CAN SCIENTIFIC AGENTS MAKE RESEARCH STATE TRANSITIONS AS EVIDENCE EVOLVES?

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific discovery unfolds through evolving research states: researchers reason from the available evidence context, identify what remains unresolved, and update conclusions as new evidence arrives. However, scientific agents are often evaluated on isolated tasks or with the full research context available at once, leaving unclear whether they can construct, advance, carry and even revise a state along a real research trajectory. To make these capabilities measurable, we collect recent biomedical studies as source-grounded discovery graphs that record what was known at each point, what evidence supported each claim, and which analyses followed. This structure enables us to derive evidence boundaries and research checkpoints naturally from each study's own trajectory, rather than impose them manually. We instantiate three settings: Initial Discovery from the data available at the outset; Continued Investigation from an source-grounded checkpoint, with only subsequent evidence revealed; and Full-Trajectory, where agents continue from the state they constructed themselves. We evaluate seven frontier agentic systems with graph-grounded rubrics that separately assess scientific setup, analysis design, and empirical endpoint, while allowing equivalent alternatives to the authors' exact implementations. Agents often identify relevant analyses but rarely carry them through to supported conclusions. Supplying a grounded research state substantially reduces stable cross-system differences, indicating that state construction is a major source of capability variation. Strikingly, on matched downstream tasks, performance drops from 61.1% with the source-grounded checkpoint to 35.2% when agents continue from their own state. The loss is concentrated in analyses missed at earlier stages, which agents rarely revisit and repair. Our work shifts evaluation from whether an agent can rediscover scientific findings to whether it can advance a scientifically valid research state as evidence evolves.

1 INTRODUCTION

Scientific discovery unfolds through evolving research states. Researchers form conclusions from the evidence currently available, identify what remains unresolved, and revise or extend those conclusions as new evidence arrives. Scientific agents (Huang et al., 2026; Gottweis et al., 2026) should therefore do more than solve isolated analysis tasks: they should be able to **construct** a warranted research state from raw evidence, **advance** an established state with new evidence, and **revise** their own state when later evidence exposes deficiencies.

However, current evaluations reveal only parts of this process. They assess isolated analysis tasks or question answering over a fixed dataset (Chen et al., 2025; Mitchener et al., 2025), or ask agents to reconstruct published findings with all relevant evidence available at once (Majumder et al., 2025; Wang et al., 2026a). These settings measure useful abilities, but they do not distinguish where failure arises: in constructing the scientific state, in advancing a valid one, or in carrying forward and correcting a state the agent produced itself.

Evaluating a research transition therefore requires two components: staged evidence, so that later data cannot inform earlier decisions, and a ground truth for each stage to specify what the available evidence could support, independent of what the agent chooses to do. Splitting a dataset by hand can stage the evidence, but where the boundaries fall and what each stage supports then reflect the evaluator's judgment rather than the structure of the science. We obtain both components from the

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.

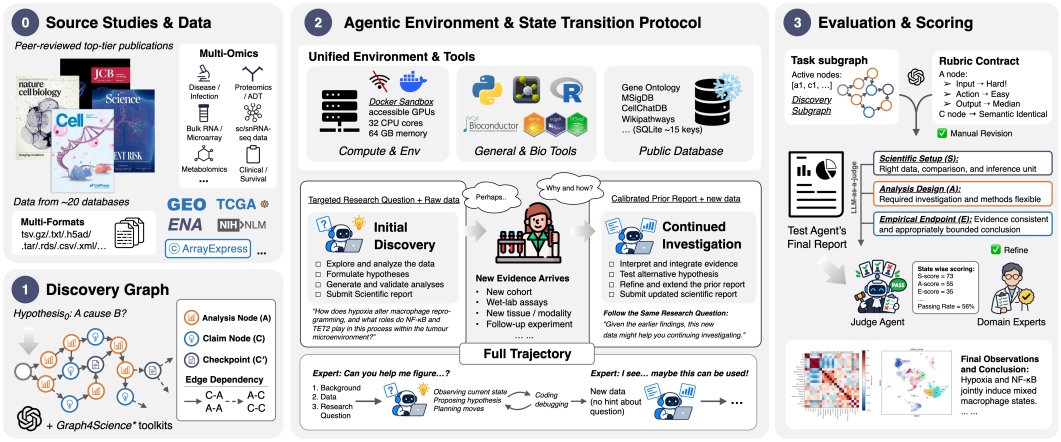

Figure 1: Overview of research state transition evaluation. Discovery graphs derived from published studies define staged evidence releases for Initial Discovery, Continued Investigation, and Full Trajectory. Agents construct and advance research states, while stage-specific rubrics assess scientific setup, analysis design, and empirical endpoints from frozen reports.

science itself, by treating a published study as a record of evidential dependencies, from which stages and their supportable conclusions can be recovered, instead of a set of answers to reproduce.

Concretely, we reconstruct each study as a source-grounded discovery graph, built from the peer-reviewed publication, its supplementary material and code, and the public data it used. Nodes represent the research setup, analyses, empirical evidence and claims; edges encode evidential dependencies, recording which evidence each claim rests on and which later evidence validates or extends it. Evidence boundaries and research checkpoints then follow from the study's own trajectory rather than being imposed. We draw on biomedical studies, which build their central claims through long chains of analyses and whose follow-up evidence typically required a new cohort, assay or platform, so that later stages add genuinely new evidence rather than more of the same.

We use these boundaries to instantiate three settings. **Initial Discovery** asks agents to establish a research state from a question and the initially available data. **Continued Investigation** supplies all systems with the same calibrated, source-grounded checkpoint and asks them to advance it using newly released evidence. **Full Trajectory Investigation** instead requires agents to continue from their own earlier reports, code, and artifacts, exposing whether upstream errors propagate and whether later evidence invites revision. We evaluate each stage with graph-grounded criteria for scientific setup, analysis design, and empirical endpoint, accepting scientifically equivalent implementations rather than requiring reproduction of the source paper' exact workflow.

Across seven frontier systems, our design exposes several agents behaviors hidden by a single endpoint score. Agents touch most relevant analyses but rarely complete the full chain from setup to the supported endpoint: in Initial Discovery, 80.6% of analysis nodes are covered but only 14.3% are complete; and this pattern holds in Continued Investigation. When all systems receive the same source-grounded checkpoint, stable cross-system differences shrink, suggesting that constructing the research state itself accounts for an important part of current capability differences. Strikingly, on identical downstream tasks, performance falls from 61.1% with a source-grounded checkpoint to 35.2% when agents must build on their own earlier state. Dependency analysis further confirms the failure comes from the missing earlier stage analyses. These results reveal an asymmetry: **current agentic systems can move research forward, but rarely reason backward to revisit the state they are building on**. Together, our contributions are:

Research-state Evaluation.We formulate scientific agents evaluation around constructing, advancing, or correcting a research state. Graph-grounded rubrics score the reasoning behind a conclusion rather than its agreement with the paper.

Source-grounded Progression Reconstruction. We recover the evidential and analytical dependencies from published studies to define stage-specific checkpoints, evidence boundaries, and evaluation criteria.

Behavior asymmetry on Scientific Agents. The state construction strongly differentiates current systems, while supplied states suppress this differentiation, and agents carrying their own state do not reopen the first-stage, even when new evidence calls for it.

2 Related Work

Research AI agents combine literature search, code execution, data analysis, and experiment design to tackle open-ended scientific tasks. Evaluations span prescribed analyses in ScienceAgentBench (Chen et al., 2025), research replication in PaperBench (Starace et al., 2025), and broader research and discovery environments in AstaBench and ASI-Bench (Bragg et al., 2026; Zhou et al., 2026). DiscoveryBench (Majumder et al., 2025) and FIRE-Bench (Wang et al., 2026a) evaluate the recovery of scientific insights from data. Claim-level rubrics in FIRE-Bench and evidence-graph scoring in SciRigor (Liu et al., 2026a) further assess whether analyses support scientific conclusions.

In particular, biomedical agents support collaborative research (Gao et al., 2024; Li et al., 2026a), from tool use and hypothesis generation to experimental design (Huang et al., 2026; Gottweis et al., 2026; Ghareeb et al., 2026), with specialized systems for spatial biology (Wang et al., 2025a; Luo et al., 2026b), single-cell analysis (Alber et al., 2026), and target discovery (Zhang et al., 2025). Knowledge and evidence graphs further guide evidence synthesis and research (Wang et al., 2026b; Ren et al., 2026). Benchmarks span bioinformatics pipelines (Guo et al., 2026), omics discovery (Luo et al., 2026a), multistage statistical reasoning (Li et al., 2026b), single-cell and spatial analyses (Diks et al., 2026b;a), autonomous and copilot settings (Johri et al., 2026), and challenges across biological scales (Liu et al., 2026c). Beyond endpoints, BiomniBench scores research processes and claims (Qu et al., 2026), BixBench3 derives tasks from study dependencies (Koch et al., 2026), and BioDSA-1K includes unresolved hypotheses (Wang et al., 2025b). These benchmarks assess within-task reasoning but not revision as new data arrive; TruthInsightBench identifies staged validation as future work beyond its single-phase protocol (Yang et al., 2026).

Meanwhile, trajectory-level and long-horizon evaluations complement these scientific benchmarks by examining how agents reach their outputs. TRACE evaluates deep-research trajectories (Chen et al., 2026); Liu et al. (2026b) diagnoses long-horizon search failures, Sun et al. (2026) focused on hierarchical evidence aggregation, and Dong et al. (2026) develops agents for extended real-world research tasks. Diagnostic evaluations also examine failures across frontier research workflows (Fei et al., 2026) and long-horizon AI research and development beyond final scores (Li et al., 2026c). Sequential diagnosis and belief-revision evaluations test updating under new evidence (Nori et al., 2025; Wilie et al., 2024), but supply findings or premises than require agents to produce the evidence through progressive and complex data analysis. In addition, more details of comparison with related and contemporaneous work are provided in Appendix A.

3 Research State Transition Evaluation

We evaluate three capabilities required to investigate a scientific question as evidence evolves: **constructing** a warranted research state from raw evidence, **advancing** an established state with newly available evidence, and **carrying** and **correcting** a state produced by the agent itself. We assess these capabilities separately through *Initial Discovery*, *Continued Investigation* and *Full Trajectory* (Figure 2). Across these settings, data are released in stages derived from the evidential dependencies of published studies. Each stage requires the agent to generate evidence through analysis, and its conclusions are assessed against what the evidence available at that stage can support.

3.1 Research States and Evaluation Settings

Research state is defined as the findings and interpretations established at a stage, together with their supporting evidence, limitations, and unresolved questions. A state transition consists of generating new evidence through analysis and determining how it supports, qualifies, or revises the prior state. Its validity therefore depends both on performing an appropriate analysis and on drawing an inference warranted by its result. Crucially, validity is stage-specific. For example, a discovery cohort established an association without supporting its generalization at the initial stage; later arrived evidence and validation can strengthen, limit, or challenge that conclusion. We therefore specify sci-

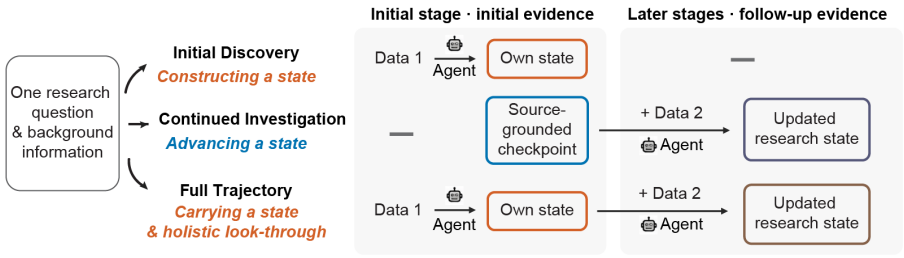

Figure 2: Illustration of state-transition evaluation design.

entific requirements including data, analysis design and empirical observations separately for each stage and freeze the agent's current report before releasing later data.

Task interface. At stage t, the agent receives $\mathcal{T}_t = \langle I_t, P_t, \mathcal{D}_t \rangle$, where I_t specifies the research question, background, and stage scope; P_t conveys prior findings, their evidence, and limitations through a research report; and $\mathcal{D}_t$ contains the data released for the current stage. The agent conducts analyses and submits a report with supporting code and artifacts. The report distinguishes inherited findings from new conclusions and states their evidence, scope, and limitations. We denote the state expressed in this report by $\widehat{\mathcal{S}}_t$. Later-stage data remain hidden until their release; in Full Trajectory Tasks, previously released data, code, and artifacts remain available as working context in addition to the prior report.

Initial Discovery. With $P_1 = \varnothing$ and a fresh workspace, the agent begins with a research question and the initial data, without a supplied finding to guide subsequent investigation. It must identify analytical objects, comparison groups, and inferential units, execute interdependent analyses, and establish findings that can support further research. An initial finding may establish only that a phenomenon exists, leaving validation or mechanism unresolved; reaching that finding can nevertheless require substantial analytical work. This setting tests whether the agent can turn a research question into analyses that establish a warranted starting state.

Continued Investigation. For a continuation stage $t > 1$, we supply $P_t = \mathcal{S}^{\mathrm{src}}_{t-1}$ through a calibrated prior report without displaying the limitation part as hints in a fresh session and workspace. Here, $\mathcal{S}^{\mathrm{src}}_{t-1}$ denotes a source-grounded checkpoint: its findings, evidential scope, and qualifications are checked against eligible predecessor evidence, and current and later target findings are excluded. All systems receive the same report for a given task, without a predecessor agent's code or artifacts. The agent must determine which prior objects, groupings, and findings remain applicable, generate evidence from the newly released data, and judge how that evidence supports, limits, or changes the prior understanding.

Full Trajectory. The agent first performs Initial Discovery, then continues with $P_t = \widehat{\mathcal{S}}_{t-1}$ in the same session and workspace. It inherits its own reports, code, and artifacts, including useful results, omissions, and unsupported interpretations. This setting tests whether discovery and continuation work together when the agent must build on its own analytical history: whether earlier findings support subsequent work, whether errors persist, and whether new evidence prompts appropriate revision. Frozen stage reports preserve earlier judgments for evaluation while allowing the agent to revise its current understanding. For corresponding continuation stages, Continued Investigation and Full Trajectory tasks release the same new data but provide different prior states and access to earlier work. Their comparison diagnoses performance under these two continuation conditions.

3.2 Source-Grounded Research Progression

Why biomedical studies? Research-state evaluation requires studies in which new evidence meaningfully changes what can be concluded about a shared scientific question. Biomedical investigations naturally provide such structure: an initial association may be followed by validation in an independent cohort, an orthogonal assay, or a different tissue or modality. These transitions change the evidential scope of earlier claims while remaining traceable through the publication, supplementary materials, code, and public data, making biomedical studies a suitable substrate for reconstructing research progression. Study selection, provenance, and data preparation are detailed in Appendix B.

Table 1: **Discovery graph relations.** Each edge type is illustrated with the same source study, in which peripheral blood from systemic sclerosis patients was profiled to study interstitial lung disease (ILD). Analysis nodes (V_A) are shown as operations and claim nodes (V_C) as quoted findings.

Relation	Edge	Example (P2. Systemic sclerosis with interstitial lung disease, SSc-ILD)
Support. The analysis provides the evidence for the claim.	$V_A \to V_C$	*Participant-level differential abundance of T-cell neighbourhoods, ILD vs. other SSc* (compare cell-type frequencies between patient groups) → *"Memory CD8 T cells are increased in SSc-ILD blood."*
Motivation. The claim raises the question that a later analysis addresses.	$V_C \to V_A$	*"CD8_TEM_T2ISG carries an IFN-γ response programme with high CXCR3/CCR5."* → *Scoring this programme in independent SSc-ILD lung data* (does the blood signal also appear in lung tissue?)
Dependency. The downstream analysis directly uses an output of the upstream analysis.	$V_A \to V_A$	*Clustering and annotation of CITE-seq profiles into T-cell subsets* (group cells and label their types) → *Differential abundance on the annotated cells* (reuses those labels)
Claim relation. One claim refines, qualifies, or extends another.	$V_C \to V_C$	*"Memory CD8 T cells are increased in SSc-ILD blood"* → *"The leading state is CD8_TEM_T2ISG, with an IFN-γ programme"* (narrows a broad population to one specific cell state)

Discovery graphs. We reconstruct each study as a discovery graph $G = (V_A \cup V_C, E)$ using Graph4Science, a toolkit supporting graph construction, source tracing, and independent node- and edge-level review. Analysis nodes V_A record scientific operations, inputs, outputs, and observations, while claim nodes V_C record findings and their evidential scope. Edges encode four types of relations: evidential support, scientific motivation, actual output dependencies between analyses, and relations between claims (Table 1). A checkpoint is anchored at a claim that closes a reportable stage; its associated findings, supporting evidence, and qualifications define the entry state for a subsequent investigation. Multiple branches may originate from the same checkpoint without depending on one another. Importantly, the graph records published evidential dependencies, not the authors' actual chronology of exploration. The graph instead makes their evidential basis traceable and constrains which data, prior findings, and target claims belong to each stage. As an external quality check, domain experts independently reviewed discovery graphs from five studies and agreed with 98% of the constructed nodes and edges. More details are in Appendix C.

Task construction. Each executable subgraph defines an entry state, an evidence boundary, and a reportable endpoint. Eligible predecessor nodes determine the calibrated prior report for Continued Investigation, which includes only the eligible predecessor state and removes downstream findings, unrelated sibling branches, and instructions for the prospective analysis. We obtain 66 tasks: 32 Initial Discovery and 34 Continued Investigation tasks; ten studies additionally support Full Trajectory evaluation. Examples of the tasks are in Appendix F.

3.3 Discovery Graph-grounded Assessment

Scientific criteria. Scientific analyses admit multiple valid implementations, so we score scientific invariants rather than reproduction of the source workflow. Each active graph node is translated into binary rubric criteria in three categories: *Scientific setup* (**S**) specifies the analysis set, data objects, populations, comparisons, time points, denominators, and independent units of inference. This category is the most stringent because changing the target or inferential unit changes the scientific question itself. *Analysis design* (**A**) specifies the scientific purposes and design safeguards required to address that question, such as metadata reconciliation, analysis at the biological unit, or a required robustness check, and equivalent algorithms and implementations are accepted. *Empirical endpoint* (**E**) specifies the resulting findings, including their object, direction, relationship, and evidential scope. Empirical endpoint criteria are assessed semantically; exact numerical agreement is required only when a value defines the study design or another binding task condition.

Assessing a transition. Because **criteria are conditioned on the current evidence boundary** rather than the source paper's final conclusions, it mitigates contamination through memorized findings. Memorized knowledge of a paper's final findings does not by itself earn credit: conclusions that require unreleased evidence are not counted as valid progress at the controlled stages.

The primary task score is the equal-weight mean of the fidelity pass rates of the three rubric categories (S, A and E). We additionally report node coverage and completion. Let V_t be the active

Table 2: **Scientific requirements and node-level capabilities.** S/A/E denote Scientific setup, Analysis design, and Empirical endpoint. Score is their equal-weight fidelity mean. Cov. and Comp. are the percentages of nodes with at least one or all criteria passing. Node metrics average nodes within tasks and tasks within each system.

Harness	Backbone	Initial Discovery						Continued Investigation					
		S	A	E	Score	Cov.	Comp.	S	A	E	Score	Cov.	Comp.
Closed-Source													
Codex	GPT-5.4	61.4	50.8	46.9	53.0	78.1	13.0	70.5	61.5	56.7	62.9	90.6	17.3
	GPT-5.6-Sol	63.9	51.5	45.5	53.6	79.9	16.8	69.7	62.1	59.7	63.8	**93.3**	25.5
Claude Code	Claude-Sonnet-5	66.1	54.1	46.5	55.6	81.4	16.8	**81.1**	**75.7**	**70.3**	**75.7**	93.0	**36.0**
	Claude-Opus-5	**77.0**	**67.1**	**63.4**	**69.2**	**90.0**	**30.9**	69.5	66.9	63.4	66.6	86.6	21.1
Open-Source													
DeepSeek	DeepSeek-V4-Flash	**54.8**	47.7	**39.5**	**47.3**	**80.3**	5.7	**65.6**	**69.4**	58.0	**64.3**	**91.1**	**17.7**
Harness	DeepSeek-V4-Pro	51.6	**48.7**	39.2	46.5	78.5	7.6	61.6	56.6	57.9	58.7	89.2	16.8
KimiCode	Kimi-K2.6	49.6	42.1	36.0	42.5	76.1	**9.3**	61.4	63.3	**58.3**	61.0	87.9	11.0

analysis nodes in task t, where an active node has at least one rubric in each of S, A, and E. Let Q_{tv} contain all rubric attached to node v, and let $f_{tvq} \in \{0, 1\}$ be the majority-voted fidelity label for criterion q. We define

$$\mathrm{Cov}_t = \frac{1}{|V_t|} \sum_{v \in V_t} \max_{q \in Q_{tv}} f_{tvq}, \quad \mathrm{Comp}_t = \frac{1}{|V_t|} \sum_{v \in V_t} \min_{q \in Q_{tv}} f_{tvq} \tag{1}$$

where we average Cov_t and Comp_t across tasks with eligible nodes, using the same task-level weighting principle as the primary score. Information recall is reported separately. Full-Trajectory stages are scored from their intermediate reports, so later revisions do not alter earlier scores.

We adopt an LLM-as-judge protocol: the judge evaluates each binary rubric criterion against sentence-indexed reports under strict instructions, with a majority vote over three runs. We use GPT-5.4 as the primary judge and re-evaluate a matched subset with DeepSeek-V4-Flash to test backbone sensitivity. The two judges agree on over 93% of decisions (Cohen's $\kappa \approx 0.87$), and yield nearly identical system rankings (Spearman's $\rho = 0.976$). To validate the LLM-as-judge protocol, a domain expert independently scored reports from three studies across all three evaluation settings for two systems (GPT-5.4 and Claude-Opus-5) using the same rubric criteria, achieving 96% agreement with the LLM judge. Full settings, and the backbone comparison and rubric number sensitivity analysis are in Appendix D.

4 Constructing, Advancing, and Revising Research States

4.1 Experiment setup

We evaluate seven frontier systems across the three research-state transition settings in Section 3.1, including Claude Code, Codex, DeepSeek Harness, and KimiCode with different LLM backbones: Claude-Opus-5 and Claude-Sonnet-5 for Claude Code; GPT-5.4 and GPT-5.6-Sol for Codex; DeepSeek-V4-Flash and DeepSeek-V4-Pro for DeepSeek Harness; and Kimi-K2.6 for KimiCode. Each agent–task stage is executed in a fresh, isolated Docker sandbox with a task-specific read-only data archive, a shared frozen public biological reference database and a separate writable workspace. Agents can autonomously use Bash, Python, and R, with 8-hour max time allocation, 32 CPU cores, 64 GB memory, and uniform GPU exposure. Direct Internet access is disabled, except for allowlisted model endpoints and package channels. Execution ends when the native runtime accepts the agent's final response, or upon failure or timeout. Further implementation and reproducibility details appear in Appendix J. Also to establish a domain-specific harness baseline and assess evaluator robustness, we also include Biomni; its detailed performance and a comparative robustness analysis against Codex are provided in Appendix I. [1]

[1] For reproducing this work, we will release the code and curated data after the double-blind review process.

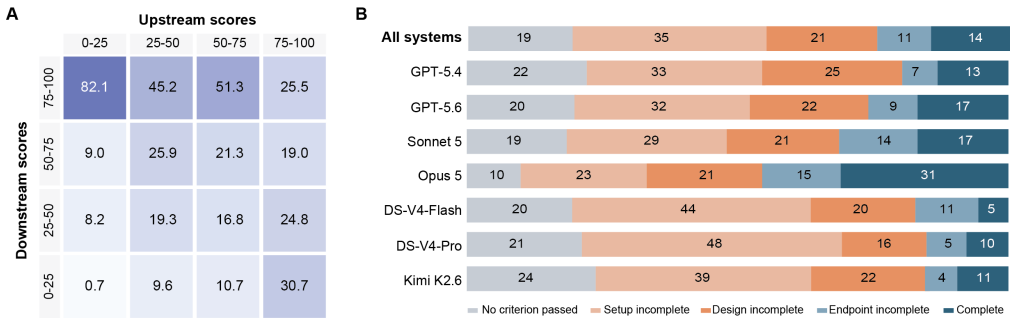

Figure 3: **Intra-task analysis of node dependency in long-horizon Initial Discovery task.** (**A**) Dependency transition between upstream and downstream analyses, showing both dependency propagation and downstream-specific failure modes. (**B**) Nodes are often covered but rarely complete. Each bar partitions S/A/E-eligible Initial Discovery analysis nodes into one mutually exclusive first-failure state under a S–A–E validity ordering.

4.2 Constructing a Research State

Systems differ substantially in their ability to construct a research state. The Initial Discovery columns in Table 2 reveal how well agents build a research state from initial raw data alone. Performance varies widely across systems: the best and worst systems differ by 26.7 points, and these differences are reasonably consistent across studies (Kendall's $W = 0.36$, split-half reliability = 0.86), suggesting a system that builds a better state on one study tends to do so on others. Within the same Claude Code harness, Opus-5 leads Sonnet-5 by 13.6 points (95% CI $[8.1, 19.2]$), indicating that the separation is not explained by scaffolding alone.

Agents often identify the relevant analyses but rarely complete them. Every system shows the same ordering across rubric categories: setup scores highest, analysis design next, and the empirical endpoint lowest (60.6, 51.7 and 45.3 on average). We count an analysis node as *covered* if at least one of its criteria passes the majority vote of three judge runs, and as *completed* if all of them pass (columns Cov. and Comp. in Table.2). The node-level diagnostics enabled by the discovery graph, reveal a sharper version of this pattern: agents cover 80.6% of analysis nodes on average but complete only 14.3%. Indeed, most analyses break down before the endpoint is reached: averaged by all tested systems, 54% of nodes fail to complete setup, and among nodes whose setup and design are both complete, 44% still miss part of the endpoint (Figure 3). Together, agents frequently enter the correct analytical path but fail to carry it through to a supported empirical conclusion.

Error propagation along dependencies We use the discovery graph to investigate the correlation between dependent analyses. When an upstream analysis scores below 0.25, the analysis that depends on it also scores below 0.25 in 82% of cases and above 0.75 in fewer than 1%. Conversely, even a strong upstream analysis does not guarantee completeness: when the upstream scores above 0.75, the dependent analysis succeeds in only 31% of cases and fails outright in 26% (Figure 3). A correct upstream analysis is therefore necessary but not sufficient for a successful downstream step.

4.3 Advancing a Supplied State

Supplying a common research state reduces stable differences between systems. The Continued-Investigation half of Table 2 measures how well agents extend a source-grounded state with newly released evidence. Under this setting, cross-system differences become less consistent: the stable component of between-system variation falls from 8.5 to 4.8 points, agreement of system rankings across tasks falls from Kendall's $W = 0.36$ to 0.09, split-half reliability drops from 0.86 to 0.50, suggesting that once that variation at constructing a research state is controlled, continuation performance becomes more task-specific.

System ranking changes once the starting state is controlled. Sonnet-5 leads every other system, overtaking Opus by 9.1 points (95% CI [1.3, 17.5]; first in 99% of bootstrap resamples). Be-

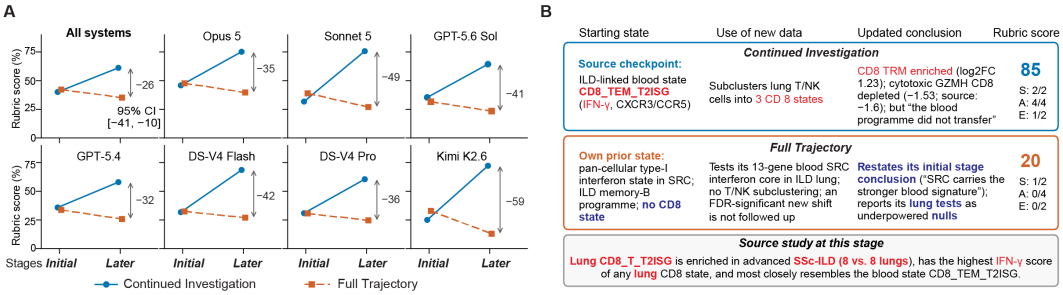

Figure 4: **Carrying its own state costs for later states analyses.** (**A**) Rubric scores at initial discovery and later stages for isolated tasks (continued investigation setting) and Full Trajectory tasks. (**B**) One example of comparison on reports generated by Opus 5 on P2 study in the indicated settings. More details are provided in Appendix F.

cause Sonnet and Opus run in the same harness, the reversal reflects the backbone rather than the scaffolding. The DeepSeek pair shows a similar, smaller shift: V4-Flash and V4-Pro are indistinguishable in initial discovery ($+0.8$ points) but Flash leads in continued investigation (+5.6, [0.5, 10.9]), whereas the two Codex backbones do not differ in either setting. This indicates that constructing a research state and advancing an established one are related but distinct capabilities: a system that is strongest at discovery is not necessarily strongest at continuation.

A grounded starting state does not remove the difficulty of reaching a correct endpoint. Continued Investigation preserves the same ordering from scientific setup to analysis design to empirical endpoint, with average scores of 68.5, 65.1 and 60.6. Even among runs in which every setup and design criterion passes, a third (21/63) still miss part of the empirical endpoint. This reveals that supplying a valid prior state does not guarantee that agents can convert an appropriate analysis into a supported scientific conclusion.

4.4 Carrying and Revising One's Own State

A self-built state imposes a large downstream cost. Full Trajectory task runs all stages of a study in a single continuous session. Each agent submits an intermediate report before new data are released and, after the last stage, a final report that must answer the full research question using all available evidence. Every report is scored with the same rubrics as the isolated tasks, so later-stage reports compare directly with continued-investigation runs on identical tasks.

Figure 4A shows the initial stage is most reproducible to Initial Discovery (r = 0.93), suggesting differences at later stages do not stem from different starting performance. The difference shows at later stages: on identical continuation tasks scored with identical rubrics, agents' performance drops from 61.1% with the source-derived checkpoint to 35.2% with its own earlier state (-26.0 points, 95% CI $[-40.6, -10.1]$). Only five studies were completed by all systems; the others had early terminations, sometimes triggered by biology-content filters. These paired studies results shows that: own-state scores are lower in 94%, and every system loses between 32 and 59 points (-41.6 pooled) (Also see Appendix G). Even the smallest loss of the best-performed system under this setting exceeds the entire best-worst gap between systems in either isolated setting (Table 2). This suggests that downstream success depends not only on whether an agent can analyze newly released evidence, but also on the quality of the scientific state it has accumulated beforehand.

The cost is not explained by long conversation history. We next examine whether a fresh start session affects agents' performance, as a natural alternative explanation is that continuous full trajectory runs simply suffer from longer context. In the paired 14 branches, one condition retains the preceding conversation, the other starts a fresh session, while both run independently and resume from the same frozen workspace at the stage boundary, including the agent's report, code and intermediate artifacts. The average performance is unchanged (36.0 vs. 36.0; an aggregate difference of approximately zero), despite variation across individual branches. Removing the conversation history thus has no consistent effect: the cost resides in what the agent has built (i.e., its conclusions, code and artifacts), rather than in the conversation length.

Table 3: **Downstream fidelity depends on upstream state, and first-stage analyses are carried forward almost unchanged. Left:** Downstream fidelity by upstream state; values are averaged over continuation analysis nodes. **Right:** State of each initial-discovery analysis node in intermediate (rows) and final (columns) reports of full trajectory runs, as row percentages. A node is complete if all criteria pass, partial if some do, and not addressed if none do. Nodes are pooled over 59 system–study pairs; duplicate runs are weighted by half.

Upstream state	Source-derived	Own state	Own state, fresh session
At least one parent is not covered	47.2	18.4	20.1
Every required parent covered	54.6	50.2	47.3

		Final report		
Intermediate report	Nodes	Not addressed	Partial	Complete
Not addressed	112	**95.5**	4.5	0.0
Partial	184	13.0	**84.5**	2.4
Complete	38	2.6	19.7	**77.6**

The cost comes from missing upstream analyses. We use the discovery graph to identify the dependencies of later-stage analyses that are built on first-stage analyses. We classify each later-stage analysis by whether the agent's own first-stage report left any of these predecessors entirely unaddressed (no criterion passing), and compare the same analyses across starting states (Table 3). When every predecessor was addressed, carrying its own state costs an agent little: downstream fidelity is 50.2% from its own state against 54.6% from the source checkpoint. When a predecessor was missing, the same analyses fall from 47.2% to 18.4%. The source-checkpoint column shows that these analyses are only slightly harder in themselves; the loss arises because the agent builds on an incomplete state, and a fresh session does not recover it (20.1%). The pattern holds for every system: on criteria that explicitly require inherited first-stage results, own-state performance is lower than with the source checkpoint for all seven systems (median difference -38.7 points). among analyses whose predecessors were addressed, predecessor fidelity is only weakly related to downstream performance ($r = 0.19$). The cost of a self-built state is thus concentrated in analyses whose foundations the agent failed to lay.

Revision rarely happened after state transition. We score each run's intermediate and final reports against the initial-discovery rubric to investigate what agents carry forward and what they correct (Table 3). Across 59 system-study pairs, final reports retained 86% of the criteria that had passed at their intermediate reports but recover only 2.8% of those that had failed. At the level of analyses, two-thirds (68%) received exactly the same credit in both reports; of those left incomplete, 1.5% were completed by the end, and of those not addressed at all, 4.5% were addressed later. Each system repairs its failed criteria between 0.8% and 8.0%, with Opus doing the most correction (Appendix H). Losses are partly omissions: 37% of the lost credit belongs to analyses absent from the final report altogether, so we do not interpret losses as reversals. Although execution traces show substantial within-stage refinement such as code editing, no attempt to reopen first-stage analyses at the later stage was observed. Two exemplary reports generated by Opus-5 on one study under these two settings show what this looks like (Figure 4B): the missed cell types in its own state was never examined at the later stage, and the final reports just repeat its first-stage conclusion.

5 Conclusion

In this work, we reframe the evaluation of scientific agents around research states: how they construct, advance, and revise a research state as evidence evolves. Source-grounded discovery graphs enable us to evaluate these transitions against stage-specific evidential constraints, while separating state construction from continuation and self-carried investigation. Across seven frontier systems, a consistent picture emerges. Agents often identify relevant analyses but rarely complete the full chain from scientific setup to a supported empirical endpoint. Supplying a calibrated research state reduces stable differences between systems, showing that much of their differentiation arises while constructing scientific understanding from raw evidence. More importantly, when agents must continue from their own earlier state, downstream performance drops sharply; this gap is not explained by longer conversation history, and previously failed criteria are rarely recovered. These results expose a limitation that static evaluation misses: current scientific agents are much better at moving an investigation forward than at revisiting and correcting the state on which that investigation depends. Reliable scientific agents will need not only stronger analysis, but the ability to recognize when new evidence should send them back.

AI USE STATEMENT

We used generative AI tools to improve the writing, fix grammatical errors, refine table formatting, and code debugging that does not involve the scientific content of the work. We did not use them to generate research ideas or design experiments. The authors reviewed every AI-assisted edit and take full responsibility for the paper's content.

REPRODUCIBILITY STATEMENT

To support reproducibility, we will release the code, curated data, and Graph4Science toolkit after the double-blind review process. The manuscript documents the research-state formulation, task interface, and evaluation settings in Section 3.1. Discovery graph construction, verification, and task derivation are described in Section 3.2 and Appendix C. Study selection, data provenance, and benchmark inventory appear in Appendix B. Rubric categories, scoring, node-level metrics, and the judge protocol and expert validation are specified in Section 3.3. Judge-backbone and rubric-count sensitivity analyses appear in Appendix D. Agent harnesses, backbones, runtime versions, sandbox configuration, resource limits, and offline references are detailed in Section 4 and Appendix J. Prior-report construction and task prompts are described in Appendices J.2 and J.3. A worked case study appears in Appendix F; Full Trajectory results, credit carry-over, and the domain-specific harness comparison appear in Appendices G, H, and I.

ETHICS STATEMENT

We introduced an evaluation framework for measuring whether scientific agents can construct, advance, and revise research states as evidence evolves. All tasks are derived from peer-reviewed biomedical studies and use only data deposited in public repositories (e.g., GEO), together with the studies' supplementary materials and publicly released code. We do not collect new human-subject data or use proprietary or controlled-access resources. The human-derived data we use were de-identified and released by the original studies under their respective ethical approvals and data-use terms, and we use them only for their intended research purposes. Our tasks assess scientific reasoning over existing studies. They do not produce clinical recommendations, diagnoses, or therapeutic interventions. We caution that strong performance in our evaluation should not be taken as evidence that an agent is reliable, safe, or valid for real-world biomedical decision-making, and agent-generated conclusions still require expert scrutiny. We hope this work supports more transparent and trustworthy assessment of AI systems for scientific discovery.

REFERENCES

Samuel Alber, Bowen Chen, Eric Sun, Alina Isakova, Aaron J Wilk, and James Zou. Cellvoyager: Ai compbio agent generates new insights by autonomously analyzing biological data. *Nature Methods*, 23(4):749–759, 2026.

Jonathan Bragg, Mike D'Arcy, Nishant Balepur, Dan Bareket, Bhavana Dalvi Mishra, Sergey Feldman, Dany Haddad, Jena Hwang, Peter Jansen, Varsha Kishore, et al. Astabench: Rigorous benchmarking of ai agents with a scientific research suite. In *International conference on learning representations*, volume 2026, 2026.

Yanyu Chen, Jiyue Jiang, Jiahong Liu, Yifei Zhang, Xiao Guo, and Irwin King. Trace: Trajectory-aware comprehensive evaluation for deep research agents. In *Proceedings of the ACM Web Conference 2026*, pp. 2524–2534, 2026.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, volume 2025, 2025.

Ian Diks, Harihara Muralidharan, Tim Proctor, and Kenny Workman. Verifiable benchmarking of long-horizon spatial biology. *arXiv preprint arXiv:2605.28065*, 2026a.

Ian Diks, Zhen Yang, Arjun Banerjee, Tim Proctor, and Kenny Workman. scbench-long: Verifiable benchmarking of long-horizon single-cell biology. *arXiv preprint arXiv:2606.26563*, 2026b.

Yao Dong, Xinglin Xiao, Liwei Dong, Xinlong Jin, Zhengbo Li, Heng Zhang, Duyun Wang, and Nan Xu. S1-deepresearch: Beyond search, toward real-world long-horizon research agents. *arXiv preprint arXiv:2606.15367*, 2026.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Licheng Zhang, Xiaorui Wang, and Zhendong Mao. Deep-research bench: A comprehensive benchmark for deep research agents. In *International Conference on Learning Representations*, volume 2026, pp. 42414–42448, 2026.

Yanlin Fei, Nazhou Liu, Xinmiao Yu, Shaolong Chen, Lei Li, Rahul Thapa, Madalina Ciobanu, Qingqing Mao, and Ritankar Das. How do agents fail on autoresearch: End-to-end diagnostic evaluation on 100 real-world frontier research tasks. *arXiv preprint arXiv:2608.14905*, 2026.

Shanghua Gao, Ada Fang, Yepeng Huang, Valentina Giunchiglia, Ayush Noori, Jonathan Richard Schwarz, Yasha Ektefaie, Jovana Kondic, and Marinka Zitnik. Empowering biomedical discovery with ai agents. *Cell*, 187(22):6125–6151, 2024.

Ali Essam Ghareeb, Benjamin Chang, Ludovico Mitchener, Angela Yiu, Caralyn J Szostkiewicz, Dmytro Shved, Gavin J Gyimesi, Jon M Laurent, Samantha M Wright, Muhammed T Razzak, et al. A multi-agent system for automating scientific discovery. *Nature*, pp. 1–3, 2026.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, et al. Accelerating scientific discovery with co-scientist. *Nature*, pp. 1–3, 2026.

Ken Gu, Ruoxi Shang, Ruien Jiang, Keying Kuang, Richard-John Lin, Donghe Lyu, Yue Mao, Youran Pan, Teng Wu, Jiaqian Yu, et al. Blade: Benchmarking language model agents for data-driven science. In *Findings of the association for computational linguistics: EMNLP 2024*, pp. 13936–13971, 2024.

Wenbin Guo, Minzhe Zhang, Bowei Han, Youjia Ma, Yang Leng, Shishir Hebbar, Xiaoyuan Zhou, Wenhao Gu, Xiao Yang, and Shashi Dhar. Promptbio-bench: benchmarking llm-based bioinformatics agents for end-to-end data analysis. *bioRxiv*, pp. 2026–05, 2026.

Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, Ryan Li, Yusuf Roohani, Lin Qiu, Shiyi Cao, Gavin Li, et al. Autonomous biomedical research with an artificial intelligence agent. *Science*, pp. eadz4351, 2026.

Shreya Johri, Erica Pimenta, Josephine Yates, Jingxin Fu, Erik L Bao, Hyeji Jun, Brendan Reardon, Sasha Bacot, Maha Shady, Doris Fu, et al. Evaluating agentic ai for biological discovery in autonomous and copilot settings. *bioRxiv*, pp. 2026–06, 2026.

Zane Koch, Asmamaw T Wassie, Javier Valdes-Aleman, Jason Lee, Michaela M Hinks, Samuel G Rodriques, Andrew D White, and Jon M Laurent. Bixbench3: Benchmarking ai agents on research-study-scale computational biology tasks. *arXiv preprint arXiv:2608.25286*, 2026.

Jon M Laurent, Joseph D Janizek, Michael Ruzo, Michaela M Hinks, Michael J Hammerling, Siddharth Narayanan, Manvitha Ponnapati, Andrew D White, and Samuel G Rodriques. Lab-bench: Measuring capabilities of language models for biology research. *arXiv preprint arXiv:2407.10362*, 2024.

Binglan Li, Anil Kumar Saini, Jose Guadalupe Hernandez, and Jason H Moore. Agentic ai and the rise of in silico team science in biomedical research. *Nature Biotechnology*, 44(5):711–725, 2026a.

Jeremy Li, Suyash Shringarpure, Edmund Wong, and Andrew Ho. Genebench-pro: Evaluating multistage statistical reasoning in genomics, quantitative biology, and translational biomedicine. *bioRxiv*, pp. 2026–06, 2026b.

Yiwei Li, Wanli Yang, Hexiang Tan, Xiangzhou Huang, Zhengyu Chen, Ziran Li, Borun Chen, Shanglin Lei, Huaisheng Zhu, Hao Tian, et al. Beyond final scores: A systematic evaluation of agents for long-horizon ai research and development. *arXiv preprint arXiv:2608.13417*, 2026c.

Bowen Liu, Shuo Nie, Bodong Du, and Xiaomeng Li. Scirigor: Evaluating open-ended scientific analysis beyond final scores. *arXiv preprint arXiv:2609.06192*, 2026a.

Qi Liu, Jiaxin Mao, Fengbin Zhu, and Tat-Seng Chua. Diagnosing search behavior and failure modes in long-horizon search agents. *arXiv preprint arXiv:2608.01913*, 2026b.

Tianyu Liu, Allen Xin Wang, Antonia Panescu, Lisa Xinyi Chen, Wenxin Long, Xinyu Wei, Yueqian Jing, Ziyao Zeng, Jihang Chen, Sihan Jiang, et al. Benchmarking ai agents for addressing scientific challenges across scales. *arXiv preprint arXiv:2606.12736*, 2026c.

Erpai Luo, Jinmeng Jia, Yifan Xiong, Xiangyu Li, Xiaobo Guo, Baoqi Yu, Minsheng Hao, Lei Wei, and Xuegong Zhang. Benchmarking ai scientists for omics data–driven biological discovery. *Bioinformatics*, 42, 2026a.

Hanwen Luo, Liliang Liu, Zhongyu Xing, Xinlu Li, Xuerui Zhang, Wei Du, Bin Liu, Jun Wang, and Guoxian Yu. Stanalyzer: Transparent spatial transcriptomics analysis via an agentic architecture. *bioRxiv*, pp. 2026–04, 2026b.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. Discoverybench: Towards data-driven discovery with large language models. In *International Conference on Learning Representations*, volume 2025, pp. 4556–4579, 2025.

Henry E Miller, Matthew Greenig, Benjamin Tenmann, and Bo Wang. Bioml-bench: Evaluation of ai agents for end-to-end biomedical ml. *bioRxiv*, pp. 2025–09, 2025.

Ludovico Mitchener, Jon M Laurent, Alex Andonian, Benjamin Tenmann, Siddharth Narayanan, Geemi P Wellawatte, Andrew White, Lorenzo Sani, and Samuel G Rodriques. Bixbench: a comprehensive benchmark for llm-based agents in computational biology. *arXiv preprint arXiv:2503.00096*, 2025.

Harsha Nori, Mayank Daswani, Christopher Kelly, Scott Lundberg, Marco Tulio Ribeiro, Marc Wilson, Xiaoxuan Liu, Viknesh Sounderajah, Jonathan Carlson, Matthew P Lungren, et al. Sequential diagnosis with language models. *arXiv preprint arXiv:2506.22405*, 2025.

Kyle O'Connell. Claroai-bench: Evaluating agentic scientific reproducibility on real biomedical papers. *bioRxiv*, pp. 2026–05, 2026.

Siba Smarak Panigrahi, Jovana Videnović, and Maria Brbic. Heurekabench: A benchmarking framework for ai co-scientist. In *International Conference on Learning Representations*, volume 2026, pp. 11029–11061, 2026.

Yuanhao Qu, Yingzhou Lu, Xinming Tu, Serena Zhang, Tianwei She, Alexander Glenn Shaw, Jou-Ho Shih, Bingqing Zhao, Minjie Shen, Haochen Yang, et al. Biomnibench: Process-level evaluation of llm agents for real-world biomedical research. *bioRxiv*, pp. 2026–05, 2026.

Zhenjiang Ren, Ruiji Li, Xujing Zhang, Ziliang Pang, Shuo Ren, and Jiajun Zhang. Evigraph: Evidence-guided autonomous research agents. *arXiv preprint arXiv:2608.04738*, 2026.

Manasi Sharma, Chen Bo Calvin Zhang, Chaithanya Bandi, Clinton Wang, Ankit Aich, Huy Nghiem, Tahseen Rabbani, Ye Htet, Brian Jang, Sumana Basu, et al. Researchrubrics: A benchmark of prompts and rubrics for evaluating deep research agents. In *International Conference on Learning Representations*, volume 2026, pp. 90447–90472, 2026.

Zachary S Siegel, Sayash Kapoor, Nitya Nadgir, Benedikt Stroebl, and Arvind Narayanan. Corebench: Fostering the credibility of published research through a computational reproducibility agent benchmark. *arXiv preprint arXiv:2409.11363*, 2024.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. Paperbench: Evaluating ai's ability to replicate ai research. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 56843–56873. PMLR, 2025.

Yubo Sun, Chunyi Peng, Yukun Yan, Zhenghao Liu, Sen Mei, Bangrui Xu, Xuanhe Zhou, Chi Chen, and Maosong Sun. Hievidr-bench: A benchmark for hierarchical evidence aggregation in deep research. *arXiv preprint arXiv:2607.25151*, 2026.

Hanchen Wang, Yichun He, Paula P Coelho, Matthew Bucci, Abbas Nazir, Bob Chen, Linh T Trinh, Serena Zhang, Ziyu Lu, Kexin Huang, et al. Spatialagent: an autonomous ai agent for spatial biology. *bioRxiv*, pp. 2025–04, 2025a.

Zhen Wang, Fan Bai, Zhongyan Luo, Jinyan Su, Kaiser Sun, Xinle Yu, Jieyuan Liu, Kun Zhou, Claire Cardie, Mark Dredze, Zhiting Hu, and Eric P. Xing. Fire-bench: Evaluating ai agents on the rediscovery of scientific insights. *arXiv preprint arXiv:2602.02905*, 2026a.

Zifeng Wang, Benjamin Danek, and Jimeng Sun. Biodsa-1k: Benchmarking data science agents for biomedical research. *arXiv preprint arXiv:2505.16100*, 2025b.

Zifeng Wang, Zheng Chen, Ziwei Yang, Xuan Wang, Qiao Jin, Yifan Peng, Zhiyong Lu, and Jimeng Sun. Empowering biomedical evidence exploration and synthesis with deep knowledge graph research. *Nature Machine Intelligence*, pp. 1–15, 2026b.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

Bryan Wilie, Samuel Cahyawijaya, Etsuko Ishii, Junxian He, and Pascale Fung. Belief revision: The adaptability of large language models reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10480–10496, 2024.

Zhibo Yang, Chen Zhang, Yuewei Zhang, and Hao Wang. Truthinsightbench: An evidence-grounded benchmark for automated evaluation of open-ended scientific discovery agents. *arXiv preprint arXiv:2609.05079*, 2026.

Zhongyue Zhang, Zijie Qiu, Yingcheng Wu, Shuya Li, Dingyan Wang, Yong Liu, Zhuomin Zhou, Yusong Hu, Yuhan Chen, Duo An, et al. Origene: a self-evolving virtual disease biologist automating therapeutic target discovery. *BioRxiv*, pp. 2025–06, 2025.

Junwei Zhou, Zhen Sun, Binyu Li, Jiangyu Zhou, Yuexi Pan, Hengyu Wang, Honghe Ren, Xiaohan Jia, Xueyang Zhou, Xiaoyu Cao, et al. Asi-bench: At the dawn of artificial superintelligence. *arXiv preprint arXiv:2608.17271*, 2026.

A Comparison with Related Work

Complementing the discussion in Related Work section, here we provide a detailed comparison with related and contemporaneous work in terms of evidence generation, claim-level scoring, dependency and process structure, and progressive evidence evaluation, as shown in Figure 4.

Table 4: Comparison with selected benchmarks for scientific and research agents. **Agent-derived evidence**: agents execute data analyses or experiments to produce empirical outputs. **Claim-level scoring**: reported findings are evaluated against reference findings or supporting evidence; △ denotes limited answer-, result-, or rubric-level coverage, and artifact or numerical-output matching alone does not qualify. **Dependency/process structure**: the reported structure used to organize analyses, evidence, or assessment; workflow stages and rubric hierarchies are distinguished from explicit dependency graphs. **Progressive evidence evaluation**: controlled releases of new data across research stages are combined with stage-specific criteria for the conclusions supported by the available evidence, with later data excluded from earlier scored reports. ✓ = satisfies the stated criterion; △ = limited coverage; ✗ (or – for structure) = not established in the compared protocol. In the final column, ✗ concerns the combined protocol and does not exclude multistep analysis, intermediate feedback, or sequential information gathering.

	Agent-derived evidence	Claim-level scoring	Dependency/process structure	Progressive evidence evaluation
Knowledge and analytical tasks				
LAB-Bench (Laurent et al., 2024)	✗	✗	–	✗
BixBench (Mitchener et al., 2025)	✓	△	–	✗
BioML-bench (Miller et al., 2025)	✓	✗	–	✗
ScienceAgentBench (Chen et al., 2025)	✓	✗	Workflow milestones	✗
BixBench3 (Koch et al., 2026)	✓	✗[c]	Artifact DAG	✗
Reproduction				
PaperBench (Starace et al., 2025)	✓	△	Rubric tree	✗
CORE-Bench (Siegel et al., 2024)	✓[a]	✗	–	✗
ClaroAI-Bench (O'Connell, 2026)	✓[a]	△	Reproduction stages	✗
Inferential decisions within an analysis				
GeneBench-Pro (Li et al., 2026b)	✓[b]	✗	Decision chains	✗[b]
BLADE (Gu et al., 2024)	✓	✗	Data-flow graph	✗
DiscoveryBench (Majumder et al., 2025)	✓	✓[d]	Hypothesis decomposition	✗
BioDSA-1K (Wang et al., 2025b)	✓	✓	Hypothesis–evidence links	✗
Claim–evidence evaluation within a run				
BiomniBench (Qu et al., 2026)	✓	✓	Decision-point rubrics	✗
FIRE-Bench (Wang et al., 2026a)	✓	✓[d]	Research-problem tree	✗
SciRigor (Liu et al., 2026a)	✓	✓	Typed evidence graph	✗
Open-ended discovery questions				
HeurekaBench (Panigrahi et al., 2026)	✓	✓[d]	Insight–code links	✗
TruthInsightBench (Yang et al., 2026)	✓	✓	Data-to-claim chain	✗[e]
Information seeking and deep research				
BrowseComp (Wei et al., 2025)	✗	✗	–	✗
DeepResearch Bench (Du et al., 2026)	✗	✓[d]	Statement–URL links	✗
ResearchRubrics (Sharma et al., 2026)	✗	△	Task-specific rubrics	✗
Ours	✓	✓	Analysis–claim DAG	✓

[a] Execution-based settings: CORE-Bench Medium/Hard and ClaroAI-Bench computational D5; CORE-Bench Easy supplies outputs, and metadata-only audits do not require execution. [b] GeneBench-Pro requires analysis of simulated data; it grades final target fields, not free-text reasoning. Its "staged files" are mounted at initialization, and auxiliary stage-level scoring is future work. [c] BixBench3 calibrates artifact thresholds to biological meaning but scores artifact agreement, not reported claims. [d] DiscoveryBench, FIRE-Bench, and HeurekaBench match findings to references; DeepResearch Bench's FACT checks statement–source support. These checks do not by themselves verify an agent's computational evidence chain. [e] TruthInsightBench uses single-phase tasks and identifies delayed validation-data release as future work.

B Data Curation and Benchmark Inventory

This section documents the data-curation decisions underlying the benchmark and inventories the selected source studies. We report the screening funnel, task coverage, graph sizes, and source-archive scale; the main paper describes the task protocol and evaluation.

Construction turns published studies into executable research transitions by recovering their analytical inputs, selecting a supported scope, and controlling which evidence is available at each stage. Table 5 summarizes the evaluated collection; Table 6 lists its source studies and task coverage.

Area	Releases	Share (%)
Cancer	23	34.8
Tissue	17	25.8
Immune	13	19.7
Infectious	13	19.7
Total	66	100.0

Collection statistic	Value
Expert-nominated candidate studies screened	53
Selected-in benchmark paper packages	32
Initial / continued investigation tasks	32 / 34
Studies supporting Full-Trajectory	10
Expanded data payload (lower bound)	285.38 GB
Expanded leaf files	17,082

Table 5: **Collection overview. Left:** Task distribution by biomedical domain; percentages are of all 66 task releases. **Right:** Summary statistics of the curated collection.

For each source study, we collect the datasets deposited in public repositories such as GEO, ArrayExpress, and TCGA/GDC, together with the accompanying supplementary files and released author code if available. After manually cross-checking sample details and metadata across sources, we only keep studies that can be fully analyzed using public data.

For each selected study, we construct a data package for its task release. The release contains source data from the original papers. We remove all author-generated results that could leak the conclusion or analysis design, such as final annotations, derived scores, statistical summaries, or answer-bearing figure data. We strictly prevent data leakage to ensure that the tasks accurately evaluate the agent's analytical reasoning rather than mere information retrieval.

We apply only the preparation necessary to make the deposited data usable, such as unpacking archives, repairing malformed tables, and restoring mappings explicitly documented by the source. Otherwise, we preserve the native file formats, directory organization, and analysis-relevant metadata irregularities. Agents must still identify the appropriate samples, reconcile identifiers, define comparisons, and select suitable analyses. Consequently, we exclude any candidate task if missing data or metadata renders its intended comparison indeterminate. Ultimately, 32 source papers were retained from the initial pool of 53 candidate studies, yielding a total of 285.38 GB of data.

B.1 Study Coverage and Data Scale

The inventory reports source graph nodes and edges alongside the availability of initial discovery and continued investigation tasks. Graph size describes the reconstructed source study, not the number of scored targets. Study numbers in this inventory identify the 32 selected source studies. Across 66 tasks, recursively expanded logical data payloads total at least 285.38 GB, comprising 17,082 identifiable leaf files.

Table 6: Source studies and task coverage. Topic descriptions link to the publications. $|V|$ and $|E|$ denote source graph nodes and edges. I / C indicates Initial Discovery / Continued Investigation tasks (✓: one task; ✓✓: two continued-investigation tasks; ×: absent). A dagger marks Full-Trajectory studies.

ID	Study topic	$\lvert V\rvert$	$\lvert E\rvert$	I / C
P1	Myeloma immune microenvironment	15	12	✓/✓
P2	Systemic sclerosis immune heterogeneity†	39	72	✓/✓✓
P3	Bacterial transcriptomics with PETRI seq	17	10	✓/✓
P4	Colorectal cancer and liver metastasis†	26	33	✓/✓✓

Continued on next page

Table 6 (continued)

ID	Study topic	$\lvert V\rvert$	$\lvert E\rvert$	I / C
P5	Mouse pancreatic beta cell atlas	53	109	✓/✓
P6	T cell exhaustion in thyroid cancer†	18	29	✓/✓
P7	Prognostic T cell signature in colorectal cancer†	23	23	✓/✓✓
P8	Myeloid cells and breast cancer metastasis	34	20	✓/✓✓
P9	Cellular communities in chronic obstructive pulmonary disease	31	16	✓/✓
P10	Macrophage reprogramming under hypoxia†	13	16	✓/✓✓
P11	Cervical organoids during Chlamydia infection†	22	36	✓/✓✓
P12	Resistance to checkpoint blockade in melanoma	26	21	✓/✓
P13	CD39 in colorectal cancer and liver metastasis†	34	28	✓/✓
P14	Interferon epsilon in ovarian cancer	28	29	✓/×
P15	Serum metabolomics of pulmonary nodules	37	36	✓/✓
P16	Replication of tuberculosis gene signatures	12	10	✓/×
P17	Striatal transcriptomics in opioid use disorder†	20	25	✓/✓
P18	Follicular remodeling in the adult ovary	21	11	✓/×
P19	Oligodendrocyte heterogeneity in multiple sclerosis	34	34	✓/✓
P20	Cellular modules in Crohn's disease	16	16	✓/✓
P21	Human retinal cell atlas	28	32	✓/✓
P22	Immune zonation of the human kidney	41	41	✓/✓
P23	Human T cells across tissues and activation states	25	21	✓/✓
P24	Progenitors and differentiation in the human thymus†	34	29	✓/✓
P25	Immune responses in COVID 19 and influenza	24	27	✓/✓
P26	Lung cell atlas of lethal COVID 19	32	32	✓/×
P27	Epidermal differentiation across skin sites	33	31	✓/✓
P28	Growth conditions and bacterial transcriptomes	15	20	✓/✓✓
P29	Microbial transcriptomics with combinatorial barcoding	32	34	✓/×
P30	Alveolar niche in chronic obstructive pulmonary disease	37	29	✓/✓
P31	Lung cell populations in pulmonary fibrosis	21	16	✓/✓
P32	Gut development and childhood Crohn's disease	29	39	✓/✓
Total		870	937	

C Discovery Graph

C.1 Definition

The graph separates scientific operations from the propositions they establish. An analysis node records its inputs, analytical operation, comparisons where applicable, output objects, and reported observations. A claim node records a proposition together with its scope and qualifications. This distinction allows several analyses to support a claim and a finding to inform multiple subsequent investigations. Source locators link the recorded content to the article or supplementary evidence.

We define a scientific discovery graph is a typed directed acyclic graph (DAG)

$$G = (V, E), \qquad V = V_C \cup V_A, \qquad V_C \cap V_A = \varnothing.$$

Here, V_C is the set of scientific claim nodes, and V_A is the set of analysis or experiment nodes. A central research question q is task metadata, rather than a node in G.

The edge set $E \subseteq V \times V$ naturally partitions into four types based on the endpoint sets V_C and V_A:

$$E \subseteq (V_C \cup V_A) \times (V_C \cup V_A) = (V_C \times V_C) \cup (V_C \times V_A) \cup (V_A \times V_C) \cup (V_A \times V_A).$$

Their meanings are as follows:

- $c_i \to c_j \in E_{CC}$: a claim-level relation. The earlier claim supports, refines, bounds, challenges, or composes the later claim.
- $c_i \to a_j \in E_{CA}$: a rationale relation. The claim c_i provides a scientific reason for performing action a_j. This edge may carry a free-text rationale.

- $a_i \to c_j \in E_{AC}$: an evidence relation. The artifact y_i produced by action a_i is used as evidence for claim c_j. This relation records evidential provenance; it does not make y_i a graph node or imply that c_j is true.
- $a_i \to a_j \in E_{AA}$: an action-dependency relation. Action a_i enables, parameterizes, selects, or composes with action a_j.

For a single task, let $c_0 \in V_C$ be a distinguished initial-context statement that summarizes the task-start knowledge $V_{C,0}$. It need not answer q and is not a checkpoint. Let $V_C^\star \subseteq V_C$ be the set of checkpoint claims. A claim is a checkpoint when it directly contributes a reportable answer to the research question and marks a local evidence closure or next-stage knowledge state:

$$c \in V_C^\star \quad \Longleftrightarrow \quad \begin{array}{l} c \text{ directly contributes a reportable answer to } q, \\ \text{and } c \text{ marks a local evidence closure or next-stage knowledge state.} \end{array}$$

Its purpose is to explain how the available answer state for q advances from the entry knowledge state to the terminal checkpoint claim or claims. Research state is defined as the findings and interpretations established at a stage, together with their supporting evidence, limitations, and unresolved questions.

C.2 Discovery Graph Construction

We introduce Graph4Science, a toolkit that be designed and developed for agents to construct and inspect candidate discovery graphs under the representation above. The toolkit can be installed in an agent's execution environment and exposes programmatic and command-line interfaces through which agents can query the reference catalogue, inspect source bundles, propose analyses, claims, relations, and bounded subgraphs, and link each proposed element to the relevant source evidence. We will release the toolkit together with the benchmark, allowing agents to independently use the same graph-construction and source-grounding workflow.

For each study, a source manifest fixes the target article and its complete supplementary files. Stable locators and content hashes are generated using the `pdf2md` library. Automated checks, implemented by designed and calibrated verifier LLMs, validate schema conformance, identifiers, source anchors, output lineage, and dependency closure. We construct paired source-faithful and jittered candidate nodes from three development studies, introduce factual, scope, or granularity errors, and test whether the verifier assigns the correct `pass`/`revise` verdict and diagnostic class using only each node's declared source closure.

We then review each candidate graph against the source bundle, including potentially omitted evidence and the scope of its claims and relations. Only the reviewed graph is used to define checkpoints, data releases, and rubrics; its source bundle and construction record are frozen with the benchmark.

Table 7: **Verifier check.** Each case is a candidate node with its declared source closure. Verdict denotes binary pass/revise agreement; class denotes exact diagnostic agreement.

Study	Cases	Pass/revise	Diagnosis
P2 (held-out)	14	13/14	12/14
P28	14	13/14	13/14
P17	16	15/16	15/16

D Robustness and Sensitivity Analysis

Judge-backbone sensitivity. We assess whether the evaluation results depend on the backbone used by the LLM judge. Specifically, we compare the primary GPT-5.4 judge with DeepSeek-V4-Flash. To isolate the effect of the judge backbone, we retain only evaluation instances with an identical report hash, task identifier, and set of rubric identifiers in both runs. This produces a strictly matched subset of 253 reports spanning 36 tasks and eight evaluated systems, including the additional Biomni baseline beyond the seven systems in the main evaluation.

Table 8: Judge-backbone robustness on subset of 253 reports. Δ denotes DeepSeek-V4-Flash minus GPT-5.4. Pearson's r and Spearman's ρ are computed across matched reports.

Metric	GPT-5.4	DeepSeek-V4-Flash	Δ	r	ρ
Overall task score	50.11	48.97	-1.14	0.950	0.944
Information recall	48.50	47.93	-0.56	0.962	0.955
Fidelity	48.27	47.38	-0.89	0.959	0.952
Initial Discovery score	48.82	47.63	-1.19	0.946	0.934
Continued Investigation score	51.76	50.68	-1.08	0.952	0.955
Within-judge agreement	95.56	95.37	-0.20	–	–

Table 9: System-level weighted scores under the two judge backbones. Only reports in the strictly matched subset are included.

Evaluated system	N	GPT-5.4	DeepSeek-V4-Flash	Δ
Claude Code + Claude Opus 5	34	61.95	61.16	-0.79
Claude Code + Claude Sonnet 5	31	58.70	58.29	-0.41
Codex + GPT-5.4	30	53.16	52.42	-0.74
Codex + GPT-5.6	36	50.59	49.78	-0.80
DeepSeek Harness + DeepSeek-V4-Flash	32	47.44	44.34	-3.10
KimiCode + Kimi-K2.6	35	46.14	44.89	-1.25
DeepSeek Harness + DeepSeek-V4-Pro	35	41.97	40.68	-1.30
Biomni Harness + GPT-5.4	20	36.69	36.19	-0.50

Table 8 summarizes the comparison. Across the matched reports, DeepSeek-V4-Flash assigns an overall score 1.14 percentage points lower than GPT-5.4, indicating a small tendency toward stricter judgments. Nevertheless, the report-level scores remain strongly correlated, and the mean differences are similar across the two stage groups. The two judge backbones agree on 93.91% of the information-recall decisions, with Cohen's $\kappa = 0.874$, and on 93.81% of the fidelity decisions, with $\kappa = 0.872$. These results indicate strong cross-backbone agreement beyond chance.

The system-level rankings are highly consistent across judge backbones (Spearman's $\rho = 0.976$): six of the eight systems retain exactly the same rank, with only the fifth- and sixth-ranked systems exchanging positions.

E Nodes and Rubrics number sensitivity

We abbreviate Initial Discovery and Continued Investigation as ID and CI, respectively in this section. To assess whether the ID–CI performance gap is driven primarily by task size, we examined papers for which the numbers of rubrics and analysis nodes were closely matched. Among the four papers with exactly matched rubric counts, CI outperformed ID in all four cases, with an average advantage of 10.9 percentage points. This suggests that the observed ID disadvantage cannot be explained solely by rubric-count imbalance. Table 10 summarizes ten papers with matched or near-matched rubric counts. Positive values of $\Delta(\mathrm{CI} - \mathrm{ID})$ indicate higher CI performance.

The exact-match subset provides the clearest evidence: CI exceeds ID in all four papers with identical rubric counts. The mixed results among near-matched papers further suggest that dependency structure and content-specific requirements, rather than rubric count alone, contribute to the observed performance differences.

F Case Study: Systemic Sclerosis

Paper 2 investigates why patients with systemic sclerosis (SSc) develop distinct organ manifestations, particularly scleroderma renal crisis (SRC) and interstitial lung disease (ILD), and whether these manifestations are associated with distinct immune-cell populations or cellular states.

Paper	# Rubric ID/CI	# Nodes ID/CI	Score ID	Score CI	$\Delta(\mathrm{CI}-\mathrm{ID})$
11	19/19	3/3	36.4%	48.1%	+11.7 pp
20	8/8	1/1	73.4%	78.6%	+5.2 pp
21	8/8	2/2	81.0%	91.3%	+10.3 pp
9	5/5	1/1	81.2%	97.6%	+16.4 pp
32	15/14	3/4	42.1%	17.5%	-24.7 pp
23	22/20	4/4	43.8%	66.2%	+22.5 pp
3	8/6	1/1	26.4%	75.4%	+49.0 pp
1	7/10	1/1	80.6%	81.2%	+0.6 pp
5	10/6	3/1	97.6%	45.6%	-51.9 pp
10	32/38	4/4	51.9%	20.9%	-31.1 pp

Table 10: ID and CI performance for papers with matched or near-matched rubric counts. Positive values of $\Delta(\mathrm{CI}-\mathrm{ID})$ favor CI.

Initial discovery: peripheral blood. At the initial stage, an agent receives PBMC CITE-seq data from 21 treatment-naive SSc participants and six matched healthy donors, with paired RNA and antibody-derived tag (ADT) measurements. Four SSc participants had SRC and 17 did not. The ILD comparison is defined as 10 ILD-without-SRC participants versus the remaining 11 SSc participants. One participant, SSc-1, contributes before-onset-after samples; these are repeated observations from one individual and must not be treated as three independent patients.

The agent should first establish a PBMC reference map before making any clinical comparison. This requires cell-level QC, biologically justified RNA/ADT-based annotation, and resolution of the broad and fine immune states needed for SRC and ILD analyses. Because cells are nested within participants, the agent should perform clinical inference using participant-level proportions, pseudobulk summaries, compositional models, or other sample-aware differential-abundance methods rather than treating cells as independent clinical replicates.

The initial investigation produces two carried-forward dependency objects:

- **SRC candidate-state edge.** `CD14_EGR1`, together with `CD14_ISG` and `CD16_ISG`, is the leading SRC-associated peripheral monocyte-state hypothesis. The `CD14_EGR1` programme is expected to involve EGR1-associated inflammation and TNF/NF-κB signalling.
- **ILD state-hypothesis edge.** A memory-T-cell signal, particularly `CD8_TEM_T2ISG`, is carried forward as the ILD-associated peripheral programme, with interferon and CXCR3/CCR5-related features.

These discovery results support peripheral-blood associations only. Agents in the continued stages must retain this scope: a blood state is not, by itself, evidence of tissue localization, blood-to-organ trafficking, cellular differentiation, or causal organ injury.

Continued investigation I: ILD lung branch. The ILD continuation consumes the `CD8_TEM_T2ISG` hypothesis from the initial report, but it does not inherit the original PBMC cells, cell labels, or matched participants. The agent instead receives an independent, unpaired lung single-cell cohort. It should independently annotate lung CD8 states, compare state abundance between advanced SSc-ILD and healthy lungs using donor- or sample-aware inference, quantify interferon/T2ISG activity, and assess molecular similarity to the peripheral programme.

The agent may conclude that blood and lung states show molecular correspondence if supported by the data. It should not conclude that the states are identical, that blood cells entered the lung, or that they caused ILD.

Continued investigation II: SRC follow-up branch. The SRC continuation consumes `CD14_EGR1` as the fixed blood-derived candidate. The added R1 blood/kidney, R3 external SSc PBMC, and R4 MPA CITE-seq datasets are parallel evidence branches rather than a linear validation pipeline. The agent should:

1. characterize `CD14_EGR1` relative to other monocyte states, including its state-specific genes and inflammatory pathways;
2. evaluate the `CD14_EGR1` programme in an independent PBMC cohort, while respecting any missing SRC, ILD, or treatment annotations;
3. compare blood and kidney myeloid programmes, including possible correspondence between EGR1-high peripheral monocytes and `THBS1_Mac`-like kidney macrophages.

Agents may query the local biological reference catalogue to define gene sets, resolve gene identifiers, and obtain pathway annotations in MSigDB.

A paired blood–kidney analysis from one SRC participant can support only a limited inference of programme correspondence. It cannot prove monocyte-to-macrophage differentiation, blood-to-kidney migration, or a general SRC mechanism.

Selected rubric requirements.

The rubric evaluates whether an agent preserves these dependencies and inferential boundaries:

- **Scientific setup (S):** Correct CITE-seq cohort definition and repeated-sample handling; state definition before SRC/ILD interpretation and retention of participant provenance.
- **Analysis (A):** Cell QC and justified annotation; participant-aware SRC/ILD abundance analysis; donor-aware lung CD8 analysis; and CD14_EGR1-specific molecular characterization.
- **Empirical endpoint (E):** SRC-associated CD14_EGR1/CD14_ISG/CD16_ISG states; ILD-associated memory/CD8 states; inflammatory CD14_EGR1 programmes; and cautious blood–kidney correspondence without claiming proven differentiation.

Thus, an agent is evaluated not only on whether it names the expected cell state, but also on whether it uses the correct dependency object, analysis unit, evidence source, and strength of inference.

G FULL-TRAJECTORY RESULTS BY SYSTEM

Table 11 gives the values plotted in Figure 4.

Sensitivity to the scoring definition. Scoring each rubric item equally instead of averaging the three category pass rates leaves the later-stage gap unchanged (-26.0, 95% CI $[-40.6, -10.1]$). An association between an agent's own initial-discovery score and its later own-state score, conditioning on task difficulty, is present under category scoring ($\beta = 0.46$, 95% CI $[0.06, 0.85]$, standard errors clustered by study) but weakens under item-level scoring ($\beta = 0.38$, $[-0.06, 0.81]$); we therefore do not rely on it.

Session history on criteria that use inherited results. On the 15 later-stage nodes whose criteria explicitly require inherited first-stage results (5 studies), the fresh session never outperformed the retained session: it scored lower on 5 nodes and tied on 10, all ties at zero. Conversation history may therefore help slightly when an agent uses inherited results, although the effect is small relative to the cost of a self-built state and rests on few nodes.

H CARRY-OVER OF FIRST-STAGE CREDIT

For each Full Trajectory run we score both the intermediate report (submitted before new data are released) and the final report against the study's initial-discovery rubric, and compare them criterion by criterion within each analysis node. A criterion passes if it passes the majority vote of three judge runs. Nodes are *complete* if all their criteria pass, *partial* if some do, and *not addressed* if none do. We analyse 61 runs covering 59 system–study pairs. Gains and losses are counted within nodes, so offsetting changes inside a node are not visible and the reported gain rates are lower bounds (see Table 12).

Table 11: Full Trajectory tasks versus the Isolated tasks. Initial discovery compares the isolated run with the first stage of the Full-Trajectory run. Later stages compare the same rubric scored with the source-derived checkpoint supplied (continued investigation) and with the agent's own state (Full Trajectory). Δ = own $-$ supplied. System rows cover the five studies completed by all systems; the pooled rows cover those five and all ten. One run per system and task.

		Initial discovery		Later stages		
Harness	Backbone	Isolated	Full traj.	Supplied	Own	Δ
Claude Code	Claude Opus 5	**45.9**	**47.7**	75.2	**39.7**	-35.5
Claude Code	Claude Sonnet 5	31.7	38.9	**75.7**	27.0	-48.7
Codex	GPT-5.6 Sol	35.4	31.6	64.3	23.4	-40.9
Codex	GPT-5.4	35.8	33.7	57.9	25.8	-32.1
DeepSeek	V4 Flash	31.8	32.4	68.5	27.0	-41.6
DeepSeek	V4 Pro	30.9	30.7	60.5	24.6	-35.9
KimiCode	Kimi K2.6[†]	25.0	32.8	72.2	13.0	-59.3
Pooled, 5 common studies		33.8	35.4	67.7	26.0	-41.6
Pooled, all 10 studies		40.1	42.2	61.1	35.2	-26.0

[†]Six of seven later-stage tasks on the common studies.

Table 12: Criterion-level carry-over of initial-discovery credit by system: share of criteria passed in the intermediate and final reports, share of criteria failed in the intermediate report that pass in the final report, share of passed criteria retained, and share of analysis nodes whose credit is unchanged. Node transitions pooled over systems are in Table 3.

		Passed		Failed $\rightarrow$	Passed	Nodes
System	Studies	Intermediate	Final	passed	retained	unchanged
GPT-5.4	7	27.0	26.4	3.2	88.8	75.6
GPT-5.6-Sol	8	31.9	30.8	2.5	91.5	80.0
Claude-Sonnet-5	6	34.5	35.8	5.3	93.8	68.6
Claude-Opus-5	9	49.1	48.4	8.0	90.4	56.2
DeepSeek-V4-Flash	10	30.5	22.6	0.8	72.6	56.4
DeepSeek-V4-Pro	10	29.6	25.3	1.5	81.8	69.1
Kimi-K2.6	9	30.1	27.1	1.3	87.3	74.5
All	10	33.1	30.4	2.8	86.3	68.3

I Domain-Specific Harness Comparison

To assess whether domain-specific harnesses offer measurable benefits, we compare Codex and Biomni using GPT-5.4 on 38 shared tasks subset: 18 Initial Discovery tasks and 20 Continued Investigation tasks. Codex scores 57.9 and Biomni 45.6, a paired difference of 12.4 percentage points (95% source-study bootstrap CI $[4.3, 20.0]$).

Setting	Biomni	Codex	Difference
Initial Discovery	41.4	50.7	$+9.3$
Continued Investigation	49.3	64.4	$+15.1$
All shared tasks	45.6	57.9	$+12.4$

Table 13: GPT-5.4 harness comparison. Scores (%) equally average the S, A, and E fidelity pass rates within each task, then average tasks equally.

The observed difference applies to these evaluated configurations and tasks. It does not establish a general advantage of general-purpose over domain-specific agents.

J EXPERIMENT SETUP

We execute every agent–task pair in a fresh, isolated Docker container. Each run receives a task-specific, read-only data archive containing only the datasets allowlisted for that task, together with a separate writable workspace for analysis code, intermediate results, figures, and the final report. For follow-up tasks, the workspace additionally includes the report produced for the corresponding first-stage investigation. Task-specific frozen biological reference resources are mounted read-only when required. Each run is allocated up to 32 CPU cores and 64 GiB of memory, with a wall-clock limit of eight hours.

Direct Internet access is disabled. Model API requests and package installation are routed through an exact-host allowlist proxy, preventing general web browsing and unrestricted external data retrieval.

Table 14: Runtime distribution of successful runs. All values are in minutes.

Evaluation	n	Min	P25	Median	Mean	P75	P90	P95	Max
Initial Discovery & Continued Investigation	507	1.2	10.0	21.0	31.4	43.6	68.6	91.3	366.9
Full Trajectory	60	5.3	46.7	80.2	105.0	129.6	206.1	243.2	524.0

Table 15 and 14 reports the pinned native-agent runtimes. For Full-Trajectory tasks, each state has eight-hours runtimes individually. All agent–task pairs were run in fresh Docker containers. Inputs and the shared `universal-2022-05` biological reference catalogue were read-only; direct Internet access was disabled except for allowlisted model and package endpoints.

Table 15: Agentic frameworks, runtime versions, and LLM backbones.

Component	Version	LLM backbone(s)
Docker Engine	29.7.2	–
Claude Code	2.1.220	Claude-Opus-5; Claude-Sonnet-5
Codex	0.147.0	GPT-5.4; GPT-5.6-Sol
DeepSeek Harness	0.1.1rc1	DeepSeek-V4-Flash; DeepSeek-V4-Pro
KimiCode	0.38.0	Kimi-K2.6

We include a Full Trajectory only if its selected stages form a continuous scientific subgraph meeting three strict criteria:

(1) **Sequential Dependency**: Each subsequent stage must build upon previously established findings, outputs, or analytical objects. (2) **Linkable Evidence**: Every new release must contribute distinct evidence that can be linked to prior materials via compatible identifiers, processing conventions, or documented mappings. (3) **Input Availability**: All inputs required for later stages must either be directly available from preceding releases or be reproducible within them.

In our experiments, only ten studies support Full Trajectory evaluation. While other studies can be used for independent tasks.

J.1 DATABASE OFFLINE DEPLOYMENT

Offline reference construction. We test agents in restricted network environment, so relying on live biological databases would introduce information leakage. We therefore constructed an offline reference catalogue from fixed upstream releases. Each source was downloaded and recorded with its release date, licence, provenance, and checksum, and converted into normalized SQLite tables. All agents run share this reference tables are mounted for evaluated agents; source archives, author code, task mappings, and precomputed results remain excluded.

Agent access. The system prompt only informs agents that a public biological reference catalogue is available at agents' home root, where they are not required to read the guides, and the benchmark does not force a particular database, table, or query. The guides provide neutral descriptions and example queries, while each agent independently decides which information is useful. The same read only SQLite files can also be queried directly with Python, R, SQLite, DuckDB, or command line tools:

Database	Frozen date	Information	Paper
Gene Ontology	2019-12-09	Terms; relations; human annotations	7, 13, 18, 19, 22, 25, 27, 28, 30
Reactome	2019-12-10	Human pathways; gene members; UniProt members	5
WikiPathways	2019-06-10	Human pathways; members; cross references	–
MSigDB	2019-10-03	Hallmark; C5; descriptions; members	2, 5, 7, 8, 13, 17, 24, 27
Ensembl gene map	2019-08-30	Gene IDs; symbols; biotypes; coordinates	5, 16, 17, 19
Ensembl Compara	2019-09-26	Human–mouse; human–rhesus; mouse–rhesus orthologs	5, 8, 17, 19
HGNC	2020-07-01	Current symbols; aliases; previous symbols	16, 31
Ramilowski	2015-06-22	Ligand–receptor pairs; sources; evidence	20
CellPhoneDB	2019-09-09	Interactions; complexes; protein and gene metadata	4, 8, 22, 26, 32
CellChatDB	2022-05-07	Interactions; complexes; cofactors; gene information	6, 10, 11, 27
GEO platforms	2016-06-23	Probes; gene symbols; Entrez IDs; source rows	4, 7, 16, 25
B. subtilis 168	2019-08-24	Locus tags; symbols; descriptions; coordinates	29
E. coli K-12	2019-08-24	Locus tags; symbols; descriptions; coordinates	3, 29
P. aeruginosa PAO1	2019-08-24	Locus tags; symbols; descriptions; coordinates	28
S. Typhimurium SL1344	2019-08-24	Locus tags; symbols; descriptions; coordinates	28

Table 16: Frozen biological reference databases available in the benchmark.

```
python query_reference.py --list
python query_reference.py --database hgnc --info
python query_reference.py --database hgnc --tables
python query_reference.py --database hgnc --example
```

Temporal freeze and leakage control. The temporal cutoff applies to the reference catalogue, not to the publication years of the source studies. All mounted database snapshots predate the 2023–2025 source papers; within this pre-2023 window, each resource is fixed to the latest documented release that provides the generic information required by the benchmark. For example, CellChatDB is frozen at version 1.4.0, released on 7 May 2022, and later releases are not mounted.

The catalogue contains generic identifiers, annotations, pathways, and interaction records only. It excludes author-generated result tables, source-specific annotations, precomputed analysis outputs, and records added after the freeze that could cite, reproduce, or incorporate a source study's findings. Thus, a 2025 source paper may be evaluated with its released biomedical data while the agent's auxiliary biological knowledge remains restricted to a pre-publication database snapshot. When the required generic information is available in an earlier release, we use that release rather than a later version; post-freeze resources are not used to fill missing information.

`msigdb.geneset` and `msigdb.member` for P2 tasks, programme characterization, GPL570 and GPL4133 probe annotations for cross platform signature validation, and bacterial feature tables. These lookups provide the identifier, gene set, and species annotation information required by the corresponding analyses without supplying their expected answers, which are essential for agents to successfully reach the endpoint.

J.2 Construction Prior Research States

For continued investigation, agents begin from a checkpoint of what the study had established before the current state. For each target subgraph, we trace the graph backward to collect the analyses and claims on which that stage depends. These preceding findings define the prior research state. Later results and parallel branches that do not contribute to the target are not included.

We also collect the relevant source passages, author's procedural numerical results if supplement tables are accessible. The data used for figuring was be excluded for causing leakage. We ask LLM, GPT-5.4 to prepare a concise prior report describing the established findings, their supporting evidence, without any limitation or quantification about current state understanding. A separate manual review on the report against the selected source materials and removes statements that depend on unreleased evidence before it being used in tasks.

J.3 Task Prompts

All independent tasks use one of the task messages in Figures 5 and 6, followed by the shared reporting contract in Figure 7. Braced fields are instantiated from the task manifest, the report format requirement is the same for all tasks.

Initial-discovery task message

{`background`}

I am interested in investigating the following research question:

{`research_question`}

To address this question, I collected some data:

{`data_summary`}

Please investigate this question using the provided data and any other relevant materials available in the workspace. You may determine which analyses are most appropriate. Your report should describe only analyses that you actually performed and results supported by the data. For each main analysis, briefly state the data included, the unit of comparison, the method used, and the main result. If an intended analysis was not performed or a different method was used, state this briefly rather than implying that it was done. Do not provide only a proposed analysis plan or an inventory of available files, and do not fabricate results.

Figure 5. Initial Discovery prompt: The task-specific message is followed by the shared reporting contract in Figure 7.

Continued-investigation task message

{`background`}

I am investigating the following research question:

{`research_question`}

A current research report from the preceding stage is available at `report.md`. Treat it as prior context and distinguish its carried-forward observations from the evidence you generate in this task.

To investigate this question, I collected additional data:

{`data_summary`}

Please investigate the research question using the materials available in the workspace. The earlier report and the additional data may provide some context or evidence, but you should determine for yourself whether, how, and to what extent each bears on the question.

Figure 6. Continued Investigation prompt: continued investigation. `report.md` is a fixed source-derived checkpoint report supplied identically to every agent; no predecessor agent workspace, code, artifacts, or session is exposed.

Shared reporting contract

Save your code and supporting artifacts in the designated workspace directories, and save the final Markdown scientific report to `/workspace/final_report.md`. Use exactly the following top-level structure:

```
# Research Report
## Introduction
## Methods and Results
## Conclusion and Discussion
```

`Introduction.` Briefly restate the scientific background and research question. Introduce the reasoning behind the approach you took and explain how the available data informed the investigation. Keep this section focused on the completed work rather than presenting a prospective analysis plan.

`Methods and Results.` Write this section as a concise record of the investigation you completed, so that a reader can follow each distinct analysis from its inputs to its saved outputs.

For every main analysis, create a separate descriptive `###` subsection, such as `### Cell-type annotation` or `### Case-control abundance comparison`. Within each analysis subsection, use all four of the following `####` headings, in this order:

`#### 1. Data`
Describe the data used for this analysis, including the groups or subsets and their size. Mention important filtering, selection, or grouping decisions. Report relevant participant, sample, and cell counts when they differ. State the comparison unit, denominator, and whether the data are paired, repeated, longitudinal, or cross-sectional. Make clear when this analysis uses a result produced in an earlier analysis.

`#### 2. Analysis`
Explain what you did with this data and why. Name the actual method used when relevant, distinguishing marker-based annotation, reference mapping, unsupervised clustering, pseudobulk analysis, deconvolution, and gene-expression scoring. For each statistical comparison, state the comparison unit and denominator where applicable, together with the test or model, important covariates, multiple-testing correction, and decision threshold.

`#### 3. Findings`
Summarize what this analysis found. State the direction and strength of the evidence, including important effect sizes, p-values, adjusted p-values, or confidence intervals when available. Explicitly report null, non-significant, or opposite-direction findings when they affect the research question. Do not imply that an analysis was performed if it was not.

`#### 4. Supporting evidence`
End this subsection with an `Evidence:` sentence listing the saved tables, figures, or outputs that directly support the findings. Briefly describe what each artifact contains and identify the relevant observation, column, row, or summary value. Filenames alone are not sufficient.

`Conclusion and Discussion.` Write exactly three concise prose paragraphs in this section.

In the first paragraph, directly answer every component of the research question. Avoid incidental exact values unless they are essential to the scientific conclusion.

In the second paragraph, provide a self-contained scientific synthesis of the answer and its principal supporting evidence. Prefer one core finding per sentence and avoid unnecessarily complex sentences.

In the third paragraph, discuss the material limitations of the current evidence and the most informative next steps. State each limitation in a separate sentence, explain which conclusion it prevents or qualifies, and identify the specific data, comparison, validation, or experiment that would address it. Include any important limitation caused by an unperformed analysis or a method different from the most direct approach. Do not ask only for "more data" or "further research."

Figure 7. All tasks share the same report format, which is also applied to prior reports.