ULTRAFASTBENCH: A DETERMINISTIC BENCHMARK FOR PREDICTIVE EXPERIMENTAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce UltrafastBench to test final-answer prediction, diagnosis, and constrained control in specified ultrafast-optics systems. Its 400 questions span 24 physical-model families, including 240 design tasks. Executable references and typed answer fields support deterministic scoring without an LLM judge. On 380 unchanged screened prompts, five models answer 88–284 completely. On a frozen 400-question fresh-parameter cohort, GPT-6 Luna and GPT-6 Sol answer 115 and 303 without tools; isolated Python access raises these scores to 339 and 399, rescuing 233/285 and 97/97 prior misses. In four category-plus-estimate families, Sol chooses all 80 categories but completes only 49 answers closed-book, rising to 80 with Python. Saved traces show both numerical rescues and tool-induced errors. The results establish sensitivity to this Python-enabled protocol on new parameter worlds within the developed families. Versioned questions, references, and traces permit field-level replay.

1 INTRODUCTION

We use *predictive experimental reasoning* for three tasks about a specified optical system: forecasting a measurement, inferring a hidden setting from measurements, and choosing a feasible component or control. Questions state the physical model, givens, answer fields, and an objective when a design choice is requested. The evaluation measures final answers under those assumptions, rather than operation of a laboratory experiment.

We selected ultrafast optics because a pulse system can link temporal and spectral measurements to a hidden phase or setting and then to a feasible shaper or compressor choice. Dispersion and nonlinear phase couple these descriptions of the pulse (Weiner, 2011; Boscolo et al., 2015). Across the benchmark, questions cover forward prediction, inverse diagnosis, and control; some couple identification to a control choice. By stating the physical models, numerical inputs, candidate settings, and tie rules, we can compute reference answers and verify every requested field deterministically under the stated assumptions.

What is measured. The 400 questions comprise 90 predictions, 70 diagnoses, 91 identification-and-design tasks, and 149 constrained designs. Prediction asks for an outcome; diagnosis asks for a hidden setting; design asks for feasible controls, sometimes after identifying the model. Strict item accuracy requires every requested answer field.

Contributions.

- **Domain-specific tasks:** 400 ultrafast-optics questions from 24 model families, including 240 design questions with feasibility or robustness constraints.
- **Deterministic evaluation:** executable references and typed scoring, with versioned checks of calculations, wording, and answer contracts.
- **Field-level evidence:** screened-prompt and fresh-parameter results that separate correct choices from numerical estimates and auxiliary fields.

The evidence concerns specified mechanisms and new parameters within the developed families; it does not test unseen optical mechanisms or laboratory execution.

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.

SymPyBench already generates parameterized physics questions with executable targets (Imani et al., 2026); PRISM-Physics evaluates physics derivations and clarifications (Zhao et al., 2026). Our distinction is the specified ultrafast-optics prediction, inverse, and constrained-control slice, particularly questions coupling identification to a subsequent control choice. SciCode and other scientific-agent benchmarks score programs or repository work rather than these short answers (Tian et al., 2024; Duston et al., 2025; Chen et al., 2025). SciCode-Verified shows why specification and test defects can change a scientific benchmark's scores (Hu et al., 2026).

2 BENCHMARK AND VALIDITY

Question construction. Each family defines physical equations and a validity regime, samples a world, computes its reference answer before any model call, and renders a question with typed answer fields. Design questions usually specify a finite candidate set, feasibility constraints, an objective, and a tie rule so the reference can enumerate valid choices. A robust compressor task, for example, selects dispersion and temporal-lens settings that obey bounds across uncertain input chirps. Continuous optima are used only where the stated idealized model determines them.

The 400 screened questions allocate 50 items each to amplification, characterization, combs and CEP, dispersion and compression, mode locking, nonlinear optics, pulse Fourier optics, and experimental spectroscopy. This is parameter coverage within recurring mechanisms: numeric normalization leaves 68 prompt patterns, 19/24 families have one pattern, and 23/24 families have one task label. Two pairs of families also share core equations. The counts therefore describe parameter variation rather than 400 independent mechanisms; task type is mostly confounded with family.

Reference and semantic checks. We distinguish three checks: whether the physical calculation replays, whether the wording determines its reference answer, and whether the scorer accepts equivalent answers. Re-evaluation and a synthetic reference response pass for all 400 source items. The release records sampler seeds, sampled world values, reference fields, question text, and scorer inputs for replay. Independent numerical checks include QuTiP versus SciPy integration for three-level absorption, two solvers for two-level pump–probe, and direct comb sums versus a pypret Fourier calculation for CEP waveforms. Replaying a calculation alone cannot establish that the question has a unique physical interpretation; Figure 1 shows the three checks.

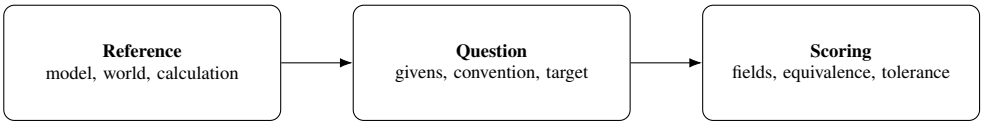


Figure 1: The executable reference, question wording, and scorer require separate checks. A solver round trip checks only the first and third.

Twenty noncollinear BBO prompts illustrate that distinction. The reference used $\Delta k = |\mathbf{K}_p - \mathbf{K}_s| - k_i$ with the idler along $\mathbf{K}_p - \mathbf{K}_s$, but the original wording did not define the scalar mismatch. A revision gave these prompts new IDs and specified the scalar; a later check still found the signal–idler angle convention implicit. We exclude all 20 from the five-model common-prompt score. The fresh version states the angle relation explicitly and received new responses. In a separate scorer defect, 38 comb/CEP prompts requested bare JSON although the common parser required a `FINAL` wrapper. Section 4 reports the effect of accepting complete JSON objects without changing model responses.

Development audit. A saved GPT-6 Pro review flagged 86 question versions for clarification and four keys as uncertain; these are leads, not established defects. Project records describe a two-reviewer return covering 780 scored prompts and 2,320 saved responses, but the available ledger has only consolidated verdicts, without separate votes or case-specific rationales. We corrected 97 no-answer verdicts using saved empty traces. The accuracy below remains deterministic agreement with executable references and typed contracts, not a separately adjudicated physical-equivalence score.

For the fresh v2 Luna run, a separate saved trace audit replayed all 400 strict scores and manually adjudicated its nine unparsed responses against their exact trace hashes. Eight gave no answer and one gave a substantively wrong answer, so this audit found no recoveries among unparsed responses: Luna remains at 115/400. Its parsed mismatches were classified by the scorer; this trace audit does not cover Sol or the Python-tool responses.

Later v3 audit leads. After the v2 evaluations, a corrected v3 retained the 400 sampled worlds and references while changing 360 question IDs to clarify wording or answer contracts. Four GPT-6 Pro packets provide per-item evidence for all 400 v3 questions; their unadjudicated labels flag 48 prompts, three keys, and 390 scorer contracts. Because v3 preserves all v2 worlds and reference targets, the Pro key calculations also check the v2 reference answers: 397 keys were marked valid and three uncertain. Only 40 questions retain identical v2 wording; the other 360 cannot inherit v3 prompt-validity judgments. These model-assisted findings are not established defects, and the packets do not audit saved v2 model responses. All fresh accuracy results below remain on v2.

3 Evaluation procedure

Cohorts. The screened 400 were assembled from an existing candidate pool: 49 earlier items were replaced, and 20 retained items were selected after a Gemini miss. The primary five-model comparison uses only the 380 prompts and answer fields unchanged by the BBO correction. For a separate test of new parameters, we sampled 400 worlds with the same family quotas. No model response selected those worlds. Validation found two wording defects after this draw; 40 prompts were revised with new IDs and new model responses while retaining their worlds and references. This scored v2 cohort has 65 numeric-normalized patterns and tests transfer within the developed families. Predetermined seeds and quotas select the first world in each family that passes its physical checks, reference replay, scorer round trip, and checks against exposed worlds and prompts. The reported fresh responses use the versioned `blind_transfer_luna_20260925_v2` dataset (SHA-256 `b7c56f1fbb7a82eac3d2f69644e695b61b620dfa608fc129836589cdc746f7f4`) in the supplement. The later v3 revision has no evaluation-model run reported here.

Model requests. Each closed-book request has one question, no prior turns, no system message, and no tools; all runs request high reasoning. GPT-6 Luna, GPT-6 Sol, DeepSeek v4.1 Flash, Gemini Flash, and Muse Spark 1.3 answer the 380 screened prompts. Luna and Sol also answer the 400 fresh prompts. Table 1 lists direct-answer caps and the Python-tool runs' cumulative cap; effective compute budgets are unequal. Direct-run request objects record exact requested and returned model IDs, settings, reported use, and responses. DeepSeek explicitly enabled thinking; Spark used the Responses `input` wrapper, and the others used Chat Completions `messages`.

Budget pilots. Earlier 32,768-token pilots motivated larger caps for DeepSeek and Spark: 19/20 saved DeepSeek traces (19 on excluded BBO prompts) and 26/77 Spark traces on unchanged prompts exhausted that cap without a final answer. On the 77 matched Spark prompts, 51 were correct at 32K and 63 in the later run: 51/63 later-correct items remained correct, while 0/14 later-noncorrect items were correct at 32K. On the same 20 DeepSeek item IDs, all 19 BBO no-answers remained no-answers in the later 65,536-token run; the other item remained correct. The Spark pilot covers only five of 23 common-prompt families, and its model alias, timeout, and runner also changed. These data cannot estimate a full 380-item common-cap score or ranking.

Python-tool comparison. Luna and Sol each answered the same frozen 400 fresh v2 questions again with access to generic Python (NumPy, SciPy, SymPy, and mpmath). The question text and final-answer contract were unchanged; the wrapper permitted Python in place of the closed-book instruction. Each item had an isolated Python process without network, benchmark files, or answer keys, at most 16 tool calls with a 45-second per-call timeout, high reasoning, and a cumulative requested 32,768-token completion cap. Saved tool calls and final answers were paired by item with the closed-book traces and replayed through the same structured scorer.

Measures. Strict item accuracy requires every typed field. We also report correct fields, category choices and numerical-estimate error for prediction, and inferred model and selected controls for

Table 1: Saved evaluation protocols. Dates are UTC in 2026; all runs request high reasoning, and DeepSeek also enabled thinking. Gemini returned `google/gemini-3.8-flash` on 49 traces and the ID shown otherwise. The Python-tool cap is cumulative across steps; other caps are per request. The 20 corrected BBO prompts are saved but excluded from the screened score. Exact completion timestamps are in `protocol.csv`.

Cohort	Requested ID	Returned ID	UTC dates	Wrapper	Output cap	Saved/scored
Screened	`gpt-6-luna`	`gpt-6-luna`	Sep 25	`messages`	32,768	400/380
Screened	`gpt-6-sol`	`gpt-6-sol`	Sep 25	`messages`	32,768	400/380
Screened	`go-ds-flash`	`deepseek-v4.1-flash`	Sep 25	`messages`	65,536	400/380
Screened	`gemini-main`	`gemini-3.8-flash`	Sep 25	`messages`	32,768	400/380
Screened	`muse-spark-main`	`muse-spark-1.3-contributor`	Sep 25–26	`input`	131,072	400/380
Fresh	`gpt-6-luna`	`gpt-6-luna`	Sep 25	`messages`	32,768	400/400
Fresh	`gpt-6-sol`	`gpt-6-sol`	Sep 25	`messages`	32,768	400/400
Fresh+Python	`gpt-6-luna`	`gpt-6-luna`	Sep 26	`messages`	32,768 total	400/400
Fresh+Python	`gpt-6-sol`	`gpt-6-sol`	Sep 26	`messages`	32,768 total	400/400

coupled design. All items remain in accuracy denominators; missing or unparsed final answers get no decision credit and no numerical-error value. Thus a strict pass is a complete answer verified against the stated numerical oracle and answer contract; the score does not incorporate later judgments about prompt ambiguity or physically equivalent answers outside that contract. Known-format scoring additionally accepts complete bare JSON on the 38 conflicting prompts and otherwise uses the same scorer.

Replay. The anonymous supplementary archive (`ultrafastbench-anonymous-v2.tar.gz`) contains versioned datasets, saved requests and answers, both Python-tool runs, the saved 32,768-token pilots, scorer and generator source, pinned generation dependencies, and `SHA256SUMS`. The included generator constructs independently seeded 400-item cohorts across the same 24 families, applies the documented v2/v3 wording and answer-contract corrections, and records executable reference checks, candidate rejections, and provenance. This supports reproducible evaluation on new parameter worlds. Its `protocol.csv` and `pilot_protocol.csv` give the complete and pilot settings. From the extracted root, `sh paper/replay-docker.sh` verifies hashes and regenerates scores, tables, and figures; the companion `.tar.gz.sha256` sidecar records the archive checksum (SHA-256 `5d93edca9aa5e71453670a8ccb768dd573c238de95f82f00ded86ecf50ce8487`).

4 RESULTS

Fresh parameter worlds and Python access. On the same 400 fresh v2 parameter worlds, Luna and Sol answer 115 and 303 items completely without tools, rising to 339 and 399 with Python (Table 2). A complete answer has every required field correct; all 400 items remain in the denominator.

Table 2: Paired strict accuracy on 400 frozen fresh v2 items per model. Rescue is a closed-book miss answered completely with Python; regression is the reverse.

Model	Closed-book	With Python	Rescued/missed	Regressions
GPT-6 Luna	115/400	339/400	233/285	9
GPT-6 Sol	303/400	399/400	97/97	1

Closed-book Luna has 276 parsed mismatches, 9 unparsed responses, and 1,158/2,190 correct fields; Sol has 70 mismatches, 27 unparsed responses, and 1,922/2,190 correct fields. Table 2 shows that Python access raises Luna's score by 56.0 percentage points and Sol's by 24.0 points; every rescue used the tool. Sol's 399/400 leaves one item of headroom under this tool-enabled protocol. Closed-book Sol is at least 80% correct in 17/24 families and perfect in ten; five families account for 50 of its 70 parsed mismatches, identified after inspecting the results. On the four choice-plus-estimate families, Sol's category choices remain 80/80 while complete answers rise from 49/80 to 80/80, consistent with a substantial numerical burden. Luna's category choices rise from 56/80 to 78/80 and complete answers from 21/80 to 77/80. The net gain of 22 category choices shows that its improvement is not limited to numerical estimates. Of Luna's 61 remaining tool misses, 31 are in two eigenpulse families and have an incorrect required-gain field. In an exploratory post hoc check,

21 of these gains match using post-gate rather than the specified pre-gate variance. The saved traces alone cannot establish whether this is a convention mismatch, model error, or equation error.

The traces show two ways Python helps. On fresh-v2 item 387, Luna chose the correct two-level pump–probe category A but guessed $\delta p = -0.45$, outside the reference -0.35749 ± 0.015. With Python it integrated the stated Bloch equations for pump-plus-probe and pump-only using SciPy's `solve_ivp`, then returned -0.3575. On item 034, it enumerated all 216 two-zone gain settings and corrected its feasible count and worst-case output. Tool-enabled runs can also introduce errors: in item 039, Sol's code compared sums across gain worlds instead of the required pulse-output ratio within each world, reporting 101 feasible settings rather than 27 while retaining the correct optimum. One Luna regression used = rather than : in four final fields and failed parsing; its other eight regressions have scored field mismatches.

Prediction and design decisions. Table 3 separates physical category decisions from numerical estimates on four prediction families. Sol chooses all 80 categories but completes only 49 category-and-estimate answers. The four estimates are dimensionless fractions with different physical definitions and tolerances, so a pooled error would obscure family differences. The categories are imbalanced: a post hoc family-majority choice gets 58/80, so uniform guessing is not a suitable comparator or a trained non-LLM baseline.

Table 3: Fresh prediction results (20 items per family and model). Category counts correct choices; complete requires both choice and estimate. Absolute estimate errors are computed over parsed answers; all estimates are dimensionless fractions.

Family	Model	Category	Complete	Median error	P90 error
CEP waveform peak	Luna	17/20	14/20	.004	.281
CEP waveform peak	Sol	20/20	19/20	.0004	.008
f-to-2f phase interference	Luna	8/20	2/20	.282	.712
f-to-2f phase interference	Sol	20/20	11/20	.020	.082
Three-level absorption	Luna	17/20	4/20	.080	.247
Three-level absorption	Sol	20/20	11/20	.016	.093
Two-level pump–probe	Luna	14/20	1/20	.098	.345
Two-level pump–probe	Sol	20/20	8/20	.017	.154
Total	Luna	56/80	21/80	—	—
Total	Sol	80/80	49/80	—	—

The two coupled design families show why a correct control selection and a complete answer need separate counts (Table 4). In two-stage SPM, the model must infer the second Kerr coefficient from two width measurements, then choose a dispersion pair among 30 candidates under two robustness worlds. Sol identifies the model on 11/20 items but selects both controls on 6/20. In two-zone gain, Sol selects all settings on 17/20 items yet completes only 7/20 because an auxiliary feasible-count field is often wrong. These are final-answer outcomes; they do not identify whether a miss came from optical reasoning or unaided calculation. Figure 2 sketches one scored two-stage SPM item.

Table 4: Fresh design decisions (20 items per family and model). Each column counts items meeting all named fields; complete also requires auxiliary fields.

Family	Model	ID correct	Controls	ID+controls	Complete
Two-stage SPM	Luna	9	3	1	0
Two-stage SPM	Sol	11	6	6	3
Two-zone gain	Luna	20	3	3	0
Two-zone gain	Sol	17	17	17	7

Example: diagnose, then control two Kerr stages. A Gaussian pulse acquires nonlinear phase in stage 1, traverses dispersion D_1, then acquires phase in stage 2 with unknown coefficient B_2 before dispersion D_2:

$$E_1 = \mathcal{F}^{-1}\left[e^{iD_1 q^2/2}\mathcal{F}\left(E_0 e^{1.5i|E_0|^2}\right)\right], \qquad E_{\text{out}} = \mathcal{F}^{-1}\left[e^{iD_2 q^2/2}\mathcal{F}\left(E_1 e^{iB_2|E_1|^2}\right)\right].$$

Infer: identify B_2 from measured spectral and temporal widths among four candidates. **Control:** choose D_1 and D_2 from a 6×5 grid, respecting a spectral-width bound in both $B_2^\star \pm 0.21$ worlds and minimizing worst-case final duration. The reference chooses control indices $(0, 3)$; Sol returned $(4, 0)$.

Figure 2: Abridged fresh-cohort item `ufb-blind-luna-v1-254-2041092605`. The scored wording, constants, and reference are in the released dataset.

Screened five-model comparison. Table 5 gives strict and known-format scores on the same 380 questions. Prior model responses influenced screened-item selection, and the runs used different effective output budgets; these are observed outcomes under specified protocols, not a general model ranking. The one-item Sol–Spark strict difference has no broader ranking implication.

Table 5: Final-answer scores on 380 unchanged screened prompts. Strict credit requires every field; known-format scoring also accepts the bare JSON requested by 38 prompts. Runs use provider-specific protocols.

Model	Strict item	Known-format item	Known-format fields
GPT-6 Luna	88/380	89/380	996/2,070
GPT-6 Sol	284/380	298/380	1,880/2,070
DeepSeek v4.1 Flash	237/380	252/380	1,571/2,070
Gemini Flash	257/380	257/380	1,698/2,070
Muse Spark 1.3	283/380	291/380	1,913/2,070

The format correction recovers 38 fully correct answers across models: one for Luna, 14 for Sol, 15 for DeepSeek, and eight for Spark. For example, a CEP response gave the correct category and a within-tolerance estimate in the requested JSON, but omitted the parser's `FINAL` wrapper. DeepSeek has 70 no-answer traces on the common subset, each exhausting its requested 65,536 completion tokens; 138 of its correct traces used more than 32,768 tokens. Spark has 94 correct traces above that threshold. Gemini has 27 no-answers and reports output above its requested cap on 198/380 traces, including 150 correct. Spark's separate full corrected 400-item run scores 288/400 strictly; its revised BBO items are outside Table 5. Figure 3 separates parsed mismatches from no-answer and unparsed responses.

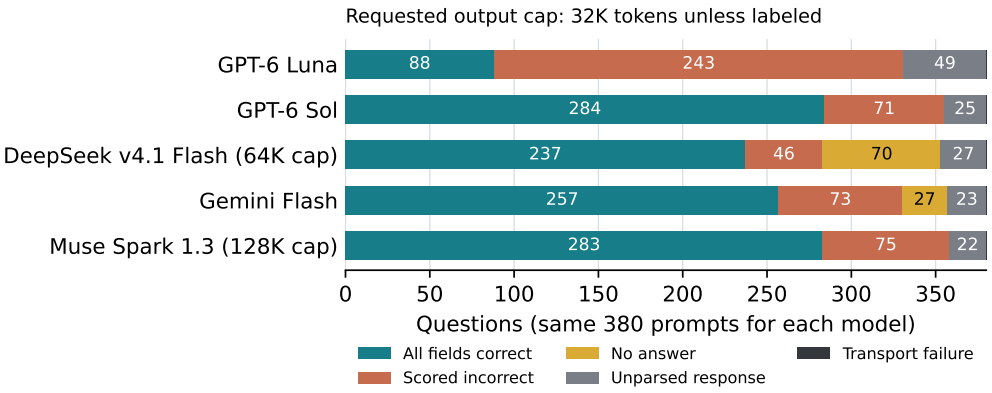


Figure 3: Strict outcomes on the 380 common screened prompts. No-answer and unparsed responses remain in each model's denominator.

Topic-level context. Sol reaches 49/50 on characterization and 48/50 on dispersion, while Luna reaches 21/50 and 20/50. Full per-topic counts for both cohorts are in the versioned result reports. These aggregates average the developed families and do not test unseen mechanisms.

5 LIMITATIONS

- **Scope.** The systems are idealized, and most families use one prompt pattern. Fresh items change parameters within developed families, not optical mechanisms. Task type is mostly confounded with family, so the results do not isolate a general difference between prediction, inference, and control. Held-out mechanisms and matched task variants on the same optical worlds, with independently verified references and contracts, are needed to test that broader claim.
- **Protocol.** Python access changes the prompt wrapper and reasoning trajectory, so its paired gain cannot be assigned uniquely to arithmetic, optics knowledge, or answer formatting. One response per item and provider-specific budgets limit cross-model comparisons.
- **Audit evidence.** The saved two-reviewer return lacks separate votes and case-specific evidence. We corrected no-answer verdicts using saved traces, but tool responses lack an independent physical audit. Scores therefore measure deterministic agreement with the declared references and answer contracts.

6 CONCLUSION

UltrafastBench supplies reproducible final-answer tasks for specified ultrafast-optics prediction, diagnosis, and constrained control. Its field-level results expose where a correct decision differs from a complete answer. On the 400 fresh v2 questions, Python access rescues 233/285 Luna and 97/97 Sol closed-book misses under a protocol that permits numerical tools. This shows protocol sensitivity in the observed deficit. The comparison does not identify a pure arithmetic fraction; Luna's remaining tool errors and the BBO and comb/CEP audits show the limits of treating replay and structured scoring as physical validation. Held-out mechanisms and matched task variants are needed to establish whether the observed differences generalize beyond parameter changes within these families.

AI USE STATEMENT

OpenAI Codex assisted with benchmark methodology and experiment design; implementation and checking of code used to construct synthetic questions, compute references, and score responses; analysis and interpretation of results; literature search and synthesis; and drafting, editing, and pre-submission review of this manuscript. The evaluated language models were study subjects: their responses were saved as experimental observations and did not set the reference answers. The authors reviewed the AI-assisted work and take full responsibility for the methods, code, data, citations, results, and text in this paper.

REFERENCES

Sonia Boscolo, Julien Fatome, Sergei K. Turitsyn, Guy Millot, and Christophe Finot. Temporal and spectral nonlinear pulse shaping methods in optical fibers. In Stefan Wabnitz and Benjamin J. Eggleton (eds.), *All-Optical Signal Processing: Data Communication and Storage Applications*, pp. 105–128. Springer, 2015. doi: 10.1007/978-3-319-14992-9_4. URL https://doi.org/10.1007/978-3-319-14992-9_4.

Ziru Chen, Shijie Chen, Yuting Ning, et al. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.05080.

Titouan Duston, Shuo Xin, Yang Sun, et al. AInsteinBench: Benchmarking coding agents on scientific repositories, 2025. URL https://arxiv.org/abs/2512.21373.

Sihan Hu, Lyuhan Huang, Youjin Deng, and Kun Chen. SciCode-Verified: How benchmark defects underestimated the scientific-coding ability of language models, 2026. URL https://arxiv.org/abs/2608.04975.

Shima Imani, Seungwhan Moon, Adel Ahmadyan, Lu Zhang, Ahmed Kirmani, and Babak Damavandi. SymPyBench: A dynamic benchmark for scientific reasoning with executable python code. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics: Industry Track*, 2026. URL https://aclanthology.org/2026.eacl-industry.8/.

Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, et al. SciCode: A research coding benchmark curated by scientists. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/36850592258c8c41cecdaa3dea5ff7de-Abstract-Datasets_and_Benchmarks_Track.html.

Andrew M. Weiner. Ultrafast optical pulse shaping: A tutorial review. *Optics Communications*, 284 (15):3669–3692, 2011. doi: 10.1016/j.optcom.2011.03.084. URL https://doi.org/10.1016/j.optcom.2011.03.084.

Wanjia Zhao, Qinwei Ma, Jingzhe Shi, Shirley Wu, Jiaqi Han, Yijia Xiao, Si-Yuan Chen, Xiao Luo, Ludwig Schmidt, and James Zou. PRISM-Physics: Causal DAG-based process evaluation for physics reasoning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/1a379b51d7ec80b32d8c14f40c281ce1-Abstract-Conference.html.