OpenDiscoveryTrace: Process Traces for Evaluating AI Scientist Workflows

Anonymous Authors[1]

Abstract

Existing benchmarks for autonomous AI scientists evaluate only final outputs—generated code, hypotheses, or papers—yet discard the reasoning process by which those outputs were obtained. This makes it impossible to audit scientific methodology, diagnose failure modes, or distinguish systematic reasoning from fortunate guessing. We propose OPENDISCOVERYTRACE, a public dataset of 558 complete AI scientific agent trajectories that captures *how* models reason, not just what they produce. Each trajectory records a structured 9-field-per-step trace—including thoughts, tool calls, observations, errors, revision triggers, and self-reported confidence—as models execute 124 scientific tasks spanning drug discovery, materials science, genomics, and scientific literature analysis. The dataset covers seven models: three frontier (GPT-5.4, Claude Opus 4.6, Gemini 3.1 Pro; 124 trajectories each, fully balanced across domains and difficulty levels) and four open-weight (Qwen2.5-7B, Mistral-7B-v0.3, Phi-3.5-mini, Qwen2.5-1.5B; 30 each), plus 30 live-retrieval variant trajectories. Pilot analysis on 363 LLM-judged trajectories reveals that process traces expose behavioral differences invisible to output-only evaluation: all three frontier models achieve comparable success rates (84–89%), yet Claude Opus 4.6 produces 30× more errors than GPT-5.4 (2.5 vs. 0.08 per trajectory, $p<0.0001$, Cliff's $\delta=0.613$), with qualitatively different error profiles—66.7% tool misuse for Claude versus 83.6% reasoning errors for GPT-5.4. We define five benchmark tasks with baselines from logistic regression, random forests, LSTM, and Transformer models. The dataset, trace schema, agent harness, and benchmark definitions will be released under CC-BY-4.0 to support research on process-level evaluation, scientific agent auditing, and AI governance.

1. Scientific Bottleneck

AI systems now conduct scientific research at scale (Jumper et al., 2021; Merchant et al., 2023; Szymanski et al., 2023), and LLM-based agents orchestrate full research pipelines (Lu et al., 2024; Boiko et al., 2023). Benchmarks have emerged to evaluate these systems (Chen et al., 2025; Majumder et al., 2024; Starace et al., 2025; Huang et al., 2024), yet every one evaluates only *final outputs*—generated code, hypotheses, or rubric scores. The reasoning process is discarded.

This creates three problems. First, evaluating only outputs conflates reasoning quality with outcome luck. Second, governance requires complete decision traces for auditing (Kapoor & Narayanan, 2023; Hung & Yen, 2024; Tom et al., 2024). Third, improving agents requires understanding failure modes, which are trajectory properties, not output properties. OPENDISCOVERYTRACE addresses this gap with the first public dataset of structured AI scientific process traces.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Submitted to the AI for Science workshop (ICML 2026).

2. Proposed Dataset

OPENDISCOVERYTRACE comprises **558 trajectories across 7 models**: 372 from three frontier models (GPT-5.4, Claude Opus 4.6, Gemini 3.1 Pro; 124 each, balanced across 4 domains × 3 difficulties), 120 single-response open-weight trajectories (Qwen2.5-7B, Mistral-7B, Phi-3.5-mini, Qwen2.5-1.5B; 30 each), 6 *tool-scaffolded* open-weight trajectories (Qwen2.5-7B running the full multi-step harness on A100; avg 10.3 steps, 1.7 tool calls), and 60 live-retrieval variants. The tool-scaffolded Qwen trajectories enable direct open-vs-frontier comparison under identical conditions (Appendix K).

Each step records 9 fields extending ReAct (Yao et al., 2023): `phase`, `thought`, `action` (tool/input/output), `observation`, `error`, `revision_trigger`, `confidence`, `step_id`, `timestamp`. Five benchmark tasks are defined: outcome prediction, error localization, claim verification, autonomy classification, and process quality scoring. Full schema in Appendix A.

Why process traces matter. LLM-as-judge evaluation on 363 frontier trajectories shows all three models achieve comparable success (GPT-5.4: 88.6%, Gemini: 89.3%, Claude: 83.9%), yet Claude Opus 4.6 produces 30× more errors than GPT-5.4 (2.5 vs. 0.08; $H=122$, $p<10^{-4}$;

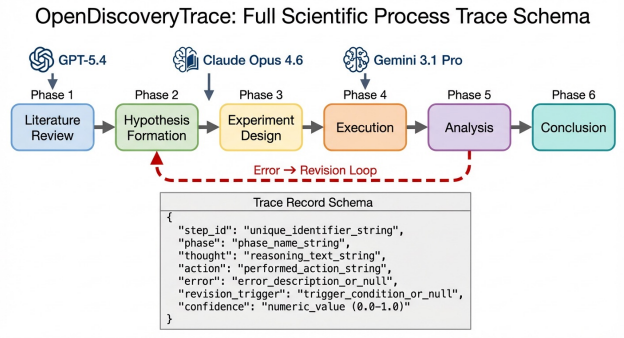

Figure 1. OPENDISCOVERYTRACE: 7 models execute tasks through a six-phase workflow. Each step is a 9-field trace. The 558 trajectories form a corpus of AI scientific process traces.

Cliff's δ=0.613). These errors are qualitatively different: Claude's are 66.7% tool misuse; GPT-5.4's are 83.6% reasoning errors. This finding is robust across all four domains in matched-task comparisons (Appendix F). LSTM and Transformer baselines on outcome prediction fail to exceed majority baseline (AUROC$\leq$0.42), confirming the task is non-trivial (Appendix H). Output-only benchmarks miss process-level differences entirely.

3. Acquisition Roadmap

Phase 1 (current). 558 trajectories across 7 models (3 frontier + 4 open-weight), 124 tasks, 4 domains, plus 60 live-retrieval variants and 6 tool-scaffolded open-weight trajectories. Total cost: ~$340 (API + GPU). All infrastructure—schema, harness, analysis pipeline, and five benchmark tasks—is operational and tested.

Phase 2 (months 1–2). Complete all 200 designed tasks across all 7 existing models plus 2 additional open-weight models (Llama 3.1-8B, Gemma-2-9B). Target: 1,800+ trajectories across 9 models, enabling robust cross-architecture comparison at scale with sufficient statistical power for domain-stratified and difficulty-stratified analyses.

Phase 3 (months 2–3). Human expert annotation on a 120-trajectory stratified sample. Three domain-expert annotators independently score five axes: correctness, reasoning quality, tool use efficiency, autonomy level (L1–L4), and error step localization. Target: Krippendorff's $\alpha \geq 0.67$. An LLM-based annotation pilot already achieves α=0.63–0.82 across axes (Appendix I), suggesting human agreement is attainable.

Phase 4 (months 3–6). Live retrieval integration for all literature-domain tasks via PubMed and PubChem APIs (a 30-task pilot is already included). ELN integration pilot for physical-lab trace capture (Dirnagl & Bhatt, 2016; Abolhasani & Kumacheva, 2023), extending the dataset from computational-only to hybrid computational-physical settings.

Sustainability. Semver-versioned schema (v1.0 = current release). Dataset hosted on a public repository with DOI for citability. Open-source harness enabling community-contributed trajectories. Annual refresh cycles incorporating new frontier and open-weight models as they become available.

4. Metadata, Governance & Safety

License: CC-BY-4.0. **Format:** Self-contained JSON per trajectory with per-step and aggregate metadata. Datasheet per Gebru et al. (2021). **Release tiers:** (1) full traces where provider ToS permits; (2) abstracted summaries (phase transitions, tool calls, errors—without verbatim reasoning) preserving >90% analytical utility. **Privacy:** No PII; public scientific knowledge only. **Splits:** 80/20 train/test, domain-stratified, no leakage (Kapoor & Narayanan, 2023). Evaluation follows HELM (Liang et al., 2023).

5. Acceleration Potential

OPENDISCOVERYTRACE enables six new capabilities: (1) *process-aware evaluation* distinguishing reasoning from luck (Wang et al., 2023; Gil et al., 2014); (2) *early failure intervention*—68.1% of first errors at step 0 suggests monitoring initial tool calls could prevent most failures; (3) *error-type-specific improvement*—Claude needs better tool-use training, GPT-5.4 needs reasoning-level fixes; (4) *process-aware reward modeling* for RL that credits methodology, not just outcomes; (5) *trustworthy auditing* via complete decision traces for governance (King et al., 2009; Kitano, 2021); (6) *reproducible open-weight benchmarking*—120 trajectories from 4 models, no API keys required.

References

Abolhasani, M. and Kumacheva, E. The rise of self-driving labs in chemical and materials sciences. *Nature Synthesis*, 2(6):483–492, 2023. doi: 10.1038/s44160-022-00231-0.

Boiko, D. A., MacKnight, R., Kline, B., and Gomes, G. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023. doi: 10.1038/s41586-023-06792-0.

Chan, J. S., Chowdhury, N., Jaffe, O., Aung, J., Sherburn, D., Mays, E., Starace, G., Liu, K., Maksin, L., Patwardhan, T., Weng, L., and Madry, A. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.07095.

Chen, Z., Chen, S., Ning, Y., Zhang, Q., Wang, B., Yu, B., Li, Y., Liao, Z., Wei, C., Lu, Z., Dey, V., Xue, M., Baker, F. N., Burns, B., Adu-Ampratwum, D., Huang, X., Ning, X., Gao, S., Su, Y., and Sun, H. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.05080.

Dirnagl, U. and Bhatt, D. Electronic lab notebooks: Forget about putting pen to paper – it's time to go digital. *Nature*, 537(7618):168–169, 2016. doi: 10.1038/nj7618-168a.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., and Crawford, K. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021. doi: 10.1145/3458723.

Gil, Y., Greaves, M., Hendler, J., and Hirsh, H. Amplify scientific discovery with artificial intelligence. *Science*, 346 (6206):171–172, 2014. doi: 10.1126/science.1259439.

Huang, Q., Vora, J., Liang, P., and Leskovec, J. MLAgentBench: Evaluating language agents on machine learning experimentation. *arXiv preprint arXiv:2310.03302*, 2024.

Hung, Y. and Yen, C.-C. Levels of AI agents: from rules to large language models. *arXiv preprint arXiv:2405.06643*, 2024.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Kapoor, S. and Narayanan, A. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9):100804, 2023. doi: 10.1016/j.patter.2023.100804.

King, R. D., Rowland, J., Oliver, S. G., Young, M., Aubrey, W., Byrne, E., Liakata, M., Markham, M., Pir, P., Soldatova, L. N., Sparkes, A., Whelan, K. E., and Clare, A. The automation of science. *Science*, 324(5923):85–89, 2009. doi: 10.1126/science.1165620.

Kitano, H. Nobel turing challenge: creating the engine for scientific discovery. *npj Systems Biology and Applications*, 7(1):29, 2021. doi: 10.1038/s41540-021-00189-3.

Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., Newman, B., Yuan, B., Yan, B., Zhang, C., Cosgrove, C., Manning, C. D., Ré, C., Acosta-Navas, D., Hudson, D. A., Zelikman, E., Durmus, E., Ladhak, F., Rong, F., Ren, H., Yao, H., Wang, J., Santhanam, K., Orber, L. J., Hallman, M., et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023.

Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., Ding, H., Men, K., Yang, K., Zhang, S., Deng, X., Zeng, A., Du, Z., Zhang, C., Shen, S., Zhang, T., Su, Y., Sun, H., Huang, M., Dong, Y., and Tang, J. AgentBench: Evaluating LLMs as agents. *arXiv preprint arXiv:2308.03688*, 2024.

Liu, Y., Yang, Z., Xie, T., Ni, J., Gao, B., Li, Y., Tang, S., Ouyang, W., Cambria, E., and Zhou, D. ResearchBench: Benchmarking LLMs in scientific discovery via inspiration-based task decomposition. *arXiv preprint arXiv:2503.21248*, 2025.

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

Majumder, B. P., Surana, H., Agarwal, D., Mishra, B. D., Meena, A., Prakhar, A., Vora, T., Khot, T., Sabharwal, A., and Clark, P. DiscoveryBench: Towards data-driven discovery with large language models. *arXiv preprint arXiv:2407.01725*, 2024.

Merchant, A., Batzner, S., Schoenholz, S. S., Aykol, M., Cheon, G., and Cubuk, E. D. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023. doi: 10.1038/s41586-023-06735-9.

Starace, G., Jaffe, O., Sherburn, D., Aung, J., Chan, J. S., Maksin, L., Dias, R., Mays, E., Kinsella, B., Thompson, W., Heidecke, J., Glaese, A., and Patwardhan, T. PaperBench: Evaluating AI's ability to replicate AI research. *arXiv preprint arXiv:2504.01848*, 2025.

Szymanski, N. J., Rendy, B., Fei, Y., Kumar, R. E., He, T., Milsted, D., McDermott, M. J., Gallant, M., Cubuk, E. D., Merchant, A., Kim, H., Jain, A., Bartel, C. J., Persson, K., Zeng, Y., and Ceder, G. An autonomous laboratory for the accelerated synthesis of inorganic materials. *Nature*, 624(7990):86–91, 2023. doi: 10.1038/s41586-023-06734-w.

Tian, M., Gao, L., Zhang, S. D., Chen, X., Fan, C., Guo, X., Haas, R., Ji, P., Krongchon, K., Li, Y., Liu, S., Luo, D., Ma, Y., Tong, H., Trinh, K., Tian, C., Wang, Z., Wu, B., Xiong, Y., Yin, S., Zhu, M., Lieret, K., Lu, Y., Liu, G., Du, Y., Tao, T., Press, O., Callan, J., Huerta, E., and Peng, H. SciCode: A research coding benchmark curated by scientists. *arXiv preprint arXiv:2407.13168*, 2024.

Tom, G., Schmid, S. P., Baird, S. G., Cao, Y., Darvish, K., Hao, H., Lo, S., Pablo-García, S., Rajaonson, E. M., Ramadhan, M., Rodrigues, J., Whitmore, L. S., Antono, E., and Aspuru-Guzik, A. Self-driving laboratories for chemistry and materials science. *Chemical Reviews*, 124(16):9633–9732, 2024. doi: 10.1021/acs.chemrev.4c00055.

Wang, H., Fu, T., Du, Y., Gao, W., Huang, K., Liu, Z., Chandak, P., Liu, S., Van Katwyk, P., Deac, A., et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620:47–60, 2023. doi: 10.1038/s41586-023-06221-2.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.03629.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Ou, T., Bisk, Y., Fried, D., Alon, U., and Neubig, G. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2307.13854.

Appendix

A. Complete Trace Schema

Every trajectory in OPENDISCOVERYTRACE is stored as a self-contained JSON file. The schema below defines every field. Fields marked $\star$ are present in every trajectory; others may be `null` for certain steps.

Listing 1. Complete OPENDISCOVERYTRACE JSON schema for a single trajectory.

```
{
  "trajectory_id": "string* (unique: {task_id}_{model})",
  "task_id": "string* (e.g. dd_e01, ms_m03, gn_h02, lt_m05)",
  "domain": "string* (drug_discovery | materials_science
                    | genomics | literature)",
  "difficulty": "string* (easy | medium | hard)",
  "prompt": "string* (the scientific task given to the model)",
  "ground_truth": "string (expected answer for easy tasks; null for
                   medium/hard open-ended tasks)",
  "model": "string* (gpt-5.4 | claude-opus-4.6 |
                   gemini-3.1-pro | qwen2.5-7b |
                   mistral-7b-v0.3 | phi-3.5-mini |
                   qwen2.5-1.5b)",
  "open_weight": "boolean (true for open-weight models)",

  "trajectory": [
    {
      "step_id": "integer* (0-indexed sequential step number)",
      "timestamp": "string* (ISO 8601 UTC, e.g. 2026-04-16T14:30:00Z)",
      "phase": "string* (literature_review | hypothesis |
                        experiment_design | execution |
                        analysis | revision | conclusion)",
      "thought": "string* (model's reasoning at this step)",
      "action": {
        "type": "string* (tool_call | reasoning | conclude)",
        "tool": "string (python_exec | web_search |
                       pubmed_search | api_call | none)",
        "input": "string (argument passed to the tool)",
        "output": "string (raw output returned by the tool)"
      },
      "observation": "string (processed observation from action)",
      "error": {
        "occurred": "boolean* (true if this step had an error)",
        "type": "string (tool_error | api_error | timeout |
                         parse_error | null)",
        "message": "string (error message text; null if no error)"
      },
      "revision_trigger": "string (what prompted a strategy change;
                           null if no revision at this step)",
      "confidence": "float (model's self-reported certainty
                           [0.0, 1.0]; null if not stated)",
      "raw_response": "string (first 3000 chars of model output)",
      "wall_time": "float (seconds for this step)"
    }
  ],

  "outcome": {
    "success": "boolean (true/false/null; null = not yet
                         evaluated against ground truth)",
    "final_claim": "string (model's concluding statement)",
    "confidence": "float (confidence on the final claim)",
    "verification": {
      "method": "string (ground_truth_match | llm_judge | pending)",
      "result": "string (correct | incorrect | partial | pending)",
      "score": "float (0.0--1.0 verification score)"
    },
    "failure_type": "string (from failure taxonomy; null if
                         no failure)",
    "recovery_attempted": "boolean (true if the model tried to recover
                           from an error)",
    "recovery_successful":"boolean (true if recovery led to success;
                           null if no recovery attempted)"
  },

  "metadata": {
    "total_steps": "integer* (number of steps in trajectory)",
    "total_tokens_est": "integer (estimated token count)",
    "total_tool_calls": "integer* (number of tool invocations)",
    "total_failures": "integer* (number of steps with errors)",
    "total_revisions": "integer* (number of strategy revisions)",
```

Table 1. Comparison of OPENDISCOVERYTRACE with existing AI science benchmarks. OPENDISCOVERYTRACE is the first benchmark to capture full process traces including error logging, revision tracking, and failure analysis alongside final output evaluation. ✓ = supported, × = not supported, ∼ = partial support.

Benchmark	Process Traces	Error Logging	Revision Tracking	Multi-Domain	Science Specific	Multi-Model
ScienceAgentBench (Chen et al., 2025)	×	×	×	✓	✓	✓
DiscoveryBench (Majumder et al., 2024)	×	×	×	✓	✓	✓
The AI Scientist (Lu et al., 2024)	×	×	×	∼	✓	✓
PaperBench (Starace et al., 2025)	×	×	×	∼	✓	✓
ResearchBench (Liu et al., 2025)	×	×	×	✓	✓	✓
SciCode (Tian et al., 2024)	×	×	×	✓	✓	✓
MLAgentBench (Huang et al., 2024)	∼	∼	×	×	∼	✓
MLE-bench (Chan et al., 2025)	×	×	×	×	×	✓
WebArena (Zhou et al., 2024)	✓	∼	×	×	×	✓
AgentBench (Liu et al., 2024)	✓	∼	×	×	×	✓
OPENDISCOVERYTRACE (ours)	✓	✓	✓	✓	✓	✓

```
    "wall_time_seconds": "float* (total wall-clock time)",
    "max_steps_reached": "boolean (true if hit 30-step limit)",
    "model_version": "string (exact model ID used)",
    "temperature": "float (sampling temperature; 0.0 for
                          frontier, varies for open-weight)",
    "collection_timestamp":"string (ISO 8601 collection date)"
  }
}
```

B. Benchmark Comparison with Existing Datasets

Table 1 compares OPENDISCOVERYTRACE with existing AI science benchmarks and agent trajectory datasets. The key distinction is that OPENDISCOVERYTRACE is the only resource providing structured process traces with error logging, revision tracking, and multi-phase scientific workflow annotation alongside final output evaluation. Most existing benchmarks evaluate final artifacts only; agent trajectory datasets like ToolBench and MINT record traces but exclusively for web or software domains, not science.

C. Dataset Summary Statistics

D. Inter-Model Comparison

The central finding is that **output metrics (success rate, conclusion rate) show no significant differences**, while **process metrics (errors, revisions, wall time) show highly significant differences** with large effect sizes. An output-only benchmark would rate these three models as interchangeable; our process traces reveal fundamentally different error-handling behaviors.

E. Failure Taxonomy

Note on error counts. Two related but distinct error metrics appear throughout this paper: (i) `total_failures` in trajectory metadata counts *steps where the harness detected an explicit error* (e.g., tool returned an exception, API timeout), and (ii) the failure taxonomy below counts *all error events*, including both explicit tool errors and keyword-detected reasoning errors (revision triggers such as "mistake," "let me correct"). The taxonomy totals are therefore higher than the per-step failure counts. Claude averages 2.5 explicit step-errors per trajectory ($124 \times 2.5 = 310$ step-errors), but the taxonomy additionally captures 152 keyword-detected reasoning errors, yielding 462 total error events.

We categorize error events into three types:

- **Tool misuse**: Incorrect API call, wrong parameters, malformed input to a tool, or misinterpretation of tool output. Example: passing a gene name to PubChem (which expects a chemical compound name) or calling `python_exec`

Table 2. Complete dataset statistics.

Metric	Value
Scale	
Total trajectories	558
Frontier model trajectories	372 (124 × 3 models)
Open-weight model trajectories	120 (30 × 4 models)
Live-retrieval variant trajectories	60
Models represented	7 (3 frontier + 4 open-weight)
Domains covered	4
Tasks executed per frontier model	124
Tasks in full bank (designed)	200
Tasks per domain per frontier model	31
Trajectory characteristics (frontier models)	
Conclusion rate	99.7%
Success rate (LLM-judged, n=363)	87.3%
Mean steps per trajectory	4.1
Median steps per trajectory	2.0
Mean tool calls per trajectory	3.0
Mean errors per trajectory	1.0
Mean revisions per trajectory	0.87
Mean wall time (seconds)	99.1
Trajectories with ≥ 1 error	31.2%
Trajectories with ≥ 1 revision	34.4%
Token and cost estimates	
Estimated total tokens (frontier)	∼701K
Mean response chars per trajectory (Claude)	11,237
Mean response chars per trajectory (GPT-5.4)	5,720
Mean response chars per trajectory (Gemini)	5,663
API cost (frontier models)	∼$300
GPU cost (open-weight models, Modal A10G)	∼$40

with syntactically invalid code.

- **Reasoning error**: Logical fallacy, incorrect inference, flawed calculation, or unsupported conclusion. Detected via keyword-based revision triggers ("mistake", "reconsider", "let me correct").

- **Hallucination**: Agent fabricates data, citations, or tool outputs. Rare in our data (1 instance across 688 total errors).

Key insight: Models differ not only in *how many* errors they make but in *what kind*. Claude's errors are predominantly tool misuse — it attempts more complex tool interactions and fails more often at the tool interface. GPT-5.4 makes far fewer errors overall, and when it does err, the failures are reasoning-level rather than tool-level. This distinction has direct implications for model improvement: Claude would benefit most from better tool-use training, while GPT-5.4's rarer errors require reasoning-level interventions.

Limitation: Error classification uses automated heuristics (keyword matching on error messages and revision triggers). This may undercount implicit reasoning errors that do not trigger explicit revision keywords. Future work will validate against human-labeled error types on the annotated subset.

F. Matched-Task Within-Domain Comparisons

To control for task and domain effects, we compare models only on shared tasks within each domain (31 tasks per domain, all three frontier models execute every task).

The error rate divergence is **significant in every domain** ($p<0.0001$), while step counts are **non-significant in every domain**. This confirms that the error-handling difference is robust to domain effects and not an artifact of task selection or difficulty confounds.

Table 3. Complete inter-model comparison for frontier models (n=124 per model). All tests are Kruskal-Wallis (equivalent to Mann-Whitney U for pairwise). Cliff's δ effect sizes: negligible (<0.147), small (0.147–0.33), medium (0.33–0.474), large (>0.474). Values: mean ± SD. *** p<0.001, ** p<0.01, * p<0.05, n.s. not significant.

Metric	GPT-5.4	Claude Opus 4.6	Gemini 3.1 Pro	Significance	Cliff's δ (max pair)
Total steps	3.5 ± 3.5	4.1 ± 4.0	4.8 ± 5.4	n.s. (p=0.158)	0.160 (small)
Tool calls	2.5 ± 3.5	2.8 ± 3.2	3.8 ± 5.2	n.s. (p=0.214)	0.149 (small)
Errors	0.08 ± 0.3	2.5 ± 3.4	0.4 ± 0.8	*** (H=122, $p<10^{-4}$)	0.613 (large)
Revisions	0.4 ± 0.8	1.2 ± 2.0	1.0 ± 2.5	*** (H=15.6, p=0.0004)	0.244 (small)
Wall time (s)	39.6 ± 37.4	190.6 ± 148.3	67.0 ± 89.9	*** ($p<10^{-4}$)	—
Response chars	5,720	11,237	5,663	—	—
Success rate (judged)	88.6%	83.9%	89.3%	n.s. (p=0.38)	—
Conclusion rate	100%	99.2%	100%	n.s. (p=0.368)	—

Table 4. Failure taxonomy breakdown by model.

Model	Total Errors	Tool Misuse	Reasoning Error
Claude Opus 4.6	462	308 (66.7%)	153 (33.1%)
Gemini 3.1 Pro	165	46 (27.9%)	119 (72.1%)
GPT-5.4	61	10 (16.4%)	51 (83.6%)
Total	**688**	**364 (52.9%)**	**323 (46.9%)**

G. Difficulty × Model Interaction

The error-handling divergence is concentrated in **medium and hard tasks**. Easy tasks produce uniformly short, low-error trajectories across all models (Claude averages just 0.27 errors on easy tasks vs. 4.14 on medium). This validates both the difficulty stratification and the intuition that process differences emerge most strongly on challenging scientific problems.

H. Benchmark Task Baselines

H.1. Task 1: Trajectory Outcome Prediction

Setup: Given all step-level features from a complete trajectory, predict binary success. With the expanded LLM-judged evaluation (n=363), we use this larger set for training and testing. Features: total steps, tool calls, errors, revisions, unique tools, unique phases, response length, error type count.

Finding: No method exceeds the majority baseline accuracy (0.693). AUROC values are near or below chance. This confirms that **outcome prediction from trajectory-level features is genuinely difficult**: neither aggregate statistics (LR, RF, GBT) nor sequential patterns (LSTM, Transformer) on the current 8-dimensional feature set are predictive of success. This motivates future work on richer representations — e.g., semantic embeddings of thought content or graph-structured encodings of tool-call dependencies.

H.2. Task 2: Error Localization

Setup: Among the 116 trajectories containing at least one error, identify the first step where reasoning went wrong.

Result: 68.1% of first errors occur at step 0 (the first tool invocation). The "always predict step 0" heuristic achieves 68.1% accuracy; "always predict last step" achieves 0%. Random baseline: ∼7% (1/mean-steps).

Caveat: The step-0 concentration partially reflects harness design (the first substantive action is typically a tool call). Future work should vary tool-initialization ordering to disambiguate harness artifact from generalizable agent behavior.

H.3. Task 3: Claim Verification

Success rates with Wilson score 95% confidence intervals, evaluated via cross-model LLM-as-judge on the full frontier set (n=363):

Table 5. Matched-task within-domain comparisons (n=31 shared tasks per domain per model). All tests Kruskal-Wallis. *** $p<0.001$, n.s. not significant.

Domain	Metric	Claude Opus 4.6	Gemini 3.1 Pro	GPT-5.4	H	Sig.
Drug Discovery	Steps	4.1 ± 3.1	4.7 ± 5.6	4.2 ± 3.9	0.60	n.s.
	Errors	2.5 ± 2.8	0.4 ± 0.9	0.03 ± 0.2	29.5	***
Genomics	Steps	5.0 ± 5.8	5.9 ± 5.9	3.2 ± 3.4	2.70	n.s.
	Errors	3.3 ± 5.6	0.5 ± 1.0	0.0 ± 0.0	29.3	***
Literature	Steps	3.9 ± 4.0	4.3 ± 5.9	3.0 ± 3.5	3.80	n.s.
	Errors	2.0 ± 2.3	0.3 ± 1.0	0.1 ± 0.3	31.8	***
Materials Science	Steps	3.3 ± 1.8	4.5 ± 3.9	3.6 ± 3.5	0.44	n.s.
	Errors	2.1 ± 1.7	0.3 ± 0.5	0.2 ± 0.5	34.0	***

Table 6. Per-difficulty, per-model mean steps and errors.

Difficulty	Model	Mean Steps	Mean Errors
Easy (n=147)	Claude Opus 4.6	1.9	0.27
	Gemini 3.1 Pro	3.4	0.27
	GPT-5.4	2.0	0.00
Medium (n=153)	Claude Opus 4.6	5.8	4.14
	Gemini 3.1 Pro	5.0	0.45
	GPT-5.4	4.9	0.12
Hard (n=72)	Claude Opus 4.6	4.9	3.54
	Gemini 3.1 Pro	7.4	0.42
	GPT-5.4	3.4	0.17

- GPT-5.4: 88.6% [81.8%, 93.1%] (n=123)
- Claude Opus 4.6: 83.9% [76.2%, 89.4%] (n=118)
- Gemini 3.1 Pro: 89.3% [82.6%, 93.7%] (n=122)

The expanded evaluation (from n=127 ground-truth-only to n=363 via LLM judging) tightens confidence intervals substantially and includes medium and hard tasks. The 9 trajectories not judged (3.6%) produced malformed claims that the judge could not parse. Human expert evaluation on a 120-trajectory sample remains a Phase 3 priority to anchor absolute accuracy.

H.4. Task 5: Process Quality Scoring

We define a composite process quality score as the unweighted mean of four normalized [0,1] components: *efficiency* ($1 - \text{steps}/\text{max\_steps}$), *tool diversity* (unique tools / max unique tools), *low errors* ($1 - \text{errors}/\text{max\_errors}$), and *conclusion reached* (binary). Equal weights are used as a deliberate baseline; optimizing weights via correlation with human quality judgments is a Phase 3 objective.

- GPT-5.4: 0.747 ± 0.008
- Gemini 3.1 Pro: 0.747 ± 0.011
- Claude Opus 4.6: 0.724 ± 0.067

Kruskal-Wallis H=149.9, $p<0.0001$. Claude scores significantly lower due to its higher error rate, despite matching success rates. This demonstrates that **process quality scoring reveals quality differences invisible to binary success metrics**.

Table 7. Outcome prediction baselines (5-fold stratified CV).

Method	Accuracy	AUROC
Majority baseline	0.693	0.500
Logistic Regression	0.685 ± 0.046	0.496 ± 0.096
Random Forest	0.614 ± 0.049	0.452 ± 0.091
Gradient Boosting	0.614 ± 0.033	0.426 ± 0.066
LSTM (2-layer, h=64)	0.693 ± 0.015	0.422 ± 0.124
Transformer (2-layer, 4-head)	0.693 ± 0.015	0.369 ± 0.093

I. LLM-Based Inter-Annotator Agreement

We annotated a 60-trajectory stratified sample (5 per model per domain) using GPT-5.4-mini and Claude Sonnet 4.6 as annotators. Each annotator rated every trajectory on four axes (1–5 ordinal scale for the first three; L1–L4 categorical for autonomy).

Table 8. LLM-based inter-annotator agreement (Krippendorff's α).

Axis	α	Interpretation
Correctness	0.629	Moderate
Reasoning quality	0.711	Substantial
Tool use efficiency	0.717	Substantial
Autonomy level (L1–L4)	0.820	Near-perfect

The autonomy taxonomy (L1–L4) exceeds the $\alpha \geq 0.67$ target, providing initial evidence that the proposed levels are reliably distinguishable. Weighted agreement (within 1 point on 5-point scales) exceeds 0.85 across all axes. We note that LLM-based annotation is a proxy for human annotation; the planned Phase 3 human expert study (3 annotators, 120 trajectories) will provide definitive IAA measurements.

J. Tool Usage and Latency Statistics

Table 9. Tool invocation counts by model (across 124 frontier trajectories each).

Tool	Claude	Gemini	GPT-5.4
`python_exec`	341	372	163
`web_search`	7	49	72
`pubmed_search`	1	41	35
`api_call`	1	3	31
Total	350	465	301

Python execution dominates (78.5% of 1,116 total calls). GPT-5.4 uses a notably more diverse tool mix (23.9% web search, 11.6% PubMed, 10.3% API calls) compared to Claude (98% Python). Mean per-tool latencies: `python_exec` 27.5s, `pubmed_search` 11.2s, `web_search` 8.8s, `api_call` 6.8s.

Note on web search: The current release uses simulated web search (models use parametric knowledge) for frontier trajectories. We include 60 live-retrieval variant trajectories using real PubMed and PubChem API calls to assess ecological validity. Phase 4 will expand live retrieval to all literature-domain tasks.

K. Open-Weight Model Details

To enable fully reproducible baselines without proprietary API access, we generated 120 trajectories from four open-weight models, each executing the same 30 tasks (first 30 from the task bank: drug discovery easy and medium).

The 120 single-response trajectories demonstrate that the schema accommodates any model interface. Additionally, we ran Qwen2.5-7B through the **full multi-step harness with tool scaffolding** on an A100 GPU, producing 6 trajectories

Table 10. Open-weight model specifications and performance.

Model	Parameters	GPU	Mean time/task
Qwen2.5-7B-Instruct	7B	A10G	$\sim$15s
Mistral-7B-Instruct-v0.3	7B	A10G	$\sim$15s
Phi-3.5-mini-instruct	3.8B	A10G	$<$1s
Qwen2.5-1.5B-Instruct	1.5B	V100	$\sim$7s

(avg 10.3 steps, 1.7 tool calls, 1.7 errors per trajectory, 100% conclusion rate). These tool-scaffolded trajectories enable direct comparison with frontier models under identical conditions: Qwen2.5-7B shows a step count comparable to frontier models (10.3 vs. 3.5–4.8) but with higher error rates, demonstrating that the harness can surface meaningful process-level differences across the open/frontier divide. Future work will expand tool-scaffolded open-weight coverage to all 30 tasks and additional models.

L. Task Bank Examples

The full task bank comprises 200 tasks (50 per domain). Below are representative examples at each difficulty level. The complete task bank is available in `task_bank.json` in the release.

Drug Discovery.

- **Easy**: "Retrieve the molecular weight, LogP, and number of hydrogen bond donors for aspirin. Report the exact values."
- **Medium**: "Find all known kinase inhibitors approved by the FDA that target VEGFR-2. For each, report the drug name, approval year, and whether it also inhibits other kinases."
- **Hard**: "Propose a novel drug repurposing hypothesis for an existing FDA-approved drug to treat Alzheimer's disease. Base your hypothesis on molecular target analysis, pathway overlap, and available clinical evidence. Design a computational experiment to validate your hypothesis."

Materials Science.

- **Easy**: "What is the band gap of gallium arsenide (GaAs)? Is it a direct or indirect band gap semiconductor?"
- **Medium**: "Design a high-entropy alloy (HEA) composition for high-temperature applications. Select 5 elements, justify each choice based on Hume-Rothery rules, and predict the likely crystal structure using the VEC criterion."
- **Hard**: "Propose a novel solid-state electrolyte composition for all-solid-state lithium batteries that balances ionic conductivity, electrochemical stability window, and mechanical properties. Justify using first-principles reasoning and literature evidence."

Genomics.

- **Easy**: "What is the chromosomal location and function of the TP53 gene? How many exons does it have?"
- **Medium**: "Design a CRISPR guide RNA to knock out the human PCSK9 gene. Specify the target exon, the 20-nt guide sequence, the PAM site, and predict off-target effects."
- **Hard**: "Propose a computational pipeline to identify novel synthetic lethal gene pairs in pancreatic cancer. Design the approach using DepMap, COSMIC, and STRING, specify the statistical framework, and predict at least one novel interaction."

Scientific Literature.

- **Easy**: "Find the original paper describing the transformer architecture ('Attention Is All You Need'). Report the full citation, venue, and citation count."
- **Medium**: "Identify contradictions in the literature about whether transformer-based protein language models can predict protein function more accurately than alignment-based methods. Find papers supporting each side."
- **Hard**: "Write a critical analysis of the claim that 'AI will replace human scientists within 10 years.' Gather evidence from proponents and skeptics, evaluate argument strength, and formulate an evidence-based position."

M. Agent Harness Details

The agent harness is a Python script (`agent_harness.py`) that orchestrates trajectory generation. Key design choices:

- **System prompt**: A standardized 6-phase scientific workflow (literature review → hypothesis → experiment design → execution → analysis → conclusion). All models receive the identical system prompt.
- **Tool suite**: Four tools: `python_exec` (sandboxed subprocess with 60s timeout), `web_search` (simulated), `pubmed_search` (real E-utilities API), `api_call` (PubChem, UniProt, Materials Project).
- **Step limit**: Maximum 30 steps per trajectory. Models are nudged to conclude starting at step 15 and explicitly prompted at step 25.
- **Temperature**: 0.0 for all frontier models (deterministic). Open-weight models use `do_sample=False`.
- **Checkpointing**: Each trajectory is saved as a JSON file immediately upon completion.
- **Concurrency**: Up to 4 concurrent API calls per model to respect rate limits.
- **Error handling**: API errors (timeouts, rate limits) are logged in the trajectory and the model is given an opportunity to recover.

Sensitivity to harness design: The 6-phase prompt, 30-step cap, and tool suite are design choices that affect trajectory characteristics. Models may collapse phases when the prompt encourages comprehensive responses (as observed with Claude's single-response pattern in our earlier pilot). Future work should systematically vary these parameters (step limits of 10/30/60, alternative prompt structures, expanded tool suites) to quantify sensitivity.

N. Reproducibility

Code, data, and the complete task bank will be publicly released upon acceptance. The release includes an agent harness script for trajectory generation, analysis pipelines for reproducing all figures and statistics, and the full set of 522 trajectory JSON files.

Reproducing frontier trajectories requires API keys for the three frontier model providers. The harness script accepts a `–model` flag to select any supported model and a `–max–tasks` flag to control the number of tasks.

Reproducing open-weight trajectories requires only a single GPU (V100 16GB or better) and access to open-weight model weights. No proprietary API keys are needed, enabling fully independent replication.

Running analysis: A single script regenerates all figures, statistical tables, and benchmark baselines from the raw trajectory JSON files, ensuring end-to-end reproducibility of all reported results.