AriadneBench: Evaluating Scientific Decisions by AI Agents in Cryo-EM Workflows

Anonymous authors
Paper under double-blind review

Abstract

Recent advances in AI agents are reshaping workflows for scientific discovery. However, reliable decision-making in cryo-electron microscopy (cryo-EM) remains challenging for agents, as they must align decisions with scientific objectives and autonomously adapt to multimodal evidence. We introduce AriadneBench, the first benchmark for evaluating these capabilities in real cryo-EM workflows. It comprises 180 tasks across eight task families and 13 workflows, targeting scientific goal alignment and adaptive autonomy. Workflow snapshots preserve the information available at each decision point, while expert annotations and auditable evidence tools enable joint evaluation of answer correctness, reference evidence coverage, and operational compliance. We evaluate seven state-of-the-art agent systems and find that the strongest achieves an overall Success Rate of only 63.21%. Agents perform better at configuring operations and recommending next steps than at particle selection and failure localization. Trajectory analysis shows that even complete reference evidence coverage does not guarantee a correct decision. To investigate whether distilled knowledge from our benchmark can improve these decisions, we induce one reusable skill per task family from training trajectories and expert annotations. On 37 held-out tasks, skill guidance improves Success Rate for five of seven agents, with an average gain of 4.25 percentage points across all seven agents.

1 Introduction

In recent years, emerging AI agents have been rapidly integrated into scientific research workflows to support experiment planning, execution, and interpretation for scientific discovery acceleration (Boiko et al., 2023; Swanson et al., 2025). In cryo-electron microscopy (cryo-EM), such agents could similarly speed up biomolecular structure determination for mechanistic studies and structure-guided drug discovery (Renaud et al., 2018). However, the reliable application faces several challenges. First, processing strategies vary across datasets, requiring domain expertise for effective decision-making. Second, these decisions must be grounded in the multimodal evidence produced throughout multi-stage single particle analysis (SPA) workflows (Lyumkis, 2019). Finally, incorrect decisions can affect multiple downstream steps (Cheng et al., 2015), increasing both time and computational costs.

These challenges have long motivated automation in cryo-EM. Conventional tools such as Warp (Tegunov & Cramer, 2019) and TranSPHIRE (Stabrin et al., 2020) automate preprocessing and particle picking, reducing human intervention. More recently, general-purpose AI agents including Codex and Claude Code can generate code, run commands, and call external tools, but their reliability in cryo-EM tasks remains unclear. For example, successful execution does not guarantee a correct scientific decision (Chen et al., 2025b), and even a correct decision may not be grounded in the necessary evidence (Shao et al., 2026). This gap motivates a benchmark grounded in real cryo-EM workflows, where different agents can tackle the same scientific decision-making tasks. In contrast, the benchmark measures both decision correctness and evidence acquisition. Such evaluation can characterize the capabilities and limitations of AI agents across cryo-EM tasks.

Specifically, the benchmark should follow three design principles: scientific grounding, workflow realism, and standardized evaluation. First, tasks and evaluations should align with the scientific objectives of cryo-EM and assess not only final results but also whether an appropriate chain of

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.

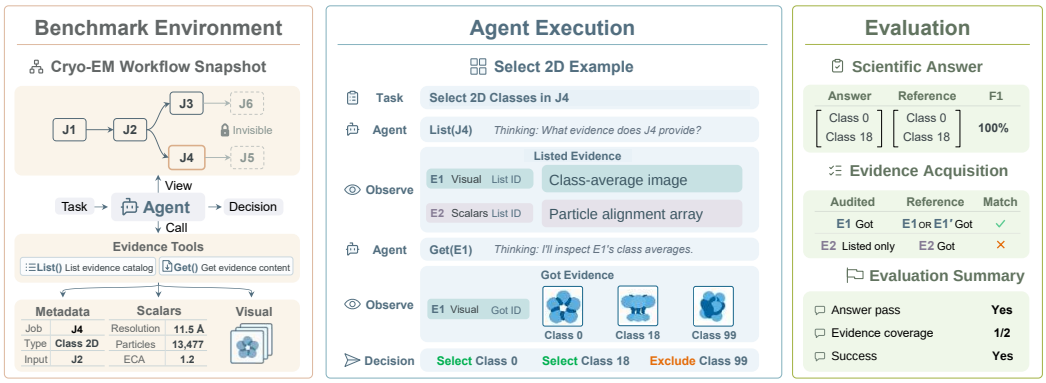


Figure 1: **Overview of AriadneBench. Left:** Workflow snapshots preserve upstream context and multimodal evidence while hiding subsequent jobs. **Middle:** A Select 2D example illustrates evidence discovery (`List`), retrieval (`Get`), and decision-making. **Right:** We evaluate answers and acquired evidence against expert annotations. The example succeeds with a correct answer, 50% evidence coverage, and operational compliance. Listing alone does not count toward evidence coverage.

evidence supports the reasoning process. Second, tasks should take place in the context of real processing workflows, requiring agents to reason about the next step adaptively over current intermediate results. Third, agents should be evaluated under consistent conditions so that performance differences reflect their true capabilities (Chen et al., 2026).

We introduce AriadneBench , a benchmark built from real cryo-EM workflows to evaluate the scientific goal alignment and adaptive autonomy of AI agents under standardized conditions. In particular, we build decision snapshots from real-world workflows, keeping upstream jobs, settings, branches, and intermediate outputs while hiding downstream steps and final answers to prevent information leakage. AriadneBench includes eight task families assessing whether agents align decisions with scientific objectives (*scientific goal alignment*: Inspect Picks, Select 2D, 3D Cleaning, and Reconstruction) and autonomously adapt actions to evidence and workflow states (*adaptive autonomy*: Interaction, Gate, Next Job, and Earliest Fail). Experts provide reference decisions and annotate the evidence needed to support them, including alternative evidence sources that lead to the same conclusion. All agents operate under the same information and resource constraints, while a shared audit tool records what they actually inspect, allowing us to evaluate decision correctness and evidence coverage separately without hallucination.

We evaluate seven agent systems on 180 tasks and find that Claude Fable 5.1 leads in Success Rate (63.21%) and Task Score (83.50). For scientific goal alignment, some agents perform well at comparing candidate reconstructions but struggle to balance quality control with the retention of useful particles. For adaptive autonomy, agents are more effective at translating explicit goals into job configurations and subsequent plans than at assessing whether current results support further analysis or diagnosing the causes of observed problems. Our trajectory analysis further shows that acquiring reference evidence does not guarantee correct scientific judgments. To improve these decisions, we induce eight reusable skills from training trajectories and expert annotations. On 37 held-out tasks, with training and test workflows separated within each task family, we observe higher Success Rates for five of seven agents, with an average increase of 4.25 percentage points across all seven. We observe improvements on some tasks in both capability dimensions, although gains vary across agents and tasks.

2 AriadneBench: Evaluating Scientific Reasoning in Cryo-EM Workflows

As illustrated in Figure 1, AriadneBench is a benchmark for evaluating agents' capability of aligning decisions with scientific objectives (scientific goal alignment) and autonomously adapting actions

to evidence and workflow states (adaptive autonomy) in cryo-EM workflows. Each task starts from a *snapshot* of a workflow in CryoSPARC (Punjani et al., 2017). The snapshot defines the permitted processing history and scientific outputs, which agents must inspect through tools to retrieve job metadata, numerical results, and visual evidence; thus, agents' actions can be faithfully recorded. During execution, agents use tool calls to locate, retrieve, and inspect relevant workflow evidence, then submit structured answers. Finally, AriadneBench 's evaluation jointly considers answer correctness, evidence coverage, and compliance with operational constraints.

2.1 Benchmark Environment Construction

Workflow collection. A *workflow* is a directed acyclic graph, in which nodes denote processing jobs and edges encode their data dependencies (Conesa et al., 2023). Our initial workflow corpus comprised 2,033 recorded cryo-EM processing workflows from in-house experiments and processing runs associated with public EMPIAR datasets (Iudin et al., 2016). After screening for workflow completeness, evidence richness, and execution stability, and consolidating overlapping workflows within the same projects, we retained 73 candidate groups. Manual review of their reconstruction results selected ten core workflow groups. Three additional groups were subsequently included for the Earliest Fail tasks, yielding 13 representative workflows in total. Together, the workflows contain 474 Job entries, as illustrated in Figure 2(b).

Task families. We organize 180 tasks into eight families (T1–T8) grouped under two capability dimensions (Figure 2(a)). *Scientific goal alignment* includes Inspect Picks (T1), Select 2D (T2), 3D Cleaning (T3), and Reconstruction (T4), assessing whether scientific objectives consistently guide agents' decisions. *Adaptive autonomy* includes Interaction (T5), Gate (T6), Next Job (T7), and Earliest Fail (T8), assessing whether agents autonomously adjust their actions in response to available evidence and the current workflow state. As demonstrated in Figure 2(c), each family contains 6–46 tasks, and the median number of jobs per task ranges from 6 to 24 across families. Appendix A details workflow selection and task construction.

Snapshot construction. A workflow *snapshot* provides the realistic context for agent execution, including the target job, its upstream workflow history, and the intermediate results, i.e., evidence, needed for decision-making. Note that agents cannot access reference answers, grading information, or subsequent jobs that could reveal the answer (Kapoor & Narayanan, 2023). For each task, different agents receive the same snapshot and reason over precomputed workflow outputs without the need for actual job submission. To this end, the benchmark substantially reduces computational cost while preserving realistic workflow context. Most task families provide read-only workflow access, but Interaction and Inspect Picks additionally allow agents to configure parameters for answer verification. Appendix A.3 details native snapshot materialization and storage requirements.

Additionally, we implement an *evidence tool* to retrieve job metadata, numerical results, and visual outputs from the task snapshot. Agents must use this tool to list or access workflow evidence, while the tool records every interaction, allowing us to audit the true trajectory of evidence acquisition.

2.2 Expert Annotation

To evaluate agent performance autonomously, we should construct structured expert annotations as references for every task (Chen et al., 2025b). To efficiently annotate hundreds of tasks, we develop a tailored annotator with a graphical user interface that allows experts to assign task-specific ground-truth labels and add explanatory notes.

For each task, the expert annotation includes an answer, a reasoning trajectory, and the corresponding supporting evidence along this trajectory (DeYoung et al., 2020), e.g., which figure has been accessed and is useful for decision-making. Each step in this trajectory specifies what to inspect (*Observation*) and records the findings and their interpretation (*Conclusion*). Each Observation refers to supporting evidence, including job metadata, numerical results, or visual content. The cited evidence comes from the same data sources that agents access through our evidence tool.

Different evidence can be equivalent for the same conclusion (Thorne et al., 2018). To annotate these alternatives and reduce evaluation error, we further introduce *evidence groups*, each containing one or more equivalent evidence. An agent accesses a group by retrieving at least one of its sources.

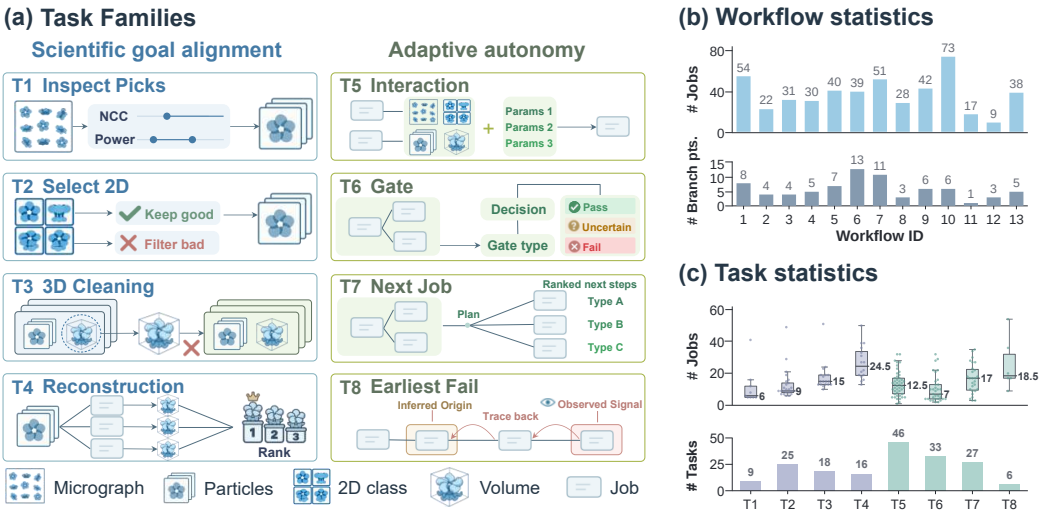


Figure 2: **Task families and dataset statistics of AriadneBench. (a)** Eight task families covering scientific goal alignment (T1–T4) and adaptive autonomy (T5–T8). **(b)** Job and branch-point counts for 13 workflows. **(c)** Snapshot job distributions (top) and counts of all 180 tasks by family (bottom). In the upper plots, dots represent tasks and labels indicate medians.

For quality control, another expert reviews each completed annotation, checking the reference answer, expert trajectory, and supporting evidence before approving it or requesting revisions.

Appendix A.4 documents the annotation interface and review process. Appendix B details evidence acquisition and auditing.

3 Evaluation Protocol

Using the snapshots and expert annotations in Section 2, we evaluate agents through three checks: answer correctness, reference evidence coverage, and operational compliance. We define their combined metrics, introduce the evaluated systems, and describe task execution and auditing.

3.1 Evaluation Metrics

Let $t \in \{1, \ldots, T\}$ index the $T = 8$ task families and $i \in \{1, \ldots, N_t\}$ index each agent's N_t tasks in family t. Experiment 1 averages within families and then equally across families; Experiment 2 averages the same run-level metrics directly over 37 matched tasks per agent and condition.

Compliance. We require all task-specific operational constraints to hold. For example, configure-only Inspect Picks permits saving the designated job's NCC and power thresholds but prohibits executing it. We define **Compliance** as

$$\sigma = \frac{1}{T} \sum_{t=1}^{T} \left(\frac{1}{N_t} \sum_{i=1}^{N_t} \mathbb{1}[\mathcal{C}_{t,i}] \right), \tag{1}$$

where $\mathcal{C}_{t,i}$ denotes compliance with all constraints, $\mathbb{1}[\cdot]$ is the indicator function, and $\sigma_{t,i} = \mathbb{1}[\mathcal{C}_{t,i}]$ is the run-level compliance indicator.

Evidence acquisition. We measure reference evidence coverage as $\varepsilon_{t,i} = M_{t,i}/K_{t,i}$, where $M_{t,i}$ of the $K_{t,i}$ eligible reference groups were acquired. Mean coverage is $\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{N_t}\sum_{i=1}^{N_t}\varepsilon_{t,i}\right)$. To allow evidence paths that differ from the expert trajectory, we define **Evidence Pass Rate** as

$$\rho = \frac{1}{T} \sum_{t=1}^{T} \left(\frac{1}{N_t} \sum_{i=1}^{N_t} \mathbb{1}[\varepsilon_{t,i} \geq \tau] \right), \tag{2}$$

with $\tau = 0.5$ in our primary evaluation. Appendix B.3.1 illustrates evidence grouping on a concrete task.

Answer correctness. Let $\mathcal{A}_{t,i}$ denote satisfaction of the task-family answer criterion and $\beta_{t,i} \in [0, 100]$ the **Answer Score**, including partial credit. We define **Answer Pass Rate** as

$$\alpha = \frac{1}{T} \sum_{t=1}^{T} \left(\frac{1}{N_t} \sum_{i=1}^{N_t} \mathbb{1}[\mathcal{A}_{t,i}] \right). \tag{3}$$

Appendix C.1 specifies the answer-pass criteria and partial-credit rules.

Task success rate. We define run-level **Success** $s_{t,i} \in \{0, 1\}$ as the conjunction of the three checks and aggregate it as **Success Rate**:

$$s_{t,i} = \mathbb{1}[\mathcal{C}_{t,i}]\mathbb{1}[\varepsilon_{t,i} \geq \tau]\mathbb{1}[\mathcal{A}_{t,i}], \qquad S = \frac{1}{T} \sum_{t=1}^{T} \left(\frac{1}{N_t} \sum_{i=1}^{N_t} s_{t,i} \right). \tag{4}$$

Task score. **Task Score** provides partial credit on unsuccessful runs. Answer credit is scaled by evidence coverage up to τ; operational violations yield zero. The run-level score and its aggregate are

$$q_{t,i} = \sigma_{t,i} \min\left(1, \frac{\varepsilon_{t,i}}{\tau}\right) \beta_{t,i}, \qquad Q = \frac{1}{T} \sum_{t=1}^{T} \left(\frac{1}{N_t} \sum_{i=1}^{N_t} q_{t,i} \right), \tag{5}$$

with $q_{t,i}, Q \in [0, 100]$.

Cost. Cost is the mean LLM inference expenditure (USD/run), including every model call and unsuccessful run. We use family-balanced averaging in Experiment 1 and direct averaging over the same 37 matched tasks per agent and condition in Experiment 2. Appendix C.2.2 details cost accounting; offline skill-induction costs are reported separately in Appendix D.4.

3.2 Evaluated Agent Systems

We evaluate seven representative modern agent systems spanning both closed-weight and open-weight models. Agents built on closed-weight models paired with their native harness include Codex with (**GPT-6 Astra**), Claude Code with (**Claude Fable 5.1**), and Grok Build with (**Grok 4.6**). Agents built on open-weight models paired with their native harness include DeepSeek Harness with (**DeepSeek V4.1 Flash**), Qwen Code with (**Qwen 3.8 Max**), Kimi Code with (**Kimi K3**), and ZCode with (**GLM-5.3 Flash**). Each evaluated agent is a fixed pairing of an LLM and its native harness. Throughout the paper, we use the LLM names to refer to these complete agent systems.

Each agent retains its native interaction interface while operating on the same workflow snapshots, evidence rules, submission contracts, and grading criteria. The results therefore compare complete agent systems rather than isolated models (Yang et al., 2024). Appendix C.2.1 reports harness versions and inference settings.

3.3 Task Execution

For each task, we initialize an isolated environment from the frozen workflow snapshot and provide the agent with the task instructions, operational constraints, and permitted tools. The agent uses the evidence tool to inspect the workflow and obtain results that inform its subsequent actions. It may also perform local analysis and, where permitted, configure job parameters. We specify the required answer fields, their meanings, and the supported submission interfaces in Appendix A.5.

We record tool interactions and operational actions throughout each run. For evaluation, we match acquired evidence to eligible expert-annotated evidence groups and check recorded actions against the task-specific constraints. Listing an evidence source alone does not count as acquiring its contents. The final answer is graded under the corresponding task-family rules, and the resulting scores are aggregated using the metrics defined in Section 3.1. Appendix B provides the tool definitions and detailed audit rules, including a correct answer with insufficient evidence coverage (Appendix B.1).

We examine direct image processing in Appendix B.4 and compare list/get use across agents in Appendix B.5.

4 Experiments

Using the evaluation protocol in Section 3, we study how well current agent systems make cryo-EM decisions and whether skills learned from previous runs improve performance on unseen workflows. We first evaluate seven agents on all 180 tasks under autonomous exploration (Section 4.1). We then select training runs and build one skill for each task family (Section 4.2). Finally, we test these skills on 37 held-out tasks and compare them with autonomous exploration, expert evidence lists, and expert inspection steps (Section 4.3). The results show clear differences across task families and improved performance for most agents.

4.1 Agentic Autonomous Exploration

Experimental setup. We evaluate seven agents on all 180 tasks across eight task families. Each agent explores the workflow snapshot, gathers evidence, and submits a decision without additional guidance.

Overall reliability. We find that Claude Fable 5.1 leads in Success Rate (63.21%) and Task Score (83.50), followed by GPT-6 Astra and Grok 4.6 (Table 1). All three exceed 90% Evidence Pass Rate and achieve full operational compliance, yet their Success Rates remain below 64%. For these agents, incorrect answers are more common than failures to meet the evidence requirement. Figure 3(c) also shows that runs with correct answers can still fail because the agent has not met the evidence or operational requirements. We examine acquisition and interpretation errors in Appendices E.1–E.2, with further cases across agents and a shared-task comparison in Appendices E.4–E.5. Appendices C.3 and C.4 give per-task scores and full results by task family.

Scientific goal alignment. We find particularly weak performance on Inspect Picks (T1), with a mean Success Rate of 11.11% (Figure 3(a)). Inspect Picks requires choosing normalized cross-correlation (NCC) and power cutoffs that remove unwanted picks while preserving useful particles (Xu et al., 2025). On one task, GPT-6 Astra achieved 100% precision against the expert selection but retained only 49.4% of the reference particles (Appendix E.3, Figure 19). The expert prioritized retaining useful particles to avoid missed picks. Astra's stricter NCC and power cutoffs discarded many reference particles, causing its answer to fail the task's selection criterion.

Adaptive autonomy. We find higher Success Rates for Interaction (T5) and Next Job (T7) than for Gate (T6) and Earliest Fail (T8) (Figure 3(a)). Interaction and Next Job ask agents to configure or recommend an operation. Gate assesses whether current outputs justify transitioning to the next workflow stage. For instance, a favorable resolution does not guarantee that a map supports atomic modeling in the target region, requiring the agent to perform localized structural validation beyond scalar metrics (Kaur et al., 2021). Earliest Fail requires agents to trace an observed failure to an upstream processing decision, such as identifying a poor refinement caused by an earlier 2D class selection. These diagnostic tasks prove harder because they require assessing scientific quality and tracing upstream history. We test whether induced skills improve performance on these tasks in Section 4.3.

Cost–performance tradeoff. We compare Task Score and Cost in Figure 3(b). Grok 4.6 scores 79.43 at $0.72 per run, compared with 83.50 at $2.15 for Claude Fable 5.1—about one-third of the cost. DeepSeek V4.1 Flash scores 67.09 at $0.07 per run. Grok 4.6 also scores higher than Kimi K3 while costing less, so Kimi K3 is not on the Pareto frontier.

4.2 Inducing Generalizable Skills from Agent Trajectories

Training trajectory selection. We hold out workflows separately within each task family, leaving 143 training tasks and 37 test tasks, approximately an 80/20 split (Appendix D.1). From 1,001 training runs, we automatically select 391 trajectories with complete records, scorable answers, and no constraint violations. For each task, we group eligible runs by answer correctness and evidence coverage. From each occupied group, we select one representative that maximizes tool diversity, keeping at most four traces in total. Crucially, experts provide only the reference annotations rather

Table 1: **Evaluation of autonomous exploration performance on AriadneBench.** We evaluate seven agents on 180 tasks. Success Rate (%) requires answer correctness, evidence coverage $\geq$ 50%, and operational compliance. Task Score (0–100) combines answer credit with evidence and compliance checks (Section 3.1). Bold indicates the best result in each column.

Agent	Overall Performance ↑		Pass Rates (%) ↑			Cost (USD/run) ↓
	Success Rate	Task Score	Answer	Evidence	Compliance	
Claude Fable 5.1	**63.21**	**83.50**	**67.84**	**94.83**	**100.00**	2.15
GPT-6 Astra	56.57	82.15	62.86	91.59	**100.00**	1.81
Grok 4.6	54.99	79.43	59.62	92.36	**100.00**	0.72
Kimi K3	42.86	67.74	54.28	80.64	96.14	0.77
DeepSeek V4.1 Flash	47.78	67.09	57.95	81.17	99.18	0.07
Qwen 3.8 Max	42.69	67.72	52.27	82.04	98.10	0.63
GLM-5.3 Flash	39.75	59.40	51.48	69.17	92.57	**0.06**

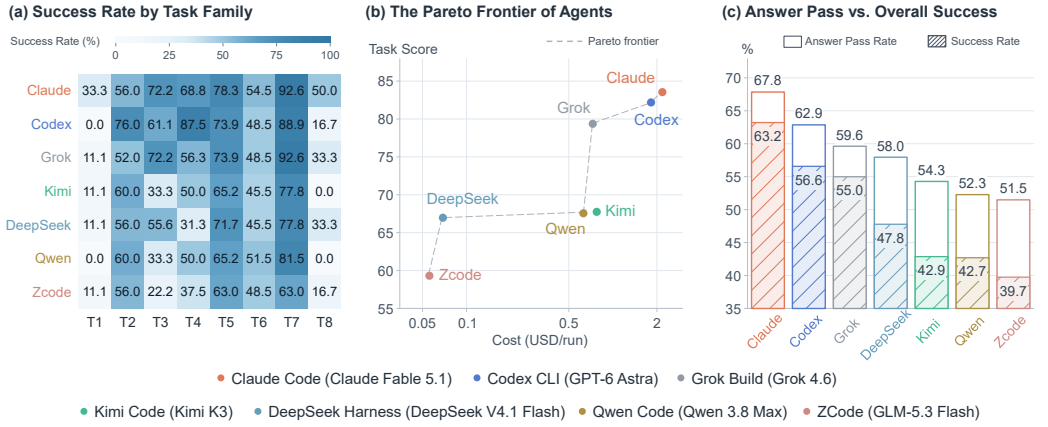


Figure 3: **Autonomous exploration results on AriadneBench.** **(a)** Agent success rates across the eight task families. **(b)** Task score–cost pareto frontier of the evaluated agents. **(c)** Comparison of answer pass rate and success rate across agents. The gap represents unsuccessful cases: answers are correct, but process requirements are not met.

than manually curating runs. Full selection protocols and counts are provided in Appendix D.1 and Table 19.

Skill induction. Building on agent experience distillation (Zhao et al., 2024; Wang et al., 2025), we independently induce one skill per task family. For each training task, we give GPT-6 Astra the question, selected agent trajectories, evaluation feedback, expert steps, and reference answer in a separate context and ask it to propose reusable rules. We merge the proposed rules by task family and check them against the original records. Each skill specifies what to inspect, how to decide, and when to stop. To prevent data leakage, we sanitize all rules by stripping task identifiers, ground-truth answers, and proprietary scoring rubrics. The resulting skills remain frozen during evaluation without updating model weights. Appendix D.2 details the complete skill induction, consolidation, and auditing protocols. All induction code, configurations, prompt templates, and final skills will be publicly released.

4.3 Effectiveness of Induced Skills

Experimental setup. We compare seven agents on the same 37 held-out tasks under four conditions: autonomous exploration (**AE**), induced skill (**IS**), evidence list (**EL**), and evidence + steps (**ES**) (Table 2). IS supplies the fixed skill for the task family to AE; EL adds an expert evidence list to AE; ES adds expert inspection and comparison steps to EL. EL/ES provide no expert findings or answers;

Table 2: **Effect of induced skills on AriadneBench.** We test seven agents on 37 held-out tasks with autonomous exploration (AE), induced skill (IS), evidence list (EL), and evidence + steps (ES).

	Induced skill comparison						Task-specific expert guidance			
	AE		IS		Δ(IS−AE)		EL		ES	
Agent	Success*	Score*	Success*	Score*	ΔSuccess	ΔScore	Success*	Score*	Success*	Score*
Claude Fable 5.1	67.57	88.36	81.08	92.94	**+13.51**	**+4.58**	70.27	89.00	75.68	88.76
GPT-6 Astra	64.86	86.02	70.27	86.63	**+5.41**	**+0.61**	72.97	87.87	72.97	88.61
Grok 4.6	59.46	81.35	62.16	82.71	**+2.70**	**+1.36**	64.86	85.56	72.97	88.90
Kimi K3	45.95	72.04	51.35	71.40	**+5.41**	**-0.64**	51.35	74.50	56.76	74.62
DeepSeek V4.1 Flash	51.35	71.78	59.46	76.51	**+8.11**	**+4.73**	62.16	76.51	62.16	78.53
Qwen 3.8 Max	59.46	76.48	54.05	76.99	**-5.41**	**+0.51**	59.46	73.86	54.05	69.98
GLM-5.3 Flash	43.24	63.75	43.24	60.43	**0.00**	**-3.32**	51.35	66.84	56.76	69.83

IS = AE + skill; EL = AE + evidence list; ES = EL + steps. EL/ES omit expert findings and answers. Success (%); Task Score (0–100); changes in percentage points / score points. *: equal task weights.

Appendix D.5 specifies the guidance and execution settings. Starred metrics weight tasks equally (Appendix C.2.2).

Skills improve performance on unseen workflows. We observe higher Success Rates with IS than with AE for five agents (Table 2). Across all seven agents, the average gains are 4.25 percentage points in Success Rate and 1.12 Task Score points. Claude Fable 5.1 improves from 67.57% to 81.08% Success Rate without access to test-task expert annotations. Gains vary across agents and metrics: Kimi K3 improves in Success Rate but loses Task Score, while Qwen 3.8 Max shows the reverse. Appendix D.6 compares evidence use and decisions in paired AE/IS trajectories on a held-out task.

Benefits of evidence lists and inspection steps. We find that Grok 4.6 improves from 64.86% Success Rate with EL to 72.97% with ES (Table 2). The extra steps specify which observations to compare while leaving the conclusion to the agent. Across seven agents, IS gives smaller average gains over AE than EL or ES, although Claude Fable 5.1 obtains its best Success Rate and Task Score with IS.

Benefits of skill guidance for workflow decisions. We find that most of the increase in successful runs from AE to IS comes from Gate tasks. Correct gate identification is unchanged, but fully correct answers increase from 19 to 24 (Appendix C.5), indicating better decisions about whether to proceed. The skill asks agents to inspect retained and excluded particles and relate map quality to the intended interpretation. Reconstruction also improves; its skill asks agents to check Fourier shell correlation (FSC) definitions (Scheres & Chen, 2012b), masks, and directional resolution (Tan et al., 2017) before ranking maps.

Remaining limitations of skill guidance. On one Earliest Fail test task, Grok 4.6 covers more reference evidence with IS but shifts its prediction from the correct failure job to a later job (Appendix C.5). In a separate task used for skill induction, the expert checks input connections and alternative classes to distinguish generating a weak class from selecting it for further refinement (Appendix E.1). This distinction still requires judgment about the scientific target. Execution and evidence checks can also fail: in Inspect Picks, two IS runs fail a configuration or evidence check, reducing mean Task Score despite nearly unchanged Answer Score (Appendix C.5).

5 Related Work

Automated cryo-EM workflows. Existing approaches to cryo-EM automation include conventional tools, integrated automated workflows, and agentic methods. Warp automates image preprocessing (Tegunov & Cramer, 2019), while Topaz and crYOLO support particle picking (Bepler et al., 2019; Wagner et al., 2019). CryoSPARC, RELION, and Scipion provide reconstruction pipelines and infrastructure for automated workflows (Punjani et al., 2017; Kimanius et al., 2021; De la Rosa-Trevin et al., 2016). TranSPHIRE adapts particle picking based on downstream classification feedback (Stabrin et al., 2020); CryoWizard combines Cryo-IEF representations (Yan et al., 2026) with

particle-quality assessment for automated 2D classification, while cryoAgent (Li et al., 2026) explores adaptive tool use across processing stages using ReAct (Yao et al., 2022). However, these methods are often tailored to specific tasks, and standardized, comprehensive evaluation of their decision-making remains limited. AriadneBench complements these methods by evaluating decisions at fixed workflow states, including intermediate-result selection, planning, and failure localization.

Scientific agent benchmarks. Existing benchmarks assess scientific programming (Tian et al., 2024; Chen et al., 2025b), data analysis and discovery (Jing et al., 2025; Majumder et al., 2025), machine learning experimentation (Huang et al., 2023; Chan et al., 2025), and research reproduction (Siegel et al., 2024; Starace et al., 2025). AstaBench spans multiple scientific tasks and tools (Bragg et al., 2026). SpatialBench and scBench ground biological analysis tasks in concrete data and verifiable outcomes (Workman et al., 2025; 2026). However, these benchmarks provide limited coverage of cryo-EM workflow decisions that require interpreting intermediate results. AriadneBench focuses on cryo-EM workflow decisions: snapshots preserve the dependencies, parameters, branches, and intermediate outputs needed to relate current observations to earlier processing choices.

Evidence and process evaluation. AgentBoard and ToolSandbox evaluate intermediate progress (Ma et al., 2024; Lu et al., 2025); ProcessBench studies the earliest erroneous step in mathematical reasoning (Zheng et al., 2025). ALCE evaluates citation correctness and completeness (Gao et al., 2023), while SeekBench examines evidence grounding and sufficiency in information-seeking agents (Shao et al., 2026). Our evidence audit records acquisition of eligible expert-reference sources, including reviewed micrograph processing for Inspect Picks. Success combines this acquisition requirement with scientific answer grading and operational compliance; it does not certify citation entailment or interpretation.

Learning from agent experience. Reflexion uses verbal feedback across trials (Shinn et al., 2023). ExpeL extracts natural-language insights from experience, while Agent Workflow Memory induces reusable workflows and supplies them as context for subsequent tasks (Zhao et al., 2024; Wang et al., 2025). SkillRL additionally combines skill distillation with policy learning (Xia et al., 2026). Our offline skill induction extracts textual procedures from experience without parameter updates. It compares trajectories within a task, uses training-only expert supervision, and consolidates source-supported procedures independently by family. The induced skills are fixed before held-out cryo-EM evaluation.

6 Discussion

Limitations. Given the time required to execute cryo-EM workflows, AriadneBench evaluates decisions from static snapshots and could be extended to dynamic workflows. AriadneBench covers a limited range of proteins and workflows owing to expert annotation costs; future extensions could broaden this diversity. AriadneBench currently audits evidence acquisition, while future work could assess the correctness of intermediate reasoning steps (Lightman et al., 2024). AriadneBench only uses skills to evaluate experience transfer at low cost and facilitate ablation analysis, future work could compare this approach with trajectory-based post-training (Zeng et al., 2024; Chen et al., 2025a).

Conclusion. We introduce AriadneBench to evaluate how well agents align decisions with scientific goals and autonomously adapt their actions to evidence in real cryo-EM workflows, using decision snapshots, expert annotations, and auditable evidence tools. Across 180 tasks, the strongest of seven agents achieves only 63.21% Success Rate. Agents perform better at configuring operations and planning next steps than at particle selection, workflow readiness, or failure localization; acquiring reference evidence does not guarantee correct scientific judgment. On 37 held-out tasks, skills induced from prior trajectories improve Success Rate for five of seven agents, though gains vary. Together, these findings show that reusable procedures help, while reliable cryo-EM decisions still require interpreting evidence against scientific goals and workflow history.

AI Use Statement

We used generative AI tools to assist with literature searches and reference checking, experimental design, data and trajectory analysis, result interpretation, code development, figure preparation, and manuscript drafting, translation, and editing. AI-assisted skill induction and source auditing are described in Section 4.2 and Appendix D. We take responsibility for the final content of this work, including all scientific claims, citations, and AI-assisted artifacts.

Ethics Statement

AriadneBench combines private experimental workflows with workflows associated with publicly available data. Agent access is restricted to task-scoped environments that exclude source-system credentials and private grading material, with permitted operations specified in Appendices A and C. Scientific images, annotations, and trajectories will be shared only to the extent permitted by data owners and participating experts.

Reproducibility Statement

Appendices A–C document benchmark construction, evidence auditing, grading rules, agent configurations, and detailed evaluation results. Appendix D describes the training/test split, trajectory selection, skill induction, and guidance conditions. To support reproducibility and reuse, we will release the trajectory-selection and skill-induction code, configurations, prompt templates, and final skills. Access to underlying scientific data and trajectories remains subject to the permissions described above.

References

Tristan Bepler, Andrew Morin, Micah Rapp, Julia Brasch, Lawrence Shapiro, Alex J Noble, and Bonnie Berger. Positive-unlabeled convolutional neural networks for particle picking in cryo-electron micrographs. *Nature methods*, 16(11):1153–1160, 2019.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

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, pp. 110136–110223, 2026.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, et al. Mle-bench: Evaluating machine learning agents on machine learning engineering. In *International Conference on Learning Representations*, volume 2025, pp. 50466–50494, 2025.

Zhixun Chen, Ming Li, Yuxuan Huang, Yali Du, Meng Fang, and Tianyi Zhou. Atlas: Agent tuning via learning critical steps. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25334–25349, 2025a.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Sahel Sharifymoghaddam, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, et al. Browsecomp-plus: a fair and disentangled evaluation benchmark for deep search agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22349–22370, 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, pp. 96934–96990, 2025b.

Yifan Cheng, Nikolaus Grigorieff, Pawel A Penczek, and Thomas Walz. A primer to single-particle cryo-electron microscopy. *Cell*, 161(3):438–449, 2015.

Pablo Conesa, Yunior C Fonseca, Jorge Jiménez de la Morena, Grigory Sharov, Jose Miguel de la Rosa-Trevín, Ana Cuervo, Alberto García Mena, Borja Rodríguez de Francisco, Daniel Del Hoyo, David Herreros, et al. Scipion3: A workflow engine for cryo-electron microscopy image processing and structural biology. *Biological Imaging*, 3:e13, 2023.

Jose Miguel De la Rosa-Trevin, A Quintana, L Del Cano, A Zaldívar, I Foche, J Gutiérrez, J Gómez-Blanco, J Burguet-Castell, J Cuenca-Alba, V Abrishami, et al. Scipion: A software framework toward integration, reproducibility and validation in 3d electron microscopy. *Journal of structural biology*, 195(1):93–99, 2016.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. Eraser: A benchmark to evaluate rationalized nlp models. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4443–4458, 2020.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488, 2023.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. Mlagentbench: Evaluating language agents on machine learning experimentation. *arXiv preprint arXiv:2310.03302*, 2023.

Andrii Iudin, Paul K Korir, José Salavert-Torres, Gerard J Kleywegt, and Ardan Patwardhan. Empiar: a public archive for raw electron microscopy image data. *Nature methods*, 13(5):387–389, 2016.

Liqiang Jing, Zhehui Huang, Xiaoyang Wang, Wenlin Yao, Wenhao Yu, Kaixin Ma, Hongming Zhang, Xinya Du, and Dong Yu. Dsbench: How far are data science agents from becoming data science experts? In *International Conference on Learning Representations*, volume 2025, pp. 32597–32649, 2025.

Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9), 2023.

Satinder Kaur, Josue Gomez-Blanco, Ahmad AZ Khalifa, Swathi Adinarayanan, Ruben Sanchez-Garcia, Daniel Wrapp, Jason S McLellan, Khanh Huy Bui, and Javier Vargas. Local computational methods to improve the interpretability and analysis of cryo-em maps. *Nature communications*, 12(1):1240, 2021.

Dari Kimanius, Liyi Dong, Grigory Sharov, Takanori Nakane, and Sjors HW Scheres. New tools for automated cryo-em single-particle analysis in relion-4.0. *Biochemical Journal*, 478(24):4169–4185, 2021.

Daoyi Li, Shaodi Yang, Qian Xiao, Tongxin Niu, Yan Zhang, Yun Zhu, and Fei Sun. cryoagent: An agentic workflow for robust and adaptive end-to-end cryo-em image processing. *bioRxiv*, pp. 2026–04, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, et al. Toolsandbox: A stateful, conversational, interactive evaluation benchmark for llm tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, 2025.

Dmitry Lyumkis. Challenges and opportunities in cryo-em single-particle analysis. *Journal of Biological Chemistry*, 294(13):5181–5197, 2019.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. Agentboard: An analytical evaluation board of multi-turn llm agents. *Advances in neural information processing systems*, 37:74325–74362, 2024.

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.

OpenAI. GPT-6 Astra model. OpenAI developer documentation, 2026. URL `https://developers.openai.com/api/docs/models/gpt-6-astra`. Model documentation and pricing; accessed 23 September 2026.

Ali Punjani, John L Rubinstein, David J Fleet, and Marcus A Brubaker. cryosparc: algorithms for rapid unsupervised cryo-em structure determination. *Nature methods*, 14(3):290–296, 2017.

Jean-Paul Renaud, Ashwin Chari, Claudio Ciferri, Wen-ti Liu, Hervé-William Rémigy, Holger Stark, and Christian Wiesmann. Cryo-em in drug discovery: achievements, limitations and prospects. *Nature reviews Drug discovery*, 17(7):471–492, 2018.

Sjors H. W. Scheres and Shaoxia Chen. Prevention of overfitting in cryo-EM structure determination. *Nature Methods*, 9:853–854, 2012a. doi: 10.1038/nmeth.2115. URL `https://www.nature.com/articles/nmeth.2115`.

Sjors HW Scheres and Shaoxia Chen. Prevention of overfitting in cryo-em structure determination. *Nature methods*, 9(9):853–854, 2012b.

Jiaqi Shao, Yuxiang Lin, Munish P Lohani, Yufeng Miao, and Bing Luo. Do llm agents know how to ground, recover, and assess? evaluating epistemic competence in information-seeking agents. In *International Conference on Learning Representations*, volume 2026, pp. 152647–152679, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Zachary S Siegel, Sayash Kapoor, Nitya Nadgir, Benedikt Stroebl, and Arvind Narayanan. CORE-bench: Fostering the credibility of published research through a computational reproducibility agent benchmark. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Markus Stabrin, Fabian Schoenfeld, Thorsten Wagner, Sabrina Pospich, Christos Gatsogiannis, and Stefan Raunser. Transphire: automated and feedback-optimized on-the-fly processing for cryo-em. *Nature communications*, 11(1):5716, 2020.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, et al. Paperbench: evaluating ai's ability to replicate ai research. *arXiv preprint arXiv:2504.01848*, 2025.

Kyle Swanson, Wesley Wu, Nash L Bulaong, John E Pak, and James Zou. The virtual lab of ai agents designs new sars-cov-2 nanobodies. *Nature*, 646(8085):716–723, 2025.

Yong Zi Tan, Philip R Baldwin, Joseph H Davis, James R Williamson, Clinton S Potter, Bridget Carragher, and Dmitry Lyumkis. Addressing preferred specimen orientation in single-particle cryo-em through tilting. *Nature methods*, 14(8):793–796, 2017.

Dimitry Tegunov and Patrick Cramer. Real-time cryo-electron microscopy data preprocessing with warp. *Nature methods*, 16(11):1146–1152, 2019.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 809–819, 2018.

Minyang Tian, Luyu Gao, Shizhuo D Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, et al. Scicode: A research coding benchmark curated by scientists. *Advances in Neural Information Processing Systems*, 37:30624–30650, 2024.

Thorsten Wagner, Felipe Merino, Markus Stabrin, Toshio Moriya, Claudia Antoni, Amir Apelbaum, Philine Hagel, Oleg Sitsel, Tobias Raisch, Daniel Prumbaum, et al. Sphire-cryolo is a fast and accurate fully automated particle picker for cryo-em. *Communications biology*, 2(1):218, 2019.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63897–63911. PMLR, 13–19 Jul 2025.

Kenny Workman, Zhen Yang, Harihara Muralidharan, and Hannah Le. Spatialbench: Can agents analyze real-world spatial biology data? *arXiv preprint arXiv:2512.21907*, 2025.

Kenny Workman, Zhen Yang, Harihara Muralidharan, Aidan Abdulali, and Hannah Le. scbench: Evaluating ai agents on single-cell rna-seq analysis. *arXiv preprint arXiv:2602.09063*, 2026.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026.

Chentianye Xu, Xueying Zhan, and Min Xu. Cryomae: Few-shot cryo-em particle picking with masked autoencoders. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3876–3885. IEEE, 2025.

Yang Yan, Shiqi Fan, Fajie Yuan, and Huaizong Shen. A comprehensive foundation model for cryo-em image processing. *Nature Methods*, 23(1):88–95, 2026.

John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. Agenttuning: Enabling generalized agent abilities for llms. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3053–3077, 2024.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Processbench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1024, 2025.

Appendix Contents

A Details of Benchmark Construction **15**

A.1 Workflow Selection and Scientific Context . . . 15
A.2 Constructing Eight Task Families . . . 15
A.3 Lightweight Snapshots in a Native CryoSPARC Environment . . . 16
A.4 Annotation Interface and Review . . . 17
A.5 Task Interface and Answer Submission . . . 19

B Details of Evidence Auditing **21**

B.1 Why We Audit Evidence Acquisition . . . 21
B.2 Evidence Tool Design . . . 21
B.3 Making Evidence Access Auditable . . . 22
B.4 A Direct Image-Processing Route . . . 23
B.5 List and Get Calls Across Agents . . . 25

C Details of Task Grading and Results **25**

C.1 Answer-Pass Criteria and Partial Credit . . . 25
C.2 Evaluation Settings and Aggregation . . . 27
C.3 Per-Task Scores and Answer-Pass Outcomes . . . 27
C.4 Complete Autonomous Results . . . 29
C.5 Guidance and Skill Results . . . 29

D Details of Skill Induction **32**

D.1 Task Split and Trajectory Selection . . . 32
D.2 Rule Induction, Consolidation, and Audit . . . 33
D.3 Skill Format and Application . . . 33
D.4 Cost of Skill Induction . . . 34
D.5 Guidance Conditions and Execution Settings . . . 34
D.6 Paired Trajectories with Skill Guidance . . . 35

E Agent Trajectories and Failure Analysis **36**

E.1 Incomplete Evidence Acquisition . . . 36
E.2 Incorrect Decisions despite Full Reference Coverage . . . 36
E.3 Expert Checks in Particle Selection . . . 37
E.4 Failure Cases across Agents . . . 38
E.5 Comparing Agents on the Same Task . . . 39

Reading guide. Appendix A describes workflow selection, task construction, native snapshots, expert annotation, and the task submission interface. Appendix B specifies evidence tools, access auditing, and observed list/get behavior. Appendix C gives grading rules, evaluation settings, aggregation, and complete results. Appendix D details trajectory selection, skill induction, its cost, and test-time guidance. Appendix E compares recorded trajectories and examines acquisition, interpretation, and submission errors.

A Details of Benchmark Construction

We describe how we select workflows, construct scientific decision tasks, build lightweight native environments, obtain expert annotations, and define the agent-facing task interface.

A.1 Workflow Selection and Scientific Context

We screen 2,033 recorded cryo-EM workflows for reconstruction or refinement activity, evidence availability, and execution stability. We consolidate overlapping branches into candidate groups of at most 200 jobs, retaining 73 high-priority candidates from 59 projects for expert inspection. The screening statistics identify candidates; they do not establish that a branch is scientifically successful or identify the cause of a failure. Experts examine reconstruction quality and processing history before selecting ten core groups. Three additional groups provide further Earliest Fail examples, giving 13 groups in the final benchmark.

Table 3 summarizes what these workflows involve. The core collection includes filament, membrane-protein, chaperonin, polymerase, and other protein-complex workflows. Curator notes document choices such as comparing picking strategies, recovering underrepresented views, selecting among 3D classes, and assessing whether further refinement improves the map. These scientific comparisons require inspection of images, numerical diagnostics, and branch history in addition to job metadata. Across the 13 groups, we retain 474 job entries and 76 branch points; a branch point has at least two distinct outgoing neighbors.

Table 3: Protein types, data sources, and goals of the 13 workflows. In-house data were supplied by collaborating laboratories. Workflows 1–10 follow Figure 2; 11–13 provide additional failure-localization tasks. Counts are per workflow.

Workflow	Jobs	Branches	Tasks	Protein type	Source	Goal
1	54	8	25	Filament	In-house	Particle selection; reconstruction
2	22	4	15	Chaperonin	EMPIAR-11170	Micrograph and particle curation
3	31	4	13	RNA polymerase	In-house	Particle selection
4	30	5	14	ATPase complex	EMPIAR-12701	Branch comparison
5	40	7	15	Membrane protein	In-house	Non-uniform refinement
6	39	13	24	Filament	In-house	Helical refinement; motion correction
7	51	11	15	Protein complex	EMPIAR-11734	Core interpretability
8	28	3	21	Membrane protein	EMPIAR-10081	Refinement; failure diagnosis
9	42	6	18	Oxidoreductase	EMPIAR-12423	Local refinement; provenance
10	73	6	17	GPCR	In-house	Reconstruction comparison
11	17	1	1	Viral capsid	EMPIAR-10341	Localize high-resolution failure
12	9	3	1	Lipid transporter	EMPIAR-10350	Trace reconstruction failure
13	38	5	1	Peptidase	EMPIAR-10980	Trace reconstruction failure
Total	474	76	180			

A.2 Constructing Eight Task Families

We locate decision points within the selected workflows and construct questions supported by the available processing history (Table 4). A workflow contributes the task families that its history supports; we do not require every workflow to supply all eight families. One workflow can contribute several questions at different stages or branches. Curators exclude a putative Reconstruction task when only one candidate exists, or the available evidence does not support a defensible comparison. The resulting 180 tasks comprise 9 Inspect Picks, 25 Select 2D, 18 3D Cleaning, 16 Reconstruction, 46 Interaction, 33 Gate, 27 Next Job, and 6 Earliest Fail tasks.

For each candidate, an expert defines the scientific question, the visible workflow state, the permitted actions, and the required answer. The expert solves the task, records the supporting observations and conclusions, and identifies acceptable evidence alternatives. A different expert reviews the complete package, and either approves it or requests revision. This work is performed for individual

tasks: a job's type or completion status alone does not determine a reference answer. For example, a completed refinement can still be scientifically unsuitable, while a weak class can be a legitimate classification output rather than the origin of a failure.

Table 4: Scientific decision points used to construct the eight task families. Experts define the question, admissible evidence, and reference decision for each instance.

Task	Decision point	Expert contribution
Inspect Picks	Candidate picks available	Set cutoffs; check identity, centering, retention
Select 2D	2D classes available	Retain useful classes; exclude artifacts
3D Cleaning	3D classes available	Select matching particle–volume pairs
Reconstruction	Alternative maps available	Rank maps by quality, validation, provenance
Interaction	Job configuration required	Specify the operation, inputs, parameters
Gate	Stage transition reached	Assess readiness to proceed
Next Job	Next operation required	Identify acceptable next operations
Earliest Fail	Scientific problem observed	Trace failure origins; compare alternatives

A.3 Lightweight Snapshots in a Native CryoSPARC Environment

We separate workflow export from task materialization. First, we export the selected native job graph and the metadata, numerical datasets, logs, visual assets, and explicitly retained volumes needed for decision-making. Large movie and particle-image payloads that are outside the task's evidence contract are omitted. The evaluated exports use manifest-only omission records rather than copying the omitted payloads. These records preserve provenance but do not supply scientific pixels; selected final volumes and masks are explicitly retained when needed. We preserve native job types, parameter names, and input/output connections so that the exported workflow can be attached to CryoSPARC rather than replaced by a synthetic graph.

We then construct each task from the permitted jobs and their required upstream dependencies. Mutable project and job-control files are copied into the task environment. Immutable scientific payloads are stored in a content-addressed cache and shared through hard links, avoiding a fresh data copy for every task. Each task is attached as a native CryoSPARC project/workspace and accessed from an isolated Docker environment. The task exposes neither future jobs that reveal the answer nor the private reference annotations. Configure-only tasks permit the specified job configuration but do not execute new reconstruction workloads.

For Inspect Picks, we additionally provide ten deterministic micrograph previews per task, sampled without consulting private particle-retention labels. The nine tasks therefore contain 90 preview entries, totaling about 131.3 MiB in logical image size. Agents receive the permitted previews, numerical outputs, visual assets, and selected final volumes; complete acquisition movies and unneeded particle-image stacks remain outside the exposed snapshot.

We obtain the source project directories through the CryoSPARC API and inspect the 474 selected job directories on the source filesystem. The 13 workflow views span nine projects and contain approximately 14.84 TiB of project-local regular files. Their native export bases occupy 391.9 GiB in logical file size, about 2.6% of that footprint. We count the same selected job sets, exclude unrelated jobs, and do not follow external file links; these values therefore do not include acquisition movies stored outside the project directories. The original and exported file compositions are shown in Figure 4. Scientific pixel payloads account for about 96.5% of source bytes, while numerical datasets account for about 78.9% of retained export bytes.

Across all 180 task-materialization records, newly copied control files have a median size of 21.7 MiB and a total of 5.1 GiB. In 127 tasks, all scientific payloads already exist in the cache, so no new cache data are added. Recorded cache additions total 225.6 GiB across the remaining task creations; this amount depends on the cache state and is distinct from the copied control-file cost. Preview sizes are reported separately above. These measurements describe snapshot files before agent execution and exclude common Docker image layers, directory overhead, and run-generated logs.

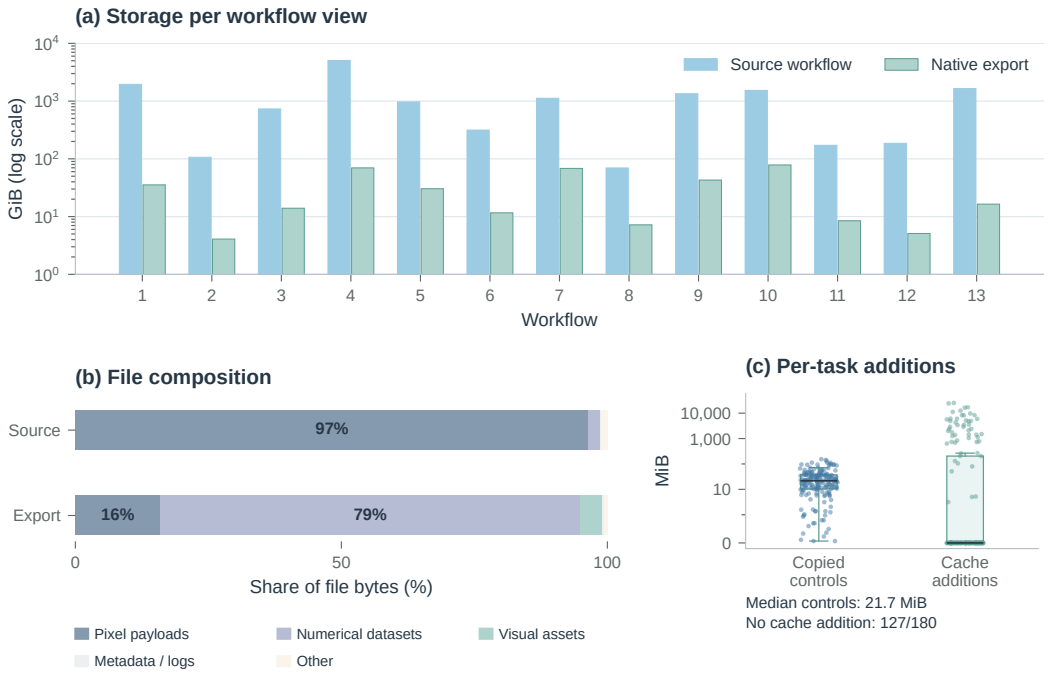


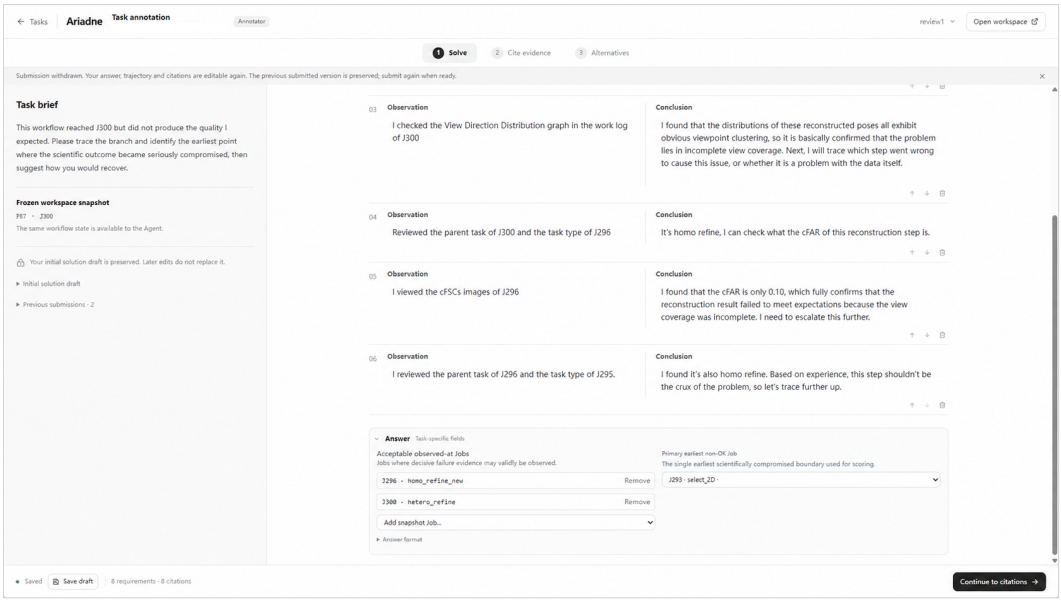
Figure 4: **Storage requirements of native workflow snapshots.** **(a)** Project-local source and exported file sizes for the same 13 workflow job sets; external file-link targets are excluded. **(b)** Composition of scientific and metadata files. **(c)** Recorded per-task additions from copied controls and newly populated shared cache objects. Cache additions depend on prior cache contents; these counters do not include filesystem-directory overhead. Storage units are binary GiB and MiB.

A.4 Annotation Interface and Review

We build a web interface for task definition, scientific annotation, evidence citation, and independent review (Figure 5). Experts work with the same underlying evidence sources and source identifiers exposed to agents. Each annotation contains the question and task scope, a reference answer, an ordered expert trajectory, and evidence groups. Each trajectory step separates an *Observation*, specifying what is inspected, from a *Conclusion*, recording the finding and its interpretation. Evidence citations attach to the observations, and experts can group alternative sources that support the same judgment.

The reviewer checks the answer, the scientific argument, evidence availability, and the compatibility of the task with its snapshot and operational constraints. The interface retains submitted versions and revision requests. The same person does not approve their own annotation; this is annotation followed by independent review, not two independent solutions or an inter-expert agreement study. The task definitions contain 471 expert steps and 961 per-task evidence groups. These annotation totals differ from the 924 groups used in the frozen scoring reference; evidence coverage uses the task-specific grading reference in Appendix B.3. Table 5 reports their distribution across task families.

(a) Task annotation



(b) Independent review

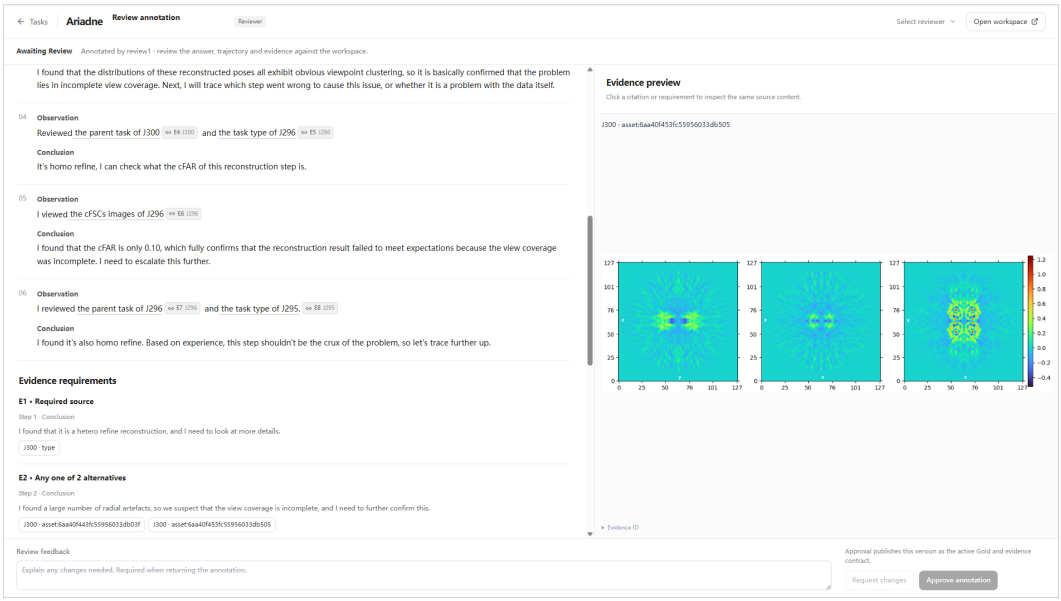


Figure 5: **Web interface for expert annotation and review.** **(a)** The annotator records observations, conclusions, and a reference answer. **(b)** A different expert reviews the trajectory, cited evidence, and alternative sources, with visual evidence available for inspection.

Table 5: Expert annotation counts by task family. Entries report the median (minimum–maximum) across tasks. Groups count distinct annotated evidence groups, source items count distinct cited sources within each task, and expert steps count entries in the reference trajectory. These are annotation statistics rather than scoring denominators.

Family	Tasks	Groups	Source items	Expert steps
T1	9	3 (3–10)	12 (12–19)	2 (2–4)
T2	25	3 (2–4)	3 (2–4)	2 (2–3)
T3	18	10 (4–19)	11 (4–19)	2 (2–4)
T4	16	10 (7–15)	16.5 (9–24)	4 (2–8)
T5	46	2.5 (1–10)	3 (1–11)	2 (1–3)
T6	33	3 (1–10)	5 (1–15)	2 (1–7)
T7	27	3 (1–13)	5 (1–13)	2 (1–6)
T8	6	19 (15–29)	26 (20–35)	9.5 (6–11)

A.5 Task Interface and Answer Submission

Answer fields and submission formats. We specify the scientific answer variables for each task family. The recorded runs contain both tool submissions and structured final answers parsed by the evaluator. Table 6 gives the tool schemas, field meanings, and submission prerequisites; every tool requires a free-text `rationale`. The answer-pass criteria are given in Table 9.

Public definitions and source identifiers. We provide public definitions for fixed vocabularies: Gate uses four stage transitions and their readiness standards, while Next Job uses a catalog of canonical job-type identifiers and display names. Other tasks refer to the snapshot's original class IDs, Job UIDs, or Job/output sources. These interfaces define the expected answer without revealing task-specific reference answers.

Deciding when to submit. We state the task-specific prerequisites in each tool description, including configuring an unqueued job or completing a candidate comparison. In this interface, agents invoke the tool when they judge their exploration sufficient; until then, they can continue gathering evidence. The tool does not assess scientific sufficiency.

Validation and termination. For tool submissions, we check the public schema, catalog identifiers, ordered numeric power bounds, and agreement between a ranking and its top choice. At most three attempts are allowed; errors reveal no scientific feedback. The first accepted submission locks the answer and ends the task. Structured final text is checked by the answer grader, including its source-reference format. Scientific correctness, evidence coverage, and compliance are evaluated separately (Appendix C).

Table 6: Task-specific submission tools. All require a free-text `rationale`; prerequisites summarize their descriptions.

Task and tool	Answer fields and their meaning	When to submit
T1 Inspect Picks `submit_inspect_picks`	`ncc_min`, `power_min`, `power_max`: lower NCC and lower/upper power bounds, matching the configured job. Each is a finite number or `null` for a native default.	After choosing thresholds and configuring the requested job, leave it unqueued.
T2 Select 2D `submit_select_2d`	`selected_templates`: distinct original class IDs from the source 2D Classification job. Preserve native numbering rather than renumbering by montage position.	After deciding which classes to retain.
T3 3D Cleaning `submit_3d_cleaning`	`pairs`: matched `particles` and `volume` sources in `J/output` form. Submit one paired entry per recommended class.	After selecting matched particle and volume inputs.
T4 Reconstruction `submit_reconstruction`	`ranked_job_uids`: every candidate exactly once, best to worst for the task objective. `best_job_uid` must equal the first entry.	After comparing all candidate reconstructions.
T5 Interaction `submit_interaction`	`job_config`: job `type`, input `connections`, and `params`. Connections map input slots to [Job UID, output name] sources; report every non-default parameter set.	After configuring the requested job, leave it unqueued.
T6 Gate `submit_gate`	`gate_id`: a publicly defined transition. `decision`: `pass`, `fail`, or `uncertain`. The rationale also states the proposed next action.	When ready to assess the applicable transition.
T7 Next Job `submit_next_job`	`job_types`: one to five distinct canonical job-type identifiers from the public catalog, ordered by predicted likelihood; use type IDs rather than display names or Job UIDs.	When ready to recommend the next processing step.
T8 Earliest Fail `submit_earliest_fail`	`observed_at`: where failure is visible. `earliest_non_ok`: the earliest job inferred to be scientifically compromised. These can differ; no execution error is required. The rationale explains the evidence, causal inference, and recovery plan.	When able to identify the failure observation and infer its origin.

Public Gate definitions and decisions.

`gate_id`	Meaning and readiness standard
`readiness_to_micrograph_gate`	Micrographs have passed quality checks and have usable CTF estimates for particle picking.
`micrograph_to_particle_gate`	Candidate particle positions cover relevant particles and views across usable micrographs.
`particle_to_volume_gate`	Cleaning removes unwanted particles while retaining useful particles and views, yielding a stack ready for reconstruction.
`volume_to_pdb_gate`	An interpretable, high-resolution volume is ready for atomic model building, without a blocking orientation-coverage problem.

`decision`: `pass` means that the selected transition's readiness standard is met; `fail` means that it is not met; `uncertain` means that the available evidence does not support a definite judgment.

B Details of Evidence Auditing

We explain why answer correctness and evidence acquisition are evaluated separately, then describe the interfaces and records used to audit access.

B.1 Why We Audit Evidence Acquisition

We distinguish agreement with a reference answer from documented inspection of its supporting evidence. Across the 1,260 autonomous runs, 74 pass the answer check but cover less than 50% of the reference evidence. A correct final label alone does not reveal which scientific checks the agent performed.

On Gate task `T6-134`, GLM-5.3 Flash reads refinement validation, local-resolution views, and map images before approving the transition to atomic modeling (Figure 6). The decision matches the expert answer. However, the trace does not show the reference half-map comparison, mask inspection, or the local-resolution job's global FSC being opened. Recorded coverage is 40%, so the run fails the evidence check and receives Task Score 80. The agent inspected relevant scientific evidence, including an approximately 2.78 Å FSC estimate and cFAR 0.89. The audit identifies incomplete reference checks that answer-only grading would leave invisible; it does not show that the omitted checks would reverse the answer.

A complementary Kimi K3 example appears in Figure 20a (Appendix E.4).

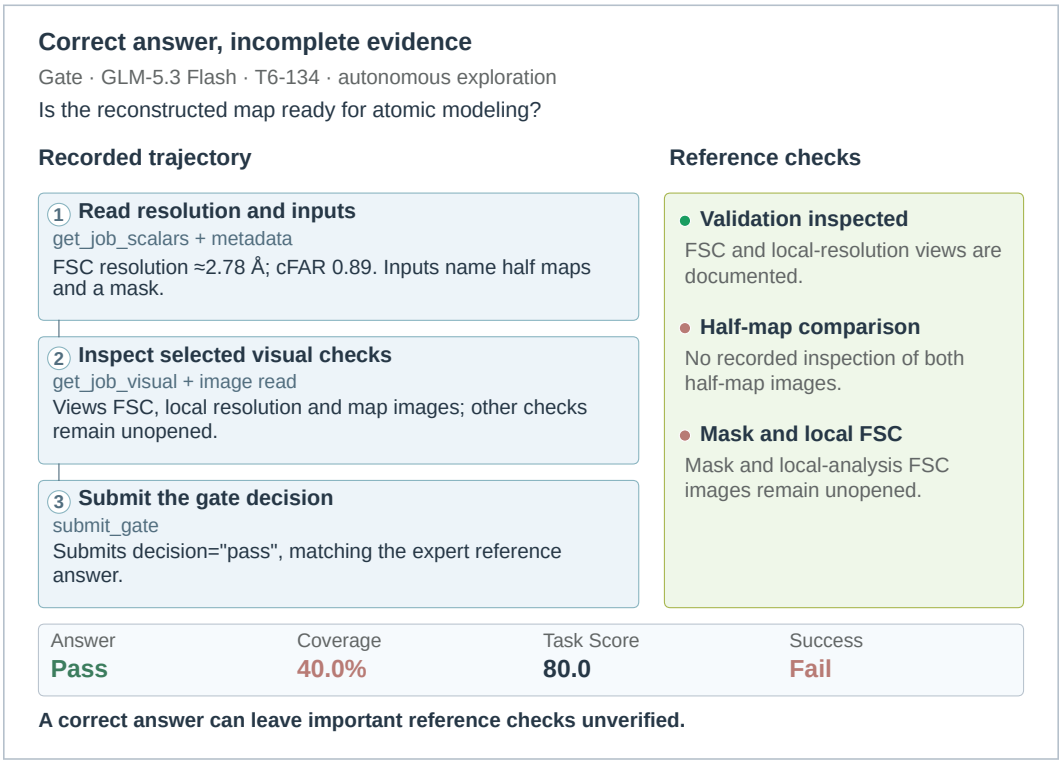


Figure 6: **A correct Gate answer with incomplete reference-evidence coverage.** Selected phases of the GLM-5.3 Flash trajectory on `T6-134`, followed by the frozen evaluation. Action and observation text is condensed from the record. Coverage denotes reference-evidence coverage; Success requires the answer, evidence, and compliance checks.

B.2 Evidence Tool Design

We provide six tools for three evidence types: job metadata, numerical results, and visual content (Table 7). Each tool is restricted to jobs in the task snapshot. The `field` argument selects type, parameters, or connections; `path` identifies a native numerical unit; and `image_id` identifies a job

image or registered micrograph preview. The listing functions help an agent discover sources without returning their contents. A retrieval can be made directly when the source is already known; a list call need not precede it.

Table 7: Read-only evidence interfaces. Listing returns source identifiers; retrieval returns content and identifiers for the source and successful read.

Function interface	Returned information
`list_job_metadata(job_uid)`	Available metadata fields and source IDs
`get_job_metadata(job_uid, field)`	Job type, parameters, or connections
`list_job_scalars(job_uid)`	Available numerical units and source IDs
`get_job_scalars(job_uid, path)`	The selected summary or numerical block
`list_job_visual(job_uid)`	Image/preview identifiers and source IDs
`get_job_visual(job_uid, image_id)`	Pixels of the selected image or preview

We use native numerical units, such as job summaries or result blocks, rather than counting every array element as separate evidence. Successful retrieval returns the content with its `evidence_id` and a fresh `retrieval_id`. Visual retrieval returns image pixels, not only a filename. Large numerical blocks can be loaded through an audited local-array reader; returning loading instructions alone is not a successful content read.

B.3 Making Evidence Access Auditable

We align expert annotation and agent access through a shared source catalog, content reader, and source-identifier scheme. An expert citation and an agent retrieval can therefore be matched to the same job metadata field, numerical block, or image. The annotation interface may present a preview, but its source identity remains the same. This avoids comparing annotations made through one evidence catalog with agent access recorded through another.

We include the following instruction in the tool descriptions:

> "Prefer the corresponding list/get evidence tools over direct cryosparc-tools API calls or file reads. If you use a direct route to discover or read evidence, also use the corresponding evidence tools to list those sources or get the same source evidence, respectively, so the access can be audited."

Agents are also instructed to inspect returned content and preserve the source and retrieval identifiers beside any quotation, summary, or computed result that uses it. If the same source cannot be exposed, they should report an audit gap rather than substitute unrelated evidence. These instructions guide behavior; they do not mechanically prevent an agent from using another route.

We record successful reads before the harness can transform their output, then match the source/retrieval pair to a subsequent accepted LLM input. The identifiers link a recorded read to the agent's input; an identifier by itself is not evidence content. Listings, failed reads, filenames, and unsupported citations do not establish acquisition. For direct visual delivery, the pixels must reach an image-capable model input. Derived numerical results remain linked to their source, but the audit does not independently establish that the calculation is scientifically correct.

We determine eligible sources from each run's frozen scoring reference. Following Section 3.1, $K_{t,i}$ is the number of reference groups for task i of family t, and $M_{t,i}$ counts the covered groups. Access to any accepted alternative covers a group; Inspect Picks additionally permits the reviewed processing record described below. We compute

$$\varepsilon_{t,i} = \frac{M_{t,i}}{K_{t,i}}. \tag{6}$$

Each group contributes at most once, and a source can cover several groups only when the reference explicitly assigns it to them. The evaluated cohort contains 924 scoring-reference groups and at least one per task. The evidence criterion is $\varepsilon_{t,i} \geq \tau$, with $\tau = 0.5$. This threshold measures reference coverage; it does not establish that the acquired evidence is sufficient for a correct scientific conclusion.

B.3.1 Evidence Groups in a Gate Task

We illustrate the grouping with Gate task `T6-118` (Figure 7). Its frozen reference contains three groups: the job type, the pick-statistics and final-thresholds plot, and picking overlays. The last group contains ten expert-annotated alternatives; acquiring any one covers that group once. These images are accepted alternatives for this particular reference check, rather than ten separately weighted requirements.

In the recorded autonomous runs, Kimi K3 covers only the job-type group. Qwen 3.8 Max also acquires the threshold plot and one of the ten annotated overlays, covering all three groups. With $K_{t,i} = 3$, their evidence coverage is $1/3$ and $3/3$, respectively. Other reads may support analysis, but neither sources outside these groups nor further members of a covered group add credit.

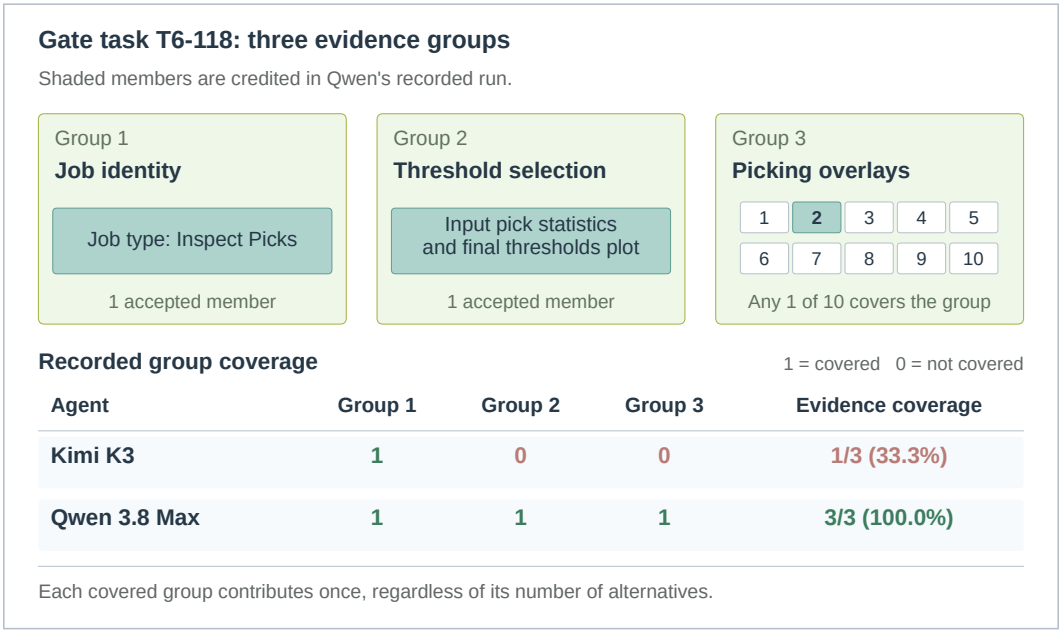


Figure 7: **Evidence-group matching on a Gate task.** The annotation contains 12 source alternatives in three equally weighted groups. Shading marks Qwen's credited members; overlay numbers are display aliases. A group receives one credit when an accepted alternative is acquired. Other source reads are omitted.

B.4 A Direct Image-Processing Route

We examine Qwen 3.8 Max on the Inspect Picks task `T1-005` (Figure 8). The run makes three metadata retrievals and six numerical retrievals, but no visual list or get call. A reviewed execution record nevertheless confirms that its Python code reads supplied micrograph pixels and successfully saves a derived image. A route based only on visual-tool counts would miss this pixel processing. The record shows what happened, but does not establish why the agent chose that route or whether the LLM viewed the resulting image.

Under the recorded Inspect Picks rule, this processing record covers one additional reference group, increasing coverage from $2/3$ to $3/3$. The run passes the evidence check at either value, so its Task Score is unchanged at 1.45, and its answer still fails. The example separates three questions: whether a tool was invoked, whether reference data were accessed or processed, and whether the scientific answer was correct.

The autonomous cohort contains 40 verified micrograph-processing records. Each is tied to its task, run, reference group, execution event, and hashed trajectory. Writing code without executing it, unsuccessful saves, and agent assertions do not qualify. We retain this route separately from verified image delivery and do not grant automatic credit to arbitrary native-file reads. Evidence coverage is consequently an audit of documented acquisition or permitted processing, not a measurement of

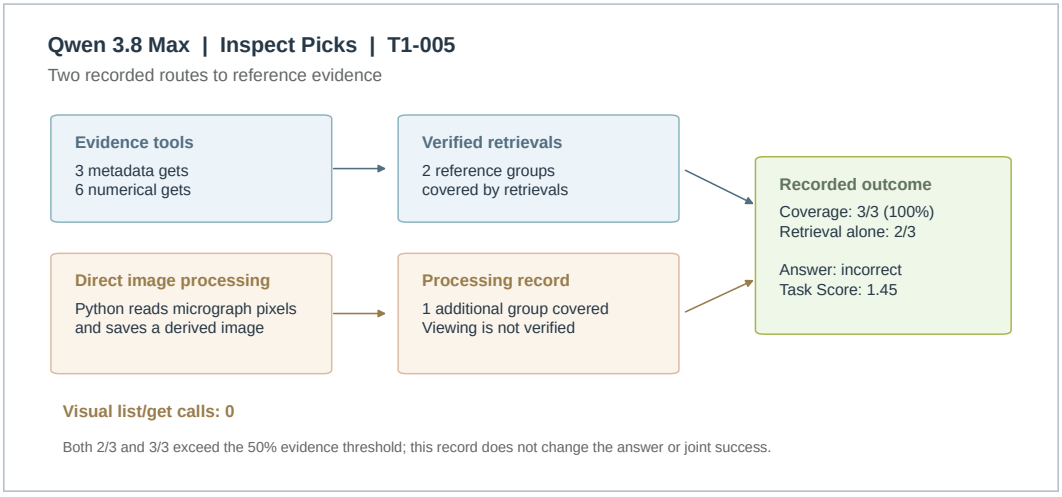


Figure 8: **Evidence access through tools and direct processing in one recorded run.** The diagram condenses the two observed access routes for Qwen 3.8 Max on task `T1-005`; it is not a complete event-by-event timeline. A reviewed processing record increases reference coverage, while image viewing remains unverified and the answer still fails.

visual attention or faithful scientific reasoning. A valid answer may also rely on useful sources outside the annotated reference.

B.5 List and Get Calls Across Agents

We count actual invocations of the six evidence tools in the primary traces of the 1,260 autonomous runs, separating list from get calls. Calls are deduplicated by their identifier and phase; failed and unfinished invocations are included as attempts. Shell/Python commands, submission tools, LLM request turns, and independent reviewer or subagent traces are excluded from these counts. Figure 9 reports medians within each task family, and Table 8 separates the three evidence types. All seven agents have their largest median get count on Earliest Fail, ranging from 39 to 84 calls per run. This task requires tracing workflow history across multiple jobs. The counts describe interface use, rather than the amount of information understood or the quality of the answer.

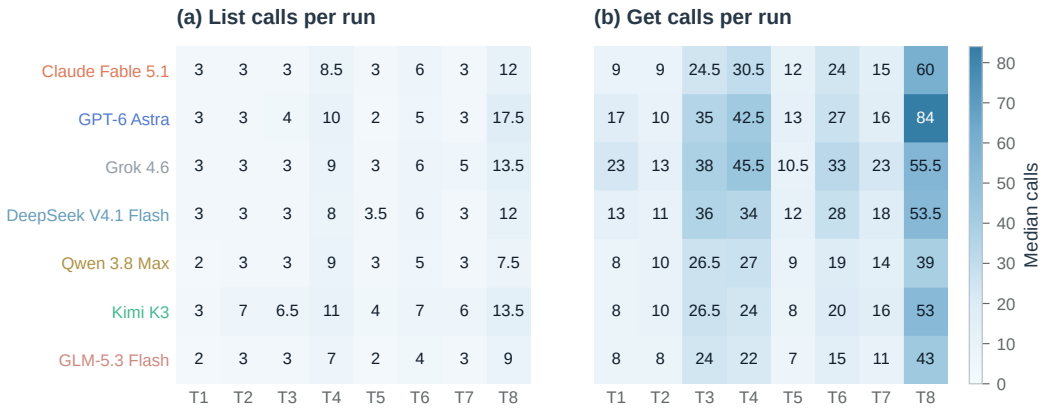


Figure 9: Median list and get invocations per run, by agent and task family. Each panel uses the same 180 tasks per agent. Counts describe attempted calls to the common evidence interface; direct API and file processing are not silently treated as tool calls.

Table 8: Median invocations per run for each evidence tool, over 180 tasks per agent. Numerical calls operate on native source blocks rather than individual array elements.

Agent	List			Get		
	Metadata	Numerical	Visual	Metadata	Numerical	Visual
Claude Fable 5.1	1	1	1	5	6	3
GPT-6 Astra	0	2	1	6	7	5
Grok 4.6	2	1	1	5	8	7.5
DeepSeek V4.1 Flash	1	2	1	6	7	3
Qwen 3.8 Max	1	1	1	5	6	2
Kimi K3	2	2	2	6	6	1
GLM-5.3 Flash	1	1	1	5	5	1

C Details of Task Grading and Results

We specify the answer-pass criteria and partial-credit rules for the eight task families, then describe evaluation settings and report per-task and aggregate results.

C.1 Answer-Pass Criteria and Partial Credit

We follow the notation in Section 3.1: t indexes task families, i indexes tasks within each family, $\mathcal{A}_{t,i}$ denotes the answer-pass event, and $\beta_{t,i}$ denotes the Answer Score. We assign $\beta_{t,i} = 100$ when the answer satisfies the task-specific pass criterion; otherwise, valid submissions receive partial credit according to Appendix C.1.2. Missing or invalid submissions cannot satisfy $\mathcal{A}_{t,i}$. We compute Task Score $q_{t,i}$ from $\beta_{t,i}$, evidence coverage $\varepsilon_{t,i}$, and operational compliance $\sigma_{t,i}$ using Equation 5.

C.1.1 Submitted Answers and Pass Criteria

Table 9: Task submissions and answer-pass criteria. All task-specific validity and job checks apply. T3/T4 use task-specific thresholds (defaults shown); T8 uses the score before the full-credit adjustment.

ID	Task	Submission	Answer-pass criterion
T1	Inspect Picks	NCC and power bounds	Precision ≥ 0.8; recall ≥ 0.9
T2	Select 2D	Retained class IDs	Particle-weighted F1 ≥ 0.8
T3	3D Cleaning	Particle–volume pairs	Exact-pair F1 ≥ 1.0
T4	Reconstruction	Candidate ranking	Accepted top choice; pairwise accuracy ≥ 0.66
T5	Interaction	Configured job and report	All configuration and reporting checks pass
T6	Gate	Gate ID and readiness	Both match the reference
T7	Next Job	Up to five ranked job types	At least one accepted recommendation
T8	Earliest Fail	Observation and boundary IDs	Accepted observation; valid lineage; score ≥ 80

C.1.2 Answer Scores and Partial Credit

The following rules specify answer credit before the full-credit adjustment in Appendix C.1. Passing answers receive $\beta_{t,i} = 100$. Task indices are omitted below for readability.

T1–T3: F1-based scoring. The score is $100F_1$, where

$$F_1 = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}.$$

Counts refer to individual particles in T1, particle-count-weighted class selections in T2, and exact particle–volume output pairs in T3. T1 compares the submitted selection with frozen retention labels without normalization by reference-threshold performance. T2 weights each class by its particle count. T3 requires both outputs in a pair to match. Scores are computed separately for each task. A forbidden job lifecycle change in T1 or an invalid class selection in T2 receives zero credit.

T4: Reconstruction. The score is $40h + 60p$, where h indicates an accepted best candidate at rank one and p is the fraction of reference candidate pairs ordered correctly. Incomplete or duplicate rankings receive no inferred ordering.

T5: Interaction. The score is

$$25z + 35\frac{k_I + c_I}{n_I + 1} + 40\frac{k_P + c_P + u}{n_P + 2}.$$

Here, z indicates the correct job type; k_I of n_I required input checks and k_P of n_P required parameter checks pass. Binary indicators c_I and c_P check whether the submitted inputs and parameters completely and accurately describe the saved configuration; u checks allowed additional parameters. The denominators are fixed by the task contract. Suppose the schema requires no input report, $c_I = 1$. Failure to configure exactly one job receives zero credit.

T6–T7: Discrete scoring. T6 assigns 100 for a correct gate and readiness decision, 50 for a correct gate with an incorrect decision, and zero for an incorrect gate. T7 assigns 100 when a valid top-five list contains an accepted job type and zero otherwise. Its separate rank-based diagnostic does not affect Answer Score.

T8: Earliest Fail. The score is $20v + 80\ell$, where v indicates an accepted observation point and ℓ is boundary credit. An exact boundary receives $\ell = 1$; predictions one, two, or three edges downstream receive 0.75, 0.5, or 0.25; predictions one or two edges upstream receive 0.5 or 0.2. Other predictions receive zero. Boundary credit is also zero unless the predicted boundary is an ancestor of, or equal to, the observation point. Causal explanations and recovery suggestions add no numerical credit.

C.2 Evaluation Settings and Aggregation

C.2.1 Cohorts and Agent Configurations

Table 10 lists the seven agent systems, harness versions, and representative reasoning-effort settings. We use the second-highest supported reasoning effort for each model–harness pair to balance reasoning capacity with evaluation cost. Appendix D.5.2 specifies the settings for the paired guidance runs.

Table 10: Agent configurations.

LLM	**Harness**	**Version**	**Reasoning effort**
GPT-6 Astra	Codex CLI	0.149.0	xhigh
Claude Fable 5.1	Claude Code	2.1.270	xhigh
Grok 4.6	Grok Build	1.0.30	high
DeepSeek V4.1 Flash	DeepSeek Harness	0.1.5-rc.1	high
Qwen 3.8 Max	Qwen Code	0.23.3	medium
Kimi K3	Kimi Code CLI	0.42.0	high
GLM-5.3 Flash	ZCode CLI	0.16.5	high

C.2.2 Aggregation and Cost Accounting

We follow Section 3.1: Experiment 1 averages within each task family and then equally across eight families; Experiment 2 averages the same 37 tasks for each agent and condition. Stars on Success Rate and Task Score denote task weighting. Family tables average only the corresponding tasks.

Cost. We sum all LLM inference charges per selected run and include unsuccessful runs. We use recorded charges when available and token-based estimates otherwise, estimating missing usage rather than counting it as zero. Cost follows the same aggregation weights; offline induction costs are reported in Appendix D.4.

C.3 Per-Task Scores and Answer-Pass Outcomes

Figures 10–16 show all 180 Task Scores $q_{t,i}$ in identical T1–T8 task order on shared axes. Color and the lower strip indicate answer pass; Task Score also incorporates evidence and compliance (Equation 5).

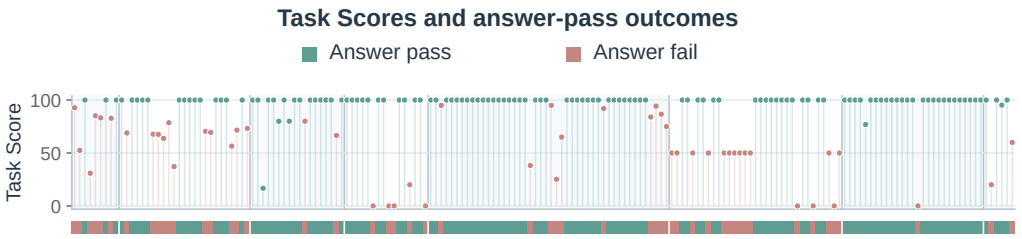


Figure 10: Claude Fable 5.1 (128/180 answers pass).

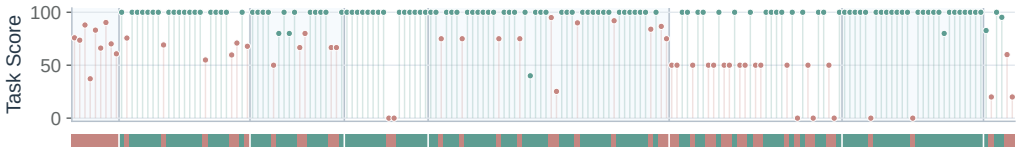


Figure 11: GPT-6 Astra (125/180 answers pass).

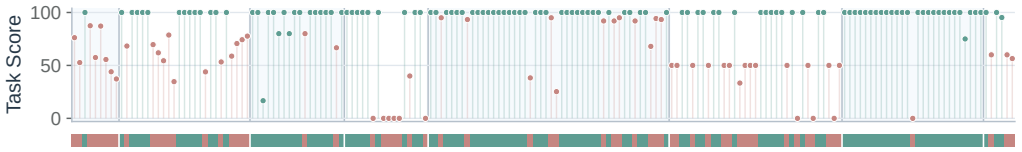


Figure 12: Grok 4.6 (118/180 answers pass).

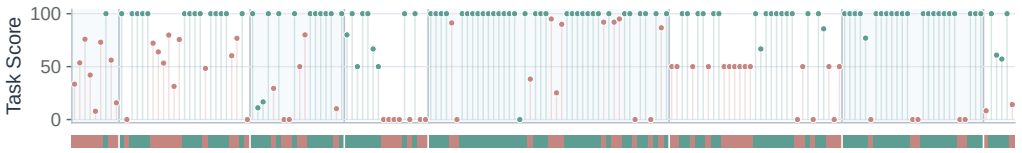


Figure 13: DeepSeek V4.1 Flash (113/180 answers pass).

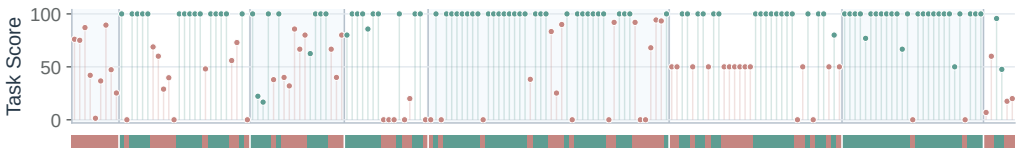


Figure 14: Qwen 3.8 Max (109/180 answers pass).

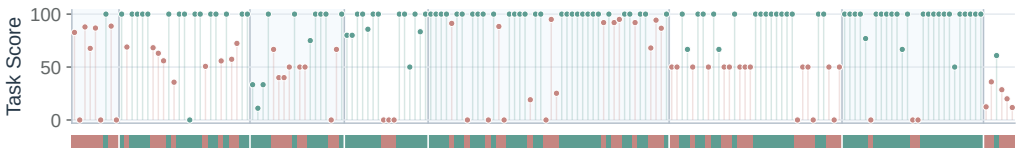


Figure 15: Kimi K3 (112/180 answers pass).

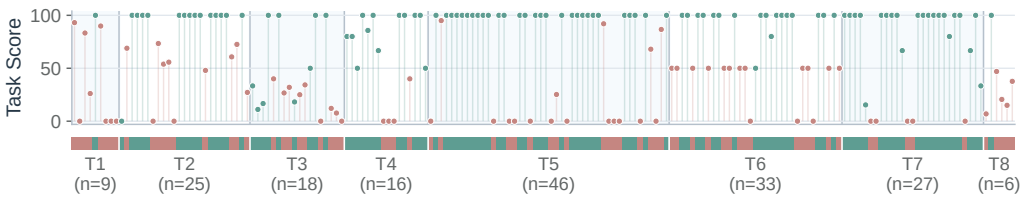


Figure 16: GLM-5.3 Flash (107/180 answers pass).

C.4 Complete Autonomous Results

We report autonomous results by task family in Tables 11–14, using the same 180-task cohort as Table 1. Each agent is evaluated on 9, 25, 18, 16, 46, 33, 27, and 6 tasks for T1–T8, respectively. Entries are average runs within each task family; metric definitions follow Section 3.1.

Table 11: Success Rate (%) by task family under autonomous exploration.

Agent	T1	T2	T3	T4	T5	T6	T7	T8
Claude Fable 5.1	33.33	56.00	72.22	68.75	78.26	54.55	92.59	50.00
GPT-6 Astra	0.00	76.00	61.11	87.50	73.91	48.48	88.89	16.67
Grok 4.6	11.11	52.00	72.22	56.25	73.91	48.48	92.59	33.33
Kimi K3	11.11	60.00	33.33	50.00	65.22	45.45	77.78	0.00
DeepSeek V4.1 Flash	11.11	56.00	55.56	31.25	71.74	45.45	77.78	33.33
Qwen 3.8 Max	0.00	60.00	33.33	50.00	65.22	51.52	81.48	0.00
GLM-5.3 Flash	11.11	56.00	22.22	37.50	63.04	48.48	62.96	16.67

Table 12: Task Score (0–100) by task family under autonomous exploration.

Agent	T1	T2	T3	T4	T5	T6	T7	T8
Claude Fable 5.1	80.83	85.01	90.19	70.00	94.57	72.73	95.44	79.21
GPT-6 Astra	71.69	91.93	88.33	87.50	93.22	69.70	91.85	63.00
Grok 4.6	66.42	81.85	90.19	58.75	95.07	69.19	95.37	78.62
Kimi K3	57.08	81.12	62.01	73.69	85.63	69.19	84.95	28.27
DeepSeek V4.1 Flash	50.87	78.46	66.53	46.67	87.07	69.77	80.63	56.73
Qwen 3.8 Max	53.37	74.98	68.35	61.61	79.92	73.64	88.65	41.28
GLM-5.3 Flash	43.60	74.41	39.28	65.77	71.02	70.61	72.67	37.85

Table 13: Answer Pass Rate (%) by task family under autonomous exploration.

Agent	T1	T2	T3	T4	T5	T6	T7	T8
Claude Fable 5.1	33.33	56.00	88.89	68.75	78.26	54.55	96.30	66.67
GPT-6 Astra	0.00	76.00	72.22	87.50	76.09	48.48	92.59	50.00
Grok 4.6	11.11	52.00	88.89	56.25	73.91	48.48	96.30	50.00
Kimi K3	11.11	64.00	55.56	81.25	65.22	51.52	88.89	16.67
DeepSeek V4.1 Flash	11.11	56.00	66.67	56.25	73.91	51.52	81.48	66.67
Qwen 3.8 Max	0.00	60.00	50.00	62.50	65.22	54.55	92.59	33.33
GLM-5.3 Flash	11.11	60.00	50.00	75.00	63.04	54.55	81.48	16.67

Table 14: Evidence Pass Rate (%) by task family under autonomous exploration.

Agent	T1	T2	T3	T4	T5	T6	T7	T8
Claude Fable 5.1	100.00	100.00	83.33	100.00	95.65	100.00	96.30	83.33
GPT-6 Astra	88.89	100.00	88.89	100.00	95.65	100.00	92.59	66.67
Grok 4.6	100.00	100.00	83.33	100.00	95.65	96.97	96.30	66.67
Kimi K3	100.00	96.00	72.22	68.75	95.65	93.94	85.19	33.33
DeepSeek V4.1 Flash	100.00	92.00	77.78	62.50	93.48	93.94	96.30	33.33
Qwen 3.8 Max	100.00	88.00	72.22	81.25	95.65	96.97	88.89	33.33
GLM-5.3 Flash	88.89	92.00	33.33	62.50	91.30	90.91	77.78	16.67

C.5 Guidance and Skill Results

We report all six metrics for AE, EL, ES, and IS on the same 37 held-out tasks per agent in Table 15. Metrics weight these tasks equally, following Table 2. The AE results cover only this test set and therefore differ from the full-cohort results in Table 1. Appendix D.5.2 documents the execution settings and differences between conditions.

Table 15: Complete results under AE, EL, ES, and IS on 37 held-out tasks per agent. All metrics weight tasks equally; stars mark Success Rate and Task Score. Rates are percentages; Task Score ranges from 0 to 100.

Agent	Setting	Overall Performance		Pass Rates (%)			Cost (USD/run)
		Success Rate*	Task Score*	Final Answer	Evidence	Compliance	
Claude Fable 5.1	AE	67.57	88.36	70.27	97.30	100.00	1.81
	EL	70.27	89.00	70.27	100.00	100.00	1.71
	ES	75.68	88.76	75.68	100.00	100.00	1.58
	IS	81.08	92.94	81.08	100.00	100.00	1.93
GPT-6 Astra	AE	64.86	86.02	70.27	94.59	100.00	1.46
	EL	72.97	87.87	72.97	100.00	100.00	1.19
	ES	72.97	88.61	72.97	100.00	100.00	1.16
	IS	70.27	86.63	72.97	97.30	100.00	1.68
Grok 4.6	AE	59.46	81.35	62.16	94.59	100.00	0.56
	EL	64.86	85.56	64.86	100.00	100.00	0.58
	ES	72.97	88.90	72.97	100.00	100.00	0.54
	IS	62.16	82.71	64.86	97.30	100.00	0.73
Kimi K3	AE	45.95	72.04	51.35	86.49	94.59	0.67
	EL	51.35	74.50	56.76	91.89	97.30	0.53
	ES	56.76	74.62	56.76	100.00	94.59	0.53
	IS	51.35	71.40	56.76	86.49	97.30	0.65
DeepSeek V4.1 Flash	AE	51.35	71.78	54.05	89.19	100.00	0.07
	EL	62.16	76.51	64.86	94.59	100.00	0.06
	ES	62.16	78.53	62.16	100.00	100.00	0.04
	IS	59.46	76.51	62.16	91.89	100.00	0.06
Qwen 3.8 Max	AE	59.46	76.48	62.16	91.89	100.00	0.52
	EL	59.46	73.86	59.46	97.30	94.59	0.43
	ES	54.05	69.98	54.05	100.00	94.59	0.40
	IS	54.05	76.99	62.16	86.49	100.00	0.57
GLM-5.3 Flash	AE	43.24	63.75	51.35	75.68	91.89	0.04
	EL	51.35	66.84	54.05	97.30	89.19	0.04
	ES	56.76	69.83	56.76	100.00	89.19	0.03
	IS	43.24	60.43	48.65	89.19	81.08	0.05

Changes in success on paired runs. We compare AE and IS on each task to identify changes that aggregate rates can hide (Table 16). IS turns 25 failed AE runs into successes, but also turns 14 successful AE runs into failures, yielding 11 additional successes overall. Claude Fable 5.1 succeeds on five additional tasks and retains every AE success. GLM-5.3 Flash has five changes in each direction, despite no change in its overall Success Rate.

Table 16: Changes in success from AE to IS on 37 paired tasks per agent. Success requires the answer, evidence, and compliance checks to pass.

Agent	**Failure → Success**	**Success → Failure**
Claude Fable 5.1	5	0
GPT-6 Astra	3	1
Grok 4.6	3	2
Kimi K3	4	2
DeepSeek V4.1 Flash	4	1
Qwen 3.8 Max	1	3
GLM-5.3 Flash	5	5
All agents	25	14

Differences across task families. We report the AE-to-IS changes by task family in Table 17. Of the 11 additional successful runs under IS, six are from Gate and two from Reconstruction. Improvements are not uniform: Inspect Picks has two additional successes but loses 13.05 Task Score points, while Earliest Fail decreases from two successful runs to one. The table includes task counts because some families have only two test tasks per agent.

Table 17: AE-to-IS changes by task family. Task counts are per agent; parentheses give paired task–agent outcomes across seven agents. Rates and scores use direct within-family means. Rate changes are percentage points, and Task Score changes are points.

Task family	Tasks (pairs)	Success Rate (%)			Task Score		
		AE	IS	Δ	AE	IS	Δ
Inspect Picks	2 (14)	7.14	21.43	+14.29	68.00	54.95	-13.05
Select 2D	5 (35)	45.71	51.43	+5.71	72.18	76.74	+4.56
3D Cleaning	4 (28)	35.71	35.71	0.00	75.09	73.78	-1.32
Reconstruction	3 (21)	61.90	71.43	+9.52	84.76	93.02	+8.25
Interaction	9 (63)	80.95	80.95	0.00	90.85	87.80	-3.05
Gate	7 (49)	36.73	48.98	+12.24	60.20	67.35	+7.14
Next Job	5 (35)	97.14	97.14	0.00	97.14	98.57	+1.43
Earliest Fail	2 (14)	14.29	7.14	-7.14	38.37	36.15	-2.22

Gate: deciding whether to proceed. Agents identify the correct gate on the same 42 of 49 runs under AE and IS, but fully correct answers increase from 19 to 24. The improvement therefore concerns the readiness decision after the gate has been identified. Combined successes increase from 18 to 24 because one correct AE answer fails the evidence check.

Inspect Picks: answer quality and execution. Mean Answer Score is nearly unchanged between AE and IS (68.00 versus 68.06). However, two IS runs lose credit through process checks: one fails the requirement to configure exactly one job, and the other fails the evidence check. Mean Task Score consequently falls from 68.00 to 54.95, even though successful runs increase from one to three out of 14.

Earliest Fail: more evidence without a better diagnosis. On task `T8-175`, Grok 4.6 covers 24 of 29 evidence groups under AE and 25 under IS. Both runs identify an accepted observation point. The predicted failure origin changes from the correct boundary under AE to a job-four dependency edge downstream under IS, reducing Task Score from 100 to 20. The additional evidence coverage therefore does not correspond to a better failure diagnosis in this case; the comparison does not identify which skill rule caused the regression.

Sensitivity to the evidence reference. One 3D Cleaning task has a revised numerical evidence reference in ES/IS relative to AE/EL. We repeat the comparison after excluding this task from every condition (Table 18). On the remaining 36 tasks per agent, IS improves mean Success Rate by 4.37 percentage points and Task Score by 1.54 points; five agents still improve in Success Rate. Within 3D Cleaning, however, the Task Score change reverses from -1.32 to $+2.94$ points. We therefore retain this sensitivity check alongside the primary 37-task results.

Table 18: Descriptive sensitivity of IS−AE to excluding task `T3-045`, whose evidence contract differs across conditions. The same task is removed for all agents and conditions; the main analysis retains 37 tasks per agent. Task counts are per agent, and pairs count task–agent outcomes. Changes are percentage points for Success Rate and points for Task Score.

Cohort	Tasks	Pairs	Success Rate (%)			Δ Task Score
			AE	IS	Δ	
Full holdout	37	259	55.98	60.23	+4.25	+1.12
Excluding T3-045	36	252	57.54	61.90	+4.37	+1.54
3D Cleaning	4	28	35.71	35.71	0.00	-1.32
3D Cleaning, excluding T3-045	3	21	47.62	47.62	0.00	+2.94

D Details of Skill Induction

We induce skills to make scientific checks reusable: which evidence to inspect, which alternatives to compare, and what findings justify a decision. We convert training trajectories and expert annotations into fixed textual procedures without updating model parameters, then apply them to held-out workflows. Appendix C.5 reports the complete evaluation; Appendix D.6 examines a paired trajectory example.

D.1 Task Split and Trajectory Selection

We hold out two or three workflows independently within each task family, yielding 143 training tasks and 37 test tasks. Each skill is induced only from its own task family, and the held-out tasks and their expert annotations are excluded. The training corpus contains 1,001 candidate autonomous runs from seven agents on the 143 training tasks. Eligibility requires a completed, gradable run with permitted operations, recorded actions, and matching supervision. Excluding 26 operationally noncompliant runs and one ungradable run leaves 974 eligible trajectories; selection retains 391 across all 143 tasks (Table 19).

For each task, candidate trajectories are stratified into four distinct groups based on two binary criteria: answer correctness and whether evidence coverage meets 50%. The detailed evaluation criteria for answer validity and evidence rules are provided in Appendices C and B.

The selection procedure is automatic; experts provide and review the reference annotations but do not hand-pick agent runs. Selection retains at most four trajectories per task. One representative is chosen from each occupied outcome group, favoring diversity relative to already selected traces. Diversity is the minimum Jaccard distance over sets of action categories, evidence-tool capabilities, and computation markers. If fewer than four traces are selected, at most one additional diverse trace is added. Ties use a deterministic hash with seed `ariadne-skill-distillation-v1`. Selection does not rank by agent identity, response length, or score; tasks without an answer-passing trajectory remain eligible.

Table 19: Task splits and trajectory counts for skill induction.

Task family	Tasks			Training trajectories		
	Total	Train	Test	Candidates	Eligible	Used
T1: Inspect Picks	9	7	2	49	44	20
T2: Select 2D	25	20	5	140	139	53
T3: 3D Cleaning	18	14	4	98	98	44
T4: Reconstruction	16	13	3	91	91	42
T5: Interaction	46	37	9	259	238	91
T6: Gate	33	26	7	182	182	65
T7: Next Job	27	22	5	154	154	61
T8: Earliest Fail	6	4	2	28	28	15
Total	180	143	37	1,001	974	391

D.2 Rule Induction, Consolidation, and Audit

We use GPT-6 Astra with ultra reasoning as the teacher for task-level induction, family-level consolidation, and source auditing, each conducted in a separate context. Each task packet contains the question and workflow context, selected agent-anonymous trajectories, evaluation feedback, ordered actions and observations, linked numerical and visual evidence, expert steps, and the reference answer. Supervision is restricted to training packets; full private particle-label tables are excluded.

Following experience extraction and workflow induction (Zhao et al., 2024; Wang et al., 2025), the teacher compares trajectories with expert supervision and proposes at most four conditional rules per task. Each rule specifies a trigger, procedure, decision criterion, uncertainty action, stopping condition, and exceptions, with supporting quotations and counterevidence. Support is identified as trajectory-based, expert-based, or both. Numerical and visual claims require inspection of the corresponding source evidence; recorded agent reasoning is not treated as independent evidence. The teacher may return no rule when reusable guidance is unsupported.

The 143 task analyses produce 521 candidate rules. Consolidation merges related rules within each family while preserving applicability conditions and conflicting evidence, retaining at most eight rules per family. Support is counted by distinct tasks and worktrees, rather than by the number of agent runs. We distinguish rules supported by at least two training workflows from rules supported by one workflow. The latter may be retained as expert-supported procedures with explicitly limited scope.

A separate audit checks source support, scientific scope, task-contract compatibility, and instance-specific information. The protocol permits one substantive revision followed by a final audit; unresolved major issues prevent inclusion. Source and format checks verify quotations, support counts, and removal of private identifiers and answers. The eight final skills contain 58 rules: 54 supported by at least two workflows and four retained with limited expert-supported scope. These checks verify support in the training records. Generalization is evaluated on held-out tasks, since generation and auditing use the same teacher.

D.3 Skill Format and Application

Each skill contains an entrypoint, `SKILL.md`, describing applicability and the core procedure, and `references/decision-rules.md`, containing conditional rules, exceptions, and uncertainty handling. Entrypoints contain 388–444 words and references 702–1,075 words.

Before evaluation, task, run, worktree, and specimen identifiers, source paths, training answers, and private grading targets are removed. Scientific terms, units, public parameter names, and answer formats remain. Each agent receives both files for its current task family, subject to the task's permissions and submission contract. Other-family skills, training trajectories, expert annotations, and support records are withheld. The 37 test tasks and their expert annotations are excluded from induction, and skills are fixed before evaluation without revision based on test outcomes.

We will release the selection and induction code, configurations, prompt templates, and final eight skills. Scientific records and annotations remain subject to the source permissions described in the paper's data statement.

D.4 Cost of Skill Induction

We account for task-level induction, consolidation, source audits, and retries, counting each teacher session and delegated call once. Total usage is 388.917 million input tokens, including 357.789 million cached tokens, and 2.760 million output tokens, including reasoning. At recorded rates of $10 per million uncached input tokens, $1 per million cached input tokens, and $50 per million output tokens (OpenAI, 2026), the estimated induction cost is $807.08. This is an API-equivalent estimate and excludes agent evaluation costs, which are accounted for separately in Appendix C.2.2.

D.5 Guidance Conditions and Execution Settings

We specify the information supplied under each guidance condition and the execution settings used in their comparison. The training/test partition is described in Appendix D.1.

D.5.1 Guidance Inputs and Information Boundaries

Autonomous exploration (AE) provides the task and workflow environment without additional guidance. The evidence-list condition (EL) adds the union of expert-annotated alternative evidence sources, identified by job, kind, and locator, deduplicated by source identity and ordered neutrally. The list reveals neither evidence-group membership nor source contents. The evidence-and-steps condition (ES) adds an ordered inspection and comparison procedure using the same source list, while withholding observed findings, intermediate Conclusions, and the reference answer. The induced-skill condition (IS) supplies the two files of the matching task-family skill described in Appendix D.3.

Guidance excludes answer-bearing information such as reference thresholds, retained-class lists, preferred reconstructions, accepted next jobs, gate decisions, and failure boundaries. Citations must resolve within the permitted snapshot; private grading targets and answer-bearing future jobs remain inaccessible. EL/ES use test-task expert annotations, whereas IS uses fixed procedures induced from training tasks. EL/ES therefore provide baselines with task-specific expert guidance; they do not guarantee an upper bound on performance.

Source pointers and inspection instructions do not count as evidence acquisition. Agents must retrieve content under the audit rules in Appendix B, comply with the task's operational constraints, and submit their own answers.

D.5.2 Paired Execution Settings

Condition AE reuses the designated autonomous runs on the held-out tasks. For guided conditions, the selected run is the latest actual attempt following infrastructure-error recovery, without ranking attempts by scientific outcome. Both successful and unsuccessful scientific outcomes are retained.

Task identities, snapshots, and scientific grading criteria are fixed across conditions. Evidence coverage uses each run's recorded scoring references, acquisitions, and reviewed processing witnesses. One 3D Cleaning task has a revised numerical evidence reference in ES/IS relative to AE/EL. The primary comparison retains its recorded outcomes; Appendix C.5 reports a sensitivity analysis excluding this task from every condition. Distinct source identifiers are not remapped or assumed equivalent.

The rollout-token limit is 150,000 for EL/ES/IS. Among the corresponding AE runs, 176 use a limit of 150,000 and 83 use 200,000. Native inference and submission settings are retained in the run records. The comparison therefore describes paired outcomes under the recorded settings rather than isolating guidance as the only changing factor. Skills remain fixed and are not revised using test outcomes.

D.6 Paired Trajectories with Skill Guidance

We compare DeepSeek V4.1 Flash on the reconstruction task `T4-063` under AE and IS (Figure 17). The two runs use the same frozen reference, harness version, reasoning-effort setting, and 150,000-token budget. The task compares two final reconstructions from different particle-cleaning histories; Candidate 2 is the expert-preferred reconstruction.

Under AE, the agent reads counts and directional diagnostics and performs additional array-based checks. It calls the maps "statistically tied" and favors Candidate 1's larger particle pool. Under IS, it retrieves each candidate's FSC and map views, traces the selected particle population, and explicitly invokes the skill rule for close comparisons. Matched auto-tight FSC values are approximately 3.10 versus 3.05 Å, and cFAR is 0.49 versus 0.59. Its final answer favors Candidate 2 while qualifying the difference as close rather than decisive.

Recorded reference coverage rises from 20% to 90%, and Task Score rises from 0 to 100. Some visual content is identical across the two runs but is accessed through different source-bound identifiers; AE also performs real map analysis. The coverage gain therefore includes improved reference matching and cannot be interpreted wholly as newly acquired scientific information. This paired trace shows additional documented checks and a revised decision aligned with the skill, but does not isolate the causal effect of an individual rule.

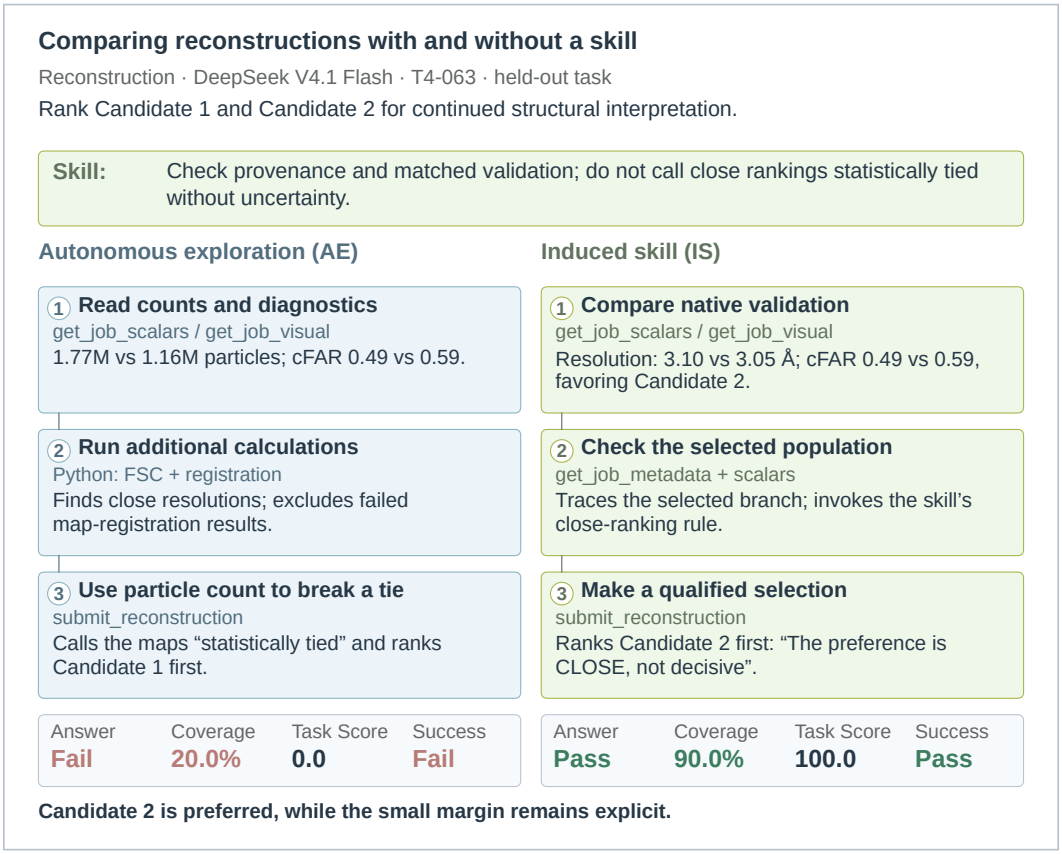


Figure 17: **A qualified reconstruction ranking with induced-skill guidance.** Paired DeepSeek V4.1 Flash runs on the same held-out task `T4-063`. Candidate labels are local aliases; the reference prefers Candidate 2. Boxes condense successive phases of each recorded trajectory, and quoted spans are exact excerpts. Coverage is the frozen reference-coverage metric, not the fraction of unique scientific information seen.

E Agent Trajectories and Failure Analysis

We use illustrative trajectories to examine acquisition, interpretation, and submission errors.

E.1 Incomplete Evidence Acquisition

On the Earliest Fail task `T8-177` (training partition), GLM-5.3 Flash inspects the fragmented class 1 used in a roughly 9.3 Å reconstruction and a coherent alternative, class 3 (Figure 18a). Other reference map and FSC comparisons remain unverified; coverage is 39.1%. The agent blames classification, while the expert identifies the later selection of class 1 for refinement, one dependency edge downstream. Here, classification also produced a coherent class; the expert's failure boundary concerns which class was selected. Missing comparisons and an incorrect boundary coexist, without establishing that a particular additional read would correct the answer (see also Figure 20b).

E.2 Incorrect Decisions despite Full Reference Coverage

On the 3D Cleaning task `T3-048` (training partition), GLM-5.3 Flash covers all reference evidence but chooses class 1 (18,153 particles) instead of the expert's structurally coherent class 2 (162,836 particles), receiving zero answer credit (Figure 18b). It inspects counts, slices, FSCs, and viewing directions, then treats full-map Fourier power as evidence of alignment-noise overfitting. That calculation measures power within individual maps; it does not measure correlation between independently reconstructed half maps. It therefore does not support the correlated-noise explanation used to reject class 2. Independent half-map validation is central to assessing overfitting (Scheres & Chen, 2012a). Another full-coverage selection error appears in Figure 20c.

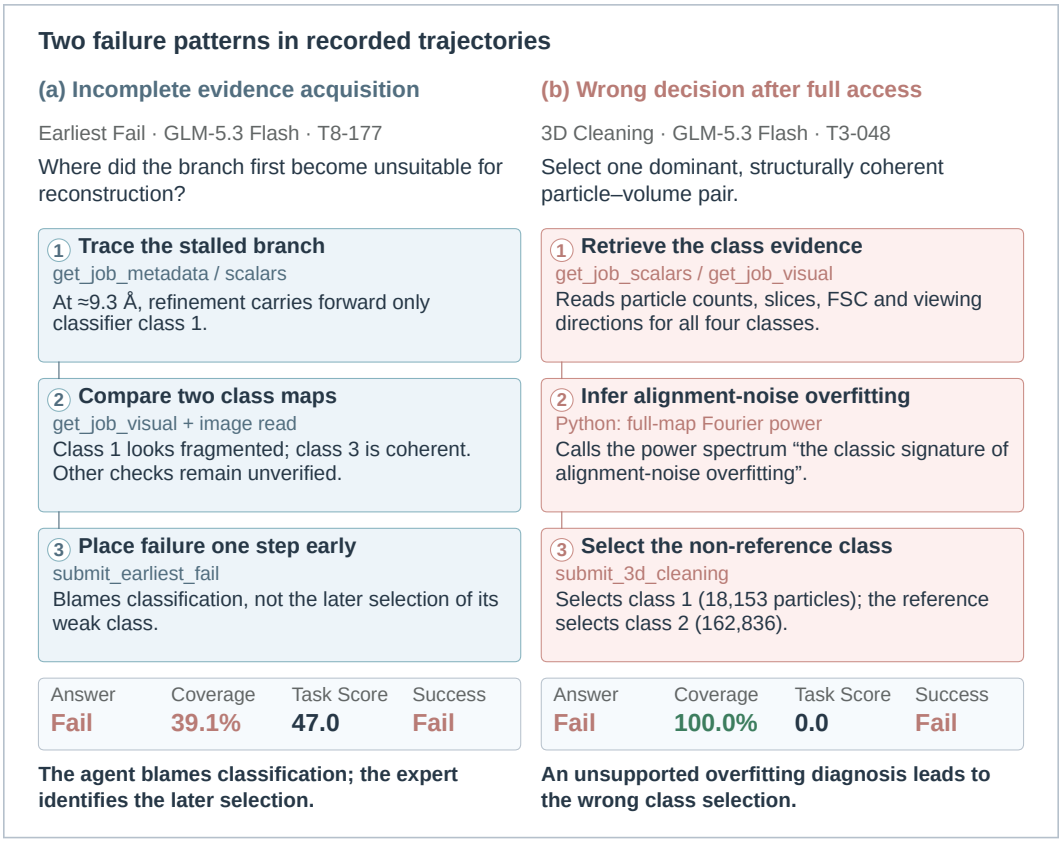


Figure 18: **Incomplete acquisition and incorrect interpretation.** **(a)** GLM-5.3 Flash places the failure boundary upstream of the damaging selection on `T8-177`. **(b)** It rejects the reference class on `T3-048` despite full reference coverage. Boxes summarize successive phases of distinct tasks; quoted text is an exact excerpt.

E.3 Expert Checks in Particle Selection

We compare particle selection with the expert annotation on a task used for skill induction. This example supports the Inspect Picks analysis in Section 4.1.

Inspect Picks: checking which particles are lost. The expert retained 99.6% of candidate picks in this task. GPT-6 Astra retained 1,822,468 particles, all matching the expert selection, but discarded another 1,863,911 reference particles. Precision was therefore 100%, while recall was only 49.4% (Figure 19). The expert checked how NCC and power cutoffs changed the picks on micrographs, balancing centering and removal of unwanted picks against particle retention. This example shows that selecting only valid particles can still discard much of the useful data. The induced skill therefore asks agents to inspect both retained and rejected picks when choosing thresholds.

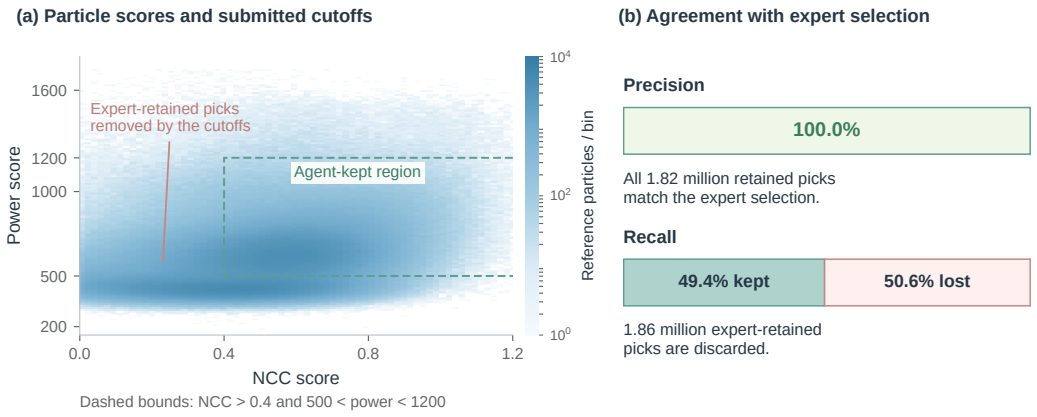


Figure 19: **High precision can hide particle loss in Inspect Picks.** An autonomous GPT-6 Astra run on a task used for skill induction. **(a)** Expert-retained particles and the agent's cutoffs. The NCC axis is zoomed to 1.2; the submitted selection has no NCC upper bound. **(b)** Precision and recall computed from the complete particle records. Particle-level reference labels were withheld from evaluated agents and skill induction.

E.4 Failure Cases across Agents

We compare three autonomous cases with different answer and evidence outcomes (Figure 20). Task partitions are marked in the panels.



Figure 20: **Additional autonomous cases across agents.** Three distinct runs, contrasting recorded checks and decisions with expert annotations. Coverage counts eligible reference groups. All three runs fail the overall Success criterion; the task partitions are indicated above each panel.

Gate: a correct answer with incomplete visual checks. Kimi K3's numerical checks support its decision but do not establish inspection of the threshold plot or picking overlays. The expert accepts residual contaminants after assessing particle coverage visually. This case shows incomplete reference verification, not evidence of guessing. Figure 7 shows the reference groups for this task.

Earliest Fail: an origin seven edges too early. GLM-5.3 Flash identifies an accepted observation point but places the origin seven dependency edges upstream of the reference. Filenames and a missing motion-correction node do not verify preprocessing history. The missed class comparison and incorrect attribution coexist; their causal relationship is not isolated.

3D Cleaning: additional classes after full acquisition. Multiple pairs are allowed, but Kimi K3's four pairs contain one reference pair and three extras, giving pair F1 0.4. Full coverage and compliance leave Task Score at 40. The error concerns class-quality assessment against the expert reference, not submission format. Appendix E.5 compares all seven agents on this same task.

E.5 Comparing Agents on the Same Task

We compare all seven agents on the same 3D Cleaning task, `T3-041` (Figure 21). All receive the same question and workflow snapshot. Six acquire all five annotated evidence groups, yet only Claude Fable 5.1, GPT-6 Astra, and Grok 4.6 pass the answer check.

Seven agents on the same 3D Cleaning task

T3-041 · Autonomous exploration

Which classes are supported for further refinement?

Agent	Evidence coverage	1 Further inspection	2 Stated judgment	3 Classes kept	Answer check Task Score
Claude Fable 5.1	100% (5/5)	FSC, viewing angles and class posteriors.	Keeps class 2; rejects diffuse maps with uncertain assignments.	0 1 **2** 3	Pass 100.00
GPT-6 Astra	100% (5/5)	FSC, directional precision and pixel-size checks.	Only class 2 supports progression; weak maps lack convincing structure.	0 1 **2** 3	Pass 100.00
Grok 4.6	100% (5/5)	FSC, directional precision and particle support.	Only class 2 has coherent density; the other maps are diffuse.	0 1 **2** 3	Pass 100.00
Kimi K3	100% (5/5)	Earlier views, FSC and viewing directions.	Treats distinct weak-map shapes as coherent populations, with caveats.	**0** **1** **2** **3**	Fail 40.00
DeepSeek V4.1 Flash	100% (5/5)	Particle-fit, class-posterior and occupancy analyses.	Self-fit and stable occupancy support all four classes, provisionally.	**0** **1** **2** **3** Wrong output notation	Fail 0.00
Qwen 3.8 Max	100% (5/5)	Initial views, FSC and viewing directions.	Stable map shapes across refinement support all four classes.	**0** **1** **2** **3**	Fail 40.00
GLM-5.3 Flash	20% (1/5)	Iteration-141 and initial class views; particle statistics.	Favors class 2; also keeps class 3 as a borderline second population.	0 1 **2** **3**	Fail 26.67

Expert reference: retain class 2 only. 0 1 **2** 3

Filled boxes = classes recommended; each class denotes a particle–volume pair.

Figure 21: **Seven agents on one 3D Cleaning task.** Autonomous runs on a training task share five reference groups: the job type and four final class views. Filled boxes mark recommended particle–volume pairs. The expert reference was withheld. Columns summarize decision stages rather than synchronized tool calls.

Different judgments after full acquisition. The three passing agents retain only class 2. Kimi K3 and Qwen 3.8 Max instead treat the weaker maps as distinct populations and retain all four classes. Their disagreement concerns whether those maps justify further refinement, not whether multiple selections are permitted.

Acquisition and submission also matter. GLM-5.3 Flash inspects initial and iteration-141 views rather than the annotated iteration-142 images; its 20% coverage does not mean an absence of visual inspection. DeepSeek V4.1 Flash names all four classes in structured final text but uses `J.output` references instead of `J/output`. No pair matches exactly, yielding Task Score zero. This reveals formatting as well as class-selection errors; submission interfaces are described in Appendix A.5.