BENCHMARKING LLM ABILITIES ON POSTDOC-LEVEL NATURAL SCIENCE PROBLEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific problems have no answer key, so evaluating an agent means deciding what counts as a better solution before any agent is run. Existing benchmarks either match a symbolic answer against a key or, where they require executable work, score it by unit tests, by a rubric, or by position on a competition leaderboard. None of these is the criterion by which a field decides that one published method beats another. STEMBENCH adopts that criterion directly. Each of its twelve research problems in high-energy physics, astrophysics and neuroscience is scored by its own field's figure of merit, and each comes with two measured anchors: a submission that uses no domain knowledge, and an expert reference reproducing a published method, both obtained by running code shipped with the benchmark. The interval between them states how much of a problem is crossed by scientific insight rather than by engineering. Across eight frontier models, task-level wins and domain-level leadership dissociate: a model that wins five of the twelve tasks leads no domain mean. In the high-effort cross-model comparison, at least one model's mean score exceeds the expert anchor on seven of twelve tasks. The remaining five span uncertainty estimation, latent and causal inference, and cross-site clinical classification. Numerical scores and official-submission validity capture different aspects of performance and are reported separately. We release the tasks, graders, anchors and harness.

1 INTRODUCTION

Asking whether an agent solved a scientific problem requires deciding, before the agent is run, what makes one solution better than another. That decision is the hard part of building an evaluation for scientific work, and it is the part that existing benchmarks answer by substituting a criterion that is easier to automate: a reference answer, a test suite, a rubric, or a position in a competition field (Section 2). None of these is the criterion by which a field decides that one published method beats another.

We present STEMBENCH, twelve research problems in high-energy physics (HEP), astrophysics, and neuroscience that adopt that criterion directly. Each problem is scored by the figure of merit its own community already uses to compare published methods, from background rejection at fixed signal efficiency to coverage-penalised interval width. Each comes with two anchors obtained by running code shipped with the benchmark: a floor produced by a submission that uses no domain knowledge, and a ceiling produced by reproducing a published method. The interval between them states how much of the problem is crossed by a scientific insight rather than by engineering, and it is against that interval, not against a perfect score, that we read what an agent achieves. Agents work offline in a sandbox over staged data of 0.1 to 12 GB and are graded host-side on withheld splits.

Our contributions are:

- **A scoring design anchored at both ends.** Every task ships a baseline and an expert reference submission, both runnable end to end on one workstation, so the interval a solution must cross is measured rather than assumed, and neither anchor is a number quoted from a paper (Section 3).
- **Twelve task contracts with community metrics.** One uniform layout across three domains, with ground truth held host-side, retrieval restricted to a curated per-task corpus, and metrics carrying coverage penalties and geometric gates as explicit defences against exploitation (Section 4).

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.

- **An evaluation of eight frontier models.** Task-level wins and domain-level leadership dissociate. At least one model's mean score exceeds the expert anchor on seven of twelve tasks in the high-effort cross-model comparison. The remaining five span uncertainty estimation, latent and causal inference, and cross-site clinical classification. Official-submission validity is reported separately from the numerical score (Section 5).

2 Related Work

Table 1: Selected related benchmarks and their evaluation targets. Rows summarise the main assessment criterion, not the only one. STEMBench reports its baseline and expert-reference scores alongside the task score.

Benchmark	Evaluation target	Assessment or performance reference
ScienceAgentBench	Data-driven scientific programs	Program and execution evaluation
Terminal-Bench-Science	Scientific workflows	Task-specific terminal verification
ScienceBoard	Scientific software workflows	Task-completion checks
LabOSBench	Scientific instrument control	Subtask completion within a step budget
MLE-bench	ML competition submissions	Competition metrics and human leaderboard results
BioML-bench	Biomedical ML pipelines	Task performance and human baselines
PRBench	Physics-paper reproduction	Reproduced quantitative results and scoring rubrics
MLR-Bench	Open-ended ML research	LLM reviewers and research-quality rubrics
COMPOSITE-STEM	Expert STEM tasks	Exact-match and rubric-based jury grading
LifeSciBench	Life-science research responses	Expert-authored task-specific rubrics
InnovatorBench	Runnable LLM-research artifacts	Correctness, performance, and output quality
NatureBench	Scientific ML solutions	Source-paper state-of-the-art performance
STEMBench (ours)	Computational solutions in HEP, astrophysics, and neuroscience	Scientific task metrics; executable baseline and expert anchors

Scientific knowledge and reasoning. Physics and mathematics benchmarks score an answer against a reference (Xu et al., 2025; Qiu et al., 2025b; Bieganowski et al., 2025), and multimodal STEM evaluations and domain question-answering collections extend this setting to visual evidence and specialised knowledge (Shen et al., 2024; Zaki et al., 2023; Rein et al., 2023). Agentic systems now reach elite-human level here: Physics Supernova attains 23.5/30 on the IPhO 2025 theory problems, placing 14th of 406 contestants (Qiu et al., 2025a). The criterion, however, is still a marking scheme for problems with known solutions. These evaluations measure important prerequisites for scientific work, but what they score is the quality of an answer, not of a method evaluated on withheld data under a community figure of merit.

Benchmarks that score executable work. SciCode scores scientific code by unit tests (Tian et al., 2024), ScienceAgentBench scores self-contained programs for 102 tasks drawn from 44 papers (Chen et al., 2025b), and Terminal-Bench-Science verifies expert-curated research workflows in a terminal, with one oracle solution per task demonstrating feasibility (Terminal-Bench-Science Team, 2026). MLE-bench scores 75 Kaggle competitions, and its multilingual and biomedical extensions and MLR-Bench carry the setting to other languages, to biomedical pipelines and to open-ended ML research (Chan et al., 2025; Trofimova et al., 2026; Miller et al., 2025; Chen et al., 2025a). These are close to the classification and anomaly-detection tasks of STEMBench, but not to its inversion and latent-parameter tasks, where predictive accuracy alone does not describe scientific validity. PaperBench, CORE-Bench and PRBench, the last with 30 tasks across 11 physics subfields, score the reproduction of published results (Starace et al., 2025; Siegel et al., 2024; Qiu et al., 2026). ScienceBoard and LabOSBench move agents into scientific software and instrument simulators, 169 tasks across four domains and 96 subtasks on eight instruments, and count a task as solved when the required operations are completed (Sun et al., 2026; Zou et al., 2026). Reproduction is also how we build expert anchors, although the evaluated agents need not recover the reference method. A test suite, a terminal check or a completion check *accepts or rejects* a solution, whereas a field *orders* competing methods on a continuous figure of merit. STEMBench adopts the ordering and scores it on a withheld split.

Scoring relative to a reference. MLE-bench reads scores against a human leaderboard, InnovatorBench against a reference solution on 20 LLM-research tasks, and NatureBench against the reported state of the art of the 90 Nature-family papers its tasks come from (Wu et al., 2026; Wang

et al., 2026). STEMBENCH ships two anchors per task, both measured by running code in the repository: a no-domain-knowledge floor and a reproduced published method. It reports raw scores beside them rather than a single normalised quantity (Section 3), and treats the expert anchor as an empirical reference, not an upper bound.

Autonomous discovery agents. A parallel line of work builds agents that run the discovery loop end to end; a recent survey organises the field into process-, autonomy- and mechanism-oriented views (Wei et al., 2025). Kosmos iterates literature search, data analysis and hypothesis generation for up to twelve hours per run and is validated by experts auditing its final reports, with 79.4% of statements judged accurate (Mitchener et al., 2025). Such assessment is made after the run, by readers, and cannot be re-executed. STEMBENCH fixes an executable acceptance criterion *before* any agent runs, bounded by two measured anchors, which makes it a regression test for the analysis stage of the discovery loop rather than a competitor to end-to-end evaluation.

Expert-rubric evaluations. COMPOSITE-STEM combines exact-match assessment with criterion-based rubrics and LLM-jury grading on 70 expert-written tasks, and LifeSciBench assesses 750 life-science workflow tasks against expert-authored rubrics (Waters et al., 2026; Liu et al., 2026). Rubrics capture aspects of scientific judgment that resist reduction to a number; STEMBENCH concentrates on computational problems where an explicit, executable scoring rule can order the submitted results. Table 1 summarises the comparison.

3 PROBLEM FORMULATION

We call a problem *scientific*, as opposed to an *exercise*, when it has the seven properties of Table 2. Each one breaks an assumption that current STEM benchmarks rely on, and each is exercised by at least one task in STEMBENCH.

Table 2: Seven properties that separate a scientific problem from an exercise, each with an example from STEMBENCH.

Property	What holds instead of an answer key
P1 Underdetermined solution set	distinct methods are all admissible and are *ordered* by a continuous figure of merit, not accepted or rejected
P2 Model and code as deliverable	an executable artefact that generalises to a withheld split; on two tasks the grader itself trains the submitted model
P3 Load-bearing domain priors	the gap to the expert is closed by a domain insight, e.g. a decorrelation condition for weak supervision or a canonical basis for Stokes profiles
P4 Data beyond the context	0.1–12 GB of staged data, reached through loaders and subsampling; deciding what to look at is part of the task
P5 Community figure of merit	gated composites and non-monotone rules: a sharper interval that under-covers loses to a wider one that covers
P6 Silent failures	plausible errors, such as site leakage or a forward model that fits the spectrum but breaks the physics, which only withheld data exposes and an LLM judge reading the output cannot see
P7 Empirical ceiling	the literature ceiling for cross-scanner ASD classification is about 0.75 AUC, so progress is distance to the best published solution, not to 1.0

Tasks and scores. A task instance is a tuple $\tau = \big(\mathcal{P}, \mathcal{D}_{\mathrm{tr}}, \mathcal{D}_{\mathrm{te}}, \mathcal{C}, \mu, b, e\big)$, where $\mathcal{P}$ is the natural-language specification (motivation, input/output contract, data description, metric description), $\mathcal{D}_{\mathrm{tr}}$ and $\mathcal{D}_{\mathrm{te}}$ are the visible and withheld splits, $\mathcal{C}$ is the submission contract accepted by the scorer, $\mu : \mathcal{C} \to [0, 1]$ is a deterministic scoring function, and b, e are two anchor scores defined below.

An agent is a policy π that interacts with a sandboxed environment under a budget B (wall-clock, tokens, messages) and returns a submission $\sigma_\pi \in \mathcal{C}$. Two anchors are shipped with every task: a *baseline* $b = \mu(\sigma_{\mathrm{base}})$, obtained by a submission that uses no domain knowledge, and an *expert reference* $e = \mu(\sigma_{\mathrm{ref}})$, obtained by reproducing a published method. Both are organiser-side artefacts. They are never staged into the agent sandbox and play no role in an episode beyond fixing the scale.

The harness reports three quantities per episode: the raw task score $\mu(\sigma_\pi)$ in the task's native definition, the gap above the baseline

$$\Delta_{\text{base}}(\pi, \tau) \;=\; 100 \cdot \big(\mu(\sigma_\pi) - b\big) \quad \text{[percentage points]}, \tag{1}$$

and `official_submission_valid`, a reproducibility gate that is independent of the numeric score and requires a syntactically valid entry point, a README with a runnable command, and every declared reproducibility file. We report raw scores with both anchors alongside, because the quantity the benchmark is designed around is the headroom

$$h(\tau) \;=\; e - b, \tag{2}$$

the part of the task that is crossed by a scientific insight rather than by engineering. A normalised score $\hat{\mu} = \text{clip}_{[0,1]}\big((\mu(\sigma_\pi) - b)/h(\tau)\big)$ can be derived post hoc from the reported triple, but we do not use it as the headline number: it discards information whenever $h(\tau)$ is narrow, and on this task set the narrow-headroom cases are exactly the scientifically hardest ones.

4 The STEMBench Benchmark

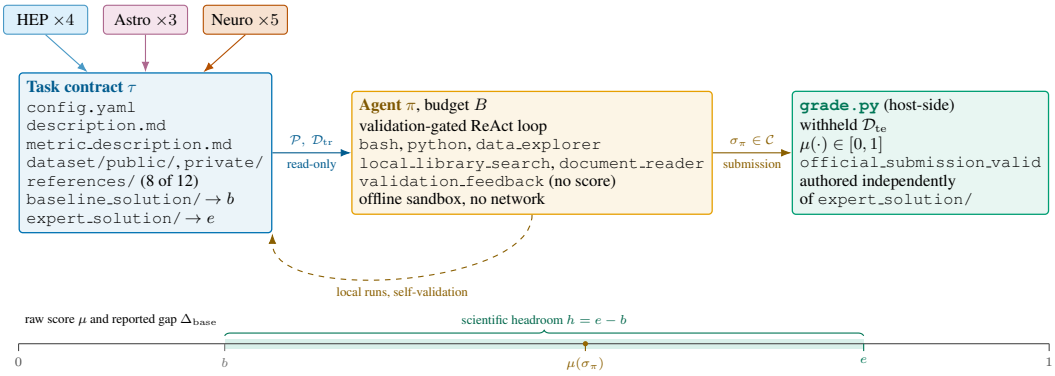


Figure 1: The STEMBench evaluation loop. The agent reads the task specification and the public split, works in an offline sandbox through six tools, and returns a submission in the task's contract. Ground truth never enters the sandbox: the grader runs host-side on the withheld split and returns $\mu \in [0, 1]$ with a reproducibility verdict, read against the anchors b and e.

STEMBench consists of twelve research problems in three domains: four in high-energy physics, three in astrophysics and five in neuroscience. Every problem ships as a self-contained contract folder with a task description, a metric specification, a role-tagged data manifest, a declared submission contract, a host-side grader, and the two anchor submissions. Figure 1 shows the resulting evaluation loop and Table 3 lists the tasks with their anchors.

The three domains were selected because they stress different parts of the capability: high-energy physics contributes large-scale data and hypothesis-free search, astrophysics contributes ill-posed inversion from remote observation with uncertainty quantification, and neuroscience contributes multi-scale forward models and cross-site generalisation on real clinical data.

4.1 Task Selection

Each of the twelve problems satisfies five criteria, applied before any solution was written. The problem falls within the scientific scope of the domain; a solution by classical or machine-learning methods is known to exist, so the task is solvable rather than open; a typical solution requires several non-trivial reasoning steps; a clear mathematical metric is definable on the submission; and the problem exercises a metric family that the others do not. The last criterion is what prevents a single agent from topping the aggregate by specialising in one scoring rule.

The neuroscience block is the one place where we revised the selection after the first round of runs. A single composite neural-mass-model task bundled three distinct inferential operations, model-family identification from generated dynamics, causal digital-twin response prediction, and latent

Table 3: The twelve STEMBENCH tasks. "Eval. units" is what the metric is computed over once. Anchors $b \to e$ are the `reference_scores` declared in each task's `config.yaml`, obtained by running the shipped baseline and expert submissions end to end on a single workstation.

Domain	Task	Eval. units	Metric family	$b \to e$
HEP	New-physics search (LHCO)	135,261 events	ROC-AUC, SIC-AUC, rejection	$0.327 \to 0.912$
	Higgs uncertainty (FAIR Univ.)	100 pseudo-exps.	coverage-penalised interval	$0.400 \to 0.687$
	Top tagging (TopLandscape)	403,999 jets	ROC-AUC, acc., rejection at ϵ_S	$0.063 \to 0.884$
	Dark Machines	34 signal cells	ϵ_S at fixed ϵ_B, AUC	$0.003 \to 0.221$
Astro	Stokes inversion, simulated	10,000 spectra	pinball loss + forward gate	$0.009 \to 0.624$
	Stokes inversion, real (Hinode)	2 rasters	+ spatial-structure term	$0.000 \to 0.598$
	SN Ia classification (PLAsTiCC)	317 objects	ROC-AUC, PR-AUC, log-loss	$0.421 \to 0.849$
Neuro	ASD classification (ABIDE)	332 subjects	held-out-site AUC, Brier, age	$0.000 \to 0.312$
	NMM family id.	45 series	chance-corrected accuracy	$0.000 \to 0.900$
	NMM digital twins	12 interventions	signed causal-response composite	$0.100 \to 0.715$
	NMM parameter recovery	20 subjects	median relative parameter error	$0.125 \to 0.515$
	BOLD deconvolution	21 cases	neural drive + two kernel banks	$0.012 \to 0.353$

parameter inference, into one weighted score, so a high score on one operation could mask a failure on another. The three now stand as independent contracts with their own hidden-target boundaries, submission schemas and anchors, and the legacy composite is retired.

Table 3 lists the twelve problems with the unit each metric is computed over, the family its scoring rule belongs to, and the two anchors. Appendix A states every task in the terms the agent receives it, and Appendix B reproduces one complete agent-facing prompt.

4.2 THE UNIFORM TASK CONTRACT

Heterogeneity across scientific domains is real, but it should not be part of what a benchmark measures. Every task is therefore organised in one layout. A single `config.yaml` is the authoritative contract: it declares the data manifest with a role per file (`training`, `test_input`, `auxiliary`, `ground_truth`), the required submission files with the format checks applied to them, the host-side scorer entry point, the two anchors, the package requirements and the compute requirements. Alongside it, `description.md` carries the scientific specification handed to the agent, `metric_description.md` pins down the scoring rule, `grade.py` implements it, `checksums.yaml` fixes the data, and `baseline_solution/` and `expert_solution/` hold the two anchor submissions with their own instructions for reproducibility. Eight tasks, all of astrophysics and neuroscience, additionally ship a curated `references/` corpus of papers and domain notes.

Only files tagged with a public role are staged into the sandbox. Ground-truth files never leave the host and are read only by the grader after the episode ends. Every active task exposes the same grading signature, `grade(submission_dir, answers_path)`, which is what makes twelve heterogeneous scientific workflows scoreable through one harness.

For the high-energy-physics family the upstream datasets are hosted externally, so the repository bundles loaders, manifests and checksums while the bulk data is fetched from the cited sources and preprocessed into the staged splits, which range from 0.1 to 12 GB per task. All datasets used are openly accessible with redistribution rights compatible with a public benchmark: Zenodo releases for the LHC Olympics R&D dataset and PLAsTiCC, public challenge releases for TopLandscape and the FAIR Universe Higgs data, a public data-use agreement for ABIDE, open solar data for the Hinode/SOT-SP rasters, and in-project generation for the synthetic neuroscience tasks.

4.3 METRICS AND DEFENCES AGAINST GAMING

Each of the twelve metrics instantiates the figure of merit by which the corresponding community already compares published methods: SIC-AUC as used to rank the LHC Olympics teams, the official coverage-and-width scoring rule of the NeurIPS 2024 FAIR Universe competition, background rejection at fixed signal efficiency as used in every modern jet-tagging paper, signal efficiency at

fixed background efficiency per (channel, signal) cell for Dark Machines, the PLAsTiCC ranking metric, the axes along which solar inversion codes are compared, the validation metrics of the Virtual Brain and Balloon model papers, the rsHRF benchmark metrics, and macro AUC over held-out acquisition sites for ABIDE. Each metric therefore inherits a literature floor and ceiling, which is what makes the anchors in Table 3 interpretable.

Three failure modes are addressed by construction. Against *metric exploitation*, composite scores bound every component to $[0, 1]$ before convex combination, so no submission can inflate the headline by driving a single component to an extreme, and the Higgs task applies an asymmetric coverage penalty so that an under-covering interval cannot win on width alone. Against *component substitution*, the two inversion tasks add a gate of our own on top of the community metric: the forward-consistency and spatial-structure terms enter as $\sqrt{R\,F}$ and $\sqrt{R\,S}$, so that a submission cannot win on spectral reconstruction while failing to recover parameters, and a declared forward model that is inconsistent with the physical law it claims to implement is caught by a dedicated validity battery and zeroes the corresponding term. Against *reward hacking on the reference*, every expert submission was produced by reproducing a published method without consulting the scorer implementation, so the metric and the reference were authored independently; a leakage attack would surface as a discrepancy between the reference score and the published score of the same method on the same task.

4.4 Anchors

The baseline anchor is deliberately weak but not degenerate: it is what a capable engineer produces without domain knowledge, for example a constant train-marginal quantile predictor on the inversion tasks, a range-midpoint parameter estimate on neural-mass parameter recovery, a constant-μ interval on the Higgs task, or a literal zero-response prediction on the digital-twin task. Several metrics normalise against exactly that predictor, which is why six baselines sit at or near 0.000 by construction rather than by accident.

The expert anchor reproduces a published method end to end. The resulting gaps are wide and scientifically meaningful, from $0.000 \to 0.900$ on neural-mass family identification to $0.009 \to 0.624$ on simulated Stokes inversion, with the deliberately narrow $0.000 \to 0.312$ on ABIDE reflecting the hard ceiling that real cross-scanner clinical data imposes (P7). In every case the gap was crossed by a domain-specific insight rather than by scaling compute or data, which is the property that makes the headroom a measurement of scientific reasoning rather than of engineering throughput.

4.5 Environment and Toolchain

The harness is built on Inspect-AI (AI Security Institute). Publication runs use per-domain container images that mount only the public files declared in the manifest, read-only, with networking disabled and every package pre-installed; Appendix C lists the images, the hardware and the per-episode limits.

The agent-facing surface is six tools, fixed across tasks. `bash` and `python` execute code in the sandbox. `data_explorer` returns a structural summary of a public data file (shape, dtype, statistics) without loading it into the model context, over eight formats, and refuses pickle-like extensions rather than implicitly unpickling them. `local_library_search` and `document_reader` expose the task-local `references/` corpus, including PDF extraction, on the eight tasks that ship one. `validation_feedback` is a pre-submission checker that reports missing files and schema violations and never reveals the numeric score. There is no internet access, which would turn problems with published solutions into retrieval exercises.

All results reported here use the **basic** agent; the **basic+references** variant is released with the harness and its evaluation is left to future work. The **basic** agent runs a validation-gated ReAct loop in which a submission is accepted only after the structural validator passes, with rejections returned as JSON feedback that the model can act on. The **basic+references** agent is identical except that the two retrieval tools are enabled where a corpus exists. Both are benchmark-pure: no hidden score, anchor or ground truth is revealed at any point. Each episode runs under one hour of wall-clock time, 10^6 model tokens and 200 messages.

The system prompt is identical across tasks and models and is reproduced in Appendix B. Two of its instructions affect how a score should be read. The agent is told to produce a valid submission early and keep it valid, so that a budget limit truncates improvement rather than erasing all work; an episode that still ends with no valid submission is therefore a substantive failure and not a formatting accident. The agent is also warned away from constant, group-midpoint and globally broadcast predictions. Several baseline anchors are exactly such predictors, so the scaffold tells the agent that the trivial solution is known to be trivial, without disclosing its score.

4.6 Evaluation Protocol

Holding the harness fixed and varying the model isolates the core LLM; holding the model fixed and varying the harness isolates the tooling. Both axes are needed, since a gain reported on one is routinely produced by the other. The harness passes a functional test suite and every contract passes checksum and anchor-grading smoke tests (Appendix C). A blinded expert-agreement study, comparing the automatic ranking of intermediate submissions between b and e with rankings by domain reviewers against targets of $\tau \geq 0.85$ and $\geq 90\%$ pairwise agreement, is the remaining validation step.

5 Results

Figure 2 reports eight frontier models under the **basic** scaffold at high reasoning effort (exact values and domain means are in Appendix D). A *retained* run is one whose model dialogue completed without infrastructure interruption. All retained runs enter every mean, including runs scoring zero.

Three observations follow. First, task-level wins and domain-level leadership differ. DeepSeek V4 Pro has the highest HEP mean (0.60), while Gemini 3.1 Pro leads astrophysics (0.84) and neuroscience (0.32). Claude Opus 5 and Gemini 3.1 Pro each lead five of the twelve tasks; GLM 5.2 leads the remaining two. Claude nevertheless leads no domain mean. In HEP, its wins on Higgs uncertainty and top tagging coexist with a lower new-physics-search score, yielding a domain mean of 0.55, below DeepSeek's 0.60. In neuroscience, its wins on digital twins and parameter recovery coexist with means displayed as 0.00 on the other three tasks.

Gemini's domain means range from 0.32 to 0.84, a spread of 0.52. The cross-model spreads of domain means are 0.37 in HEP, 0.64 in astrophysics, and 0.26 in neuroscience, using the displayed values. Thus, this within-model spread exceeds the cross-model spread in HEP and neuroscience, but not in astrophysics. A descriptive two-way sum-of-squares decomposition of the 96 unrounded task–model means assigns 60.5% of the total variation to task main effects, 11.5% to model main effects, and the remaining 28.0% to departures from additivity. This residual includes both task - model interaction and sampling variability in the three-run cell means.

Second, at least one model's mean score exceeds the expert anchor on seven of the twelve tasks: new-physics search, top tagging, Dark Machines, both Stokes inversions, SN Ia classification, and neural-mass family identification. All model means remain below the expert anchor on the other five: Higgs uncertainty (best mean 0.48, $e = 0.687$), ASD classification on ABIDE (0.29, $e = 0.312$), neural-mass digital twins (0.31, $e = 0.715233$), neural-mass parameter recovery (0.42, $e = 0.515$), and BOLD deconvolution (0.19, $e = 0.353$). On digital twins, the best mean remains closer to the baseline ($b = 0.100$) than to the expert anchor. Aggregate scores alone do not establish a single cause for these gaps. Counting individual runs rather than means, nine tasks exceed the expert anchor: ABIDE reaches 0.314 and parameter recovery 0.583.

Third, numerical scores and official-submission validity are distinct outcomes. Of the 288 retained runs, 168 have an affirmative official-submission flag, 64 a negative flag, and 56 an unavailable flag. Among the 88 runs scoring exactly zero, these counts are 15, 17, and 56, respectively. Zero scores therefore cannot be interpreted automatically as failures to produce an official submission.

Four model means on real Stokes are exactly zero, yet all twelve underlying flags are unavailable; two ABIDE means round to 0.00 but are positive, with negative flags for Claude and affirmative ones for Gemini 3.1 Pro. Exposure to the wall-clock cap varies across models in the retained campaign. It is recorded for 14 of 36 MiMo V2.5 runs (38.9%), 15 of 36 DeepSeek V4 Pro runs (41.7%), and 2 of 36 Gemini 3.7 Flash runs (5.6%). These frequencies describe observed stopping conditions;

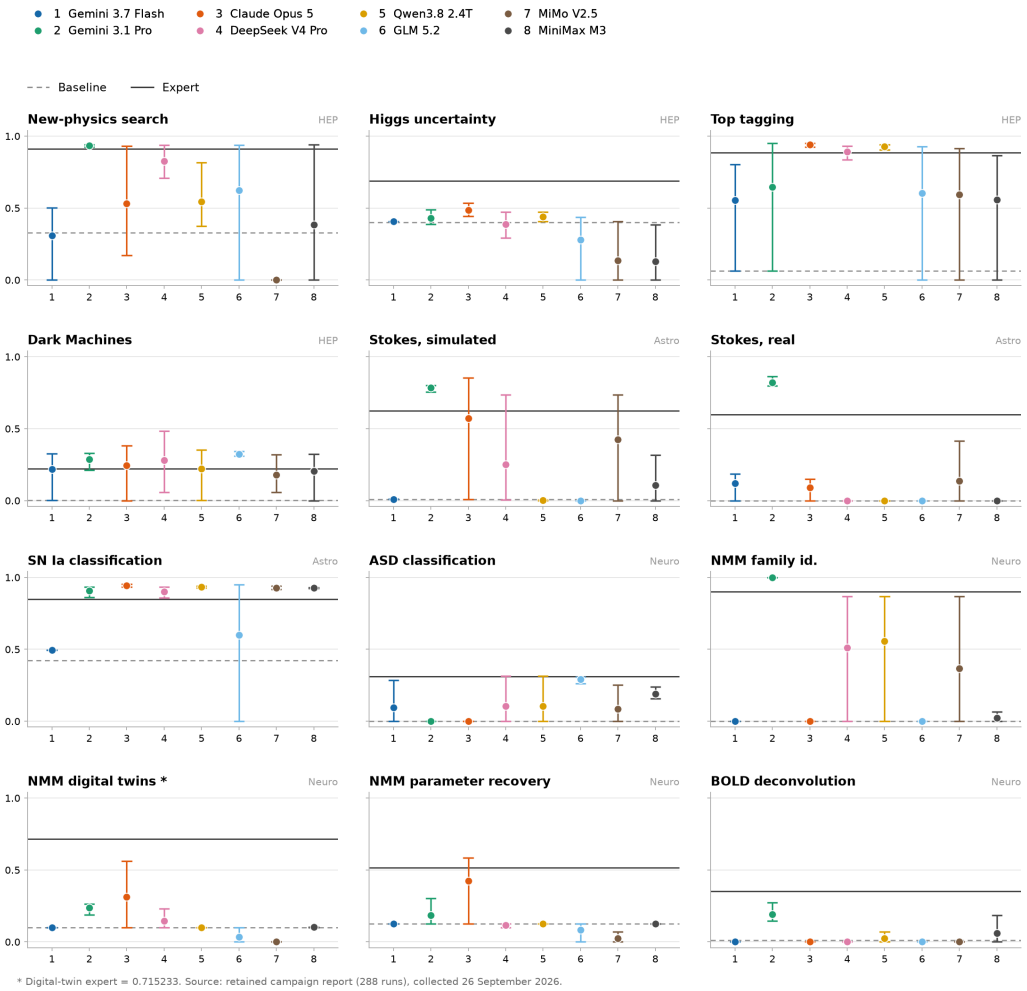


Figure 2: Raw task scores of eight frontier models using the **basic** agent at high reasoning effort. Each marker is the mean of the three retained scored runs, including zeros. Vertical bars show the minimum and maximum of these three scores, not standard deviations or confidence intervals. Horizontal lines mark the baseline b (dashed) and expert reference e (solid). A numerical zero does not by itself identify an invalid submission. The digital-twin expert anchor is $e = 0.715233$.

they do not quantify the score improvement that a larger budget would produce. Because the cap is wall-clock, it also absorbs provider latency, so these runs measure capability under a fixed time budget rather than per-token capability.

Pilot effort sweep. Before the campaign, a pilot sweep ran Gemini 3.1 Pro at `low`, `medium` and `high` reasoning effort on all twelve tasks (Appendix E). It used an earlier task revision and predates a fix to GPU access in the containers, so its absolute scores are not comparable with Figure 2. Qualitatively, most of the gain came between `low` and `medium`, several tasks were bimodal, with runs landing either near the best score or at zero, and the effect of effort was not monotone on every task. These observations motivated fixing the campaign at `high` effort and repeating every cell three times.

6 Conclusion

STEMBench scores scientific problem solving using community metrics and two measured anchors: a baseline and a reproduced expert reference.

In the high-effort comparison of eight frontier models, at least one model's mean score exceeds the expert anchor on seven of twelve tasks, while on others, such as digital-twin response prediction and BOLD deconvolution, the best model remains far below it. The five remaining gaps span uncertainty estimation, latent and causal inference, and cross-site clinical generalisation. Task wins and domain leadership differ, and numerical scores must be distinguished from official-submission validity. This range of difficulty lets STEMBENCH differentiate models by domain and task type rather than by a single aggregate score.

These conclusions have limits.Stable model rankings will require more runs per cell and confidence intervals; a pilot effort sweep (Appendix E) covers one model on an earlier task revision; and agreement between the automatic metrics and blinded expert rankings remains to be measured.

We release the tasks, graders, anchors and harness (Anonymous, 2026a;b).

REPRODUCIBILITY STATEMENT

Every claim in this paper is tied to an artefact in the benchmark repository. The twelve task contracts, their graders, both anchor submissions per task and the harness are released; Section 4 describes the contract layout and Appendix B reproduces the agent-facing prompt surface verbatim. Anchors quoted in Table 3 are the `reference_scores` in each task's `config.yaml` and are reproduced by running the shipped baseline and expert submissions. Appendix D reports aggregate per-model scores for the cross-model comparison. Appendix E reports a pilot effort and budget sweep on an earlier task revision. Data provenance and licences are given in Section 4, and per-domain container images pin the software environment. The task contracts, graders, both anchor submissions and staged data are archived at Anonymous (2026b), and the evaluation harness at Anonymous (2026a).

ETHICS STATEMENT

The ABIDE task uses de-identified human neuroimaging data obtained under the public ABIDE data-use agreement; no new human-subject data was collected and no participant is identifiable from the staged files. The task predicts a clinical label from resting-state fMRI, and we state explicitly in the task description and in Appendix A that cross-scanner transfer of such a marker is a known hard limit: the benchmark measures methodological competence on a public research dataset and the scores reported here carry no clinical validity and must not be read as evidence that any method is fit for diagnostic use. The neural-mass digital-twin task is a controlled synthetic benchmark and is likewise not a patient-specific model. All other datasets are simulated or openly released physical-science data.

AI USE DISCLOSURE

We used generative AI tools in the preparation of this paper. Specifically: to draft Sections 3 and 4 from internal project reports authored by us; to develop the conceptual framing of those sections, including the taxonomy of properties P1–P7 and the formalisation of a task instance with its two anchors; to search for and verify related benchmarks; to aggregate per-run scores into Tables 7 and 2 and to draft the accompanying interpretation; and to edit prose for readability throughout. We did not use generative AI tools to formulate or prove mathematical claims, to generate any benchmark data, or to write any task, grader, baseline or expert solution in the benchmark itself. The twelve problems, their metrics, the reference implementations and all experimental runs are the authors' own work. All AI-assisted output was reviewed by the authors. The cross-model task and domain means in Table 5 and Figure 2 were recomputed from the 288 retained per-run scores in the campaign report. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI. The language models evaluated in Section 5 and Appendix E are the object of study, not tools used to produce this paper.

REFERENCES

Thea Aarrestad, Melissa van Beekveld, Marcella Bona, Antonio Boveia, Sascha Caron, Joe Davies, Andrea De Simone, Caterina Doglioni, Javier M Duarte, Amir Farbin, et al. The dark machines anomaly score challenge: benchmark data and model independent event classification for the large hadron collider. *arXiv preprint arXiv:2105.14027*, 2021.

UK AI Security Institute. Inspect AI: Framework for Large Language Model Evaluations. URL https://github.com/UKGovernmentBEIS/inspect_ai.

Anonymous. Stembench: an inspect-ai benchmark harness for research-grade stem agent tasks, September 2026a. URL https://doi.org/10.5281/zenodo.22959834.

Anonymous. Stembench, September 2026b. URL https://doi.org/10.5281/zenodo.22960135.

Wahid Bhimji, Ragansu Chakkappai, Po-Wen Chang, Yuan-Tang Chou, Sascha Diefenbacher, Jordan Dudley, Ibrahim Elsharkawy, Steven Farrell, Aishik Ghosh, Cristina Giordano, et al. Fair universe higgsml uncertainty dataset and competition. *Advances in Neural Information Processing Systems*, 38, 2026.

Bartosz Bieganowski, Daniel Strzelecki, Robert Skiba, and Mateusz Topolewski. Putnam-like dataset summary: Llms as mathematical competition contestants. *arXiv preprint arXiv:2509.24827*, 2025.

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.

Hui Chen, Miao Xiong, Yujie Lu, Wei Han, Ailin Deng, Yufei He, Jiaying Wu, Yibo Li, Yue Liu, and Bryan Hooi. MLR-Bench: Evaluating AI agents on open-ended machine learning research, 2025a. URL https://arxiv.org/abs/2505.19955.

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.

Gustavo Deco, Adrían Ponce-Alvarez, Patric Hagmann, Gian Luca Romani, Dante Mantini, and Maurizio Corbetta. How local excitation–inhibition ratio impacts the whole brain dynamics. *Journal of Neuroscience*, 34(23):7886–7898, 2014. doi: 10.1523/JNEUROSCI.5068-13.2014.

Jose Carlos del Toro Iniesta and Basilio Ruiz Cobo. Inversion of the radiative transfer equation for polarized light. *Living Reviews in Solar Physics*, 13:4, 2016. doi: 10.1007/s41116-016-0005-2.

Mariia Demianenko, Konstantin Malanchev, Ekaterina Samorodova, Mikhail Sysak, Aleksandr Shiriaev, Denis Derkach, and Mikhail Hushchyn. Understanding of the properties of neural network approaches for transient light curve approximations. *Astronomy & Astrophysics*, 677:A16, 2023.

Karl J. Friston, Andrea Mechelli, Robert Turner, and Cathy J. Price. Nonlinear responses in fMRI: The Balloon model, Volterra kernels, and other hemodynamics. *NeuroImage*, 12(4):466–477, 2000. doi: 10.1006/nimg.2000.0630.

Gary H. Glover. Deconvolution of impulse response in event-related BOLD fMRI. *NeuroImage*, 9 (5):416–429, 1999. doi: 10.1006/nimg.1998.0419.

Daniel A. Handwerker, John M. Ollinger, and Mark D'Esposito. Variation of BOLD hemodynamic responses across subjects and brain regions and their effects on statistical analyses. *NeuroImage*, 21(3):1639–1651, 2004. doi: 10.1016/j.neuroimage.2003.11.031.

Justine Y. Hansen, Golia Shafiei, Ross D. Markello, et al. Mapping neurotransmitter systems to the structural and functional organization of the human neocortex. *Nature Neuroscience*, 25: 1569–1581, 2022. doi: 10.1038/s41593-022-01186-3.

Xuan Kan, Wei Dai, Hejie Cui, Zilong Zhang, Ying Guo, and Carl Yang. Brain network transformer. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Gregor Kasieczka, Benjamin Nachman, David Shih, Oz Amram, Anders Andreassen, Kees Benkendorfer, Blaz Bortolato, Gustaaf Brooijmans, Florencia Canelli, Jack H Collins, et al. The lhc olympics 2020 a community challenge for anomaly detection in high energy physics. *Reports on progress in physics*, 84(12):124201, 2021.

Amelia Liu, Andrew Ho, Anne Marie Droste, David Martin, Edmund Wong, Edward Zhou, Isabelle Zhou, Joshua Park, Joy Jiao, Katie-Rose Skelly, Kenny Kim, Kevin Rao, Masatoshi Uehara, Max Marion, Nicole Fitzgerald, Rachel Dias, Suyash Shringarpure, Yuan Yuan, and Yunyun Wang. LifeSciBench: Evaluating language models on realistic, expert-level tasks in the life sciences. Technical report, OpenAI and Tacit Labs, 2026. URL https://cdn.openai.com/pdf/b4299379-0a97-4ffa-8b9b-c3fbb299caa9/lifescibench_preprint.pdf.

Henry E. Miller, Matthew Greenig, Benjamin Tenmann, and Bo Wang. BioML-bench: Evaluation of AI agents for end-to-end biomedical ML. *bioRxiv*, 2025. doi: 10.1101/2025.09.01.673319. URL https://doi.org/10.1101/2025.09.01.673319.

Lukia Mistryukova, Andrey Plotnikov, Aleksandr Khizhik, Irina Knyazeva, Mikhail Hushchyn, and Denis Derkach. Stokes inversion techniques with neural networks: Analysis of uncertainty in parameter estimation. *Solar Physics*, 298:98, 2023. doi: 10.1007/s11207-023-02189-4.

Ludovico Mitchener et al. Kosmos: An AI scientist for autonomous discovery. *arXiv preprint arXiv:2511.02824*, 2025.

Lia Papadopoulos, Christopher W. Lynn, Demian Battaglia, and Danielle S. Bassett. Relations between large-scale brain connectivity and effects of regional stimulation depend on collective dynamical state. *PLoS Computational Biology*, 16(9):e1008144, 2020. doi: 10.1371/journal.pcbi.1008144.

Pavel Popov, Usman Mahmood, Zening Fu, Carl Yang, Vince Calhoun, and Sergey Plis. A simple but tough-to-beat baseline for fMRI time-series classification. *NeuroImage*, 303:120909, 2024. doi: 10.1016/j.neuroimage.2024.120909.

Jiahao Qiu et al. Physics supernova: AI agent matches elite gold medalists at IPhO 2025. *arXiv preprint arXiv:2509.01659*, 2025a.

Shi Qiu, Shaoyang Guo, Zhuo-Yang Song, Yunbo Sun, Zeyu Cai, Jiashen Wei, Tianyu Luo, Yixuan Yin, Haoxu Zhang, Yi Hu, et al. Phybench: Holistic evaluation of physical perception and reasoning in large language models. *arXiv preprint arXiv:2504.16074*, 2025b.

Shi Qiu, Junyi Deng, Yiwei Deng, Haoran Dong, et al. PRBench: End-to-end paper reproduction in physics research, 2026. URL https://arxiv.org/abs/2603.27646.

Huilin Qu, Congqiao Li, and Sitian Qian. Particle transformer for jet tagging. In *International Conference on Machine Learning*, pp. 18281–18292. PMLR, 2022.

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.

Paula Sanz-Leon, Stuart A. Knock, Andreas Spiegler, and Viktor K. Jirsa. Mathematical framework for large-scale brain network modeling in The Virtual Brain. *NeuroImage*, 111:385–430, 2015. doi: 10.1016/j.neuroimage.2015.06.016.

Jianhao Shen, Ye Yuan, Srbuhi Mirzoyan, Ming Zhang, and Chenguang Wang. Measuring vision-language stem skills of neural models. *arXiv preprint arXiv:2402.17205*, 2024.

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

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. Paperbench: Evaluating AI's ability to replicate AI research. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=xF5PuTLPbn.

Qiushi Sun, Zhoumianze Liu, Chang Ma, Zichen Ding, Fangzhi Xu, Zhangyue Yin, Haiteng Zhao, Zhenyu Wu, Kanzhi Cheng, Zhaoyang Liu, et al. Scienceboard: Evaluating multimodal autonomous agents in realistic scientific workflows. In *International Conference on Learning Representations*, volume 2026, pp. 75694–75731, 2026.

Terminal-Bench-Science Team. Terminal-Bench-Science: Evaluating AI agents on research workflows across scientific domains, 2026. URL https://github.com/harbor-framework/terminal-bench-science.

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.

Ekaterina Trofimova, Zosya Shamina, Maria Selifanova, Artem Zaitsev, Remi Savchuk, Maxim Minets, Daria Ozerova, Emil Sataev, Denis Zuenko, and Andrey Ustyuzhanin. Ml2b: Benchmarking llms on cross-lingual ml pipeline generation. 2026.

Yuru Wang, Lejun Cheng, Yuxin Zuo, Sihang Zeng, Bingxiang He, Che Jiang, Junlin Yang, Yuchong Wang, Kaikai Zhao, Weifeng Huang, et al. Naturebench: Can coding agents match the published sota of nature-family papers? *arXiv preprint arXiv:2606.24530*, 2026.

Kyle Waters, Lucas Nuzzi, Tadhg Looram, Alessandro Tomasiello, Ariel Ghislain Kemogne Kamdoum, Bikun Li, Damien Sileo, Egor Kretov, Francesco Fournier-Facio, Georgios Soloupis, Haile Kassahun, Hew Wolff, Jiaqi Cai, Lianghui Li, Marc Roth, Mohinder Naiya, Naixu Guo, Qicheng Tang, Richard Wheeler, Samuele Sala, Serguei Popov, Steven Dillmann, and Yuqi Li. COMPOSITE-Stem, 2026. URL https://arxiv.org/abs/2604.09836.

Jiaqi Wei et al. From AI for science to agentic science: A survey on autonomous scientific discovery. *arXiv preprint arXiv:2508.14111*, 2025.

Hugh R. Wilson and Jack D. Cowan. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical Journal*, 12(1):1–24, 1972. doi: 10.1016/S0006-3495(72)86068-5.

Guo-Rong Wu, Wei Liao, Sebastiano Stramaglia, Ju-Rong Ding, Huafu Chen, and Daniele Marinazzo. A blind deconvolution approach to recover effective connectivity brain networks from resting state fMRI data. *Medical Image Analysis*, 17(3):365–374, 2013. doi: 10.1016/j.media.2013.01.003.

Yunze Wu, Dayuan Fu, Weiye Si, Zhen Huang, Mohan Jiang, Keyu Li, Shijie Xia, Jie Sun, Tianze Xu, Yang Xiao, et al. Innovatorbench: evaluating agents' ability to conduct innovative ai research. In *International Conference on Learning Representations*, volume 2026, pp. 36144–36185, 2026.

Xin Xu, Qiyun Xu, Tong Xiao, Tianhao Chen, Yuchen Yan, Jiaxin Zhang, Shizhe Diao, Can Yang, and Yang Wang. Ugphysics: A comprehensive benchmark for undergraduate physics reasoning with large language models. *arXiv preprint arXiv:2502.00334*, 2025.

Mohd Zaki, NM Krishnan, et al. Mascqa: A question answering dataset for investigating materials science knowledge of large language models. *arXiv preprint arXiv:2308.09115*, 2023.

Anqi Zou, Han Deng, Chengyu Zhang, Junquan Hu, Yu Wang, Yuxiang Xing, Aokai Zhang, Hanling Zhang, Zhaoyang Liu, Ben Fei, et al. Labosbench: Benchmarking computer use agents for scientific instrument control. *arXiv preprint arXiv:2606.16802*, 2026.

A Task Descriptions

This appendix states each of the twelve problems as the agent receives it: what is visible in the sandbox, what the submission contract requires, and the property that keeps a competent generic pipeline well below the expert anchor. Evaluation units, metric families and the two anchors are given once, in Table 3, and are not repeated here. Identifiers are the directory names in the release.

A.1 HIGH-ENERGY PHYSICS

New-physics search (`lhco_anomaly`). Half a million unlabelled dijet events, seven observables per jet; submit one anomaly score per test event. No labels exist at any point, so the discriminant must come from the unlabelled mixture itself at $S/B \sim 10^{-2}$. The resonance masses are disclosed, as in the public LHC Olympics release, so the difficulty is statistical rather than one of guessing the signal model Kasieczka et al. (2021).

Higgs uncertainty (`higgsml_uncertainty`). Ten million weighted simulated events; submit a point estimate of the signal strength and a central interval per pseudo-experiment. Each pseudo-experiment is drawn at its own hidden values of six nuisance parameters, so the test distribution is shifted. Only coverage and width are scored: a sharp interval that under-covers loses to a wider one that covers Bhimji et al. (2026).

Top tagging (`top_quark_tagging`). Jets as unordered variable-length sets of constituent four-momenta plus global kinematics; submit a top-jet probability. Two of the four score components are background rejections on a logarithmic scale, so a percent-level gain in discrimination moves the headline far more than the same gain in accuracy. Rewards permutation-invariant pooling, penalises flattening Qu et al. (2022).

Dark Machines (`darkmachines_anomaly`). Four channels of background-only events, up to 7.7 million in a ragged text format. The submission is a model class with `fit` and `predict` rather than a prediction file: the grader instantiates a fresh model per channel, trains it on that channel's background and applies it to hidden files holding every BSM signal. No signal is visible to the agent and the metric cannot be computed locally Aarrestad et al. (2021).

A.2 ASTROPHYSICS

Stokes inversion, simulated (`stokes_inversion`). Simulated Stokes spectra around one Fe I line; submit three quantile arrays over eleven Milne–Eddington parameters *and* your own Unno–Rachkovsky forward model (del Toro Iniesta & Ruiz Cobo, 2016; Mistryukova et al., 2023). Azimuth is identifiable only modulo 180°, exact singular points make some parameters unobservable, and filling factor trades against field strength. The grader resynthesises the spectrum from the submitted parameters in an isolated container.

Stokes inversion, real (`hinode_inversion`). Real Hinode/SOT-SP rasters at a two-line pair; the same submission plus complete parameter maps (del Toro Iniesta & Ruiz Cobo, 2016; Mistryukova et al., 2023). The reference is a Merlin inversion that is itself noisy, real instrument effects are present, and the source-function convention needs a gain the forward model must apply. The spatial term is scored on the parameter maps rather than on synthesised spectra, so smoothing the output cannot buy it.

SN Ia classification (`lightcurve_classification`). Irregularly sampled, heteroscedastic six-band light curves with fewer than five hundred labelled objects; submit a per-object probability Demianenko et al. (2023). The work lies in representing a sparse, gappy, multi-band series rather than in model capacity. The log-loss component requires calibration, so a good ranking with over-confident scores is penalised.

A.3 NEUROSCIENCE

ASD classification (`abide_classification`). Resting-state fMRI on a 200-region atlas from sixteen training sites; submit an autism probability and a chronological age for every subject of four entirely held-out sites (Kan et al., 2022; Popov et al., 2024). Scored on discrimination plus calibration against the training prevalence, with an auxiliary age term. The ceiling is set by cross-scanner transfer, not by method quality.

NMM family identification (`nmm_forward_modelling`). Forty-five anonymous three-node recordings; name the generating neural-mass family for each (Sanz-Leon et al., 2015; Deco et al.,

2014). Equations, constants and parameter ranges are withheld, only the recorded observable per family is disclosed, and several series sit in corners of one family's parameter space that mimic another. The score is chance-corrected, so hedging earns nothing.

NMM digital twins (`nmm_digital_twins`). Spontaneous baseline activity of a 30-region delayed Wilson–Cowan system (Wilson & Cowan, 1972), its connectivity and delays, and a future intervention; submit the signed, time-resolved causal difference between activity with and without it. Matching resting means, spectra or connectivity does not determine the response (Papadopoulos et al., 2020), and the shared self-excitation gain and per-node drive parameters that set the operating regime are withheld; the model's remaining coefficients, numerics and noise scale are published in a manifest.

NMM parameter recovery (`nmm_parameter_recovery`). z-scored BOLD per subject; submit global coupling and two receptor-weighted gain parameters (Deco et al., 2014; Hansen et al., 2022). The generator is fully specified in the task description, so the whole difficulty is the inverse problem: inference over a stochastic delayed mean-field model seen through a slow, noisy, standardised channel. Standardisation removes amplitude, so only the regional pattern carries the parameters.

BOLD deconvolution (`rs_bold_deconvolution`). Two paired hemodynamic views of one neural realisation, with no stimulus timing (Handwerker et al., 2004; Wu et al., 2013); submit the shared latent neural drive, a double-gamma kernel bank and a Balloon–Windkessel parameter bank (Glover, 1999; Friston et al., 2000). The two kernel terms dominate the score and are gated, so a regionally constant template scores zero. Balloon–Windkessel parameters are credited only up to functional equivalence, through a frozen ODE on four probes.

B Agent-Facing Prompt Surface

Two texts fully determine what an agent sees before it begins: a system prompt shared by all twelve tasks, and the task's own `description.md`. Neither reveals the grader implementation, the anchors, or any ground truth. We reproduce the system prompt in full, followed by one complete task prompt. The remaining eleven are released with the benchmark.

B.1 System prompt

```
You are an autonomous scientific problem-solving agent.

Your primary objective is to maximize the scientific score using only the visible workspace
     data and the tools available in the sandbox. Structural validity and reproducibility are
      acceptance requirements, not substitutes for solving the scientific problem.

Before writing solution files, read the complete task brief and relevant public metadata.
     Identify the scored scientific target, the variation the metric rewards, and every
     stated numerical or modelling convention. Inspect the actual data, select a
     scientifically appropriate method, implement it, and test the scored outputs. A
     submission can pass structural validation and still score zero. Do not rely on constant,
      group-midpoint, globally broadcast, or data-ignoring predictions when the metric
     rewards continuous, sample-specific, regional, or temporal variation.

Do not search outside the workspace for existing solutions, hidden answers, grader code, or
     private data.

After inspecting the task and data, create a valid initial submission before starting long
     training, simulation, or optimization jobs. Keep the required files valid as you improve
      the scientific method so a budget limit cannot erase all completed work.

After that checkpoint is valid, use available budget for substantive scientific improvement.
     Diagnose the current method from public data, identify its largest unresolved
     uncertainty, and test a materially different method, modelling choice, or numerical
     setting. Keep an improvement only when public-data diagnostics support it. Do not spend
     budget on repeated summaries, unchanged tests, or speculative prose; do not repeat an
     experiment unless its hypothesis, implementation choice, or diagnostic changes.

Once the scientific method produces working scored outputs, create the smallest
     reproducibility package required by the task. Reuse the tested implementation; do not
     spend budget polishing documentation or package structure. Produce every scored output
```

```
     and required artifact at its exact relative path, preserving required subdirectories.
     The final `run.py` must read only visible task data and regenerate the scored outputs in
      the requested output directory, and `README.md` must give concise usage instructions.

Before finishing:

1. Run the solution entry point and fix execution errors.
2. Confirm that every required scored output and artifact exists at its exact path.
3. Call `validation_feedback()`, fix every reported structural problem, and validate again.
4. Call `submit()` only after validation succeeds. `submit()` independently rechecks the
     submission and rejects incomplete or malformed output.

If the full method cannot be completed within the budget, submit the strongest scientifically
      defensible working solution available. Do not replace the scientific method with an
     empty or deliberately invalid stub.
```

B.2 EXAMPLE TASK PROMPT: STOKES_INVERSION

We reproduce this task because it exercises four of the seven properties of Section 3 at once: an underdetermined solution set with exact structural non-identifiability (P1), a submission that includes the agent's own forward model rather than predictions alone (P2), a composite geometrically gated metric (P5), and a failure mode in which a spectrum is reproduced while the physics is violated (P6).

```
# Simulated Stokes Inversion

Given a simulated Stokes spectrum on a fixed 56-point wavelength grid around the Fe I 6302.5
      line, predict the 11 MilneEddington atmospheric parameters for each pixel independently
     , together with an 80% central prediction interval (the 10th / 50th / 90th percentiles).

## Scientific motivation

The Sun's photospheric magnetic field governs every form of solar activity that matters at
     human timescales  sunspots, the 11-year cycle, flares, coronal mass ejections, and the
     space-weather drivers that disrupt satellites and power grids. Its only direct remote
     diagnostic is **spectropolarimetry**: in the presence of a magnetic field, spectral
     lines split into polarized components through the Zeeman effect, and the four Stokes
     parameters $(I, Q, U, V)$ encode the full magnetic vector together with the
     thermodynamic and velocity state of the line-forming layer. Recovering atmospheric
     parameters from observed Stokes profiles  the **inversion problem**  is therefore the
     central numerical task of modern solar physics.

In the MilneEddington (ME) approximation, the line-forming atmosphere is summarised by 11
     parameters: magnetic-field strength $B$, inclination $\theta$, azimuth $\chi$, Doppler
     width $\Delta_D$, damping $a$, line-strength ratio $\eta_0$, the source-function pair $(
     S_0, S_1)$, line-of-sight velocity $v_\text{LOS}$, magnetic filling factor $\alpha$, and
      stray-light velocity $v_\text{stray}$. The forward map from these eleven scalars to the
      four Stokes profiles is the analytical UnnoRachkovsky solution of the polarized
     radiative-transfer equation. Participants are expected to implement their own forward
     map from the published solar-physics literature (e.g. Landi Degl'Innocenti & Landolfi
     2004).

This benchmark isolates this inverse problem in a **fully simulated, fully controlled**
     setting. The training and test data are produced from the validated ME forward model
     with channel-dependent Gaussian noise; generating parameters are exactly known to the
     organizer, and test targets remain private. The challenge is pixelwise: each spectrum is
      processed independently, with no spatial context. This isolates the per-pixel inversion
      competence as a clean probe of a method's ability to invert a polarised radiative-
     transfer model.

### Physical difficulties of ME inversion

Even with perfect training data, the inverse problem has structural difficulties that any
     method must address:

1. **Non-linearity.** The UnnoRachkovsky forward map composes Voigt profiles, FaradayVoigt
     anomalous-dispersion profiles, and trigonometric functions of the field angles. Small
     parameter changes can produce discontinuous-looking changes in the spectrum (e.g. when
     an absorption profile transitions from un-split to split as $B$ crosses a Doppler-width-
     dependent threshold), and large parameter changes can produce visually identical spectra
     .
2. **The 180-degree azimuth ambiguity.** The azimuth $\chi$ enters Stokes $Q$ and $U$ only
     through $\sin(2\chi)$ and $\cos(2\chi)$. The pair $\chi$ and $\chi + 180$ produce **
     identical** Stokes profiles, so there is no information in the spectrum that
     distinguishes them. The benchmark treats $\chi$ as a **circular variable with period 180
     ** everywhere in scoring (submission validation, pinball loss, interval coverage,
     interval width).
```

```
3. **Structural non-identifiability at singular points.** A few exact generator-defined
     sentinels make specific parameters genuinely unobservable for a given pixel, as a matter
      of physics rather than noise:
   - $\alpha = 0$ (no magnetic component): $B, \theta, \chi, v_\text{LOS}$ are unobservable.
   - $B = 0$ (zero field): $\theta, \chi$ are unobservable (no transverse or longitudinal
     direction).
   - $\alpha = 1$ (no stray component): $v_\text{stray}$ is unobservable.
   These exact-singularity samples are **masked out** of the score for the affected
     parameters only; near-singular cases remain scored and an honest method can protect
     itself through appropriately wide conditional intervals.
4. **Polarization signal-to-noise varies with $B$, geometry, $\alpha$, and noise.** Weak-
     field, near-longitudinal, or low-$\alpha$ pixels have faint polarization and the
     recovered field direction is noisy  but this is governed by the polarization SNR
     actually present in the spectrum, not by a fixed $B$ threshold. The dataset contains
     weak-field pixels with detectable polarization; do not assume a blanket cutoff.
5. **Filling-factor / field-strength degeneracy.** A weak field filling the entire pixel ($\
     alpha = 1$, small $B$) can produce a circular-polarization amplitude very similar to a
     strong field filling a fraction of the pixel ($\alpha < 1$, large $B$). When $\alpha \to
      1$, $v_\text{stray}$ becomes mathematically undefined.

## Dataset

The published dataset lives under `dataset/`:

| File | Shape | dtype | Role |
|---|---|---|---|
| `spectra_train.npy` | `(100000, 56, 4)` | float32 | absolute-scale Stokes $[I, Q, U, V]$,
     with noise |
| `params_train.npy`  | `(100000, 11)`   | float64 | ground-truth training parameters |
| `spectra_test.npy`  | `(10000, 56, 4)` | float32 | test spectra (the only test input you
     receive) |
| `wavelength_grid.npy` | `(56,)` | float64 | m offsets from Fe I 6302.5  line center |

`params_test.npy` (shape `(10000, 11)`) is the withheld ground truth used for scoring; it is
     **not** an input. Load the published arrays directly with `numpy`:

'''python
import numpy as np
spectra_train = np.load("spectra_train.npy")   # (100000, 56, 4) float32
params_train  = np.load("params_train.npy")    # (100000, 11)   float64
spectra_test  = np.load("spectra_test.npy")    # (10000, 56, 4) float32
wavelength    = np.load("wavelength_grid.npy") # (56,) m offsets from Fe I 6302.5
'''

The harness stages these four arrays plus the `public_manifest.json` dataset manifest  and
     nothing else  into the agent workspace; the exact mount path is supplied by the
     STEMBench instruction wrapper. Ground-truth files marked `role: ground_truth` in `config
     .yaml` (`params_test.npy`, `test_metadata.npz`) are never mounted into the agent sandbox
     .

### The 11 physical parameters (canonical order  never re-order)

| Idx | Symbol | Name | Unit | Range |
|---:|---|---|---|---|
| 0 | `B` | Magnetic field strength | G | $[0, 5000]$ |
| 1 | `theta` | Field inclination from line of sight | deg | $[0, 180]$ |
| 2 | `chi` | Field azimuth in plane of sky (period 180) | deg | $[0, 180)$ |
| 3 | `Delta_D` | Doppler width | m | $[20, 90]$ |
| 4 | `a` | Damping (Voigt parameter) |  | $[0, 1.5]$ |
| 5 | `eta_0` | Line-strength ratio |  | $[0.01, 100]$ |
| 6 | `S_0` | Source function at $\tau = 0$ | intensity | $[0, 38\,603]$ |
| 7 | `S_1` | Source function gradient | intensity | $[0, 60\,464]$ |
| 8 | `v_LOS` | LOS velocity (magnetic component) | km/s | $[-10, 10]$ |
| 9 | `alpha` | Magnetic filling factor |  | $[0, 1]$ |
| 10 | `v_stray` | LOS velocity (field-free component) | km/s | $[-10, 10]$ |

### Train/test split composition

The split is **globally group-disjoint**: no exact 11-parameter vector and no source row is
     shared between training, the organizer-only calibration set, and the test set. Both
     train and test are **balanced across four magnetic-field regimes** of equal test weight:

| Regime | $B$ range (G) |
|---|---|
| weak | $[0, 200)$ |
| intermediate | $[200, 500)$ |
| moderate | $[500, 1500)$ |
| strong | $[1500, 5000]$ |
```

```
Each regime contributes an equal share of train and of test, so changes in sample composition
      cannot silently reweight the scientific question. The four regimes are averaged equally
      in the score.

The test set additionally contains a **stress** subset (20% of test, 2,000 rows). The stress
     designation and per-row core/stress labels are **private** (organizer-only); only
     aggregate core/stress results are reported after evaluation. The stress split preserves
     the $B$-regime  filling-factor balance and the one-dimensional parameter marginals (
     audited against the core split) but is a **stratified random diagnostic**, not an
     interpolation probe: the audit documents that stress joint cells do recur in training
     within each B regime. The headline score evaluates probabilistic recovery ($R$) and
     forward consistency ($F$) over the four equally weighted magnetic regimes; core vs
     stress is reported as a stratified random split diagnostic, not a prediction about which
      method classes should succeed or fail.

### Spectra, noise, and units

Spectra are stored in **absolute intensity units** (not continuum-normalised), so the
     continuum level directly anchors the source-function parameters $S_0$ and $S_1$. Channel
     -dependent additive Gaussian noise is already baked into the released spectra with per-
     channel sigma

$$\sigma_{I,Q,U,V} = [109,\,2,\,2,\,44]\,/\,20\,000 \;\cdot\; (S_0 + \mu S_1), \qquad \mu =
     1,$$

in continuum-intensity units. This is the benchmark noise model  a frozen, reproducible
     channel-dependent model; it is not an exact Hinode/SOT-SP calibration.

## Submission contract

A submission is a directory containing **three numpy arrays** of shape $(10000, 11)$ **plus a
      forward-model entry point**:

- `<submission_dir>/pred_q10.npy`  10th-percentile prediction
- `<submission_dir>/pred_q50.npy`  median (point prediction)
- `<submission_dir>/pred_q90.npy`  90th-percentile prediction
- `<submission_dir>/forward_model/entry_point.txt`  text file with one line of the form `<
     module>:<attr>` (e.g. `me:me_model`), naming a callable inside `<submission_dir>/
     forward_model/`
- `<submission_dir>/forward_model/<module>.py`  your implementation(s) of the callable

The grader **rejects** (scores 0, before any metric is computed) any submission with **
     structural** violations:

- A missing or extra-shaped array; an object, boolean, integer, NaN, or infinity value.
- Wrong test length or parameter count.

**Parameter-level violations are scored, not rejected**  the pinball metric penalizes them
     naturally, and the grader report lists them as warnings:

- Any non-circular parameter outside its physical bounds in any of the three quantile files.
- Reversed quantiles: `q10 > q50` or `q50 > q90` for any non-circular entry.
- For `chi`: `q50_chi` outside $[0, 180)$; `q10_chi > q50_chi` or `q50_chi > q90_chi`; either
      side extending more than $90$ from `q50_chi`.

`validation_feedback` reports these parameter issues as warnings so you can fix them before
     submitting.

### Forward-model contract (for the $F$ term)

The forward-consistency term $F$ synthesizes a noiseless spectrum from your submitted `q50`
     parameters using **your** forward model  not an organizer-supplied one  and compares it
     to the observed noisy test spectrum. `forward_model/entry_point.txt` is a **required
     submission file**; a submission missing it is rejected at validation. To compute $F$,
     the grader runs your forward code inside an **isolated Docker container** (non-root,
     read-only filesystem, no network, no access to private ground truth or the task root):

1. Reads `forward_model/entry_point.txt`, parses `<module>:<attr>`.
2. Imports `<module>` from your `forward_model/` directory inside the container.
3. Calls `<attr>(params)` where `params` is `(N, 11) float64`  your `q50`.
4. Expects the return to be `(N, 56, 4) float64` **absolute-scale noiseless** Stokes $[I,Q,U,
     V]$.

If the callable is present but fails to import, raises, or returns the wrong shape, $F = 0$.
     $R$ (recovery) is unaffected  you keep the credit for calibrated parameter quantiles
     even with a non-working forward. Line constants ($\lambda_0 = 6302.5$ , $g_\text{eff} =
     2.5$, $\mu = 1$; the 56-point wavelength grid in m spans 6302.0696303.254 ) must be
     embedded in your forward model code  either hard-coded or imported from a constants
     module shipped inside your `forward_model/` directory. At grade time the grader's own `
     grader/constants.py` is available on the import path inside the synthesis container; the
```

```
      canonical values are `LINE_VEC = (6302.5, 2.5, 1)` and `LINE_ARG = 1000 * (np.linspace
     (6302.0692255, 6303.2544205, 56) - 6302.5)` (in m).

Provide a physical synthesis model that preserves the meanings and units of the 11 inputs.
     The data were generated with a MilneEddington / UnnoRachkovsky atmosphere, but your
     implementation may use a richer, stratified or multicomponent atmosphere. Supply any
     additional atmospheric quantities deterministically within your forward model and
     document their relationship to the 11 inputs. The grader does not compare your
     implementation with a prescribed solver.

Submitted forward models also pass physical-validity probes (the $P_\text{phys}$ factor of
     the score); their specification is in `metric_description.md`. They check the physical
     meanings of the declared interface  not agreement with any particular ME solver.

### Chi (circular) interval convention

$\chi$ keeps the three-file contract but its endpoints live on a local unwrapped coordinate:

- `q50_chi` is the circular center, in $[0, 180)$.
- `q10_chi  q50_chi  q90_chi`, and each side may extend at most $90$ from `q50_chi`.
- `q10_chi` and `q90_chi` **may fall outside** $[0, 180)$ so an interval can cross the $0/180
     $ boundary.

During scoring, the true $\chi$ is lifted to the equivalent value nearest `q50` before the
     pinball loss, interval coverage, and interval width are computed. This makes every $\
     chi$ operation circular while preserving the three-file submission format. Predictions
     that differ only by a $180$ shift in $\chi$ score identically.

If you only produce a point estimate, set `q10 = q50 = q90` (a zero-width interval). This is
     valid but earns an uncertainty penalty under the pinball metric  calibrated intervals
     score better.

## Allowed Tools and Libraries

The sandbox has no network access, so only the packages pre-installed in the sandbox image
     can be used. Packages relevant to this task:

- `numpy`  numerical arrays and core math for the spectral cubes.
- `scipy`  scientific routines (optimization, interpolation, statistics).
- `scikit-learn`  classical machine-learning models and preprocessing.
- `xgboost`  gradient-boosted decision trees.
- `catboost`  gradient boosting on tabular features.
- `torch`  neural network models.
- `numba`  JIT compilation for fast pixelwise numerical loops.

## Scoring

A submission is scored on the test split as a single scalar in $[0, 1]$, **higher is better
     **, perfect = 1:

$$\text{FinalScore} \;=\; 0.80 \cdot R \;+\; 0.20 \cdot \sqrt{R \cdot F_\text{eff}}$$

where $R$ is a baseline-normalized, null-gated **probabilistic recovery** score and $F$ is a
     **forward-consistency** score. A constant train-marginal predictor scores at the floor (
     its raw skill is ˜0 by construction and the null gate floors the residual); an oracle
     scores $1$.

### $R$  probabilistic recovery (primary)

For each magnetic-field regime $r$ and parameter $k$, the grader computes the mean pinball
     loss across $\tau \in \{0.1, 0.5, 0.9\}$ over the observable sampleparameter pairs (
     exact singularity pairs are masked; $\chi$ is handled circularly):

$$L_{r,k} = \frac{1}{3}\sum_{\tau \in \{0.1, 0.5, 0.9\}} \overline{\rho_\tau(y_{i,k},\, \hat
     q^{(\tau)}_{i,k})}, \qquad
\rho_\tau(y, \hat q) = \max\!\bigl(\tau(y-\hat q),\, (\tau-1)(y-\hat q)\bigr).$$

Let $L_{0,r,k}$ be the pinball loss of the constant train-marginal quantile predictor (the no
     -skill baseline) on the test split. The raw skill is baseline-normalized:

$$S^\text{raw}_{r,k} = \mathrm{clip}\!\left(1 - L_{r,k} / L_{0,r,k},\ 0,\ 1\right),$$

then null-gated and rescaled above a frozen null threshold $t_{r,k}$ (the 95th percentile of
     raw skill across $\geq 1000$ null submissions: label permutations, shuffled good
     predictions, random bounded predictions):

$$S_{r,k} = \begin{cases} 0 & S^\text{raw}_{r,k} \le t_{r,k} \\ (S^\text{raw}_{r,k} - t_{r,k
     })/(1 - t_{r,k}) & \text{otherwise.}\end{cases}$$

Within each regime, skills are grouped by physical role and weighted:
```

```
$$S_\text{mag} = \mathrm{mean}(S_B, S_\theta, S_\chi, S_\alpha),\quad
S_\text{thermo} = \mathrm{mean}(S_{\Delta_D}, S_a, S_{\eta_0}, S_{S_0}, S_{S_1}),\quad
S_\text{vel} = \mathrm{mean}(S_{v_\text{LOS}}, S_{v_\text{stray}}),$$
$$R_r = 0.50\,S_\text{mag} + 0.30\,S_\text{thermo} + 0.20\,S_\text{vel}, \qquad
R = \mathrm{mean}(R_\text{weak}, R_\text{intermediate}, R_\text{moderate}, R_\text{strong}).
    $$

The grader applies frozen monotone transforms internally before computing pinball loss  $\
    log1p$ on $B$ and $\eta_0$, identity on the rest  so quantiles are equivariant and the
    score is invariant to test-composition changes. Pearson correlation is **not** a
    component of the score.

### $F$  forward consistency (physical-equivalence gate)

For each test row, the grader synthesizes a noiseless spectrum from the submitted `q50`
    parameters using **the participant's submitted forward model** (declared in `
    forward_model/entry_point.txt`), and compares it to the observed noisy test spectrum
    with the per-sample channel noise:

$$D_i = \mathrm{mean}_{\lambda, c}\ \mathrm{Huber}\!\left(\frac{x_\text{obs} - x_\text{pred}(
    q_{50})}{\sigma},\ \delta\!=\!3\right).$$

Per regime, $F_r = \mathrm{clip}\!\left((D_0 - D_r)/(D_0 - D_\text{oracle}),\ 0, 1\right)$,
    where $D_0$ is the constant-baseline residual and $D_\text{oracle}$ the hidden-truth
    residual; then $F = \mathrm{mean}_r F_r$. The task is to provide a genuine physical
    synthesis model, and forward consistency is scaled by physical validity $P_\text{phys} \
    in [0, 1]$ ($F_\text{eff} = F \cdot P_\text{phys}$). $F$ rewards a physically equivalent
     inversion (good spectral fit) but the geometric gate $\sqrt{R \cdot F_\text{eff}}$
    ensures a pure forward fit with arbitrary latent parameters cannot win on its own  $F$
    only helps when recovery is already non-trivial.

### Reported diagnostics (not additional score terms)

The grader report includes, beyond `FinalScore`: $R$ and $F$; per-parameter pinball skill;
    the four $B$-regime scores; core vs stress scores; $q_{10}$$q_{90}$ coverage and width;
    circular-$\chi$ coverage and width; median MAE in physical units; polarization-SNR
    strata; exact-singularity counts and masks; forward residual by Stokes channel.

## Evaluation

The STEMBench harness scores a submission automatically by calling
`grade(submission_dir, answers_path)` from `grade.py`. No scoring command is
issued by the participant. To produce a scoreable submission, write exactly
three files to your solution directory:

- `pred_q10.npy`  shape `(10000, 11)`, float32/float64
- `pred_q50.npy`  shape `(10000, 11)`, float32/float64
- `pred_q90.npy`  shape `(10000, 11)`, float32/float64

Ground truth is never mounted into the agent sandbox; the harness reads it
host-side.
```

C Run Configuration

Table 4 specifies the per-episode configuration used for every run reported in Sections 5 and E, unless explicitly stated otherwise (the extended-limits ablation of Table 8 is the only exception). All fields are harness-level run policy, declared at launch and logged in every episode's metadata; none of them is part of a task's `config.yaml`; the wall-clock, token and message budgets are restated to the agent in its instructions.

All campaign episodes executed on single-GPU instances: one NVIDIA Tesla T4 (16 GB) on an Intel Ice Lake platform with 8 vCPU and 32 GB RAM, Ubuntu 24.04, with the per-domain CUDA container image holding the only software the agent can touch.

Model access uses an OpenAI-compatible protocol through the OpenRouter gateway; sampling parameters (temperature, top-p) are not exposed on this route for reasoning models and were not set, which is why run-to-run variability is measured by repetition.

Harness and contract tests. The harness is covered by a functional test suite over the tool layer, which passes 17/17 checks: the six-tool registry, the eight file formats that `data_explorer` advertises, the refusal of unsafe deserialisation, valid and invalid submission checks (including split-keyed JSON and per-key `.npz` constraints), and the mapping from the submission checks

Table 4: Per-episode run configuration. "Value" is the setting used for all reported runs; "Semantics" states what the field controls and how the harness enforces it.

Field	Value	Semantics
`sandbox`	`docker-cuda`	Offline per-domain container (network disabled, public data mounted read-only, writable solution directory).
`time_limit_sec`	3 600	Wall-clock budget per episode. The agent loop is cancelled at the deadline and whatever submission is present at that moment is graded; a single `bash`/`python` execution may run for at most two thirds of the budget.
`token_limit`	10^6	Cumulative model tokens (input plus output) across all model calls of the episode; the next model call is refused once the budget is spent.
`max_messages`	200	Total conversation messages (user, assistant and tool results) per episode; the episode ends when the limit is reached.
`max_tool_calls`	200	Declared tool-call budget, recorded in the run plan and episode metadata. It is not an independent stop condition: every tool call adds messages, so the message limit binds first.
`effort`	low / medium / high	Reasoning-effort tier forwarded to the model API as `reasoning.effort` through the OpenRouter gateway; the experimental variable of Section E, with `high` the harness default.

in `config.yaml` to validator specifications. All twelve task contracts additionally pass dataset-checksum verification and smoke tests that grade the shipped baseline and expert submissions and reproduce the anchors in Table 3.

D PER-MODEL RESULTS

Table 5 reports eight frontier models under the **basic** scaffold at high reasoning effort. Scores are the task's native headline metric in $[0, 1]$ under the current task revisions, and each cell is the mean over three retained scored runs, including zeros.

Table 5: STEMBENCH results, **basic** agent, raw task scores. Each cell is the mean over three scored high-effort runs retained in the current report (288 runs in total), including zeros. Domain means are unweighted means of the task means, computed before rounding. Bold denotes the highest unrounded mean in each row. b/e are the baseline and expert anchors. †The digital-twin expert anchor is $e = 0.715233$ (displayed as 0.72).

Task	b	e	Gemini 3.7 Flash	Gemini 3.1 Pro	Claude Opus 5	DeepSeek V4 Pro	Qwen3.8 2.4T	GLM 5.2	MiMo V2.5	MiniMax M3
New-physics search	0.33	0.91	0.31	**0.93**	0.53	0.83	0.54	0.62	0.00	0.38
Higgs uncertainty	0.40	0.69	0.41	0.43	**0.48**	0.39	0.44	0.28	0.14	0.13
Top tagging	0.06	0.88	0.56	0.65	**0.94**	0.89	0.93	0.60	0.59	0.56
Dark Machines	0.00	0.22	0.22	0.29	0.25	0.28	0.22	**0.32**	0.18	0.20
HEP mean	—	—	0.37	0.57	0.55	**0.60**	0.53	0.46	0.23	0.32
Stokes, simulated	0.01	0.62	0.01	**0.78**	0.57	0.25	0.00	0.00	0.43	0.11
Stokes, real	0.00	0.60	0.12	**0.82**	0.09	0.00	0.00	0.00	0.14	0.00
SN Ia classification	0.42	0.85	0.50	0.91	**0.94**	0.90	0.93	0.60	0.93	0.93
Astro mean	—	—	0.21	**0.84**	0.53	0.38	0.31	0.20	0.50	0.34
ASD classification	0.00	0.31	0.09	0.00	0.00	0.10	0.10	**0.29**	0.08	0.19
NMM family id.	0.00	0.90	0.00	**1.00**	0.00	0.51	0.56	0.00	0.37	0.02
NMM digital twins†	0.10	0.72	0.10	0.24	**0.31**	0.14	0.10	0.03	0.00	0.10
NMM param. recovery	0.12	0.51	0.12	0.18	**0.42**	0.12	0.12	0.08	0.02	0.12
BOLD deconvolution	0.01	0.35	0.00	**0.19**	0.00	0.00	0.02	0.00	0.00	0.06
Neuro mean	—	—	0.06	**0.32**	0.15	0.18	0.18	0.08	0.09	0.10

Table 6 complements Table 5 with the *maximum* of the same three retained runs per cell. The two views separate ability from reliability: on the HEP tasks six of the eight models reach a best-run mean of 0.63–0.71, so the spread of the means in Table 5 is driven largely by how often a model fails, not by whether it can solve the task at all. The individual-run records quoted in Section 5 appear here as cell maxima: ABIDE at 0.31 (0.314 unrounded, DeepSeek V4 Pro) and parameter recovery at 0.58 (0.583, Claude Opus 5).

Table 6: STEMBENCH results, **basic** agent, *maximum* raw task scores. Each cell is the maximum over the same three retained high-effort runs as in Table 5, including zeros. Domain rows are unweighted means of the task maxima, computed before rounding. Bold denotes the highest unrounded maximum in each row. b/e are the baseline and expert anchors. †The digital-twin expert anchor is $e = 0.715233$ (displayed as 0.72).

Task	b	e	Gemini 3.7 Flash	Gemini 3.1 Pro	Claude Opus 5	DeepSeek V4 Pro	Qwen3.8 2.4T	GLM 5.2	MiMo V2.5	MiniMax M3
New-physics search	0.33	0.91	0.50	**0.94**	0.93	0.94	0.82	0.94	0.00	0.94
Higgs uncertainty	0.40	0.69	0.41	0.49	**0.53**	0.47	0.47	0.44	0.41	0.38
Top tagging	0.06	0.88	0.80	0.95	**0.95**	0.93	0.94	0.93	0.91	0.87
Dark Machines	0.00	0.22	0.33	0.33	0.38	**0.48**	0.35	0.34	0.32	0.32
HEP mean	—	—	0.51	0.68	0.70	**0.71**	0.64	0.66	0.41	0.63
Stokes, simulated	0.01	0.62	0.01	0.80	**0.85**	0.74	0.01	0.00	0.74	0.32
Stokes, real	0.00	0.60	0.18	**0.86**	0.15	0.00	0.00	0.00	0.41	0.00
SN Ia classification	0.42	0.85	0.50	0.93	**0.95**	0.93	0.94	0.95	0.94	0.93
Astro mean	—	—	0.23	**0.87**	0.65	0.56	0.31	0.32	0.70	0.42
ASD classification	0.00	0.31	0.28	0.00	0.00	**0.31**	0.31	0.31	0.25	0.24
NMM family id.	0.00	0.90	0.00	**1.00**	0.00	0.87	0.87	0.00	0.87	0.07
NMM digital twins†	0.10	0.72	0.10	0.26	**0.56**	0.23	0.10	0.10	0.00	0.10
NMM param. recovery	0.12	0.51	0.12	0.30	**0.58**	0.12	0.12	0.12	0.07	0.12
BOLD deconvolution	0.01	0.35	0.00	**0.27**	0.00	0.00	0.07	0.00	0.00	0.18
Neuro mean	—	—	0.10	**0.37**	0.23	0.31	0.30	0.11	0.24	0.14

E PILOT REASONING-EFFORT SWEEP

This appendix reports a pilot sweep run before the cross-model campaign of Section 5. Its absolute scores are not comparable with that campaign for two reasons: several task contracts were revised afterwards, and the runs predate a fix to GPU access in the task containers. We report it for what it shows about run-to-run variance and the non-monotone effect of reasoning effort within one model, not as a measurement of the current benchmark. After the pilot, all campaign runs were fixed at `high` effort.

We evaluate `google/gemini-3.1-pro-preview` with the `basic` scaffold and the limits of Appendix C, varying only the *reasoning effort* (`low`, `medium`, `high`): 12 tasks × 3 levels × 3 runs, with the `top_quark_tagging` high-effort cell extended by 11 runs to estimate single-run variance, for 119 graded runs. Runs interrupted by infrastructure (proxy) outages were flagged automatically and excluded; zeros in retained runs are genuine episode outcomes.

Table 7: Agent scores by task and reasoning effort, `google/gemini-3.1-pro-preview` on the `basic` scaffold. Cells give mean ± standard deviation over three runs, except `top_quark_tagging` at high effort ($n = 14$, marked †). "Best" is the maximum over all runs of the task; "0-runs" counts runs scoring exactly zero: a missing submission, a submission that failed validation, or a valid submission that did not outperform the no-skill baseline; the count does not distinguish these cases. The last rows aggregate the twelve per-task means.

	Task	low	medium	high	Best	0-runs
HEP	New-physics search	0.3 ± 0.3	0.3 ± 0.5	0.6 ± 0.5	0.932	4/9
	Higgs uncertainty	0.44 ± 0.05	0.46 ± 0.16	0.5 ± 0.1	0.640	0/9
	Top tagging	0.806 ± 0.006	0.86 ± 0.03	$0.8 \pm 0.2^{\dagger}$	0.941	0/20
	Dark Machines	0.22 ± 0.19	0.31 ± 0.03	0.32 ± 0.05	0.359	0/9
Astro	SN Ia classification	0.73 ± 0.02	0.87 ± 0.03	0.91 ± 0.04	0.935	0/9
	Stokes, simulated	0.07 ± 0.11	0.74 ± 0.07	0.5 ± 0.4	0.815	0/9
	Stokes, real	0.000 ± 0.000	0.3 ± 0.3	0.81 ± 0.03	0.831	4/9
Neuro	NMM family id.	0.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000	3/9
	NMM digital twins	0.27 ± 0.15	0.16 ± 0.09	0.18 ± 0.05	0.381	0/9
	NMM param. recovery	0.125 ± 0.000	0.2 ± 0.3	0.10 ± 0.04	0.468	2/9
	BOLD deconvolution	0.16 ± 0.14	0.28 ± 0.13	0.06 ± 0.09	0.410	1/9
	ASD classification	0.13 ± 0.16	0.327 ± 0.000	0.000 ± 0.000	0.327	4/9
Mean of task means		$\mathbf{0.272}$	$\mathbf{0.484}$	$\mathbf{0.484}$		
Median of task means		0.190	0.333	0.499		

Table 7 aggregates the grid. Three patterns are visible. First, reasoning effort buys most of its improvement between low and medium (mean of task means $0.272 \rightarrow 0.484$) and nothing on average

beyond it (0.484 at high), although the median continues to rise ($0.190 \rightarrow 0.333 \rightarrow 0.499$), so high effort helps the typical task while losing ground on a few. Second, the dispersion is not noise around a single mode: several tasks are bimodal, with runs landing either near the agent's best score or exactly at zero, which is what drives the large standard deviations on `lhco_anomaly` (± 0.53 at both medium and high) and `stokes_inversion` (± 0.44 at high). Four of nine runs on each of `lhco_anomaly`, `hinode_inversion` and `abide_classification` produced nothing that cleared validation or the no-skill baseline. Third, the effort–score relation is not monotone per task: `abide_classification` scores 0.327 ± 0.000 at medium and 0.000 in all three high-effort runs, and `rs_bold_deconvolution` degrades from 0.276 at medium to 0.055 at high. These aggregate scores do not establish whether the declines arise from invalid submissions, valid low-scoring predictions, or truncation; distinguishing these explanations requires run traces and validity flags.

Table 8 compares the main and extended-budget runs. Under extended budgets all three high-effort runs on `top_quark_tagging` land in $[0.937, 0.946]$ (mean 0.941 ± 0.004), above the main-budget mean, and the lower mode visible in the fourteen-run variance study of Table 7 (runs at 0.80 and 0.063) does not appear. At low effort the extended limits change nothing (0.802 ± 0.000), since low-effort agents stop early on their own. This is consistent with truncation contributing to high-effort variance, but the extended series also switched to Inspect's native OpenRouter provider, and the missing GPU acceleration may itself have caused part of the truncation; the ablation therefore does not isolate the effect of the budget.

Table 8: Budget ablation on `top_quark_tagging`. "Main" is the primary grid (1 h, 10^6 tokens, 200 messages, 200 tool calls); "Extended" raises these to 2.5 h, 2×10^6 tokens and 500/500, executed through Inspect's native OpenRouter provider. Mean $\pm$ standard deviation, with the observed range in brackets; $n = 3$ per cell. The fourteen-run variance study of the Main/high cell is reported in Table 7.

Effort	**Main**	**Extended**
low	0.806 ± 0.006 $[0.802, 0.813]$	0.802 ± 0.000 $[0.801, 0.802]$
medium	0.86 ± 0.03 $[0.829, 0.892]$	0.85 ± 0.05 $[0.802, 0.901]$
high	0.84 ± 0.16 $[0.652, 0.931]$	0.941 ± 0.004 $[0.937, 0.946]$