PARTICLEBENCH: CAN AI AGENTS CLOSE THE REAL-TO-SIM-TO-REAL LOOP IN EXPERIMENTAL SCIENCE?

Anonymous authors
Paper under double-blind review

ABSTRACT

Agents now contribute to research mathematics because a proof assistant verifies every candidate cheaply. Experimental science verifies by experiment, which is slow and costly; where data are too scarce to learn from, search at scale needs a simulation built and calibrated against the instrument, the real-to-sim-to-real loop. We present ParticleBench, a benchmark for this ability built in particle physics, the field with the most mature simulation stack. Its 10 task families, modelled on 7 experiments across 5 measurement types, each have a generator with hidden truth, a reference solution written by the task's author and a verifier whose gates encode what the field's procedure guarantees, with a score only when every gate passes. The agent submits a program, not numbers, may query the verifier up to 10 times on a study draw, and is graded once on an exam draw with new truth after its program is frozen, as in a blind analysis. Unlike benchmarks graded against a fixed target, feedback is rationed so that a simulation of the agent's own is the only cheap checker left.

1 INTRODUCTION

Research agents built on frontier language models have begun to contribute to frontier mathematics [18], and a key enabler is cheap, reliable verification: once a problem is formalized, a proof assistant such as Lean 4 [8] checks every candidate proof automatically, so an agent can search at scale. In experimental science – physics, chemistry, biology – the checker is the experiment itself, and its cost and latency make testing every candidate impractical.

Two routes lead around this bottleneck. AlphaFold illustrates the first: train a deep neural network on a large amount of data and predict the folded structure directly [12]. Here experimental verification is paid up front, in decades of solved structures: the network compresses those data into a map from sequence to structure, and neither causal analysis nor a per-candidate experiment is needed. When massive data are unavailable for the system or the intervention of interest, the practical alternative is a simulation, which refines theories and hypotheses and makes verification cheaper.

In this second regime a research agent must close the loop that robotics calls real-to-sim-to-real (R2S2R) [13]: build a simulation of the experiment that keeps the factors that move the result and drops the rest, refine and calibrate it against experimental results, then test research hypotheses in it before verifying them on the instrument. The reasons are two-fold. First, a real experiment is full of surprises and imperfections, so an agent that cannot close the loop is naive and weak, and produces only noise. Second, without the ability to build a reliable simulation, the scalable search that AI is good at is throttled by the throughput and cost of experimental verification. Closing the loop is also the human experimentalist's core skill.

Research agents could change the trade-off that leaves many laboratories without a simulation of their own. Although simulation is established across the sciences [6], building and maintaining a simulator for one particular experiment can demand more expert effort than a limited program of trials justifies, so many laboratories maintain none and their experimentalists rely on experience and intuition. They would change it in three ways: by automating parts of model construction and calibration, by condensing and consolidating the experience and intuition of how to make an experiment succeed, and by amortizing the resulting investment over much larger searches. This change could be as transformative as the invention of the printing press.

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.

Particle physics is the best-placed discipline for benchmarks and RL environments that target the ability to close the R2S2R loop, because it has the most mature simulation stack, shared across the community and validated against data. Detector simulation toolkits such as Geant4 have been developed and tuned for decades [1, 2]. Because an experiment takes a decade or more and a mistake is expensive, every large collaboration maintains, on top of them, a simulation good enough for the numbers that enter a published measurement: detector response, efficiencies, backgrounds. The methodology of building such a simulation, tuning it and balancing accuracy against speed has been crystallized into code.

This motivates ParticleBench, a benchmark of particle-physics research tasks. The agent submits an analysis program, may query the environment for feedback at most 10 times, and is graded once on a blind exam with new truth (Section 3). The grade is gate-then-score: a field accepts a measurement on its procedure, since a laboratory has no truth to score against, so gates are pass/fail checks encoding what the field's standard operating procedure (SOP) guarantees, such as the bias of a reconstruction, and a score counts only when every gate passes. The exam redraws the truth as well as the fluctuations, so a program tuned to the study draw loses there, while a procedure that holds keeps its grade. The chances are few, and the pressure of exhausting them should drive the agent to build its own simulation and test its hypotheses there first.

The closest benchmarks grade against a target fixed before the agent starts (Section 2). Stargazer returns the verdict on the instance the agent iterates against [14]; Collider-Bench has the agent build a simulation pipeline but scores it against a published yield [9]; DiscoverPhysics holds out particles but not the law that governs them [20]; the FAIR Universe challenges grade the coverage of submitted intervals, but as a competition on one fixed dataset [3]. A fixed target can be reached by iterating against it, which is engineering however hard the task; research is work whose target is unknown, and blind analysis, the freeze before the data that count, is how physics makes that testable. Here the truth that grades the agent is drawn after its program is frozen, and feedback is rationed so that a simulation of its own is the only cheap checker left.

ParticleBench has 10 task families modelled on 7 experiments: the Planck CMB survey, the Fermi-LAT γ-ray telescope, the IceCube neutrino telescope, the GERDA $0\nu\beta\beta$ search, the JUNO reactor-neutrino detector, the Hyper-Kamiokande water-Cherenkov detector and the BESIII detector at the BEPCII e^+e^- collider. They span 5 measurement types: reconstruction, calibration and resolution, parameter estimation, hypothesis discrimination, and search with an upper limit; each family ships its generator, verifier and oracle, a reference solution by the task's author (Section 3). The authors include experts from the particle-physics domain; each oracle was written with the generator in view but as if blind to it, so it is the yardstick of Q1 rather than the experiment's own best analysis.

Our contributions are:

- **Artifact.** 10 task families with generator, verifier and oracle, released trajectories and the ledger of every verdict (Section 3).
- **Method.** Blind analysis as a benchmark protocol: a program, not a number, is submitted; a run is up to 10 study submissions on one draw and one exam draw with new truth; the verdict is gate-then-score, with three gate kinds – contract, performance and statistical – and the coverage gate of the upper-limit search as the worked example; and the verifier is validated by oracle passes (Sections 3.3–3.6).
- **Findings.** Section 5 reports two main findings. First, frontier agents remain below the oracle in aggregate exam performance. Second, modeling quality shows a stronger observed association with exam performance than SOP ratings, suggesting that adequate representations of measurement-relevant relationships matter alongside procedural execution.

Section 2 discusses related work, Section 3 the benchmark design, and Sections 4 and 5 the setup and the results.

2 Related work

Scientific reasoning benchmarks. Model releases measure scientific capability with questions that have reference answers: multiple choice in MMLU and GPQA [10, 16], harder multiple-choice and short-answer questions graded by a model in Humanity's Last Exam [15], and expert rubrics that credit intermediate results in FrontierScience [19]. An experimental analysis must also turn

	Generator hidden	Truth redrawn	Held-out exam	Feedback while exploring	Chances capped	Program re-run	Gate before score
Stargazer	$\circ^a$	✓	$\times^a$	G	✓	×	$\circ^a$
Collider-Bench	$\times^b$	×	×	W	$\circ^b$	×	$\circ^b$
DiscoverPhysics	✓	$\times^c$	$\circ^c$	W	✓	✓	$\times^c$
ReplicationBench	–	×	$\circ^d$	W	$\circ^d$	×	×
gwBenchmarks	$\circ^e$	$\circ^e$	$\circ^e$	W	×	✓	✓^{e}
FAIR Universe HiggsML	×	✓	✓	G+W	$-^f$	✓	$\times^f$
ParticleBench	✓	✓	✓	G	✓	✓	$\circ^g$

Table 1: Grading protocols of the closest benchmarks, read from the paper bodies. ✓ holds in the main protocol; ∘ holds in part; × absent; – not applicable or not stated. Feedback: G, the grader's verdict on a submission; W, the agent's own runs or probes of the world. Notes: [a]the Keplerian model to fit is given; the best of 3, 5 or 10 submissions on the same instance counts; four thresholds on the answer, with no score after. [b]a public approximate simulator is provided; one submission within 2.5 hours; format and provenance gates only. [c]22 curated worlds with fixed laws; held-out particles of the same world; a pass is two thresholds on error and explanation. [d]no generator; target values masked from the manuscript but published; 6 hours and 5M tokens, submissions uncounted. [e]public data and tools, with hidden test waveforms in 2 of 8 tasks; an integrity gate on format and completeness precedes outcome scoring; no common budget. [f]leaderboard feedback plus local simulation; a private final set; coverage is a penalty inside the score; the submission cap is not stated. [g]a contract gate in every family and a physics or statistical gate in three of 10.

observations into a physical model, account for detector response and backgrounds, and quantify uncertainty by computation.

Benchmarks for scientific analysis. SciCode and ScienceAgentBench score scientific programs against reference solutions and tests [17, 5]. ReplicationBench grades the replication of astrophysics papers against their results [21], and Collider-Bench the reproduction of LHC analyses against their published yields [9]. Stargazer generates radial-velocity worlds from a seed and grades the fitted parameters by forward modelling; its agents often fit the data while recovering the wrong physics [14]. gwBenchmarks re-runs the agents' artifacts through grader-owned metric code after catching proxy metrics and fabricated results [11]. NewtonBench and DiscoverPhysics ask agents to uncover altered or hidden laws, and DiscoverPhysics rolls the inferred law out on held-out particles [22, 20]. The FAIR Universe challenges come closest to grading a procedure: they score submitted inference algorithms by interval width and coverage over pseudo-experiments, for the Higgs signal strength [3] and for cosmological parameters [7], and an agent-driven entry shows language models building such pipelines [4].

Relation to ParticleBench. All of these benchmarks fix the target before the agent starts (Table 1). The target is a reference answer, a published result, the instance the agent iterates against, or the law that also governs the held-out particles. A fixed target can be reached by iterating against it, so such a test measures engineering, however hard the task. Three precedents come within one step of ours. Stargazer draws its worlds from a seed but grades the instance it gives feedback on. DiscoverPhysics holds out particles but not the law. FAIR Universe grades coverage, the guarantee our gates check, on one fixed dataset. ParticleBench takes the remaining step in each case. Its truth is redrawn after the program is frozen. Its feedback is rationed, so the agent's own simulation is the only cheap checker. Its gates check the procedure's guarantees before any score counts. Its verifier's error is measured in both directions (Section 3). Held-out draws are ordinary in machine learning; for science agents the draw carries new truth, so only a procedure that holds across instances passes.

3 BENCHMARK DESIGN

ParticleBench separates task-independent execution from task-specific physics. The task-independent harness manages the agent environment, the submission protocol, verifier invocation, and the run ledger. Each task provides a generator G, a task-specific verifier V consisting of gates and a score, and an expert oracle O. The same harness is used for every task.

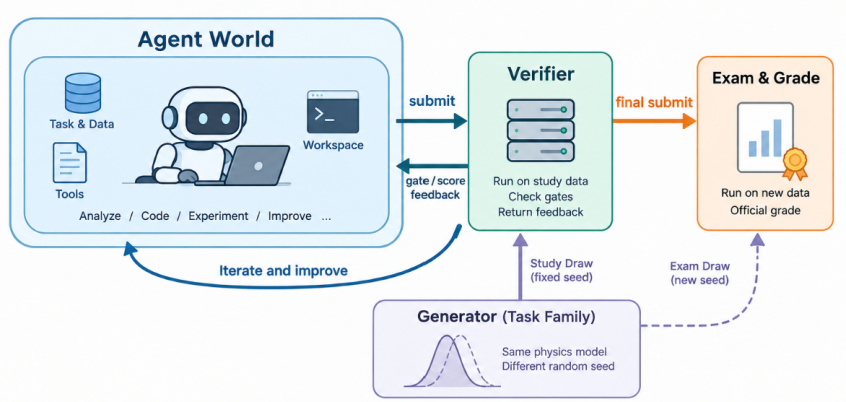


Figure 1: A ParticleBench run. The agent remains in the same world throughout development and may submit successive versions of its analysis program. Study submissions are evaluated on a fixed study draw and return gate outcomes and, conditional on passing all gates, a score. The final submission freezes the program and is evaluated once on an independent exam draw from the same generator with a new seed. The exam verdict is the official grade.

During a run, the agent remains in a persistent world containing the task statement, public data, a base software environment, and a writable workspace. The agent may issue study submissions while continuing to work in that world. Each study submission snapshots the current analysis program and is evaluated on a fixed study draw. The verifier returns the gate outcomes and, if all gates pass, the score. A final submission freezes the current program and evaluates it once on an independent exam draw produced by the same generator with a new seed. The exam verdict is the official grade of the run. Figure 1 summarizes this protocol.

3.1 Task specification

A task is specified by

$$T = (W, Q, B), \tag{1}$$

where W is the agent world, Q is the research question, and B is the run budget. The question Q defines the scientific objective and the required outputs, such as a reconstruction, a parameter estimate, a hypothesis decision, or an upper limit. The world W defines the agent-facing environment: the public inputs, the base software environment, and the writable workspace. It excludes hidden truth, private generator settings, verifier state, and run records. The budget B specifies the wall-clock and submission limits for the run; verifier resource limits are specified by the evaluation configuration.

3.2 Task suite

ParticleBench contains 10 tasks derived from 7 experiments. The suite covers 5 measurement types: reconstruction, calibration and resolution, parameter estimation, hypothesis discrimination, and upper-limit search. Figure 2 illustrates the modeled experiments. The full task cards in Appendix A specify the public inputs, submission interface, evaluation rules, and oracle development effort.

3.3 Verifier: gates and score

An evaluation draw consists of observations D and hidden truth θ. The verifier executes the submitted analysis program on D without access to θ, producing an output $\hat{y}$. A separate judging stage then evaluates $\hat{y}$ against θ. Each evaluation runs in an isolated container in which the submission and verifier code are read-only and private data are available only to the judging stage.

The verifier returns a gate vector $\mathbf{g}$ and a scalar score S. Contract gates, which are common to all tasks, check execution limits, output validity, reproducibility, and confinement to declared inputs. Performance gates enforce task-specific numerical requirements. Statistical gates test guarantees

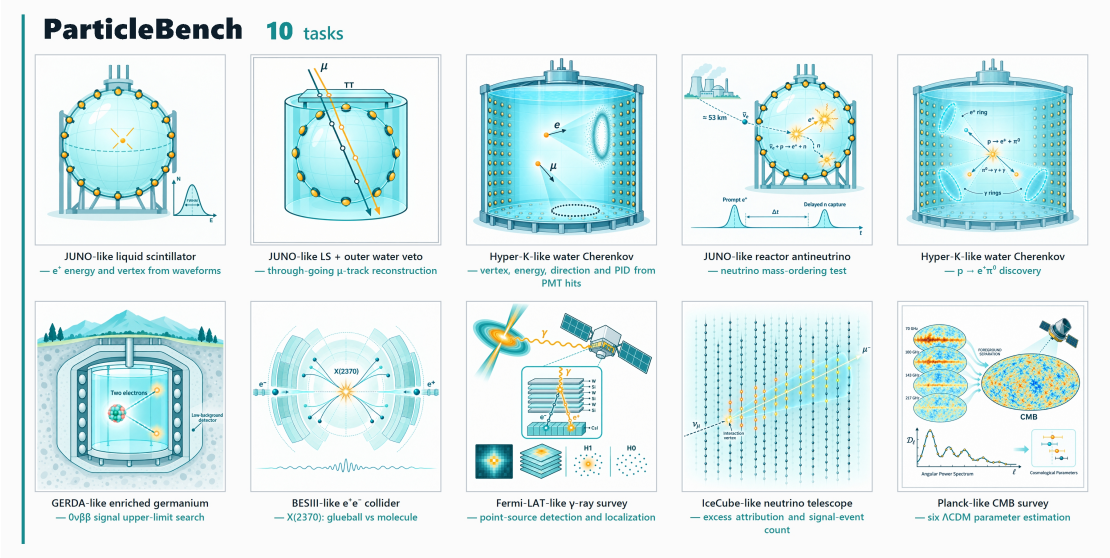


Figure 2: The 10 ParticleBench tasks. Each panel shows the modeled experiment and the measurement the task poses, in table order (left to right, top row then bottom): JRB, JMRB, HRB, JHB, HHB, GHB, BHB, FHB, IRB, PRB.

defined over the instances in a draw, such as reconstruction bias or interval coverage. The task card states the exact gates and thresholds for each task.

The score measures result quality among submissions that satisfy all gates. During the study phase, the score is reported only if every gate passes; otherwise the verifier returns the individual gate outcomes without additional diagnostic information. The exam uses the same rule, but its verdict is recorded as the official grade rather than returned for further development.

The verdict has three possible states. A submission passes if all gates hold, in which case its score is recorded. It fails if at least one gate does not hold. An evaluation error is recorded separately when the verifier or its execution environment fails to produce a valid verdict.

The gate–score separation is necessary when validity and quality are different properties. For example, a tightness score alone can favor an upper limit that undercovers, whereas a coverage gate alone accepts arbitrarily conservative limits. A coverage gate followed by a tightness score first enforces the required statistical guarantee and then rewards informative limits.

3.4 Generator and study–exam separation

Each task defines a forward generator G_ϕ, where ϕ denotes the public configuration of the physical process and detector model. A seed s produces a draw

$$G_\phi(s) = (D_s, \theta_s). \tag{2}$$

The agent receives the observations and public metadata specified by the task card. The verifier retains the seed, hidden parameters, and truth required for evaluation.

All study submissions are evaluated on one fixed study draw. The final submission is evaluated once on an exam draw generated with the same G_ϕ and a different seed. The exam draw resamples the stochastic quantities defined by the task, including hidden physical parameters when the task varies them. Consequently, the exam tests whether the submitted procedure transfers beyond the particular truth and fluctuations encountered during development. Tasks may additionally shift hidden parameters away from published reference values when this is required to distinguish measurement from recall.

3.5 Oracle and run ledger

Each task includes an oracle implemented by a human domain expert. The oracle uses the same submission interface and is evaluated by the same verifier and gates as an agent submission. Oracle success on the reported seed set provides empirical evidence that the task is solvable as specified, and its score provides a reference for interpreting agent performance. The oracle may be developed with access to the generator, but its source code and development materials are excluded from the agent world. We report the expert development effort as a descriptive task statistic rather than as a formal lower bound.

The ledger records each submission identifier, submission type, verdict, gate vector, score when defined, runtime, and timestamp. It distinguishes study submissions from the single exam submission and preserves both successful and failed evaluations. The official grade is the exam entry in the ledger. The same records support the aggregate metrics reported in Section 5, including anytime performance and cost of pass.

3.6 Validity evidence

We evaluate three properties of each task and verifier. First, oracle evaluations test empirical solvability (the task validity of Zhu et al. [23]): the oracle should pass all gates on the designated study and exam seed sets. Second, truth-access mutants test the execution boundary: a submission that attempts to read private truth should be rejected before its scientific output is scored. Third, targeted mutants test whether the verifier responds to known analysis faults.

Mutants that violate a gate-protected requirement are expected to fail the corresponding gate. Examples include omitting calibration, violating blinding, or reporting intervals with inadequate coverage. Mutants that remain admissible but degrade result quality are expected to retain gate eligibility while receiving a worse score. We report gate catch rates and score degradation separately, together with the tested seeds and the complete fault list in Appendix B. These measurements characterize the faults covered by the validation suite; they do not establish that the verifier detects every possible invalid analysis.

4 Experimental Setup

We evaluate models as autonomous research agents on ParticleBench, with persistent workspaces and access to standard research tools. Fable 5-1, DeepSeek 4.1 Flash, Kimi-K3, and GLM-5.3 use the Claude Code harness, while Astra uses Codex. Results therefore describe model–harness systems. All models are configured at the `high` reasoning effort level.

Each run has a 24-hour wall-clock limit and no additional token budget. We conduct three runs per model and task family, following the study–exam protocol described in Section 3. We record model and harness configurations, trajectories, submitted programs, evaluation verdicts, and final scores for reproducibility.

5 Results

We report exam performance and examine its association with scientific procedure and modeling quality.

Metrics. The *all-gate pass rate* is the fraction of runs passing every required gate. The *oracle-relative score* is the mean exam score, with failed runs assigned zero, divided by the task's oracle reference score. We conduct three runs per configuration and task family, and average both metrics equally across ten families.

Overall performance. Astra achieves the highest mean oracle-relative score (78.80%), followed by Fable 5-1 (76.33%). Both pass all gates in 29 of 30 runs. Kimi-K3 achieves a mean relative score of 54.54% and passes 26 runs. GLM-5.3 and DeepSeek 4.1 Flash score 47.12% and 43.96%, respectively, each passing 21 runs. The highest- and lowest-performing configurations therefore

Table 2: Exam performance on ParticleBench. Each entry reports the mean oracle-relative score across three runs, with failed runs assigned zero, followed by the number of runs passing all gates. Pass rate and Oracle-relative denote the task-macro-averaged all-gate pass rate and oracle-relative score, respectively. Bold scores indicate the highest value in each row, including ties.

Task	Astra	Fable 5-1	DeepSeek 4.1 Flash	Kimi-K3	GLM-5.3
bhb	**62.69%** (3/3)	49.02% (2/3)	2.04% (1/3)	39.33% (3/3)	14.95% (2/3)
fhb	86.20% (3/3)	92.10% (3/3)	54.20% (3/3)	82.10% (3/3)	**92.70%** (3/3)
ghb	**98.08%** (3/3)	83.52% (3/3)	14.08% (1/3)	23.84% (1/3)	0.00% (0/3)
hhb	82.83% (3/3)	**93.59%** (3/3)	80.80% (3/3)	80.51% (3/3)	82.25% (3/3)
hrb	64.80% (3/3)	**73.34%** (3/3)	61.61% (3/3)	65.58% (3/3)	60.07% (3/3)
irb	63.95% (3/3)	**80.17%** (3/3)	64.53% (3/3)	63.18% (3/3)	53.82% (3/3)
jhb	91.67% (3/3)	94.79% (3/3)	**98.96%** (3/3)	89.58% (3/3)	90.62% (3/3)
jmrb	**81.33%** (3/3)	72.12% (3/3)	56.85% (3/3)	35.15% (3/3)	73.33% (3/3)
jrb	**70.88%** (2/3)	39.08% (3/3)	6.50% (1/3)	25.29% (2/3)	3.50% (1/3)
prb	**85.60%** (3/3)	**85.60%** (3/3)	0.00% (0/3)	40.88% (2/3)	0.00% (0/3)
Pass rate ↑	**96.67%**	**96.67%**	70.00%	86.67%	70.00%
Oracle-relative ↑	**78.80%**	76.33%	43.96%	54.54%	47.12%

Table 3: LLM-assessed scientific process ratings. Scores are averaged over runs within each task family and then equally across ten families. Each stage uses a 1–3 ordinal scale; $\overline{W}$ is the mean six-stage rating. Bold indicates the highest value in each column.

Model	D	C	P	S	B	I	$\overline{W}$
Astra	2.259	2.037	1.889	1.964	1.444	1.852	1.907
Fable 5-1	**2.556**	**2.260**	**2.334**	**2.334**	**1.851**	**2.074**	**2.235**
DeepSeek 4.1 Flash	2.037	1.815	1.926	1.963	1.556	1.816	1.852
Kimi-K3	2.000	1.963	1.963	1.889	1.666	1.889	1.895
GLM-5.3	2.111	1.963	2.000	2.074	1.704	1.889	1.957

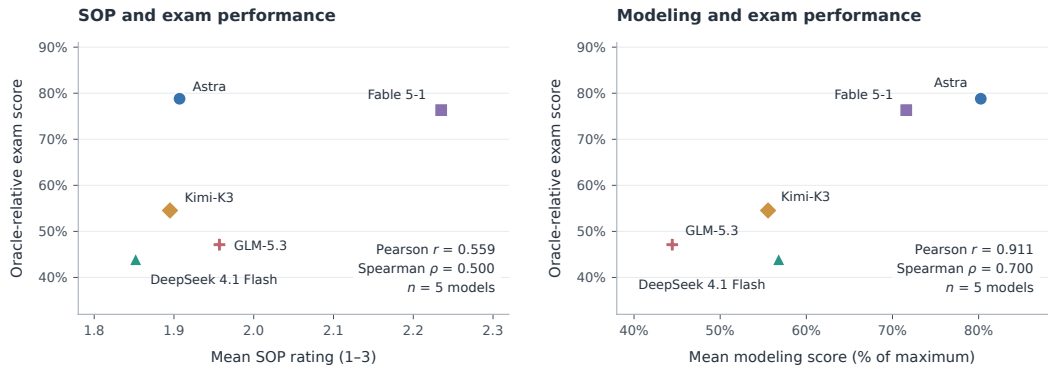
differ by 34.85 percentage points in mean oracle-relative score and 26.67 percentage points in all-gate pass rate. Even the strongest configuration remains below the oracle in aggregate score, and no configuration passes every run.

5.1 Scientific Procedure

An LLM judge evaluates six stages from trajectories, code, execution results, and validation evidence: **D**, detector modelling; **C**, detector-model calibration; **P**, component analysis; **S**, signal/background separation; **B**, background measurement; and **I**, signal extraction and hypothesis testing. Each stage receives an ordinal rating: 1 for basic or missing implementation, 2 for quantitative diagnosis or revision with incomplete validation, and 3 for an effectively validated method used in the analysis.

We average the six ratings within each run, then over runs within each task and equally across tasks. These descriptive ratings do not determine the benchmark verdict.

Similar process ratings coexist with different outcomes. Mean SOP ratings range from 1.852 to 2.235, indicating broadly similar levels of procedural execution. Their association with exam scores is moderate (Pearson $r = 0.559$; Spearman $\rho = 0.500$; Figure 3(a)). Fable 5-1 has the highest SOP rating, whereas Astra has the highest exam score. With only five model means, the Pearson correlation is not statistically significant ($p = 0.327$). SOP ratings alone therefore provide an incomplete account of performance differences.



(a) SOP ratings and exam performance.

(b) Modeling scores and exam performance.

Figure 3: Associations between analysis ratings and final performance. Each point represents one model, with both axes averaged over the same ten task families. **(a)** Mean SOP rating versus oracle-relative exam score. **(b)** Mean modeling score, expressed as a percentage of the task-specific rubric maximum, versus the same exam score.

5.2 Modeling Quality

We assess whether final programs effectively represent the key relationships identified from each task's generator and measurement objective. Each relationship receives 0–3 points: 0 for absence or a confirmed critical error, 1 for an overly idealized representation, 2 for a reasonable but incomplete simplification, and 3 for a faithful or justified equivalent treatment.

For each run, the total is divided by the rubric maximum. Scores are then averaged within tasks and equally across tasks. The percentage measures rubric attainment, not physical fidelity.

Modeling quality more closely tracks final performance. Modeling scores show a stronger observed association with exam scores (Pearson $r = 0.911$; Spearman $\rho = 0.700$; Figure 3(b)). Astra and Fable 5-1 lead in both modeling quality and final performance. Compared with the SOP results, this suggests that how agents represent measurement-relevant relationships is more closely associated with success than their overall procedural rating. The distinction is between performing scientific steps and implementing an adequate model. This comparison is descriptive: five model means do not establish causality or a significant difference between the correlations.

6 Limitations

ParticleBench currently has several important limitations. All worlds are Monte Carlo surrogates rather than collaboration-grade simulations, so the benchmark does not yet test calibration against real instrument data. Only three of ten families include a physics or statistical gate beyond contract checks; in the remaining families, physics quality is reflected only in the score. Verifier soundness is supported by truth-access and wrong-solution controls, but SOP-skip mutants have not yet been evaluated, so coverage of analysis failures remains incomplete. In addition, Table 2 combines owner versions with non-uniform protocols: hhb and jhb select the best study submission, jhb reuses the study draw, and graded truth was reachable from the agent environment in four families, although trace review found no access. The results therefore characterize the evaluated model–harness systems on these task versions, rather than a fully uniform blind-exam protocol. The evaluation covers ten families, five models, and three runs per family, making per-family comparisons noisy. Finally, the oracle is an author reference rather than an upper bound; two oracles reuse model-written code; and the trajectory ratings are retrospective single-LLM judgments, unblinded to score and not systematically validated by humans. The reported associations should therefore be interpreted as descriptive.

7 CONCLUSION

ParticleBench evaluates whether research agents can produce analysis procedures that transfer to a fresh draw under limited verifier feedback. Across ten task families, the strongest model–harness systems achieved 78.80

AI USE STATEMENT

Generative AI entered this work at four points. First, the evaluated agents are the subject of study, and their trajectories are data. Second, the task families were built with AI coding assistants: the repositories that keep a version history carry assistant attributions, the others carry agent working notes, and the port of every family to the uniform protocol was done by agent sessions supervised by the authors. Three oracles contain AI work (Appendix A): the oracle of jmrb is built on the program of an Astra run, that of irb reuses reconstruction code from an earlier Astra run, and that of ghb was produced by an agent session. Third, the trajectory analyses behind Q2–Q4 – the SOP rubric ratings, the failure diagnoses and the ratings of each agent's simulation and delivered program – were produced by LLM raters, one rater per run, retrospective and not blind to the exam score. In some families the rater is one of the evaluated models: Fable 5.1 rated the bhb rubric and the program ratings of hhb and jhb. No rating was checked systematically by a human; only isolated items were. The per-run tables behind Q4 were extracted from these reports by scripts. Fourth, the text of this paper was drafted and edited with an AI assistant. Gates and scores are code and involve no model judgement; the authors take responsibility for all content.

ETHICS STATEMENT

The work involves no human subjects and no personal data. All data are Monte Carlo, drawn by the task generators; the generators are surrogates of the experiments named, built from public software and published inputs, and no result here speaks for any collaboration or measures its analyses. In the released copies the delivered program runs in a sandbox without network access and without the truth. The released trajectories are scrubbed of credentials and host identifiers before publication, since an agent's working directory can capture both.

REPRODUCIBILITY STATEMENT

Every family ships its generator, verifier, oracle and tests, with a container recipe or the shared base image it runs in. The datasets are not shipped but regenerated from two seeds per family, and `generate --check` verifies each split against its sealed hash. The released copies were accepted by running the oracle, a wrong solution and a truth-reading solution through the full protocol on both draws (Appendix B). We release the ledger of every verdict, the trajectories of every run in Table 2, and the scripts that compute the table from the ledger. The runs in Table 2 were made on each family's owner version, which differs from the released copy in data, submit policy and wall clock.

REFERENCES

[1] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, G. Barrand, et al. Geant4—a simulation toolkit. *Nuclear Instruments and Methods in Physics Research Section A*, 506(3):250–303, 2003. doi: 10.1016/S0168-9002(03)01368-8.

[2] J. Allison, K. Amako, J. Apostolakis, P. Arce, M. Asai, T. Aso, E. Bagli, A. Bagulya, S. Banerjee, G. Barrand, et al. Recent developments in Geant4. *Nuclear Instruments and Methods in Physics Research Section A*, 835:186–225, 2016. doi: 10.1016/j.nima.2016.06.125.

[3] Lisa Benato, Wahid Bhimji, Paolo Calafiura, Ragansu Chakkappai, Po-Wen Chang, et al. FAIR Universe HiggsML uncertainty dataset and competition. *arXiv preprint arXiv:2410.02867*, 2024.

[4] Thomas Borrett, Licong Xu, Andy Nilipour, Boris Bolliet, Sebastien Pierre, et al. Competing with AI scientists: Agent-driven approach to astrophysics research. *arXiv preprint arXiv:2604.09621*, 2026.

[5] Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, et al. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, 2025.

[6] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020. doi: 10.1073/pnas.1912789117.

[7] Biwei Dai, Po-Wen Chang, Wahid Bhimji, Paolo Calafiura, Ragansu Chakkappai, et al. FAIR Universe weak lensing ML uncertainty challenge: Handling uncertainties and distribution shifts for precision cosmology. *arXiv preprint arXiv:2604.14451*, 2026.

[8] Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28*, Lecture Notes in Computer Science, pp. 625–635. Springer, 2021. doi: 10.1007/978-3-030-79876-5_37.

[9] Darius A. Faroughy, Sofia Palacios Schweitzer, Ian Pang, Siddharth Mishra-Sharma, and David Shih. Collider-Bench: Benchmarking AI agents with particle physics analysis reproduction. *arXiv preprint arXiv:2605.13950*, 2026.

[10] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

[11] Tousif Islam, Digvijay Wadekar, and Zihan Zhou. gwBenchmarks: Stress-testing LLM agents on high-precision gravitational wave astronomy. *arXiv preprint arXiv:2605.11269*, 2026.

[12] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2.

[13] Vincent Lim, Huang Huang, Lawrence Yunliang Chen, Jonathan Wang, Jeffrey Ichnowski, Daniel Seita, Michael Laskey, and Ken Goldberg. Planar robot casting with Real2Sim2Real self-supervised learning. *arXiv preprint arXiv:2111.04814*, 2021.

[14] Xinge Liu, Terry Jingchen Zhang, Bernhard Schölkopf, Zhijing Jin, and Kristen Menou. Stargazer: A scalable model-fitting benchmark environment for AI agents under astrophysical constraints. *arXiv preprint arXiv:2604.15664*, 2026.

[15] Long Phan, Alice Gatti, Ziwen Han, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

[16] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[17] Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, et al. SciCode: A research coding benchmark curated by scientists. *arXiv preprint arXiv:2407.13168*, 2024.

[18] George Tsoukalas, Anton Kovsharov, Sergey Shirobokov, Anja Surina, Moritz Firsching, Gergely Bérczi, Francisco J. R. Ruiz, Arun Suggala, Adam Zsolt Wagner, Eric Wieser, Lei Yu, Aja Huang, Miklós Z. Horváth, Andrew Ferraiuolo, Henryk Michalewski, Edward Lockhart, Codrut Grosu, Thomas Hubert, Matej Balog, Pushmeet Kohli, and Swarat Chaudhuri. Advancing mathematics research with AI-driven formal proof search. *arXiv preprint arXiv:2605.22763*, 2026.

[19] Miles Wang, Robi Lin, Kat Hu, Joy Jiao, Neil Chowdhury, Ethan Chang, and Tejal Patwardhan. FrontierScience: Evaluating AI's ability to perform expert-level scientific tasks. *arXiv preprint arXiv:2601.21165*, 2026.

[20] Matt L. Wiemann, Lindsay M. Smith, Peter Melchior, Siddharth Mishra-Sharma, Andrew Gordon Wilson, Pavel Izmailov, and Carolina Cuesta-Lázaro. DiscoverPhysics: Benchmarking LLMs for out-of-the-box scientific thinking. *arXiv preprint arXiv:2605.26087*, 2026.

[21] Christine Ye, Sihan Yuan, Suchetha Cooray, Steven Dillmann, Ian L. V. Roque, et al. ReplicationBench: Can AI agents replicate astrophysics research papers? *arXiv preprint arXiv:2510.24591*, 2025.

[22] Tianshi Zheng, Kelvin Kiu-Wai Tam, Newt Hue-Nam K. Nguyen, Baixuan Xu, Zhaowei Wang, et al. NewtonBench: Benchmarking generalizable scientific law discovery in LLM agents. In *International Conference on Learning Representations*, 2026.

[23] Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun, Andy Zhang, Shu Liu, Sasha Cui, Sayash Kapoor, Shayne Longpre, Kevin Meng, Rebecca Weiss, Fazl Barez, et al. Establishing best practices for building rigorous agentic benchmarks. *arXiv preprint arXiv:2507.02825*, 2025.

A TASK CARDS

Each family is modelled on an experiment and asks one measurement of its raw data. The agent may submit its program 10 times on the study draw and learns which gates passed and, when all did, the score; its last program is graded once on the exam draw. In hhb and jhb the official score is instead the best study submission, and jhb grades both on the same draw; prb returns the gate bits only.

bhb BESIII-like

Experiment	J/ψ radiative decays; the nature of the X(2370).
Raw data	Reconstructed events in five hadronic and one two-photon channel; six worlds per draw.
Task	Exclude the glueball or the $\Sigma\bar{\Sigma}$-molecule interpretation, or neither.
Gates	Schema; determinism; runtime.
Score	Verdict, evidence recomputed by the verifier, reasoning.

fhb Fermi-LAT-like

Experiment	One-day γ-ray sky observations.
Raw data	Photon lists (direction, energy, time) and exposure; 128 observations per draw.
Task	Probability of a faint point source; its position and flux.
Gates	Informative probabilities (permutation test); localization and flux quality.
Score	Brier skill against a constant 0.5.

ghb GERDA-like

Experiment	$0\nu\beta\beta$ decay of ^{76}Ge.
Raw data	ADC waveforms with calibration, pulser and blind audit events; 128 experiments per draw.
Task	Event energies and a 90% upper limit on the signal strength.
Gates	Energy bias and variance; coverage; informative limits.
Score	Limit loss relative to the best constant.

hhb Hyper-Kamiokande-like

Experiment	Proton decay $p \to e^+\pi^0$.
Raw data	Events, rings and a Michel-electron calibration stream per multi-year exposure.
Task	Decide whether proton decay is present.
Gates	Contract; determinism.
Score	Balanced accuracy.

jhb JUNO-like

Experiment	Eight years of reactor antineutrinos at 53 km.
Raw data	Event windows, calibration sources, reactor history; 40 campaigns.
Task	Selection, calibration, backgrounds and spectra under both orderings.
Gates	Contract.
Score	Fraction of orderings decided correctly.

hrb Hyper-Kamiokande-like

Experiment	Atmospheric-neutrino events in a water-Cherenkov detector.
Raw data	PMT hits (tube, charge, time) and calibration samples; 4950 events per draw.
Task	Vertex, energy, direction and class (e, μ, γ, π^0).
Gates	Contract; determinism.
Score	Mean quality of the four outputs.

irb IceCube-like

Experiment	20-day in-ice neutrino observations of a candidate direction.
Raw data	Pulses (sensor, time, charge) and an auxiliary run; 90 or 135 fields per draw.
Task	Source, instrumental artifact, or inconclusive.
Gates	Contract, 120 s per field.
Score	Mean one-versus-rest power over classes.

jrb JUNO-like

Experiment	Positron events in a liquid-scintillator detector.
Raw data	1 ns waveforms of 17 612 PMTs per event and calibration pools; 6250 events per draw.
Task	Calibrate, then reconstruct energy and vertex.
Gates	Finite output; valid resolution fits.
Score	Milestones on bias and resolution.

`jmrb` JUNO-like

Experiment	Cosmic muons through the water-Cherenkov veto.
Raw data	First-hit time and charge per veto PMT; about 4000 events per draw.
Task	Every muon track crossing the detector, bundles included.
Gates	Contract; determinism.
Score	Spallation-neutron veto efficiency at 1% signal loss.

`prb` Planck-like

Experiment	Full-sky CMB survey in seven bands.
Raw data	HEALPix T, Q, U maps per band; one sky per draw.
Task	Six ΛCDM parameters, 10–20σ from Planck 2018.
Gates	Each parameter within tolerance of the truth.
Score	Mean normalized parameter error.

B HACK BANK

The verifier is validated in the second direction of Section 3.6 – that it rejects unsound work – by acceptance controls; a bank of mutants that each skip one SOP step is defined but not yet run.

Acceptance controls. On the released copy of every family, two mutants of the oracle were run through the full protocol on both draws. A *wrong* solution – a malformed or fabricated answer – fails a gate in 9 of the 10 families; in bhb it passes all four gates, which check form and execution only, and scores 0. A *truth-reading* solution, which probes every path where the truth or the verifier's state could lie, finds none of them in the execution stage and fails a gate on both draws in all 10 families. These controls were run on the released copies; the versions run for Table 2 record them only as unit tests.

C TRAJECTORY SAMPLES

Four excerpts from runs in Table 2, verbatim from the run's trace up to elisions marked `[...]`, line wrapping and transliteration to ASCII; `agent:` marks the agent's own text.

prb Fable 5.1; exam 0.89, all gates

The agent fits skies it generated with known parameters and reads the pulls.

```
s101: chi2=7528.5  pulls=[ 4.41 -0.58 -0.89 -0.13 -0.63 -1.98] [...]
s005: chi2=495.4  pulls=[ 12.06  13.16 -12.2   16.35   0.69   0.64] [...]
[...]
s101: chi2=60.6  pulls=[ 0.54  0.   -0.77 -0.91 -0.53 -0.72] [...]
agent: s101 is now fixed (all pulls <1 sigma). s005 remains badly off
       (H0 pinned at 90, chi2=426). Let me look at it.
[...]
s005: chi2=84.9  pulls=[ 0.64  0.88 -0.6   2.06  2.19  2.32] [...]
agent: s005 is fixed too (chi2=85/53, all pulls within ~2 sigma).
       Let me wait for the rest, then look at the pull statistics
       across all sims.
```

hrb GLM-5.3; exam 0.32

The agent fits events generated from its own detector model.

```
agent: The synthetic test was unphysical (flat charges vs model
       expectation). Let me generate model-consistent synthetic events:
[...]
  -> [...] E= 16.9 PE=  196 err   0.27 m ang   93.4 deg [...]
  -> grid: err   1.00 | seed: ang   21.0 |
     joint: err   0.08 ang   93.1 [...]
agent: Found the direction bug: the gtab lookup index is reversed --
       idx=(1-ct)*800 maps ct=1 to gtab[0], but gtab[0] was built for
       ct=-1. The fit was using a mirrored G(theta), placing rings at
       ~139 deg instead of 41 deg. Fixing all lookup sites:
```

fhb DeepSeek 4.1 Flash; exam 0, all gates

The agent compares its simulations with the gate results on the study draw.

```
agent: So across ALL regimes I can simulate, localization passes. Yet
       on the real batch it fails badly.
       This strongly suggests the real data differs from my simulator
       in KIND, not degree. [...]
       2. localization_quality FAILS and flux_quality FAILS on every
          submit that claimed positions.
       3. All my simulations (~20 configurations) show localization
          1.000 whenever I claim.
```

bhb Fable 5.1; exam 0

The agent rewrites the decision rule of its analysis program; the comment it writes states the new rule.

```
# We therefore read Z off the fitted produced-level yield,
# Z_gg = 6 * N_prod / 10000, and place the operational thresholds where a
# wrong exclusion is unlikely: N_prod >= 11100 (molecule-like) and
# N_prod < 3500 (glueball-like, compatible with zero).
```