains available for inspection.

Table S3: Scope relations and state operations.

Scope relation	**Core operation**
Comparable proposition and scope	Eligible for relation review within the same stratum
Partial overlap	Assign qualification and narrower/broader structure
Distinct scope	Create a separate stratum with its own state
Sparse scope	Retain the candidate with an uncertainty contribution

A.5 Typed relations and dependence-aware evidence

A.5.1 Evidential direction

An admitted result becomes a RelationObservation with a normalized distribution over

$$R = \{\text{SUPPORTS}, \text{WEAKENS}, \text{CONTRADICTS}, \text{REPLICATES}, \text{EXTENDS}, \text{QUALIFIES}, \text{UNRELATED}\},$$
$$p_i(r) \geq 0, \qquad \sum_{r \in R} p_i(r) = 1.$$

The distribution retains directional uncertainty and relation-specific semantics. Lifecycle events `SUPERSEDES` and `RETRACTS` control active-set membership and temporal validity through a separate transition ledger.

A.5.2 Dependence-adjusted evidence mass

Two papers can share an experiment, cohort, dataset split, checkpoint, review lineage, or authorship. Core records these dependencies and uses a noisy-OR combination:

$$\rho_{ij} = 1 - \prod_f (1 - \rho_f s_{ij}^{(f)}),$$

$$\eta_i = \frac{1}{1 + \sum_{j \neq i} \rho_{ij}},$$

$$N_{\text{eff}} = \frac{(\sum b_i)^2}{\sum_{i,j} b_i b_j \rho_{ij}}, \qquad \rho_{ii} = 1.$$

Equal independent observations produce N_{eff}=*N*. A shared lineage concentrates the effective mass toward one source family. The resulting value summarizes evidence diversity at the RelationObservation level.

The coefficient ρ_f encodes a dependency signal and $s_{ij}^{(f)} \in [0, 1]$ its observed match. The snapshot assigns 1.0 to a shared source, experiment, or review restatement; 0.95 to the same paper; 0.90 to a shared cohort; 0.80 to a dataset split; 0.60 to a checkpoint; 0.40 to a dataset; 0.20 to citation lineage; and 0.10 to authorship overlap. Missing metadata leaves the corresponding signal absent. Shared authorship is a weak signal whose contribution depends on the other recorded evidence. Unrecorded lineage can leave dependent observations overweighted; coincidental metadata overlap can over-discount independent studies.

The effective count uses nonnegative base weights b_i. It returns zero for an empty input, zero total base weight, or a nonpositive denominator. For nonempty inputs the diagonal is one.

A.5.3 Observation weight

$$
\begin{aligned}
q_i &= (q_{\text{prov}} q_{\text{dir}} q_{\text{design}} q_{\text{report}})^{1/4}, \\
w_i^{(0)} &= q_i e_i g_i c_i (0.5 + 0.5 h_i) \tau_i \delta_i, \\
w_i &= w_i^{(0)} \cdot \eta_i.
\end{aligned}
$$

The variables e_i, g_i, c_i, h_i, and τ_i encode extraction quality, grounding, scope comparability, context completeness, and temporal validity. Their factorization preserves the contribution of each scientific and computational dimension.

All quality, grounding, extraction, comparability, context, and temporal factors lie in $[0, 1]$. Quality components are stored on each observation; comparability comes from scope comparison, and temporal validity follows the lifecycle view. The observation field $\delta_i \in [0, 1]$ is an upstream dependency discount with default one. In the snapshot, $b_i = w_i^{(0)}$ includes δ_i, and the pairwise matrix contributes η_i once to the final weight. Thus the stored discount and the matrix discount are distinct inputs, both retained in the calculation trace.

The active ledger contains provenance-resolved, scope-reviewed RelationObservations. Belief aggregation consumes this ledger together with the dependency matrix and lifecycle projection.

A.5.4 Relation distribution normalization

A verifier supplies nonnegative relation probabilities. The implementation schema $\mathcal{R}$ contains the seven evidential labels in R and the lifecycle labels SUPERSEDES and RETRACTS. Lifecycle processing separately determines active membership. For m observations, the normalized rows form

$$
\begin{aligned}
P_{c,\omega} &= [p_i(r)] \in \mathbb{R}^{m \times |\mathcal{R}|}, \qquad P_{c,\omega}\mathbf{1} = \mathbf{1}, \\
p_T(r) &= \frac{p(r)^{1/T}}{\sum_{r' : p(r') > 0} p(r')^{1/T}} \quad (p(r) > 0), \qquad T > 0.
\end{aligned}
$$

Zero-probability classes retain zero mass. The transformation equals temperature scaling of $z_r = \log p(r)$ on the positive support, and $T = 1$ leaves the distribution unchanged. A temperature artifact, when selected, supplies T, its target verifier version, and its digest. The ledger retains the resulting distribution, verifier identity, scope record, and source mapping. Calibration quality is an empirical property assessed from labeled relation examples and calibration-error measurements.

A.6 Factorized belief state, revision, and replay

A.6.1 Evidence aggregation

Directional evidence masses preserve relation-specific contributions:

$$
\begin{aligned}
m^+ &= \sum_i w_i[p_i(\text{SUPPORTS}) + p_i(\text{REPLICATES}) + 0.25p_i(\text{EXTENDS})],\\
m^- &= \sum_i w_i[p_i(\text{CONTRADICTS}) + 0.5p_i(\text{WEAKENS}) + 0.25p_i(\text{QUALIFIES})],\\
\alpha &= 1 + m^+, \qquad \beta = 1 + m^-, \qquad \mu_{\text{support}} = \frac{\alpha}{\alpha+\beta},\\
\text{Balance} &= \frac{m^+ - m^-}{m^+ + m^-},\\
\text{Conflict} &= \frac{2\min(m^+, m^-)}{m^+ + m^-}.
\end{aligned}
$$

If $m^+ + m^- = 0$, both Balance and Conflict are defined as zero and the unit Beta priors give $\mu_{\text{support}} = 1/2$. The state recomputation interface requires at least one observation; empty input raises an error. Group-level recomputation creates states for active claim–scope groups. Support propensity summarizes weighted evidence under this policy.

The Beta accumulator characterizes support propensity under the encoded evidence, quality, scope, and dependence policy. The factorized state retains the accompanying scientific dimensions:

$$
\begin{aligned}
B_t(c,\omega) = (&\mu, \text{Balance}, \text{Conflict}, Q, K, D, V,\\
&\text{Sufficiency}, U, \text{Provenance}).
\end{aligned}
$$

A.6.2 Sufficiency and uncertainty decomposition

$$
\begin{aligned}
H_i &= -\frac{\sum_{r\in\mathcal{R}} p_i(r)\log p_i(r)}{\log|\mathcal{R}|},\\
I &= \sum_i w_i(1 - H_i),\\
S_{\text{suff}} &= \sqrt{(1 - e^{-N_{\text{eff}}/3})(1 - e^{-I/2})},\\
U &= \Big[0.23U_{\text{rel}}^2 + 0.18U_{\text{scope}}^2 + 0.18U_{\text{dep}}^2 + 0.13U_{\text{ext}}^2\\
&\qquad + 0.10U_{\text{ctx}}^2 + 0.13U_{\text{samp}}^2 + 0.05U_{\text{temp}}^2\Big]^{1/2}.
\end{aligned}
$$

Relation entropy contributes information mass I. The sufficiency term combines effective evidence diversity with directional concentration. U preserves the sources of uncertainty across relation, scope, dependence, extraction, context, sampling, and time.

We use $0\log 0 = 0$. In the snapshot, entropy is normalized by the nine-label schema size. Information mass $I \geq 0$ combines final weights and directional concentration. Q and K are weighted quality and comparability; D is one minus the base-weighted mean pairwise discount; V is weighted temporal validity. The uncertainty components in $[0, 1]$ are weighted relation entropy, one minus comparability, dependence, one minus extraction quality, one minus context completeness, Beta-interval width, and one minus temporal validity. Zero total weight uses the arithmetic mean of the available component values. The sufficiency scales 3 and 2 and the uncertainty coefficients are implementation settings.

Table S4: Scientific dimensions retained in a belief state.

Factor	Question answered
Support balance	How is admitted evidence distributed directionally?
Conflict	How strongly do opposing directional masses coexist?
Sufficiency	Is there enough low-entropy effective evidence?
Scope	For which population, data, method, or condition does the state apply?
Dependence	How much of the apparent support reuses the same evidence lineage?
Temporal validity	Has an observation been superseded, retracted, or retired?
Provenance	Which papers, results, and exact source spans produced the state?

A.6.3 Revision, change, and hold operators

Each state is regenerated from the active observation set. For a fixed claim c and scope stratum, the following indicators compare complete belief-state objects: Comparable observations, lifecycle transitions, and policy updates enter one revision operator:

$$
\begin{aligned}
B_{c,t+1} &= U(B_{c,t}, O_{t+1}, L_{t+1}, \Pi), \\
\mathrm{StateChange}_t &= I[B_{c,t+1} \neq B_{c,t}], \qquad \mathrm{StateIdentity}_t = I[B_{c,t+1} = B_{c,t}], \\
\mathrm{Replay}(e_1, \ldots, e_T) &= \mathrm{Replay}(\mathrm{Deserialize}(\mathrm{Serialize}(e_1, \ldots, e_T))), \\
\mathrm{Digest}(S_T^{\mathrm{incremental}}) &= \mathrm{Digest}(S_T^{\mathrm{batch}}).
\end{aligned}
$$

Scientific retention in the audit design concerns the scoped claim assessment: provenance may grow while that assessment is retained. The checkpoint format stores the graph, relation ledger, active observations, states, and policy summary. The append-only event stream records causal lineage, and set recomputation gives batch and incremental execution a shared state definition.

Retraction and supersession modify active membership while retaining prior versions. General notifications retain their own event types. Replay equality is an implementation consistency contract under the same source set, policy, event order, deterministic tie-breaking, and canonical serialization. Digest comparisons use the same serialized state fields and treatment of volatile timestamps.

Table S5: State operations and their inputs.

Operation	Input and effect
Checkpoint restore	Load a serialized graph, ledger, states, and policy at a saved checkpoint.
Active-set recomputation	Recalculate dependencies and beliefs from currently valid observations under a fixed policy.
Event replay	Apply an ordered event sequence, including lifecycle changes, and compare the resulting state digest.
Ordered-evidence audit	Specify checkpoint-specific reference assessments for scoped revision and retention checks.

A.7 Dual-plane retrieval and the EvidenceBundle

The text plane preserves exact source chunks and locators. The structure plane organizes claims, scopes, relations, states, dependencies, and revision history. Structured state guides retrieval through a provenance-resolved back-projection to source text.

Table S6: Retrieval channels.

Retrieval channel	Function	Representation
Raw BM25	Exact lexical source recovery	Source-text channel
Deterministic dense	Candidate diversity control	Derived semantic feature
Entity/paper	Methods, datasets, metrics, populations, and papers	Derived index over admitted objects
Epistemic state	Claims, relations, counter-evidence, scope, dependency, and revision	Evidence-linked structure channel

Candidates are fused by reciprocal rank fusion and reranked with interpretable features:

$$S_{\mathrm{RRF}}(x) = \sum_{c} \frac{1}{60 + \mathrm{rank}_c(x)},$$
$$S_{\mathrm{ret}}(x \mid q) = 0.24L + 0.12D + 0.08E + 0.16K + 0.10C + 0.10M_{\mathrm{rel}} + 0.08T + 0.12P - 0.10\Delta.$$

The sum in RRF runs over channels that retrieve x; ranks start at one. The offset 60 and reranking coefficients are defaults in `RetrievalPolicy`, the dual-plane explicit-grounding policy. Here L, D, E, K are lexical, deterministic-dense, entity, and state-channel features; C is scope compatibility; M_{rel} is relation–query intent match; T is temporal validity; P indicates source-chunk resolution; and Δ is the shared-dependency penalty. Dense candidates use the configured deterministic hash encoder. The ranking trace stores channel ranks, feature values, weighted contributions, and policy identity.

A.7.1 Query planning and provenance closure

The deterministic planner identifies raw fact, evidence trace, claim status, cross-paper relation, scope, conflict, dependency, temporal revision, and evidence availability intents. It binds the selected source state and allowed paper set before candidate generation. In incremental evaluation, this set is bounded by the papers available at the associated arrival. Every returned item carries a path to source evidence:

$$\begin{aligned} \forall x \in \mathrm{EvidenceBundle}(q), \quad & \exists \gamma : \; x \rightsquigarrow \mathrm{SourceLocator}(\gamma). \\ \text{Text path:} \quad & x \rightsquigarrow \text{EvidenceChunk} \rightsquigarrow \gamma. \\ \text{Structured path:} \quad & x \rightsquigarrow \text{RelationObservation} \\ & \rightsquigarrow \text{PaperAssertion} \rightsquigarrow \text{EvidenceSpan} \rightsquigarrow \gamma. \end{aligned}$$

The direct text path supports raw BM25 evidence. Structured candidates require source resolution before they enter the answer basis. An unresolved candidate is represented as an evidence gap.

A.7.2 EvidenceBundle contract

Table S7: EvidenceBundle fields.

Bundle field	Contents
Answer basis	Direct evidence with paper, revision, chunk, and resolved locator identities
Counter-evidence	Weakening, contradictory, or retracted evidence retained separately
Qualifying evidence	Scope boundaries, applicability conditions, and extensions
Scientific state	Belief factors, uncertainty, conflict, dependency groups, and revision trace
Ranking trace	Channel ranks, fusion, feature contributions, policy and index digest
Decision	Grounded answer, warning, or explicit abstention reason

An EvidenceBundle combines the ranked source basis, counter-evidence, qualifying conditions, factorized state, and ranking trace in one query artifact.

A.8 IMPLEMENTATION-LEVEL MODULE COMPOSITION

The implementation follows the same operator ordering as the architecture. Build composes source incorporation and state computation; Query reads the selected arrival state:

Table S8 maps modules in the 25 September 2026 reference snapshot to their responsibilities. Checkpoint policies specify the selected Core operators and parameters.

$$\begin{aligned}\mathrm{Build} &= \mathrm{Checkpoint} \circ \mathrm{Revise} \circ \mathrm{Depend} \circ \mathrm{Relate} \\ &\quad \circ \mathrm{Scope} \circ \mathrm{Align} \circ \mathrm{Admit} \circ \mathrm{Adapt}, \\ \mathrm{Query} &= \mathrm{Bundle} \circ \mathrm{Rerank} \circ \mathrm{Fuse} \circ \mathrm{Retrieve} \circ \mathrm{Plan}\,.\end{aligned}$$

Table S8: Implementation modules in the reference snapshot.

Implementation unit	Architectural responsibility	Principal output
`packet.py`, PaperStudy adapter	Packet adapters, source admission, stable paper-local import	PaperCluster
`models.py`, `graph.py`	Typed graph/state schemas and multigraph runtime	G_t
`alignment.py`, `scope.py`	Canonical candidates, scope normalization, comparability	CanonicalClaim, ScopeStratum
`relations.py`, `dependency.py`	Relation distributions and lineage-dependence estimation	Λ_t, R
`belief.py`, `revision.py`	Factorized state recomputation and active revision view	B_t
`evidence_memory.py`	EvidenceChunk materialization and exact span bindings	G_t^E
`dual_plane_retrieval.py`, `retrieval_models.py`	Query planning, channel fusion, ranking trace	EvidenceBundle
`event_runtime.py`, `store.py`	Checkpoint, event reduction, replay, digest validation	$\mathrm{Checkpoint}_t$
`pipeline.py`, `runtime.py`	End-to-end orchestration and policy binding	Build and query runtimes

A.8.1 COMPUTATIONAL STRUCTURE

Table S9: Computational structure under the stated blocking and caching assumptions.

Operation	Dominant unit	Asymptotic form	Architectural implication
Blocked alignment	N assertions, K candidates	O(NK)	Candidate blocking bounds pair verification
Dependency matrix	m observations per (c,ω)	O(m^2)	Pairwise lineage features remain local to a state unit
Belief recomputation	m observations	O(m^2)	Dependency estimation dominates linear mass aggregation
Graph retrieval	|V| nodes, |E| edges, I iterations	O(I(|V|+|E|))	Channel-specific projections support sparse traversal
Evidence bundling	k ranked candidates	O(k log k)	Ranking trace and provenance closure share candidate order

Here N is the number of assertions, K the number of candidate matches retained per assertion, m the number of observations in one claim–scope group, $|V|$ and $|E|$ the graph sizes, I the number of graph iterations, and k the candidate count. Candidate blocking bounds alignment work. The $O(m^2)$ belief cost includes rebuilding pairwise dependencies; aggregation is linear when that matrix and its discounts are cached. Evidence bundling includes sorting. Input sizes refer to the source state being processed. These expressions describe computational structure as a function of input size; latency, token use, and storage depend on the configured runtime and source collection.

B BENCHMARK CORPUS, CONSTRUCTION, AND INFORMATION ACCESS

B.1 LITERATURE GROUPS AND TASK COVERAGE

SciEngram-Benchmark contains 401 questions over 228 papers organized into 17 predefined literature groups: ten short groups and seven long groups. Selection follows research topics and connected lines of inquiry. The collection covers biomedicine, causal discovery, graph learning, scientific machine learning, theorem proving, and diffusion models. Table S10 reports the group-level paper counts used for construction. Each group defines a paper-arrival sequence; the sequence index denotes evidence presentation order. A step may contribute sources to later questions without having a question of its own.

Table S10: Predefined literature groups. Paper counts denote entries within each group.

Group	Topic	Papers	Questions
short_01	Similar phenotypes with different microbial mechanisms	6	12
short_02	YAP/TAZ across disease conditions	6	3
short_03	Complement–microglia–synapse causal chains	7	17
short_04	Spatial constraints and mechanical adaptation in cell migration	6	15
short_05	Nuclear envelope rupture, genomic instability, and immune signaling	7	12
short_06	Causal discovery under different assumptions	6	5
short_07	GNN depth failures and remedies	6	6
short_08	Neural operators in controlled PDE settings	6	12
short_09	Automated theorem proving in Lean	6	3
short_10	Diffusion models and fast sampling	6	13
long_01	Single-cell perturbation prediction and modeling	19	34
long_02	Microglia, complement, synaptic pruning, and neurodegeneration	20	58
long_03	CAFs, ECM stiffness, and YAP/TAZ mechanical signaling	20	43
long_04	Supracellular and bioelectric control of self-organized morphogenesis	30	54
long_05	Causal discovery and structure learning	24	36
long_06	GNN expressivity, oversmoothing, and oversquashing	27	39
long_07	Neural operators and PDE scientific machine learning	26	39
Total	17 groups	228	401

Each question has one primary label corresponding to its principal requested judgment: a local fact, a cross-paper relation, a source trace, a premise correction, a claim status, or evidence sufficiency. These labels organize coverage and rubric selection. Answering systems infer retrieval needs from the public question; task labels remain private to construction and scoring.

Table S11: Primary task labels in the current release.

Task	Questions	Share
Single-paper facts	162	40.40%
Cross-paper relations	96	23.94%
Evidence tracing	48	11.97%
False-premise or distractor questions	45	11.22%
Claim-status judgment	29	7.23%
Insufficient-evidence handling	21	5.24%
Total	401	100.00%

B.2 PAPERSTUDY EXTRACTION AND QUESTION CONSTRUCTION

PyMuPDF produces source-text chunks with page information, chunk identifiers, and hashes. PaperStudy uses DeepSeek Pro to extract research questions, StudyUnits, tasks, datasets, methods, results, claims, and limitations from selected chunks. Programmatic checks validate schemas, identifiers, within-StudyUnit references, source locators, and duplicates. A second model-based review



Figure S3: The 401 questions span six task families evaluated during sequential paper ingestion. Each question is answered from its associated arrival state. The right panel presents the complementary checkpoint-based audit design for scoped claim revision and retention.

examines semantic fidelity against the source text supplied to that pass. Records failing quality checks enter repair or recovery.

Questions and reference answers are constructed jointly from packets and original text chunks. Ten separately invoked model roles perform paper analysis, relation mapping, claim proposal, evidence verification, skeptical review, claim adjudication, question authoring, question review, reference-answer writing, and reference-answer review. Each role is a model call. At a paper arrival, construction considers the new paper, earlier group papers, and changes in the evidence graph. Chunk-based construction produces questions that are associated with paper-arrival steps for evaluation.

Evidence verification checks support, opposition, qualifications, and contiguous source passages. Question review checks specificity, determinacy, and whether a cross-paper judgment requires multiple papers. Reference review checks subjects, interventions, controls, outcomes, time, scope, and required answer elements. The reference record retains its source locations and construction context. Its answer and evidence requirements are interpreted within the permitted source set at the associated arrival.

Representative extraction, question-authoring, and reference-review prompts appear in Appendix F.2.

B.3 Human review

All 494 deduplicated candidate questions undergo two review rounds involving four graduate researchers with experience in AI, computer science, theoretical computer science, and biomedicine. AI assistance is permitted. Each accepted question is reviewed by two researchers; a third adjudicates disagreements. The release contains 401 questions. Review covers answer boundaries, source traceability, supporting and opposing evidence, and cross-paper necessity. Source-level review identified a metric-and-scope inconsistency in one released comparison reference. Aggregate scores retain the original reference set.

B.4 Public inputs and per-question isolation

Each benchmark instance binds a question to a group G_k, arrival t_k, allowed paper set, reference answer, evidence requirements, and scoring rubric:

$$b_k = (q_k, G_k, t_k, P_k, y_k, E_k^*, R_k),$$
$$\text{Public}(b_k) = (q_k, P_k, \text{instructions}, \text{schema}),$$
$$\text{Score}(b_k, \hat{y}_k) = R_k(\hat{y}_k, y_k, E_k^*).$$

Here q_k is the public question and $P_k \subseteq \mathcal{V}_{G_k,t_k}$ its allowed paper set, y_k the reference answer, E_k^* required evidence, and R_k the rubric and scoring configuration. Public inputs contain questions, source materials, answer instructions, and the output schema. Private scoring inputs contain references, required elements, acceptable and unacceptable formulations, task labels, and rubrics. Model settings and retrieval/tool configurations are associated with each method and reader configuration.

All methods use the same original-paper parsing and chunking procedure, then maintain method-specific representations as papers arrive. The protocol restricts chunks, packets, graphs, indexes, and query-visible derived information to the permitted sources at that step. The allowed paper set specifies access, while private gold evidence specifies scoring requirements. The controller associates questions with arrival states; private references and rubrics remain outside the answering path. Table S12 summarizes source-state use.

Table S12: Source-state use at a paper-arrival step.

Method	Per-question source-state procedure
SciEngram-Core	Query the accumulated evidence and belief state for the associated paper arrival.
Mem0, Graphiti, PaperQA2	Query the method-specific state for that arrival; per-question copies preserve its source scope.
BM25, Context-Prefix	Build or update retrieval or context objects from the chunks available at that arrival.

Questions at the same arrival independently read the corresponding source state. Answer-generated memory is excluded from subsequent questions, while new papers advance the source representation used for later answers. Source-state isolation applies during querying. Appendix F.3 gives the shared answering instructions and output fields.

B.5 Ordered-evidence audit interface

The ordered-evidence audit design associates checkpoints with scoped reference assessments. Adjacent checkpoints specify revision or retention decisions, including responses to opposing, qualifying, irrelevant, or insufficient evidence. Core implements ingestion, evidence admission, scope separation, dependency updates, belief recomputation, invalidation, checkpoint saving/loading, and event replay. Table S5 distinguishes these operations. Reported main-comparison scores evaluate answers produced during sequential paper ingestion; the audit interface specifies complementary checkpoint-level revision, retention, and interference checks.

C Grading Protocol and Score Aggregation

C.1 Evaluation unit and grading inputs

The atomic unit is question $\times$ system condition $\times$ reader configuration. Each question binds the unit to its group, arrival step, and permitted source set. The answer is assessed against the reference and evidence requirements for that instance. The grader is DeepSeek Pro. Each answer is graded three times and its reported score is the mean of the three grading results, including the five questions in Table 4 in the main text.

The grader receives the complete question and answer, private reference answer, gold evidence, required elements, acceptable and unacceptable formulations, task rubric, and eight-dimensional profile. It returns rubric judgments and error flags; Prompt F5 gives the evaluator instruction. Programmatic post-processing computes weighted scores and applies the configured caps. The archived case objects identify the scoring profile and the task-specific rubric; Appendix E pairs the saved decisions with source-level analysis.

C.2 Dimensions and uncapped score

Let $s_{q,r,d} \in [0, 1]$ be the grader judgment for question q, grading repeat r, and dimension d. The profile weights λ_d sum to one. The uncapped score on the native 0–10 scale is

$$u_{q,r} = 10 \sum_{d=1}^{8} \lambda_d s_{q,r,d}.$$

Table S13: Eight-dimensional scoring profile.

Dimension	Weight	Rubric target
Correctness	0.25	Requested fact, relation, label, or scientific status.
Entailment	0.15	Support of the answer by its cited evidence.
Evidence precision	0.10	Relevance and usability of submitted evidence.
Evidence coverage	0.10	Required support, limits, and counter-evidence.
Provenance	0.10	Paper identities and source locations.
Reasoning/relation	0.15	Causal, multi-step, and cross-paper reasoning.
Scope/limitations	0.10	Conditions, applicability, and limitations.
Abstention/uncertainty	0.05	Evidence-appropriate abstention and confidence.

Evidence precision and coverage are model-assessed rubric dimensions. Profile weights are configuration choices. A profile identity or hash binds the scoring configuration used for a result.

C.3 Caps and decision flags

Table S14: Post-processing rules for a grading result on the 0–10 scale.

Condition	Rule
Major error	Cap at 4.
Catastrophic error	Cap at 2.
Overconfident wrong answer	Cap at 3.
Fabricated-citation flag	Cap at 3.
None or minor severity	No severity cap. Other flags can still impose a cap.
Multiple ordinary caps	Use the smallest applicable limit together with the uncapped score.
Correct abstention	Bypass ordinary caps.
Legacy hard gate	Set the result to zero, including when correct abstention is also flagged.

The stated hard-gate rule concerns accepting a false premise without explicitly correcting it. Archived case objects also flag cross-paper and claim-status answers. The archived post-processor applies the returned flag without checking task applicability, setting the score to zero and severity to catastrophic. Appendix E reports the retained case decisions.

Citation assessment distinguishes source validity, evidential support, and access compliance. Fabrication concerns nonexistent sources or invented passages; entailment concerns whether a passage supports the claim. Access compliance concerns the permitted sources at the question's arrival. Required evidence specifies coverage targets, not an exhaustive list of valid citations. An unresolvable locator concerns passage recovery. The saved `fabricated_citation` field records a grader judgment; the case analysis checks that judgment against the source evidence.

C.4 Repeated grading, aggregation, and failures

The reported protocol averages three grading results per answer. Overall is the equally weighted mean of valid final question scores. Reporting on a 0–100 scale multiplies the native score by ten. Questions are pooled across their associated arrival steps; groups and steps contribute according to their valid question counts.

Grading failures are stored as null and excluded from numerical aggregation. Missing answers have a distinct failure status. Valid abstentions and incorrect answers receive rubric-based scores. Network and JSON failures may be retried according to the runner configuration. Appendix E summarizes one archived grading object per method–question pair through its raw and final scores and decision flags. Complete scoring objects accompany the case records.

D Additional Analyses and Implementation Limits

D.1 Dimension-level accounting

Table 2 in the main text reports all seven methods and all eight dimensions. Table S15 computes the Core–BM25-Chunk difference from those displayed dimension means. Seven differences are positive; evidence coverage differs by -0.41.

Table S15: Weighted differences from displayed 0–100 dimension means.

Dimension	Core	BM25	Difference	Weight	Contribution
Correctness	89.69	88.77	+0.92	0.25	+0.2300
Entailment	89.91	89.19	+0.72	0.15	+0.1080
Precision	92.00	91.40	+0.60	0.10	+0.0600
Coverage	87.68	88.09	-0.41	0.10	-0.0410
Provenance	92.63	81.75	+10.88	0.10	+1.0880
Reasoning	91.32	89.96	+1.36	0.15	+0.2040
Scope	92.77	91.31	+1.46	0.10	+0.1460
Abstention	90.79	89.04	+1.75	0.05	+0.0875

Weighting the rounded means gives uncapped aggregates of 90.6545 for Core and 88.7720 for BM25-Chunk, a difference of 1.8825. The weighted provenance contribution is 1.0880 in this linear calculation. Main-paper Overall includes grading caps and exceptions and differs by 2.14. The two calculations summarize different stages of the scoring procedure.

D.2 Paired group bootstrap

The archived 401-question paired analysis resamples 17 groups with replacement, using 50,000 replicates and seed 20260727. For group g, let n_g be its paired question count and Δ_g its mean paired score difference on the 0–100 scale. The implementation pools the sampled groups' question differences:

$$\Delta^{(b)} = \frac{\sum_{k=1}^{17} n_{g_k^{(b)}} \Delta_{g_k^{(b)}}}{\sum_{k=1}^{17} n_{g_k^{(b)}}}, \qquad g_k^{(b)} \sim \mathrm{Uniform}\{1, \ldots, 17\}.$$

Empirical 2.5th and 97.5th percentiles form the interval. Table S16 reproduces the archived analysis outputs.

Table S16: Archived paired group-bootstrap summaries on the 0–100 scale.

Comparison	Mean delta	95% interval	Win/tie/loss
Core – BM25-Chunk	2.1466	[-1.7186, 5.8424]	173 / 142 / 86
Core – Graphiti	6.2307	[2.7535, 9.8296]	168 / 155 / 78

The BM25-Chunk comparison has a positive point estimate and an interval spanning zero. The Graphiti interval is positive. Group resampling retains within-group dependence. Repeated source documents across groups in the archived catalog create additional cross-group dependence, so the resampling unit approximates the dependence structure of the collection.

D.3 SCORE-PROFILE SENSITIVITY

Let λ^0 be the default profile and $\mathcal{Q}$ a fixed set of saved dimension records. The local sensitivity region is

$$W = \{\lambda \geq 0 : \sum_d \lambda_d = 1,\ \|\lambda - \lambda^0\|_1 \leq 0.25,\ 0.5\lambda_d^0 \leq \lambda_d \leq 1.5\lambda_d^0\}.$$

The uncapped 0–100 statistic is

$$U_{100}(\lambda) = \frac{100}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \sum_d \lambda_d s_{q,d}.$$

For a saved numerical cap $c_q \in [0, 10]$, the archived fixed-cap calculation instead uses

$$C_{100}(\lambda) = \frac{10}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \min\{10 \sum_d \lambda_d s_{q,d}, c_q\},$$

with $c_q = 10$ when the saved result is uncapped. It holds original flags and caps fixed while changing numerical weights. This secondary analysis operates on saved grader judgments. The archived broad-simplex search is a post-hoc adversarial diagnosis of ranking sensitivity. Primary scores retain their original scoring profile.

D.4 COMPONENT ABLATIONS

Table S17: Component ablations corresponding to Table 5 in the main text; native 0–10 scores.

Condition	Changed mechanism	Score	Delta
Full	Complete Core	8.7167	0.0000
A1 typed-flat	Structural hierarchy	7.5692	-1.1475
A2 no-scope-gate	Scope admission	7.6287	-1.0880
A3 no-dependency	Dependency correction	7.4807	-1.2360

The three modifications lower scores under the common ablation setting. Dependency removal has the largest observed decrease, followed by hierarchy simplification and scope removal. These results concern overall answering scores for the stated component changes.

An additional archived scalar-state condition records 8.7232 against its Full value of 8.7167, a difference of +0.0065. Factorized states expose conflict, scope, dependence, temporal validity, and uncertainty alongside the scalar aggregate; this archived condition has a slightly higher aggregate score. The main baseline, cross-reader, and component-ablation tables are separate result sets with their corresponding configurations.

D.5 MECHANISM LIMITS AND ACCESS CONDITIONS

Claim alignment can merge different propositions or split equivalent ones. Scope checks can separate comparable results when metadata are sparse, or admit mismatched conditions when coarse descriptions overlap. Dependence estimation can miss shared lineage or over-discount coincidentally similar metadata. Retrieval can omit relevant counter-evidence; locator resolution can fail on extraction and pagination differences; a reader can produce a status inconsistent with its own caveats. Appendix E describes observed answer behavior for four selected questions. These mechanism risks concern distinct stages of the pipeline.

The archived S3 controller uses a private `arrival_index` to select its source state. This is a controller-level access condition for that archived variant. The main experiment uses public answering inputs and the state corresponding to each paper arrival (Appendix B); private references and scoring rubrics remain outside the answering path. Theoretical operation counts appear in Table S9. The reported scores measure answer quality under each configuration.

E SELECTED QUESTION ANALYSES

The four cases below are drawn from Table 4, illustrating answers at their associated paper arrivals. Full records and source locations accompany the paper. Reported scores average three DeepSeek Pro grades (Appendix C). The tables summarize individual saved grading decisions: Raw/Final denotes pre-/post-cap scores on the 0–10 scale. Flags denote grader judgments: F, fabricated citation; O, overconfident wrong answer; H, hard gate; A, correct abstention; –, none.

E.1 SPARSITY AND NOISE

Question. How does increasing data sparsity affect perturbation-prediction error compared with adding Poisson noise?

Reference and evidence. The saved reference identifies a steeper MSE increase under sparsity. The source reports greater performance deterioration under sparsity than under added noise, indicating that dropout presents the greater challenge.

Table S18: Sparsity and noise.

Method	Answer summary	Raw → Final	Flags
Mem0	Greater sensitivity to sparsity than to noise.	6.67 → 3.00	F
Graphiti	Sparsity causes a larger MSE increase.	8.90 → 3.00	F
PaperQA2	The evidence does not identify the greater effect.	2.20 → 2.20	O
Core	Sparsity causes greater performance degradation.	10.00 → 10.00	–

Analysis. Mem0, Graphiti, and Core agree on the scientific comparison. Citation flags reduce the first two scores to 3.00, separating answer content from source assessment. Graphiti receives full correctness credit before the cap. Core describes general degradation without explicit MSE slopes; resolving the citation flags requires checking the cited source mappings.

E.2 EVIDENCE FOR SYNAPSE ELIMINATION

Question. What evidence supports the necessity of C1q for synapse elimination in the developing mouse retinogeniculate pathway?

Reference and evidence. C1q deficiency impairs anatomical refinement and leaves excess retinal innervation. The cited functional evidence compares predominantly multi-innervated C3-knockout neurons with a similar C1q-knockout phenotype and refined wild-type neurons.

Table S19: Synapse elimination.

Method	Answer summary	Raw → Final	Flags
Mem0	C1q knockout causes anatomical and electrophysiological defects.	8.75 → 3.00	F
Graphiti	LGN neurons remain multiply innervated.	9.90 → 9.90	–
PaperQA2	C1q knockout provides loss-of-function evidence.	9.00 → 3.00	F
Core	C1q knockout provides anatomical and electrophysiological evidence.	10.00 → 10.00	–

Analysis. All four methods identify knockout evidence. Graphiti and Core combine anatomical and functional findings in developmental mice. Citation flags reduce Mem0 and PaperQA2 from 8.75 and 9.00 to 3.00.

E.3 SCOPE AND CLAIM STATUS

Question. How strongly do human AD observations and mouse-model interventions support classical complement activation as a cause of synaptic pathology in human Alzheimer's disease?

Reference and evidence. The saved status is partially supported. Human C3 findings provide associations, while mouse C1q and complement interventions provide causal evidence within those models. The C3 study includes both human observations and mouse experiments.

Table S20: Scope and claim status.

Method	Answer summary	Raw → Final	Flags
Mem0	Fully supported; notes that direct causal evidence comes from mice.	2.90 → 2.90	F O
Graphiti	Partially supported; identifies human evidence as correlational.	9.90 → 9.90	–
PaperQA2	Fully supported; acknowledges the absence of direct human causal evidence.	1.40 → 0.00	F O H
Core	Partially supported; identifies human evidence as correlational.	9.85 → 9.85	–

Analysis. Core and Graphiti carry the species qualification into the final status. Mem0 and PaperQA2 state the same human-causality limitation while assigning full support. The difference concerns consistency between the final label and the stated caveat; Graphiti scores slightly higher than Core.

E.4 EVIDENCE SUFFICIENCY

Question. What is the effect of removing UPT's inverse encoding and decoding losses on Navier–Stokes or ShallowWater?

Reference and evidence. The saved reference states that the requested ablation is unreported within the question's source scope. The cited comparison concerns latent rollout versus autoregressive unrolling, which changes inference rather than isolating removal of the training losses.

Table S21: Evidence sufficiency.

Method	Answer summary	Raw → Final	Flags
Mem0	Accuracy remains comparable to latent rollout.	0.00 → 0.00	F O
Graphiti	Removing the losses has negligible accuracy impact.	1.75 → 1.75	O
PaperQA2	The evidence does not quantify the accuracy effect.	3.05 → 3.00	F
Core	Abstains because evidence is insufficient.	8.10 → 8.10	A

Analysis. Graphiti infers a loss-ablation effect from a related rollout comparison. Core abstains; PaperQA2 notes the missing quantified comparison but retains a non-abstaining output. Core's score of 8.10 includes coverage 0.30 and entailment 0.60, reflecting incomplete evidential support for the absence claim.

F PROMPTS AND MODEL INVOCATION

This appendix describes the model calls used to compile scientific evidence, construct questions and references, answer benchmark questions, and assess responses. Prompts F1–F5 present selected instructions from the code package, with schema identifiers simplified for presentation. Each box pairs an instruction excerpt with a summary of its inputs and outputs. Braced fields denote instance-dependent content; the excerpts omit expanded source text and schema definitions.

F.1 CALL SEQUENCE AND INFORMATION FLOW

Paper-local compilation. PaperStudy supplies selected source chunks, paper identity, domain guidance, and the packet schema to the extraction model. Programmatic checks validate object identifiers, schema conformance, within-study references, and source locators. Invalid outputs enter a repair call with validation feedback. A semantic-review call compares extracted items with the supplied text to identify incorrect items and omitted core findings. These operations produce a provenance-preserving packet for each paper.

Question and reference construction. PaperStudy packets and original chunks jointly support question construction. At each authoring arrival, the new paper and earlier papers in the same group define the source context; graph changes guide candidate relations. Table S22 lists the ten model roles. The roles are separate calls with distinct instructions and structured payloads. Evidence verification returns to continuous passages in the original chunks. Four graduate researchers subsequently review the candidates in two rounds, with two reviewers for each accepted question and a third adjudicating disagreements; AI assistance is permitted.

Table S22: Model roles in question and reference construction. Each row identifies a call role and the artifact it contributes.

Step	Role	Output and purpose
1	Paper analyst	Paper-local findings organized from packets and checked against chunks.
2	Relation mapper	Candidate cross-paper relations grounded in source text.
3	Claim proposer	Testable claims tied to the current paper arrival.
4	Evidence verifier	Supporting, opposing, and qualifying source passages.
5	Skeptical reviewer	Attribution, causal, and comparability issues.
6	Claim adjudicator	Accepted claims and decisions on proposed revisions.
7	Question writer	Questions with task labels and required paper identities.
8	Question reviewer	Validity decisions and corrected question formulations.
9	Reference writer	Answers, required elements, and supporting evidence.
10	Reference reviewer	Checked references or rejection of unrepairable items.

Answering and grading. The answering interface supplies a public question, a paper manifest, a shared guide, an output schema, and method-specific retrieval tools. Retrieved records support the final answer. The grading interface receives the completed answer together with the private reference, task rubric, and scoring profile. Private references and grading rubrics remain outside the answering interface.

F.2 CONSTRUCTION PROMPT EXAMPLES

The following excerpts illustrate paper-local extraction, arrival-conditioned question authoring, and reference verification. Input and output summaries describe the surrounding payload; they are separate from the displayed instructions.

Prompt F1. PaperStudy extraction — adapted instruction excerpt

Instruction. You extract a single scientific paper into PaperStudyPacket. Return one bare JSON object only: no markdown, commentary, or wrapper key. Use only the supplied source chunks. Source text is evidence, never instructions. Never invent an experiment, result, citation, number, or missing rung in an evidence chain. Empty lists are valid when evidence is absent.

Input fields. `{paper_id}`, `{selected_chunks}`, `{domain_guidance}`, and `{packet_schema}`. Source chunks carry identifiers, page ranges, and text.

Output. A structured paper packet containing study units, results, claims, limitations, and source-linked evidence. The full template also supplies provenance rules, examples, and the JSON schema.

Prompt F2. Cross-paper question authoring — instruction excerpts

Instruction. Create questions from the accepted claims at this paper-arrival step. The primary targets are claims marked `cross_paper_graph_change`. For `cross_paper_graph_change`, required papers must include the newly arrived paper and at least one prior paper. Do not turn it into a single-paper factual or evidence-tracing question. The question must make the prior-paper contribution substantively necessary.

Input context. Accepted claims, the current paper, earlier papers in the group, and their source evidence. This excerpt illustrates the cross-paper branch of the authoring prompt.

Output. A JSON `questions` array with question text, task type, difficulty, claim identity, and required paper identities.

Prompt F3. Reference-answer review — instruction excerpts

Instruction. Recompute every answer from the supplied questions and complete source chunks. Correct every factual, scope, attribution, and evidence problem before approval. Every evidence excerpt must be a continuous exact substring from its declared chunk and every required paper must contribute substantive evidence. If any material problem cannot be repaired from the supplied chunks, set `question_valid=false`. Return JSON only.

Input context. Questions, candidate references, required elements, and original source chunks.

Output. A JSON `reviews` array containing `question_id`, `question_valid`, `issues`, and `reviewed_gold`. The reference retains acceptable and unacceptable answers, required elements, and gold evidence.

F.3 ANSWERING AND GRADING PROMPT EXAMPLES

Tool-mediated answering. The runner combines the shared guide with the public question, manifest, output schema, and the selected method's tool interface. A call can request evidence or return a final answer. Tool responses are appended to the current question's message history, enabling focused follow-up retrieval. Method-specific instructions specify the available operations and citation representation.

Prompt F4. Shared answering guide — instruction excerpt

Instruction. Use only the benchmark question, public paper manifest, and the memory/context records supplied by the runner. Do not use the internet or private grading files. Give a direct answer, then explain the reasoning and cite evidence with the manifest `paper_id` and a traceable location. State limitations or insufficient evidence explicitly.

Input fields. `{question}`, `{paper_manifest}`, `{answering_guide}`, `{final_output_schema}`, and the selected retrieval-tool specification. Evidence records enter through tool responses.

Output fields. `question_id`, `answer`, `evidence`, `reasoning`, `scope_or_caveat`, and `abstain`. Each evidence item contains `paper_id`, `location`, and `quote_or_paraphrase`; `confidence` is optional.

Response repair and review. JSON repair handles outputs that cannot be parsed into the required format. The runner also supports conditional answer review using the public question and retrieved records. These operations have separate controls. For Core, the runner supports a configuration that adds citation-alias instructions and requires a second retrieval/read step; the shared guide in Prompt F4 remains the common instruction layer.

Prompt F5. Answer grading — adapted task instruction

Instruction. Grade the model answer using every dimension in scoring_profile. Return JSON only. Judge the answer against the private gold and the question-specific rubric. Do not reward a confident unsupported guess. A correct abstention is better than a confident materially false answer.

Input fields. `question`, `gold`, `question_specific_rubric`, `scoring_profile`, and `model_output`.

Output. Eight dimension scores in $[0, 1]$, severity and error flags, a question identifier, and a short explanation. Programmatic processing applies the configured weights, caps, and exceptions to each grading result.

Each answer is graded three times. Its reported score is the mean of the three results after applying the scoring rules described in Appendix C. Model-based judgments and programmatic score processing form separate stages of the evaluation.

F.4 Model configuration and invocation settings

Construction and review models. PaperStudy extraction uses DeepSeek Pro. The ten authoring roles in Table S22 also use DeepSeek Pro, with a configured base temperature of 0, an output budget of 12,000 tokens, and a context budget of 128,000 tokens. The authoring provider requests JSON outputs. The additional extraction-review pass uses GPT. These model-review stages precede the candidate-question human review described in Section F.1.

Table S23: Reader families, variants, and per-call output budgets in the reported reader configurations. Repair budgets apply to JSON-repair calls.

Reader	Model variant	Answer tokens	Repair tokens
DeepSeek	DeepSeek Flash	6,000	6,000
Gemini	Gemini Flash	32,768	32,768
Qwen	Qwen Plus	32,768	32,768
Claude	Claude Haiku	12,000	12,000

Answer and grading calls. Table S23 pairs model families and variants with reader-specific output budgets. The reader API requests temperature 0. The grader uses DeepSeek Pro, temperature 0, a 6,000-token output budget, and disabled thinking in the DeepSeek request. Three successful grading results contribute to the answer-level mean; failed API attempts are handled by the retry mechanism.

Retrieval configuration. The Core dual-plane explicit-grounding policy specifies the settings in Table S24. These are retrieval-policy parameters: they govern evidence selection and remain distinct from the reader's generation budget. The reference native-memory configuration separately names `bge-m3` embeddings and the six-layer MS MARCO MiniLM reranker for its supported baseline routes.

Table S24: Settings in the Core retrieval-policy configuration. The policy uses an active revision view and source-grounded evidence.

Parameter	Configured value
Channel candidate limit / final limit	30 / 12
Raw / counter / qualifying evidence quotas	6 / 2 / 1
BM25 parameters	$k_1 = 1.5, b = 0.75$
Reciprocal-rank fusion constant	$k = 60$
Dense-channel encoder	Deterministic hash encoder, 512 dimensions, ℓ_2 normalization
Query planner	Deterministic scientific query planner
Evidence reranker	Deterministic epistemic reranker

Source templates. The extraction and semantic-review templates are defined in the PaperStudy package. Authoring roles are stored as separate Markdown prompts in the benchmark runtime; the pipeline combines the role instructions with a JSON payload. The public release supplies the answering guide and output schema. The grading script constructs the evaluator request from the private reference, task rubric, eight-dimensional profile, and completed answer. The code package preserves these templates and their configuration files.