A Multimodal Literature Agent as Substrate for Autonomous Biology Research

M. Brkić [1] **L. Weidener** [1] **M. Jovanović** [1] **E. Ulgac** [1] **A. Meduri** [1]

Abstract

Autonomous science systems for biology, covering hypothesis generation, experimental design, and data analysis, depend on a literature module they can trust. Most existing modules treat scientific publications as text, missing the figures, gels, and tables where biological evidence is encoded. We present a literature agent that is multimodal from the ingestion layer up: figures and tables are first-class retrieval artifacts, and figure inspection is delegated to a multi-round vision-language-model (VLM) zoom loop with an explicit abstention action that flags incorrect retrievals rather than answering from them. The agent reaches the strongest publicly reported scores on three components of LAB-Bench 2: 62.5% on FigQA2, 88.6% on LitQA3, and 88.8% on TableQA2, exceeding the best public baselines by 4.4, 4.1, and 9.5 points respectively. Provenance is tracked at the chunk level (DOI, page, character offsets) and per-publication metadata (retraction status, journal, citation count) is surfaced alongside each answer, making the agent a credible grounding layer for the action-taking agents that complete an autonomous discovery pipeline. As laboratory automation expands, a trustworthy literature module (one that reads the actual evidence rather than only the prose around it) becomes the substrate the rest of the system builds on.

1. Introduction

Recent autonomous-science systems span hypothesis generation, experiment design, code execution, and manuscript drafting (Lu et al., 2024; Gottweis et al., 2025; Yamada et al., 2025), with biology-specific demonstrations including drug repurposing in a lab-in-the-loop (Ghareeb et al., 2025), twelve-hour autonomous discovery campaigns (Mitchener et al., 2025), and chemistry agents wired to robotic synthesis (Boiko et al., 2023; Bran et al., 2024). Every one of these systems depends, somewhere in the loop, on a *literature module* that turns the published record into evidence the agent can act on. The quality of that module sets an upper bound on the science the rest of the fleet can perform.

Today's literature modules are textual (Lála et al., 2023; Skarlinski et al., 2024; OpenAI, 2025). That assumption is wrong for biology: decades of biological knowledge are encoded in multi-panel figures, gels, micrographs, and tables of measurements, and an agent that reads only parsed prose can describe what a publication states about its results but is structurally blind to those results themselves. Concurrent multimodal document-retrieval work embeds page images directly (Faysse et al., 2024; Yu et al., 2024; Cho et al., 2024), but trades the exact-token retrieval that biology questions about gene names, compound identifiers, and numeric values demand.

We present a literature agent that treats text, tables, and figures as first-class retrieval artifacts simultaneously, and delegates figure inspection to a multi-round VLM zoom loop with an explicit abstention action. The agent is designed as the literature substrate for a broader fleet of action-taking scientific agents (data analysis, novelty detection, domain-specific predictors); this submission focuses on the read-only literature side, with safeguards described in §4.

2. System

Orchestration. A reasoning LLM dispatches typed tool calls and decides termination over a small set: `search_papers`, `search_pdfs` (hybrid retrieval over chunked text, tables, and figures), `inspect_figure` (the multi-round zoom loop below), `grep_document` and `grep_chunks` (exact-match search), `prune_chunks`, and `conclude_search`. Domain lookups (`search_pubchem`, `search_uniprot`, `search_trials`, `search_patents`) sit at the same level. We chose full LLM autonomy over a fixed search–rerank–inspect–answer pipeline because life-sciences questions vary too much for one path: a single reported value is one lookup, whether two studies' results agree

[1]Applied Scientific Intelligence, Inc.. Correspondence to: M. Brkić <marko@appliedscientific.ai>.

Accepted at the AI for Science workshop (ICML 2026) as a poster presentation.

is a multi-publication comparison, and a question about a sub-panel demands zooming into pixels that a textual retriever cannot see at all. The orchestrator may also conclude from abstracts and metadata alone, returned by `search_papers`, without escalating to full-text chunk retrieval via `search_pdfs`. This is a deliberate affordance that lets the agent answer broad questions efficiently (which publications exist on a topic, what a study reports at a glance), reserving deep full-text retrieval and figure inspection for questions requiring evidence buried in the body of a publication.

Ingestion. PDFs are parsed page-by-page into reading-order body text, tables (rendered to markdown), and figures (image plus caption). Each chunk is contextualized before embedding (Anthropic, 2024): text chunks are prepended with title and abstract; table chunks carry the caption plus a VLM-generated summary; figure chunks carry the caption, a VLM-generated description, and parser-extracted in-figure text (axis labels, panel names). Contextualized chunks are written to a vector index supporting hybrid (dense + BM25) retrieval, with metadata for filtering, reference-graph traversal, and chunk-level provenance (DOI, page, character offsets).

Retrieval and two-axis reranking. Dense and BM25 (Robertson & Zaragoza, 2009) searches run in parallel and fuse with Reciprocal Rank Fusion (Cormack et al., 2009), so chunks anchored on either semantic similarity or exact-token matches surface together. A pretrained cross-encoder (Nogueira & Cho, 2019) reduces the candidate set; an LLM reranker then scores each chunk on a 0–10 scale for answer relevance. For figure-linked chunks the reranker emits a second 0–10 score: an inspection priority. A 10 forces inspection; 8–9 strongly biases the orchestrator toward it; below 8 is left to discretion. Separating "the text answers the question" from "the figure answers the question" matters because the two are often dissociated.

Figure inspection with abstention. The dominant failure mode of off-the-shelf agents on figure questions is handing the entire multi-panel PNG to a VLM in a single pass: small axis labels, data points, and panel-level details are illegible at full-figure resolution. `inspect_figure` runs a multi-round zoom loop. Each round, the VLM receives the question, the surrounding chunk text, and the current view stack (full figure plus prior crops), and chooses one of three actions: *answer* (with structured fields including evidence and confidence); *request a zoom* (return a normalized bounding box and a target hint, e.g., "panel D2, red curves at -40 mV", rendered at higher resolution and pushed onto the view stack); or *flag the wrong figure*: an explicit abstention path so the agent does not expend further zoom rounds (and the inspection budget) on an erroneously promoted figure. Up to three rounds per inspection; if confidence is not reached, a final round runs at maximum reasoning effort with the full view stack in context.

Table 1. Accuracy (%) on LAB-Bench 2 components. Baselines: PaperQA2 (Skarlinski et al., 2024); o3 Deep Research (OpenAI, 2025); Edison Literature, normal and high-effort variants (White et al., 2026). Our system is the mean of three runs.

System	FigQA2	LitQA3	TableQA2
PaperQA2	17.5	–	–
o3 Deep Research	29.4	84.5	74.3
Edison Literature	42.6	80.4	70.7
Edison Literature high	58.1	82.3	79.3
Ours	**62.5**	**88.6**	**88.8**

3. Evaluation

We report results in two parts: headline accuracy against the strongest publicly reported baselines (§3.1), and ablations isolating the contribution of the two most distinctive pipeline components (§3.2).

3.1. Headline results

We evaluate on three components of LAB-Bench 2 (Laurent et al., 2026): FigQA2 (figures), LitQA3 (text), and TableQA2 (tables).

The three benchmarks reveal distinct patterns. **LitQA3** shows the smallest spread between systems (all baselines achieve accuracy in the 80% range), and our $+4.1$ over o3 derives from cumulative incremental improvements: contextualized embeddings, the LLM reranker scoring for answer relevance not just semantic similarity, and the grep tools handling exact-token queries that embedding-based retrieval imperfectly resolves. **TableQA2** admits larger gains; our $+9.5$ margin is mostly in table *retrieval*: chunks carry both markdown rendering and a VLM summary prepended with title and abstract, so they retrieve on the question itself rather than on keyword overlap. **FigQA2** is the most informative. Parser-text-only systems plateau (17.5%, 29.4%): no amount of reasoning can recover unread pixels. Treating figures as first-class artifacts produces the first substantial improvement (Edison 42.6%, 58.1% with a larger reasoning budget); our additional $+4.4$ derives from the retrieval pipeline, the grep tools, and the zoom loop's abstention action. The remaining 37.5% gap is almost entirely figure *reading*, not retrieval (figure recall exceeds 78%, DOI recall exceeds 94%), so the bottleneck has moved from *can the system see the figure* to *can the model interpret what it sees* and the score will track VLM progress.

Table 2. Component ablations (mean of three runs). Each ablation is compared against the full system scored on the same clean subset.

Configuration	Accuracy (%)	Δ (pp)
FigQA2 (clean subset, $n \approx 46$)		
Full system	63.9 ± 2.4	–
− zoom loop	50.4 ± 5.4	-13.5
TableQA2 (clean subset, $n \approx 44$)		
Full system	91.3 ± 0.9	–
− reranker	83.0 ± 1.0	-8.3

3.2. Ablations

To isolate the contribution of the two most distinctive pipeline components (the multi-round zoom loop and the two-stage reranker), we disable each in turn and re-run the affected benchmark. The zoom loop is the FigQA2-relevant component, so we ablate it there. The reranker governs how answer-bearing chunks are promoted out of hybrid retrieval, which is exercised heavily on TableQA2 because tables are directly embedded in the chunks, so we ablate it there. Each ablation runs on a fixed 50-question random subsample of its benchmark, three times, and we report the mean ± population standard deviation, matching the headline protocol. Because the agent does not always invoke the ablated component on every question (for the reranker, it sometimes answers from abstracts alone without retrieving chunks, as described in §2; for figure inspection, some FigQA2 questions are answerable from surrounding text without opening the figure), we report accuracy on the *clean subset*, the questions on which the ablated component in fact fires. This isolates the component's direct effect rather than diluting it with questions the component never touched.

Removing the zoom loop costs 13.5 points on FigQA2. Without on-demand magnification the VLM operates on a downsampled full-figure view and fails on exactly the fine-detail tasks the loop was built for: single-digit misreads of small overlay labels, miscounts of densely packed data points, and unresolvable cluster boundaries in embedding plots (examples in Appendix C.3). The failure is not merely lower confidence: in several cases the single-pass VLM returns a confidently wrong reading because at full-figure resolution it cannot see that what it is reading is wrong. The agent does not compensate by inspecting more figures (it inspects approximately the same number; see §C.2), so the loss reflects the reading itself.

Removing the two-stage reranker costs 8.3 points on TableQA2. Hybrid retrieval still ranks chunks by similarity to the query, but similarity is not answerability: a table cell holding the exact value and a paragraph of prose discussing the same assay are both topically on-target, and the retrieval score alone does not separate them. The reranker exists to make that second judgment, scoring each candidate for whether it answers the question, and without it the agent cannot reliably distinguish the answer-bearing chunk from the surrounding discussion. In a representative case it settles for a qualitative hedge ("$<$ 10 nM") drawn from a prose chunk, where the exact value (3 nM) sat in a table chunk that ranked no higher on similarity alone. We report this effect on the clean subset because, without the reranker, the agent also answers a larger share of questions from abstracts without calling the `search_pdfs` tool. The effect is several times the run-to-run standard deviation of the baseline, so the ranking of component importance is unambiguous even given benchmark stochasticity.

4. Governance, Provenance, and Outlook

Every answer is bound to a source chunk (DOI plus page and character offsets), with each retrieval, inspection, and zoom crop logged so a scientist can audit every step. A separate per-publication store (retraction status, corrections, journal, citation count) is fetched on demand and surfaced with the answer; folding it into the reranker is the next concrete step. The agent only *reads*, so its misuse surface is bounded by what is already public in the biomedical literature; the orchestrator refuses queries resembling operational instructions for harmful work (e.g., pathogen synthesis). The action-taking agents this module is built to ground will need their own policy layer, out of scope here. Open directions include distilling the orchestrator and tools into smaller specialized models for lower cost and latency.

References

Anthropic. Introducing contextual retrieval. Anthropic Research Blog, September 2024. https://www.anthropic.com/news/contextual-retrieval.

Boiko, D. A., MacKnight, R., Kline, B., and Gomes, G. Autonomous chemical research with large language models. *Nature*, 624:570–578, 2023.

Bran, A. M., Cox, S., Schilter, O., Baldassari, C., White, A. D., and Schwaller, P. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6: 525–535, 2024.

Cho, J., İrsoy, O., Mahata, D., He, Y., and Bansal, M. M3DocRAG: Multi-modal retrieval is what you need for multi-page multi-document understanding. *arXiv preprint arXiv:2411.04952*, 2024.

Cormack, G. V., Clarke, C. L. A., and Büttcher, S. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Devel-*

opment in Information Retrieval (SIGIR), pp. 758–759. ACM, 2009.

Faysse, M., Sibille, H., Wu, T., Omrani, B., Viaud, G., Hudelot, C., and Colombo, P. ColPali: Efficient document retrieval with vision language models. *arXiv preprint arXiv:2407.01449*, 2024.

Fell, J., Fischer, J., Baer, B., Blake, J., Bouhana, K., et al. Identification of the clinical development candidate MRTX849, a covalent KRAS G12C inhibitor for the treatment of cancer. *Journal of Medicinal Chemistry*, 63: 6679–6693, 2020.

Ghareeb, A. E., Chang, B., Mitchener, L., Yiu, A., Szostkiewicz, C. J., Laurent, J. M., Razzak, M. T., White, A. D., Hinks, M. M., and Rodriques, S. G. Robin: A multi-agent system for automating scientific discovery. *arXiv preprint arXiv:2505.13400*, 2025.

Gottweis, J., Weng, W.-H., Daryin, A., Tu, T., et al. Towards an AI co-scientist. *arXiv preprint arXiv:2502.18864*, 2025.

Kedzierska, K., Crawford, L., Amini, A., and Lu, A. Assessing the limits of zero-shot foundation models in single-cell biology. *bioRxiv preprint 2023.10.16.561085*, 2023.

Lála, J., O'Donoghue, O., Shtedritski, A., Cox, S., Rodriques, S. G., and White, A. D. PaperQA: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559*, 2023.

Latorraca, N., Wang, J., Bauer, B., Townshend, R., Hollingsworth, S., et al. Molecular mechanism of GPCR-mediated arrestin activation. *Nature*, 557:452–456, 2018.

Laurent, J. M., Bou, A., Pieler, M., Igoe, C., Andonian, A., Narayanan, S., Braza, J., Vassopoulos, A. S., Steenwyk, J. L., Lash, B., White, A. D., and Rodriques, S. G. LABBench2: An improved benchmark for AI systems performing biology research. *arXiv preprint arXiv:2604.09554*, 2026.

Lu, C., Lu, C., Lange, R. T., Foerster, J., Clune, J., and Ha, D. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

Miller, T., Pestronk, A., David, W., Rothstein, J., Simpson, E., et al. An antisense oligonucleotide against SOD1 delivered intrathecally for patients with SOD1 familial amyotrophic lateral sclerosis: A phase 1, randomised, first-in-man study. *The Lancet Neurology*, 12:435–442, 2013.

Mitchener, L., Yiu, A., Chang, B., Bourdenx, M., Nadolski, T., Sulovari, A., et al. Kosmos: An AI scientist for autonomous discovery. *arXiv preprint arXiv:2511.02824*, 2025.

Nogueira, R. and Cho, K. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*, 2019.

OpenAI. Introducing deep research. OpenAI Blog, February 2025. https://openai.com/index/introducing-deep-research/.

Robertson, S. and Zaragoza, H. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

Skarlinski, M. D., Cox, S., Laurent, J. M., Braza, J. D., Hinks, M., Hammerling, M. J., Ponnapati, M., Rodriques, S. G., and White, A. D. Language agents achieve superhuman synthesis of scientific knowledge. *arXiv preprint arXiv:2409.13740*, 2024.

White, A. D., Braza, J. D., Pieler, M., Skarlinski, M., and Narayanan, S. Introducing PaperQA3: A frontier multimodal deep research agent for science. Edison Scientific Blog, February 2026. https://edisonscientific.com/articles/edison-literature-agent.

Yamada, Y., Lange, R. T., Lu, C., Hu, S., Lu, C., Foerster, J., Clune, J., and Ha, D. The AI scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025.

Yu, S., Tang, C., Xu, B., Cui, J., Ran, J., Yan, Y., Liu, Z., Wang, S., Han, X., Liu, Z., and Sun, M. VisRAG: Vision-based retrieval-augmented generation on multi-modality documents. *arXiv preprint arXiv:2410.10594*, 2024.

A. System architecture in detail

This appendix provides additional architectural detail not included in the main text: the ingestion pipeline (Figure A.1), an illustrative multi-panel scientific figure that motivates the contextualization design (Figure A.2), and the full retrieval pipeline (Figure A.3).

Ingestion and contextualization. We contextualize different chunk types differently. Text chunks require minimal additional context because the contextual embedding already incorporates surrounding scientific context; only the title and abstract are prepended. Tables receive more: we convert the LaTeX rendering to markdown (a format native to LLMs) and generate a short description of the table's contents, so that queries asking about values inside the table can retrieve it rather than only queries matching the column headers. Figures require the most contextualization. A single complex figure from a scientific publication can be the answer to many distinct questions, since multi-panel figures bundle several plots into one artifact. Embedding the caption alone will not surface the figure for all such questions, so we index three pieces of information together: the caption, parser-extracted in-figure text (axis labels, panel identifiers), and a VLM-generated description capturing the full semantic content of the figure. These three views cover three distinct query shapes: queries phrased as the authors framed the figure (caption), queries naming text printed on the figure (parser text), and queries about what the figure depicts (VLM description).

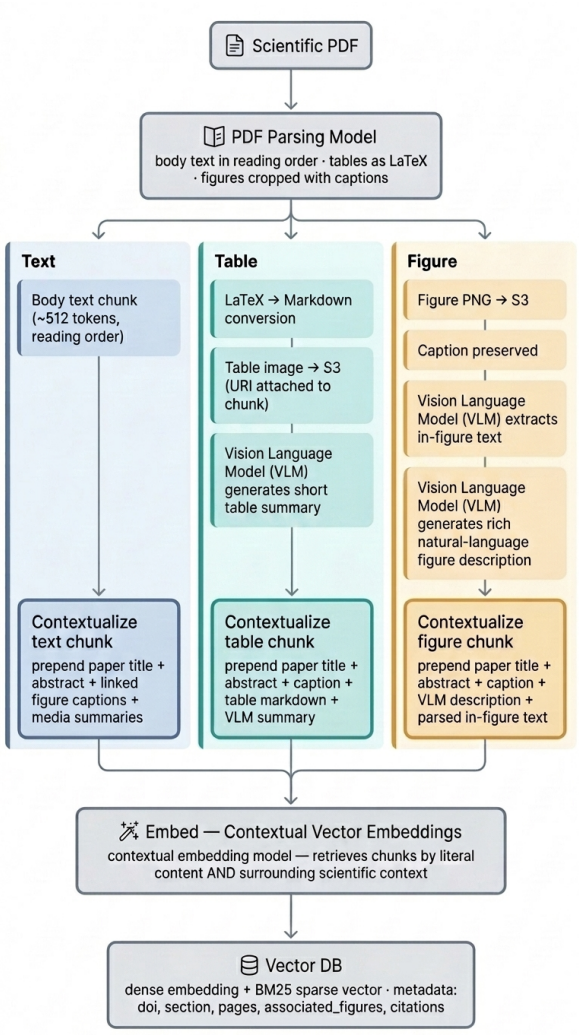

Figure A.1. Ingestion pipeline. PDFs are parsed into text, table, and figure chunks, each contextualized according to its type before embedding and storage in a vector database supporting hybrid retrieval.

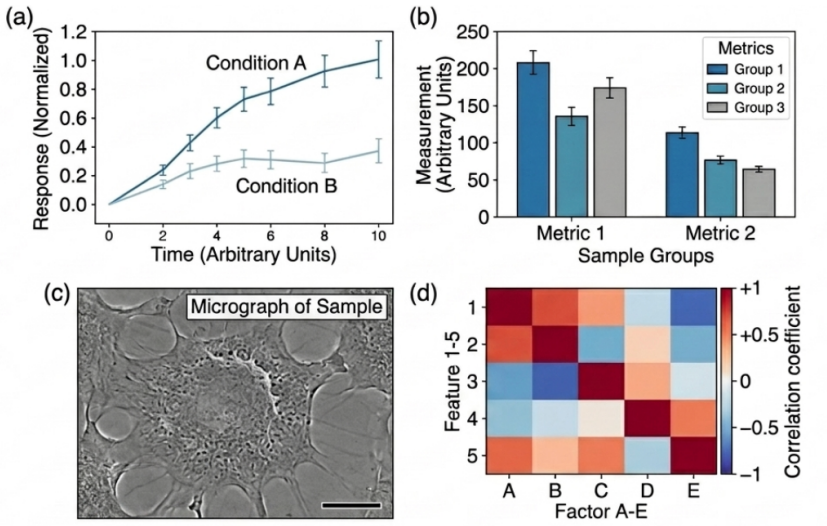

Figure A.2. A typical multi-panel scientific figure. Four sub-panels show different data types: a time-course (a), a grouped bar chart with metrics (b), a micrograph (c), and a correlation heatmap (d). A single embedding of the caption ("Figure 1. Experimental results.") would not surface this figure for the majority of queries about any of the subpanels.

Retrieval and reranking. We use hybrid search because dense retrieval and BM25 fail in different ways on biology queries. Dense retrieval imperfectly resolves exact tokens (gene names, compound identifiers, mutation strings, numeric values), which are precisely the surface forms scientists query on; BM25 retrieves these directly. Conversely, BM25 misses queries phrased in terms the chunk does not contain literally, where dense retrieval succeeds. Reciprocal Rank Fusion combines both ranked lists without requiring the two retrievers to agree: a chunk strong on either axis surfaces near the top of the fused candidate set.

The two reranking stages perform different roles, which motivates the two-stage design. The cross-encoder is inexpensive and effective at filtering the candidate pool from hundreds of chunks to tens, but it scores on textual similarity rather than on whether a chunk answers the question. The LLM reranker addresses that second question: given the query, does this chunk contain the evidence the agent requires? This judgment is what retrieval ultimately needs to deliver, but the LLM reranker is too expensive to run over hundreds of candidates, so the cross-encoder performs the initial pruning.

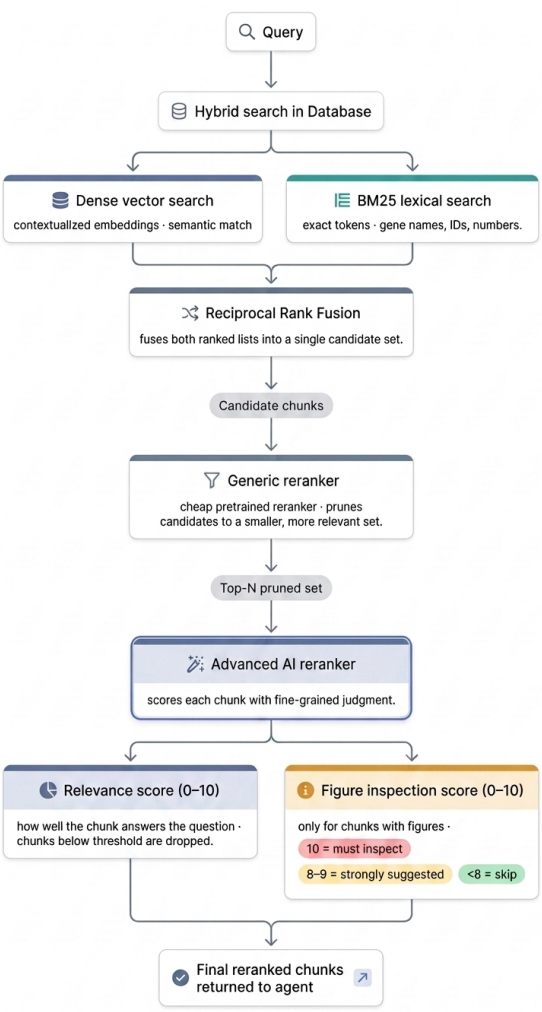

Figure A.3. Full retrieval pipeline. Hybrid dense and BM25 retrieval is fused via Reciprocal Rank Fusion, followed by cross-encoder pruning and an LLM reranker that emits both an answer-relevance score (0–10) and, for figure-linked chunks, a separate inspection-priority score.

B. Benchmark deep-dives

B.1. FigQA2

The reported 62.5% on FigQA2 is the mean of three independent runs over the same 101 questions. The runs produced 65.8%, 60.9%, and 60.9%, with a population standard deviation of 2.3 percentage points. This spread is consequential: 13 questions changed verdict between runs, so any single-run number is noisy at the question level. The mean is the appropriate summary statistic; a single run is not.

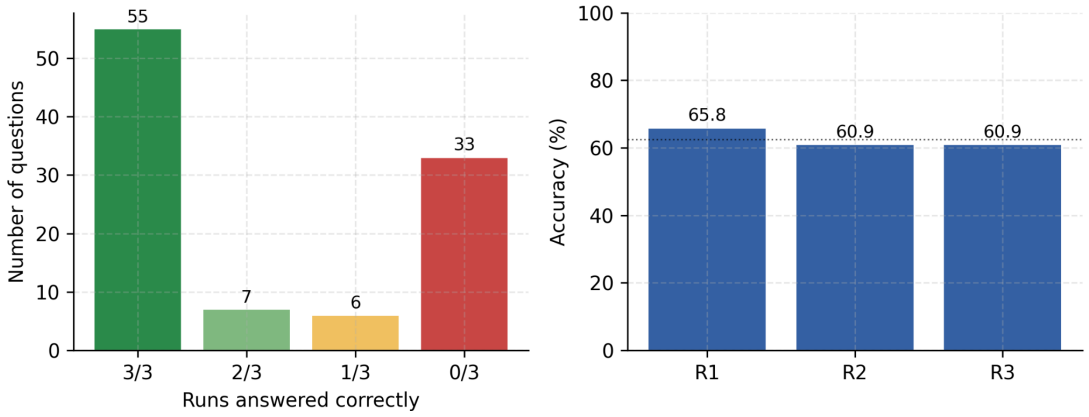

Figure B.1.1. Run-to-run agreement on FigQA2. Left: number of questions correctly answered on each of 0, 1, 2, or 3 runs. Right: per-run accuracy.

The bimodal distribution (most questions either solved on every run or missed on every run, with relatively few in between) indicates where the agent's accuracy is concentrated. The 55 stable wins are questions the architecture handles confidently; the 33 stable losses constitute the failure set analyzed below. The 13 intermediate questions are those where stochasticity in the VLM's interpretation of a figure determines the outcome.

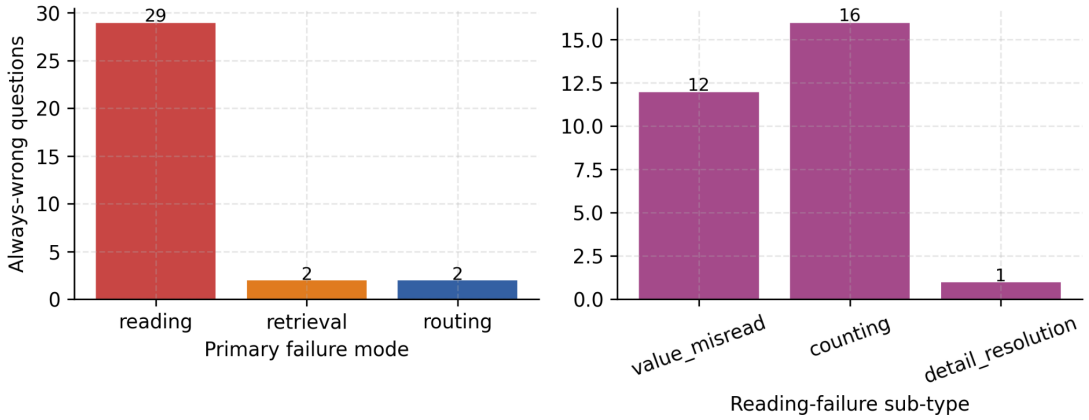

Figure B.1.2. Failure breakdown on FigQA2. Of the 33 questions answered incorrectly on every run, 29 are reading failures (correct figure inspected, pixels read incorrectly) and only 4 are upstream (2 retrieval, 2 routing). Right panel: reading-failure subtypes.

Across the 303 (run $\times$ question) trials, the agent makes a median of 6 tool calls per question (P90 $=$ 11, max $=$ 21). Approximately two of these are `inspect_figure` calls and two are zoom requests within those inspections. The 5–6 tool-call median matches the canonical successful trajectory: one `search_papers` to narrow the corpus, one `search_pdfs` to retrieve the relevant chunks, one or two `inspect_figure` calls on the surfaced figures, and a final `conclude_search` to terminate.

When the agent reaches an answer in 5–6 calls (the typical case), correctness is roughly even on a per-trial basis. When it continues for 10 or more calls, it is predominantly retrying on a question it ultimately answers incorrectly: retries are the agent's response to non-converging evidence, and they do not consistently rescue the answer.

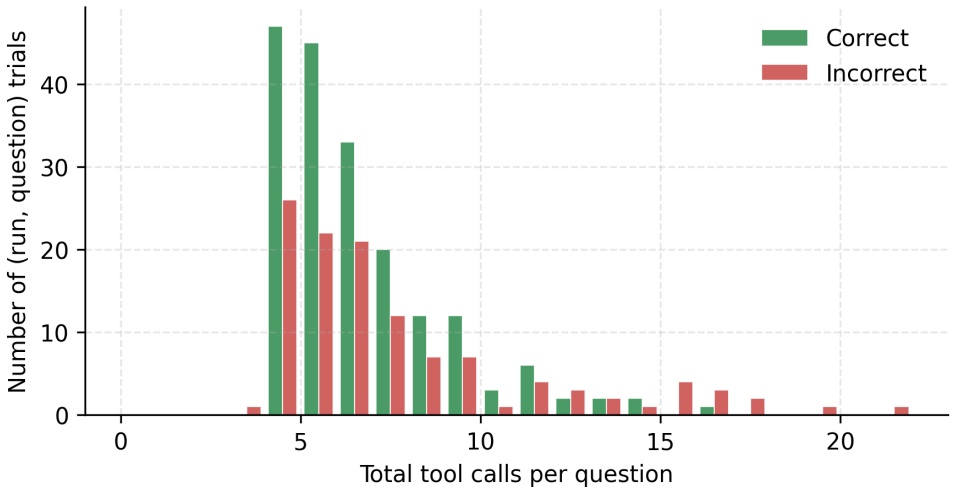

Figure B.1.3. Tool calls per question on FigQA2, split by correctness.

Wall-clock time per question is approximately 6 minutes at the median and up to 24 minutes in the tail; correct answers complete marginally faster than incorrect ones (median 306 s versus 372 s), suggesting the agent continues iterating when the evidence is thin. The VLM driving figure inspection accounts for the majority of per-question latency. Replacing the foundation vision model with a smaller model post-trained specifically for the zoom loop would substantially reduce latency.

B.2. LitQA3

The reported 88.6% on LitQA3 is the mean of three independent runs over the same 168 questions. The runs produced 90.5%, 89.3%, and 86.0%, with a population standard deviation of 1.9 percentage points. Run-to-run agreement is substantially tighter on LitQA3 than on FigQA2: 138 of 168 questions are correct on every run and only 14 are wrong on every run. The remaining 16 flip between runs and account for nearly all of the 4.5-point spread between the best and worst runs.

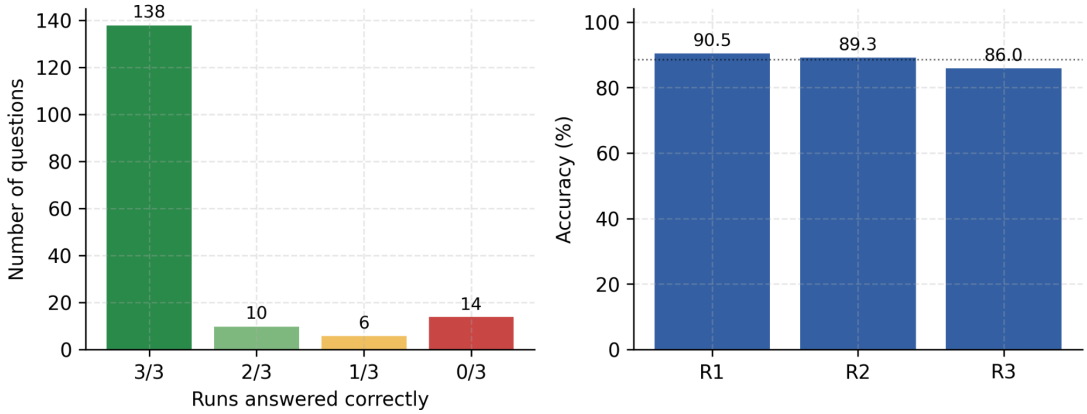

Figure B.2.1. Run-to-run agreement on LitQA3. Left: number of questions correctly answered on each of 0, 1, 2, or 3 runs. Right: per-run accuracy.

Of the 14 questions wrong on every run, 12 are reading failures and 2 are retrieval failures. There are no routing failures on LitQA3 because routing is largely trivial: once a chunk is surfaced, the agent reads it. The retrieval failures are the architecturally relevant cases: in 2 of 168 questions the source publication's DOI never surfaced as a top-1 chunk in any of three runs, an irrecoverable failure of the retrieval pipeline.

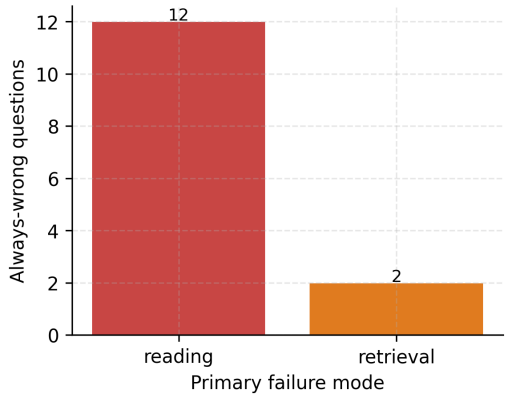

Figure B.2.2. Failure breakdown on LitQA3 across the 14 questions answered incorrectly on every run.

Across the 504 (run × question) trials, the agent makes a median of 5 tool calls per question (P90 = 14, max = 32). The canonical successful trajectory is shorter than on FigQA2 (`search_papers` to narrow the corpus, one or two `search_pdfs` calls to retrieve the relevant chunks, and `conclude_search` to terminate) because there is usually no figure-inspection loop to traverse. Wall-clock time per question is approximately 3 minutes at the median, substantially faster than FigQA2 because per-question latency is no longer dominated by VLM inference on figures.

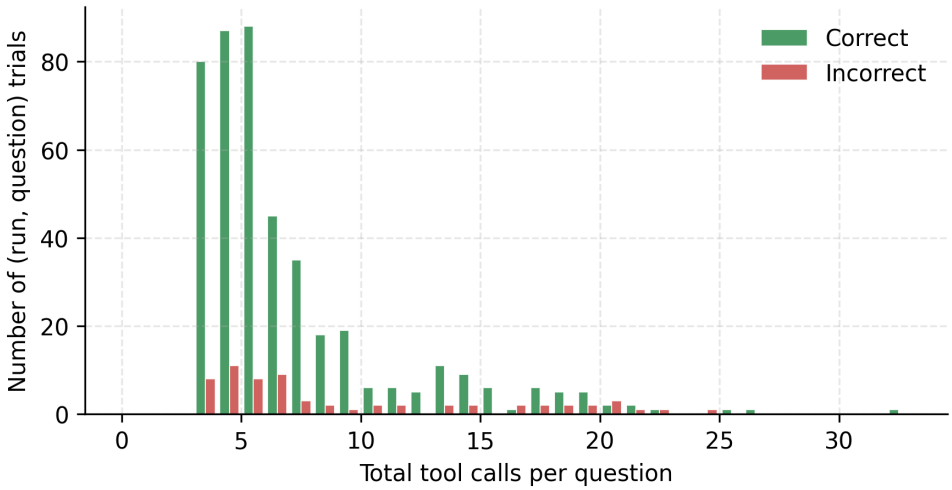

Figure B.2.3. Tool calls per question on LitQA3, split by correctness.

B.3. TableQA2

The reported 88.8% on TableQA2 is the mean of three independent runs over the same 100 questions. The runs produced 89.5%, 88.5%, and 88.5%, with a population standard deviation of 0.5 percentage points (the tightest of the three benchmarks). Eighty-five of 100 questions are correct on every run and only 9 are wrong on every run; the system's behavior on tables is essentially deterministic.

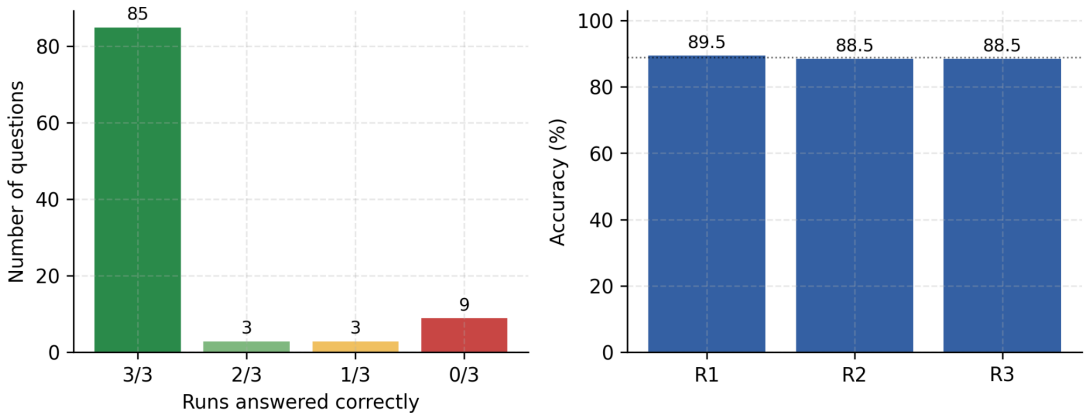

Figure B.3.1. Run-to-run agreement on TableQA2. Left: number of questions correctly answered on each of 0, 1, 2, or 3 runs. Right: per-run accuracy.

Of the 9 questions wrong on every run, 7 are reading failures, 1 is a routing failure, and 1 is a retrieval failure. The dominant failure mode is the same as on the other two benchmarks: the correct table was retrieved and inspected, but the agent extracted the wrong value. The specific failure types vary, including wrong row, wrong column, transposed numerator and denominator in a ratio, and selecting a value from a control row when the question asked for the experimental condition. Tables are substantially more stable across runs than figures, because table cells are discrete and well-localized whereas figure panels often require sub-pixel inference, the source of most FigQA2 stochasticity.

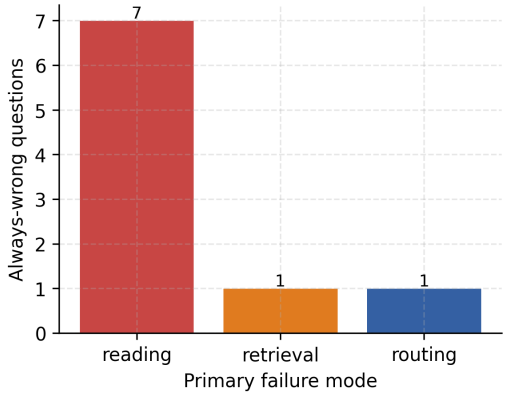

Figure B.3.2. Failure breakdown on TableQA2 across the 9 questions answered incorrectly on every run.

Across the 300 (run × question) trials, the agent makes a median of 4 tool calls per question (P90 = 12, max = 21), the shortest median trajectory of the three benchmarks. The canonical successful trajectory is `search_papers` → `search_pdfs` → `conclude_search`, with the answer extracted directly from the table's markdown rendering inside the retrieved chunk. `inspect_figure` is invoked only on the subset of questions where the markdown conversion lost information that matters (a complex header, a multi-line cell, a value embedded in a footnote), and these cases drive the long-trajectory tail of the distribution. Wall-clock time per question is approximately 2 minutes at the median, faster than both FigQA2 and LitQA3 because most questions are answered without any VLM inspection in the loop.

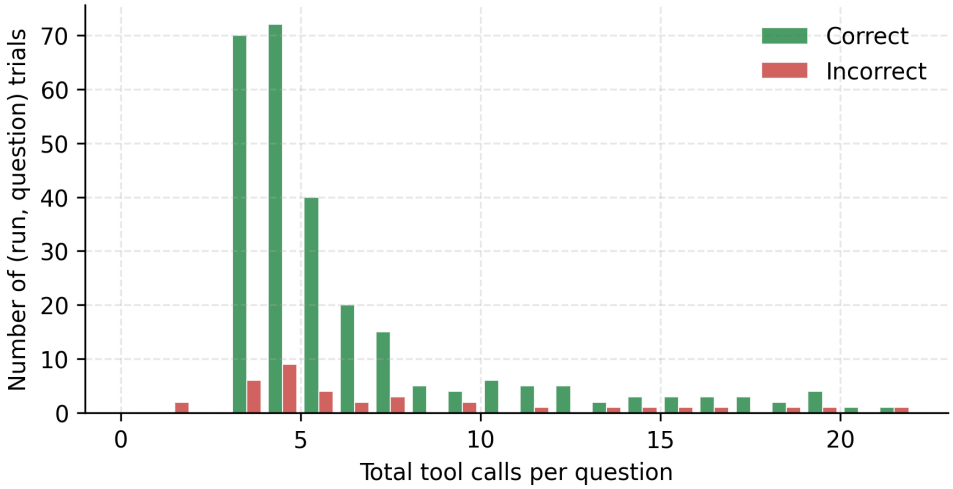

Figure B.3.3. Tool calls per question on TableQA2, split by correctness.

C. Models, costs, and ablation cases

C.1. Models and settings

Table C.1 lists every model in the pipeline. For the headline runs reported in this paper, all LLM and VLM roles were served by a single model (gpt-5.4); reasoning effort was set to medium for the orchestrator and the figure-inspection loop and low for the remaining roles. Three non-LLM components sit outside the per-question token budget: a dense embedding model, a cross-encoder used for first-stage reranking, and the ingest-time PDF parser.

Table C.1. Pipeline models (gpt-5.4 snapshot: 2026-03 release).

Role	Provider	Model	Reasoning
Orchestrator	OpenAI	gpt-5.4	medium
Figure-inspection VLM	OpenAI	gpt-5.4	medium
Answer synthesis	OpenAI	gpt-5.4	low
LLM reranker	OpenAI	gpt-5.4	low
Paper selector	OpenAI	gpt-5.4	low
Judge (evaluation)	OpenAI	gpt-5.4	low
Dense embedding	Perplexity	pplx-embed-context-v1-4b	–
Cross-encoder reranker	Jina	jina-reranker-v3	–
PDF parser (ingest)	NVIDIA	Nemotron-Parse-v1.2	–

C.2. Compute cost and what it buys

Table C.2 reports per-question token usage and model calls by pipeline role, on the full instrumented pipeline for both benchmarks. Five roles account for essentially all gpt-5.4 spend: the orchestrator (next-step reasoning), the figure-inspection VLM, the LLM reranker, answer synthesis, and the paper selector. The non-LLM components (the dense embedding model, the cross-encoder, and the PDF parser) run on separate providers and are outside the gpt-5.4 budget. We report the amortized

mean per question over all 50 questions, so roles that fire on only a subset (most notably the figure-inspection VLM) appear at their true per-question cost rather than their cost-when-fired. The accuracy figures referenced below are the clean-subset ablation deltas from §3.2 (the effect attributable to each component on the questions where it fires); the token figures come from single instrumented runs used only for cost estimation.

Table C.2. Per-question tokens and model calls by role (mean over all 50 questions). "Fires on" is how many questions invoke each role.

Role	Input tok	Output tok	Calls	% tok	Fires on
TableQA2 (full instrumented pipeline)					
orchestrator	65,349	314	5.0	61.0	50/50
LLM reranker	15,567	1,097	2.2	15.5	49/50
answer synthesis	8,845	485	1.0	8.7	50/50
figure-inspection VLM	8,422	814	1.6	8.6	21/50
paper selector	5,291	245	1.3	5.1	50/50
FigQA2 (full instrumented pipeline)					
orchestrator	111,411	498	7.9	64.5	50/50
figure-inspection VLM	22,198	2,688	3.8	14.3	45/50
LLM reranker	14,965	1,393	2.7	9.4	49/50
answer synthesis	10,224	747	1.0	6.3	50/50
paper selector	7,681	453	2.0	4.7	50/50

The orchestrator dominates on both benchmarks: it re-reads its growing trajectory context on every step, and dominates more on FigQA2 (111K vs 65K input tokens) because figure questions take more reasoning steps (7.9 vs 5.0 calls per question). The figure-inspection VLM is the next-largest consumer on FigQA2 and the reranker on TableQA2, exactly as the benchmark composition predicts: the VLM fires on 45 of 50 FigQA2 questions but only 21 of 50 on TableQA2.

Both ablated components carry a token premium. Table C.3 pairs each component's cost (the change in total per-question tokens when it is enabled) with the accuracy it buys. Enabling the zoom loop adds approximately a fifth to the FigQA2 token bill and recovers 13.5 accuracy points on the questions where it fires; enabling the reranker adds about a quarter to the TableQA2 bill and recovers 8.3 points. For a scientific-literature agent the trade is straightforward: a confidently wrong answer is more costly than the marginal compute because it can mislead a researcher or propagate into downstream work, so the more accurate, somewhat more expensive configuration is the right default for most users.

Table C.3. Token premium of each ablated component (amortized mean per-question total tokens, with vs. without; medians tell the same story). Accuracy returned, on the clean subset of §3.2: zoom +13.5 pp on FigQA2, reranker +8.3 pp on TableQA2.

Benchmark	Component	Tok./q with	Tok./q without	Premium
FigQA2	zoom loop	173,476	142,525	+22%
TableQA2	reranker	107,614	85,952	+25%

The two premiums have different sources. For the zoom loop, the cost concentrates inside each inspection rather than across more of them: with zoom enabled the agent inspects approximately the same number of figures (median 2 `inspect_figure` calls either way), but each inspection costs roughly twice as much (about 13.0K input tokens versus 6.7K without). The reason is the zoom mechanism itself. Without zoom, an inspection is a single full-figure pass; with it, the VLM averages 2.26 rounds per figure (versus 1.33), and because each round re-sends the accumulated view stack (the full figure plus every prior higher-resolution crop) the later rounds carry progressively more image tokens. The per-question VLM spend roughly doubles as a result (24.7K vs 12.0K input tokens), and that magnification is what makes fine detail (axis ticks, residue labels, individual scatter points) legible enough to read correctly.

For the reranker, the scoring pass costs about 15K input tokens per question, but removing it cuts the per-question total by more, roughly 22K (108K to 86K). The difference is downstream: with the reranker's answerability scores, the agent makes more search calls (about 12 model calls per question versus 9 without), continuing to retrieve and read because the scores signal there is answer-bearing material worth pursuing. Without them, it more often judges the corpus unhelpful and stops early, answering from abstracts. So the reranker's spend buys two things: a ranked, answer-bearing chunk set, and the additional search activity that finds the answer when it is there, each extra call itself costing tokens through the orchestrator and retrieval tools.

C.3. Ablation failure analysis

The ablation deltas in §3.2 are driven by specific, legible failure modes. We give representative cases below; each pairs the same question under the full system (correct) and under the ablation (wrong).

Zoom loop. On a structural schematic, the full system reads the labelled residues as R169 and D290 (correct); without zoom the single-pass VLM reports R175, D296, and three further labels, single-digit misreads of small overlay text it cannot resolve at full-figure resolution, returned at high confidence (Latorraca et al., 2018). On a single-cell UMAP, the full system resolves two acinar/ductal clusters; without zoom the VLM oscillates between three and five and the orchestrator commits to five, because cluster boundaries in a dense embedding are not separable without magnification (Kedzierska et al., 2023).

Reranker. Asked for a mean cellular IC50, the full system promotes the table chunk and returns the exact value (3 nM) from the relevant table; without reranking, the agent sees the hybrid candidates ranked by similarity only (with no answerability signal to separate the table cell from prose discussing the same assay), settles on a prose chunk, and reports the qualitative bound ("$<$ 10 nM") instead (Fell et al., 2020). On a pharmacokinetics question, the answer-bearing PK-table row for the queried dose is never promoted; the ablated agent reasons over narrative chunks from a related trial and concludes, wrongly, that the dose cohort does not exist (Miller et al., 2013).

Cross-model zoom check. To confirm the zoom effect is architectural rather than specific to gpt-5.4, we substituted an independent open VLM (NVIDIA Nemotron Nano Omni 3) into the same loop on FigQA2. Enabling zoom raised its accuracy from 39.6% to approximately 47.5%, a smaller gain than gpt-5.4 obtains but evidence that the zoom loop helps a weaker model as well as a stronger one, amplifying a model's existing spatial-grounding ability rather than supplying it.