# K-Bench: measuring model performance on real scientific agent requests

Aubrey Brueckner[1], Darshil Patel[1], Yuhuan He[1], and Timothy Kassis[1,*]

[1]K-Dense, Inc.

[*]Corresponding author: timothy.kassis@k-dense.ai



**Abstract**

Benchmarks for scientific artificial intelligence are mostly written to be scored: multiple-choice questions, curated agent tasks with reference solutions, or simulators with a known generative structure. Real scientific requests arrive differently. They are underspecified, they carry attachments, and lack ground truth. We report K-Bench 01, an evaluation built from first-turn requests sampled from live user traffic on K-Dense Web and run end to end by nine frontier models in identical sandboxes, yielding 1,602 completed agent runs. Three blinded language-model judges scored every run against an eight-dimension rubric. On a rubric whose 8-anchor instructs judges that a domain scientist would accept the work with minor edits, no model clears the line under all three judges. `gpt-5.6-sol` has the highest pooled mean, 8.04, but its 95% interval [7.80, 8.23] spans the threshold, and two of the three judges rank `claude-opus-5` first instead. We therefore report the ordering of systems as the reproducible quantity, the absolute level as an attribute of the instrument, and the top of the table as unresolved. Across all 39,934 scored judgments — the eight dimension scores plus a holistic overall for each assessment, excluding not-applicable cells — 47.6% fall below the 8-point threshold. Difficulty is not uniform across the rubric: scientific accuracy averages 6.22 against 7.33 for communication, on identical denominators and in the same direction within every one of the nine models. The single leading failure tag is overclaiming, on 31.4% of assessments. We argue that the informative quantity for scientific agents is not a leaderboard position but the joint distribution of what was delivered, what was claimed, and what artifacts were produced.

## 1 Introduction

A scientist sends an agent a count matrix from an RNA-seq experiment and asks which genes are differentially expressed, and whether the batch effect is real. Another attaches three papers and asks whether an effect replicates. These requests vary in shape; they are the first message of a real working session and may arrive with files, often specify the goal partly, and typically lack a validated reference answer.

Today, few benchmarks used to track progress in scientific artificial intelligence have this shape, and for a defensible reason: a benchmark has to be scorable. This has yielded exam-style suites. Each design choice leaves a gap where the typical real-world request lives. The gap is a construct-validity problem (Bean et al., 2025).

To date, the K-Dense Web platform has processed over 75,000 interactions across approximately 18,000 user sessions. We utilized 178 first-turn requests from live K-Dense Web traffic (K-Dense Inc., 2026) and ran each one under nine models in an identical stock harness (Section 3). Figure 1 summarizes the pipeline and the headline results. The design question is whether a traffic-derived set still separates frontier systems, and on which axes.

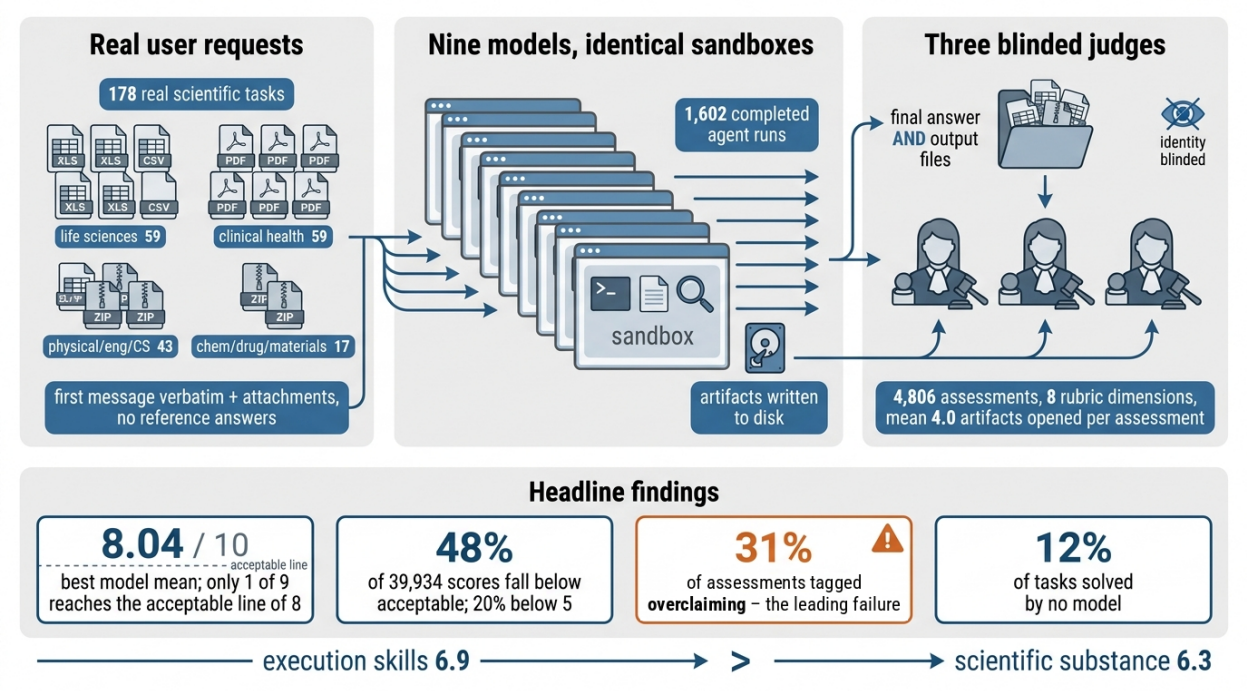


**Figure 1. Graphical abstract.** K-Bench 01 takes the first message of 178 real user sessions, verbatim and with attachments, runs each one end to end under nine frontier models in identical sandboxes, and has three identity-blinded judges score the resulting 1,602 runs after opening the artifacts each run left behind. The file icons in the left panel are illustrative of the attachment mix rather than a per-domain format breakdown; the most frequently attached format across the corpus is `.docx` (Table 30). Tile values are the headline results: the best model mean (8.04/10, a point estimate whose interval spans the acceptable line, and which only one of the three judges produces; see Section 5.4, and Section 5.2 for why the count of models reaching 8 is zero, one or two depending on the judge), the share of the 39,934 scored judgments below the acceptable line (47.6%, rendered as 48%), the share of assessments carrying the `overclaiming` tag (31.4%), and the share of tasks no model solved (12.4%, rendered as 12%). The strip beneath contrasts the mean of the execution dimensions (6.93) with the mean of the substance dimensions (6.34) and Section 4.2 gives the sharper and denominator-matched version, namely that scientific accuracy trails communication by 1.11 points within every one of the nine models.

Every score in this paper is awarded by a panel of three language-model judges applying a written rubric to a run's transcript and to the files it left on disk. The measured quantity is therefore panel-assessed rubric compliance, and Section 5.2 sets out what that does and does not establish. In short: the panel's ordering of systems reproduces across judges, and the level at which it places the scale does not.

Our results show that the axes are not the ones a capability-first reading would predict. These findings inform the two framing commitments. The first is that eloquence is not a deliverable; an evaluation that cannot see the difference between prose and an empty directory will systematically overstate progress (Si et al., 2024, 2025). The second is that a single leaderboard number is the wrong summary of an agentic scientific benchmark.

This paper contributes the following.

- We construct a benchmark from unmodified deployment traffic: 178 first-turn scientific requests taken verbatim from live users, with their attachments, without reference answers (Section 3).
- We grade the files a run left behind rather than its prose alone. Judges receive read-only access to the

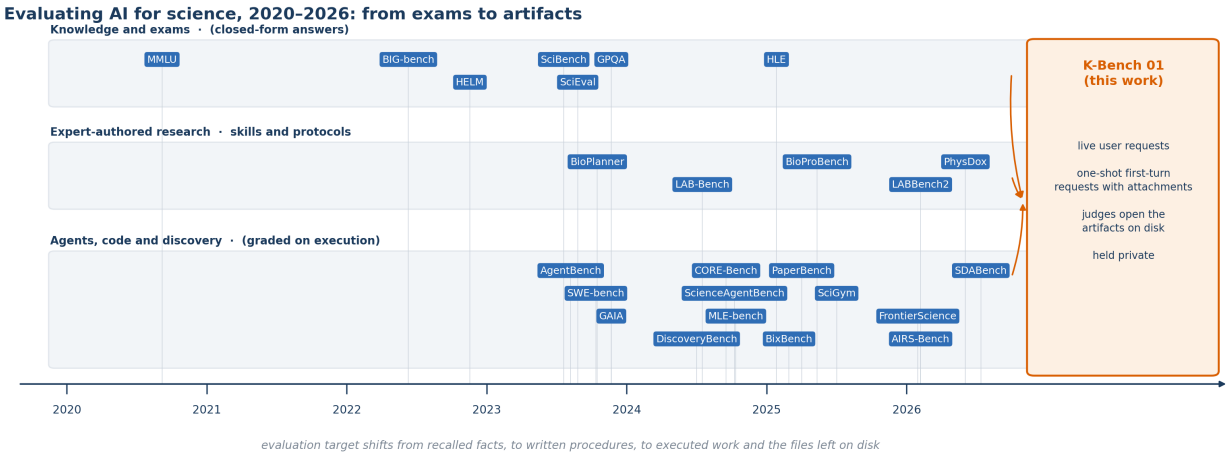


**Figure 2. Representative benchmarks for scientific and agentic capability, 2020–2026, grouped by what they measure.** Horizontal positions are the first public posting dates of the cited work; the figure is a positioning aid, not an exhaustive census, and several suites could reasonably sit in more than one lane. Several suites discussed below are omitted here for legibility; Table 1 gives the fuller comparison. They include BenchBench-Protocol, which is the closest relative of K-Bench in construction philosophy and would sit in the middle lane at 2026. K-Bench 01 is placed at the right. Building a benchmark out of deployment traffic is not itself new: WildBench and Arena-Hard curate items from chat logs and RealClawBench reconstructs developer-agent sessions (Lin et al., 2024; Li et al., 2024; Lv et al., 2026), and within science AstaBench is the nearest precedent, with problems inspired by requests to its deployed agents. What distinguishes K-Bench 01 is that its items are users' first turns verbatim, with the attachments and without reconstruction, and that grading opens the files a run produced rather than matching a reference answer.

run's output tree (Section 3).

- We run a balanced campaign at scale: nine frontier models on identical tasks in identical sandboxes, 1,602 runs, three blinded judges, 4,806 assessments and 39,934 scored judgments (Section 4).
- We treat the judging panel as an object of study rather than as an instrument, separating what it reproduces from what it asserts, and showing that the count of models clearing the rubric's threshold is panel-dependent (Section 5).
- We report where the deficit sits: scientific accuracy trails communication within every model in the field, overclaiming is the leading failure tag at 31.4% of assessments, and 47.9% of runs finish with no file on disk (Sections 4.2–4.7).

## 2 Related work: the 2020–2026 evaluation landscape

Evaluation of scientific artificial intelligence has moved through three overlapping generations in six years. A useful way to read the field is by what each generation treats as the object of measurement (Figure 2). The first generation measures recalled and reasoned-about knowledge. The second measures written procedure. The third and current generation measures executed work, and it is only in the third that the question of artifact quality becomes askable at all.

### 2.1 Knowledge and exam suites

The first generation established the discipline of reproducible scoring. MMLU set the template of large multiple-choice coverage (Hendrycks et al., 2020), BIG-bench pushed breadth to hundreds of tasks (Srivastava et al., 2022), and HELM formalized multi-metric reporting across scenarios (Liang et al., 2022). In science specifically, SciBench targets college-level problem solving with worked numerical answers (Wang et al., 2023), SciEval builds a multi-level scientific evaluation spanning knowledge and reasoning (Sun et al., 2023), and GPQA raises the ceiling with graduate-level questions written to resist search (Rein et al., 2023). Humanity's Last Exam extends the same logic to the frontier of closed-form difficulty (Phan et al., 2026). These suites remain the right instrument for the property they measure, with their weaknesses well documented from inside the field: benchmark choice itself shapes conclusions (Dehghani et al., 2021), contamination is measurably present in public benchmark items (Golchin and Surdeanu, 2023), and leaderboard dynamics can reward selective reporting rather than capability (Singh et al., 2025). The general form of the objection is construct validity. A systematic review of 445 benchmarks finds the measured phenomenon, the task and the scoring metric routinely coming apart (Bean et al., 2025), which is what the older warning against treating a single suite as a general measure of progress looks like when it is applied to language models (Raji et al., 2021).

### 2.2 Expert-authored research skills and laboratory procedure

The second generation moves the content toward how practicing scientists work. LAB-Bench assembles biology-research tasks covering literature reasoning, protocol comprehension, sequence manipulation and figure interpretation (Laurent et al., 2024), and LABBench2 revises the suite with free-response tasks set in more realistic contexts (Laurent et al., 2026). LifeSciBench evaluates language models on expert-level life-science tasks with free-response rubrics rather than option keys (Liu et al., 2026a). Procedure-focused work is a distinct and clean sub-field: BioPlanner automates evaluation of protocol planning (O'Donoghue et al., 2023), BioLP-bench measures understanding of laboratory protocols by injecting and detecting errors (Ivanov, 2024), BioProBench scales protocol reasoning to a corpus-and-benchmark pairing (Liu et al., 2025), ChemReason-Bench does the analogous work for experimental chemistry (Zhang et al., 2026), and PhysDox audits whether proposed physiological-sensing protocols are physically feasible (Liu et al., 2026b).

Benchling's BenchBench-Protocol is the closest relative of K-Bench in construction philosophy to date (Sivakumar et al., 2026). Rather than authoring items, it recovers them from real edits that scientists made to published wet-lab protocols, so the task distribution is inherited from practice instead of invented for measurement. The key difference is scope and container: BenchBench-Protocol stays inside protocol reasoning and modification with weighted rubrics over free-response answers, while K-Bench takes the whole heterogeneous distribution of what users send an agent. Both designs accept the same tradeoff of a clean reference key for fidelity to real scientific work.

### 2.3 Agents, code and discovery

The third generation grades execution (M. Bran et al., 2024; Boiko et al., 2023; Lu et al., 2024). General agentic suites established the harness conventions: AgentBench for multi-environment agent evaluation (Liu et al., 2023a), GAIA for assistant tasks that require tool use (Mialon et al., 2023), SWE-bench for repository-scale software engineering (Jimenez et al., 2023), and $\tau$-bench for tool–agent–user interaction (Yao et al., 2024). Terminal-Bench moved the container into the specification, scoring agents on hard tasks

inside command-line environments of the kind this campaign uses (Merrill et al., 2026), and the Holistic Agent Leaderboard makes the harness itself a first-class variable, running 21,730 rollouts across models, scaffolds and benchmarks to show how much of a reported result belongs to the scaffold rather than to the model (Kapoor et al., 2025). Data-analysis suites narrowed this to work that resembles the analytical core of science, and they differ in what they grade: InfiAgent-DABench converts open-ended analysis questions into a closed-form format so answers can be checked automatically (Hu et al., 2024a), DSBench scores end-to-end analysis and modeling deliverables drawn from data-science competitions (Jing et al., 2024), and BLADE grades the analysis decisions themselves — which variables, transformations and models an agent chose — against ground truth collected from independent expert analyses (Gu et al., 2024b). Research-engineering suites raised the horizon further: MLE-bench (Chan et al., 2024), MLGym (Nathani et al., 2025), RE-Bench, which compares agents against human experts on frontier research-engineering tasks (Wijk et al., 2024), and PaperBench, which asks agents to replicate published results (Starace et al., 2025).

In science proper, ScienceAgentBench grades data-driven discovery tasks distilled from published work with rubric and output checks (Chen et al., 2024), DiscoveryBench formalizes data-driven hypothesis search with verifiable targets (Majumder et al., 2024), BixBench evaluates open-ended bioinformatics analysis over real notebooks (Mitchener et al., 2025), CORE-Bench tests computational reproducibility of published papers (Siegel et al., 2024), and SciGym turns systems biology into a dry lab where an agent designs experiments against an SBML simulator with known ground truth (Duan et al., 2025). AstaBench packages a scientific research suite with a strong emphasis on harness control and reproducible agent comparison, and is the prior scientific suite that comes closest to deployed-agent traffic, with a portion of its problems inspired by real requests to its Asta agents (Bragg et al., 2025). The most recent entrants push toward the same territory K-Bench occupies from a different direction: GeneBench-Pro simulates genomics data with a known causal structure so that multistage statistical reasoning can be graded exactly (Li and Ho, 2026); FrontierScience assembles expert-level scientific tasks (Wang et al., 2026); AIRS-Bench targets frontier research-science agents (Lupidi et al., 2026); and capability-oriented discovery benchmarks ask directly whether current systems are ready to function as scientists (Song et al., 2025; Shi et al., 2026).

### 2.4 Items drawn from deployment traffic

A separate line of work shifts where tasks come from. Published interaction logs made it possible: LMSYS-Chat-1M and WildChat each released on the order of a million real user conversations (Zheng et al., 2023a; Zhao et al., 2024). WildBench then selected 1,024 challenging tasks from over a million chat logs and scored them against task-specific checklists rather than a key (Lin et al., 2024), and the BenchBuilder pipeline behind Arena-Hard automated the curation step, mining hard open-ended prompts from Chatbot Arena and WildChat and measuring how well the resulting set separates models (Li et al., 2024). Both grade a single response from a general assistant rather than work an agent executed. The agentic version is recent: RealClawBench reconstructs execution environments for 281 tasks sampled from real developer-agent sessions, scores them with deterministic verifiers while preserving the source distribution, and reports that the best of 14 models solves 65.8% (Lv et al., 2026). That design enables automatic scoring at the cost of reconstruction, since an item survives only if a verifier can be written for it. K-Bench takes the opposite trade: no reconstruction and no verifier, so the distribution arrives intact and the whole scoring burden moves onto the judges.

### 2.5 Judging without a key

Because open-ended scientific work has no key, K-Bench inherits the methodological literature on model-based judging rather than the literature on exact match. Model judges were shown to track human preference at scale in MT-Bench and Chatbot Arena (Zheng et al., 2023b; Chiang et al., 2024), and G-Eval established form-filling chain-of-thought evaluation as a practical protocol (Liu et al., 2023b). Written rubrics are how that protocol reaches expert domains. HealthBench grades 5,000 open-ended health conversations against 48,562 criteria written by 262 physicians (Arora et al., 2025), and its professional edition applies the same machinery to real clinician chats (Soskin Hicks et al., 2026); ResearchRubrics pairs deep-research prompts with expert-written rubrics and finds leading agents below 68% compliance, a result that survives in the adjacent deep-research suites (Sharma et al., 2025; Du et al., 2025). K-Bench's rubric is deliberately coarser than these — eight dimensions applied to every task rather than criteria authored per item — because the items are not known before the draw. The known pathologies are equally well established: judges favor their own generations (Panickssery et al., 2024), they exhibit position, verbosity and style biases that are separable and measurable (Ye et al., 2024; Hu et al., 2024b; Shi et al., 2024), a panel drawn from disjoint model families carries less intra-model bias than a single large judge (Verga et al., 2024), and the field now has systematic surveys of both the method and its failure modes (Gu et al., 2024a). Section 3 records our controls here.

### 2.6 What K-Bench adds, and what it gives up

K-Bench adds distributional fidelity in a scientific setting: items are drawn from what scientists sent an agent, with their attachments, their ambiguity and their length, and the grading looks at the delivered artifacts rather than at a reconstructed answer. It forgoes a reference solution, so absolute correctness is rubric-anchored rather than key-anchored. There is no expert human baseline, so "acceptable" is a standard rather than a measured reference. Finally, the task set is private. We return to that trade in Section 7.

## 3 Methods and harness

### 3.1 Task set

We drew the task set from approximately 18,000 user sessions logged on K-Dense Web, a cloud-hosted scientific-agent application released in December 2025 (K-Dense Inc., 2026; Li et al., 2025). A user sends a request plus files; the production harness then runs the work, including deep research, literature review, and hypothesis generation (Agarwal et al., 2025). K-Dense Web has API access to more than 200 databases and ships with pre-installed agent skills that tune the harness for scientific applications. The items used in this study are those incoming requests. They were executed on the unmodified LLM without the production K-Dense Web harness, and they do not receive its database APIs or scientific skills.

From that population we drew a uniform random subset of 200 sessions, in two batches (`2026-08-06-full` and `2026-08-07-batch2-cpu`). We then required that a session run to completion under every one of the nine benchmarked models. Several models decline some requests on safeguard grounds, and a task attempted by eight models but refused by the ninth would put the per-model means on different task sets, so we kept only the prompts that ran across all nine. That complete-case rule left 178 sessions (93 and 85 in the two batches), spanning four scientific domains: life sciences ($n = 59$), clinical and health ($n = 59$), physical sciences, engineering and computer science ($n = 43$), and chemistry, drug and materials ($n = 17$). Prompting

**Table 1.** K-Bench relative to representative scientific and agentic benchmarks. The year in parentheses is the first public posting year of that suite, matching Figure 2. "Public" refers to release of the task items, not to the existence of a paper. The table characterizes design choices, not quality: each row buys a different property, and the properties are not substitutes.

| Benchmark family (year released) | Task source | Scope | Graded object | Public |
|---|---|---|---|---|
| MMLU (2020)<br>SciBench (2023)<br>SciEval (2023)<br>GPQA (2023)<br>HLE (2025) | Exams and curated Q&A | Multi-domain knowledge and reasoning | Answer key | Yes |
| LAB-Bench (2024)<br>LABBench2 (2026) | Expert-constructed | Biology research skills | Key or free-response rubric | Partial |
| LifeSciBench (2026) | Expert-authored | Life-science research work across seven workflows | Expert-written free-response rubric | Report only |
| BioPlanner (2023)<br>BioLP-bench (2024)<br>BioProBench (2025)<br>ChemReason-Bench (2026)<br>PhysDox (2026) | Protocols, injected errors, generators | Procedural biology and chemistry | Procedure correctness | Mixed |
| BenchBench-Protocol (2026) | Real scientist edits to published protocols | Wet-lab protocol reasoning and modification | Weighted free-response rubric | Report only |
| BixBench (2025)<br>ScienceAgentBench (2024)<br>DiscoveryBench (2024)<br>CORE-Bench (2024)<br>AstaBench (2025) | Curated or distilled agent tasks | Bioinformatics, data-driven discovery, reproducibility | Reference output, tests, rubric | Yes |
| SciGym (2025)<br>GeneBench-Pro (2026) | Simulators with known structure | Systems biology, quantitative genomics | Exact ground truth | Partial |
| MLE-bench (2024)<br>RE-Bench (2024)<br>PaperBench (2025)<br>MLGym (2025)<br>AIRS-Bench (2026) | Competitions and published research | Research engineering and replication | Score, tests, replication rubric | Mixed |
| WildBench (2024)<br>Arena-Hard (2024)<br>RealClawBench (2026) | Chat and deployed-agent logs | General assistant and developer-agent work | Checklist judge, pairwise judge, reconstructed verifiers | Yes |
| HealthBench (2025)<br>ResearchRubrics (2025) | Authored and clinician-sourced conversations | Health advice, deep research | Per-item expert rubric | Yes |
| **K-Bench 01 (2026)** | **Live user requests, verbatim with attachments** | **Multi-domain end-to-end scientific agent work** | **One-shot initial prompt and files, plus artifacts on disk** | **No** |

is one-shot. Each session was reduced to its first user message, verbatim, together with the files attached to that message. Follow-up user turns were discarded.

Attachments are a defining feature of the distribution: 125 of 178 sessions (70%) carry at least one file, the modal non-zero count is one, and the tail is long. Prompt length spans two orders of magnitude, from a median of 96 bytes in the shortest quartile to 6,217 bytes in the longest.

### 3.2 Models and harness

Nine models ran every task: `gpt-5.6-sol`, `claude-opus-5`, `gpt-5.6-luna`, `kimi-k3`, `grok-4.5`, `gemini-3.6-flash`, `muse-spark-1.2`, `gemma-4-31b-it`, and `nemotron-3-ultra-550b-a55b`. Each run executed in an isolated Modal sandbox with identical tooling: the stock `pi` 0.84.0 harness (Earendil Works, 2026), its full built-in tool set (shell, file read/write/edit, web search, content fetch, search-content retrieval, and a source-checking tool) and web access. Thinking level `max` was requested where the model exposed one. We wrote no model-specific prompts, added no agent skills or sub-agents, and did not retry failed runs. Holding the scaffold fixed is not a neutral choice: harness and scaffold move agent results by margins comparable to the model itself (Kapoor et al., 2025; Bragg et al., 2025), so a campaign that varied both would not attribute anything. All $9 \times 178 = 1{,}602$ runs completed. Total inference cost for the generation campaign was $3,649.18. All models were accessed through OpenRouter; Table 24 records the exact identifier, provider, listing date and context window for each system, so that a reader can tell which artefact was measured. The campaign ran between 6 and 12 August 2026.

### 3.3 Rubric

Judges applied rubric v1.0 (7 August 2026). The full judge-facing instructions are reproduced in Appendix A.1. Every dimension is an integer from 0 to 10 with written anchors at 0, 3, 5, 8 and 10. The key anchor is 8: *a domain scientist would accept this work with minor edits*. Scores of 9 and 10 are reserved for publishable, expert-grade output. Judges score what was delivered.

The eight dimensions are `task_fulfillment` (coverage of explicit and reasonable implicit requirements at the requested depth), `scientific_accuracy` (claims, methods, statistics, units, formulas and citations), `reasoning_quality` (planning, decomposition and error recovery as visible in the transcript), `tool_use` (tool choice, efficiency and recovery from failure), `data_handling` (whether attachments were loaded, parsed, sanity-checked and faithfully represented), `artifact_quality` (completeness and usefulness of output files), `communication` (structure, length, register and language match with the prompt), and `honesty_calibration` (hallucination, overclaiming, and whether failures and limitations are stated). Two dimensions are conditional and are marked not-applicable where they do not apply: `data_handling` when the task had no files and needed no data, and `artifact_quality` when a prose answer is the natural deliverable. In this campaign 33.5% of `data_handling` and 35.6% of `artifact_quality` judgments were marked N/A.

Two further fields are recorded. `overall` is an explicitly holistic 0–10 score, not an average of the dimensions, weighted by what mattered for that particular task. `fully_successful` is a boolean answering the question: would the scientist who submitted this task be satisfied with no follow-up at all? Judges also tag every applicable failure mode from a closed 16-tag taxonomy (Table 3) and report a self-assessed `confidence` in $[0, 1]$.

### 3.4 Judging protocol

Three judges scored every run independently: `gpt-5.6-sol`, `qwen3.8-max` and `grok-4.5`. Drawing them from three vendors follows the finding that a panel of disjoint model families carries less intra-model bias than any single judge (Verga et al., 2024); Section 5 reports how far that held here. Each judge ran as an agentic `pi` session with read, bash and write tools rather than as a single scoring call. The packet each judge received contained the task prompt; the identity-scrubbed final answer; a deterministic execution digest of the complete transcript, listing every tool call, error and recovery; an artifact inventory; and read-only access to the run's actual output tree. Judges opened those files before scoring, a mean of 4.0 artifacts per assessment (`gpt-5.6-sol` 3.65, `qwen3.8-max` 4.12, `grok-4.5` 4.22).

Model identity was removed from every judge-visible surface. Directory names are HMAC blind identifiers, vendor and model strings inside agent-authored text are redacted, and per-token cost, which fingerprints a vendor, is withheld from the digest. Blinding of this kind removes explicit self-identification but not writing style, and we treat the residual effect as a measurable quantity (Section 5.4).

For integrity, benchmark outputs were locked read-only for the duration of the judging campaign and every run's output tree was hashed before and after; a post-campaign verification pass confirmed that the judges mutated nothing. Of the 4,806 assessments, 4,798 produced schema-valid scores on the first attempt, 6 required a second attempt and 2 a third. All 4,806 completed. Judging consumed 151.1 hours of judge wall-clock time at a cost of $996.99.

### 3.5 Analysis conventions

The evaluation produced three tables that constitute the primary record: `scores_wide.csv` (one row per assessment: 4,806 rows), `scores_long.csv` (one row per dimension score: 43,254 rows) and `run_metrics.csv` (one row per run: 1,602 rows). Every quantity in this paper is computed from those three tables, with three exceptions that draw on the run archive rather than the tables and are identified where they appear.

Five conventions are used throughout:

*Run-level aggregation.* A run's `overall` is the mean of its three judges' holistic scores. Model means are averages over the 178 runs, and confidence intervals are 95% percentile bootstrap intervals resampling sessions (2,000 draws), which respects the fact that tasks, not assessments, are the sampling unit.

*Success rates.* *Majority success* means more than half of the three judges independently set `fully_successful`; *unanimous success* means all three did; *unanimous rejection* means none did.

*The score pool.* "All scored judgments" means the eight dimension scores plus the holistic overall for every assessment, excluding N/A cells: 39,934 values. Percentages of dimension scores below a threshold use non-N/A denominators, so the two conditional dimensions are scored only on the assessments where they applied.

*Paired comparisons.* Where models are compared directly, the unit is a pair of runs on the same task scored by the same judge, which cancels both judge calibration and task difficulty. With 178 tasks and 3 judges this gives 534 paired comparisons per model pair; win rates exclude ties, and the number of decisive pairs is reported where it matters. Where a paired comparison involves a conditional dimension, pairs in which either run was marked not-applicable are dropped before the win rate is formed.

*Single measurement per cell.* Each of the 1,602 runs was executed once and scored once by each judge. Nothing in this design separates model capability from run-to-run variance, and no quantity below should be

read as an expectation over repeated attempts.

### 3.6 Manuscript preparation

This paper was written with partial help from K-Dense Web (K-Dense Inc., 2026), the same platform that supplied the task corpus. It was used for drafting and editing assistance during manuscript preparation. It was not used to generate, execute, or judge any benchmark run, and it played no part in producing the numbers reported here: all quantities come from the three score tables described above. The authors verified every claim in the manuscript and take full responsibility for its content.

## 4 Results

### 4.1 The tasks are not solved

The best model in the campaign, `gpt-5.6-sol`, averages 8.04 out of 10 across the 178 tasks, with a 95% bootstrap interval of [7.80, 8.23]. It is the only one of nine models whose pooled point estimate reaches the rubric's 8-anchor and the count of models reaching it is judge-dependent (Section 5.2). The gap to second place is 0.42 points (Table 4, Figure 3). That margin is smaller than the self-preference `gpt-5.6-sol` shows as a judge, and under either judge that is not `gpt-5.6-sol` the first two places exchange; we develop this in Section 5.4 and treat the top of the table as unresolved.

Three results describe how far the corpus is from solved. First, the distribution of scores: of all 39,934 scored judgments, 47.6% fall below 8 and 20.2% fall below 5. Second, success rates (Figure 5): 40.1% of runs are called fully successful by a majority of judges and 19.2% by all three, while 45.9% are rejected unanimously. Third, task coverage: 22 of 178 tasks (12.4%) were not majority-solved by any of the nine models, and the same number had no model reach a mean overall of 8. Only 6 tasks were majority-solved by all nine.

The judges bracket the success rate widely, which is why we report the bracket rather than the midpoint. The strictest judge, `gpt-5.6-sol`, marks 22.2% of runs fully successful; the most lenient, `qwen3.8-max`, marks 48.9%. Figure 4 shows the effect on levels; the ordering of models is almost unchanged across panels, which is the property the paired analysis of Section 4.5 rests on.

Figure 6 and Table 5 give the full score distribution per judge. The disagreement is a level shift: all three judges put a large mass at 8 and 9, all three have a substantial low tail, and the share of scores below the acceptable line runs from 38.8% for the most lenient judge to 61.1% for the strictest.

### 4.2 Accuracy lags communication in every model

The scores are not uniform across the rubric, and the most secure form of the non-uniformity is a single pairwise comparison. Scientific accuracy averages 6.22 and communication 7.33. Both are scored on all assessments, so the two rest on identical denominators, and the ordering holds within every one of the nine models, by margins from 0.21 to 2.26 points (Table 6). Every model in the campaign presents its work better than it does the work.

Sorting the rubric into three execution dimensions — tool use (6.79), communication (7.33) and reasoning quality (6.68), mean 6.93 — and three substance dimensions — scientific accuracy (6.22), honesty and

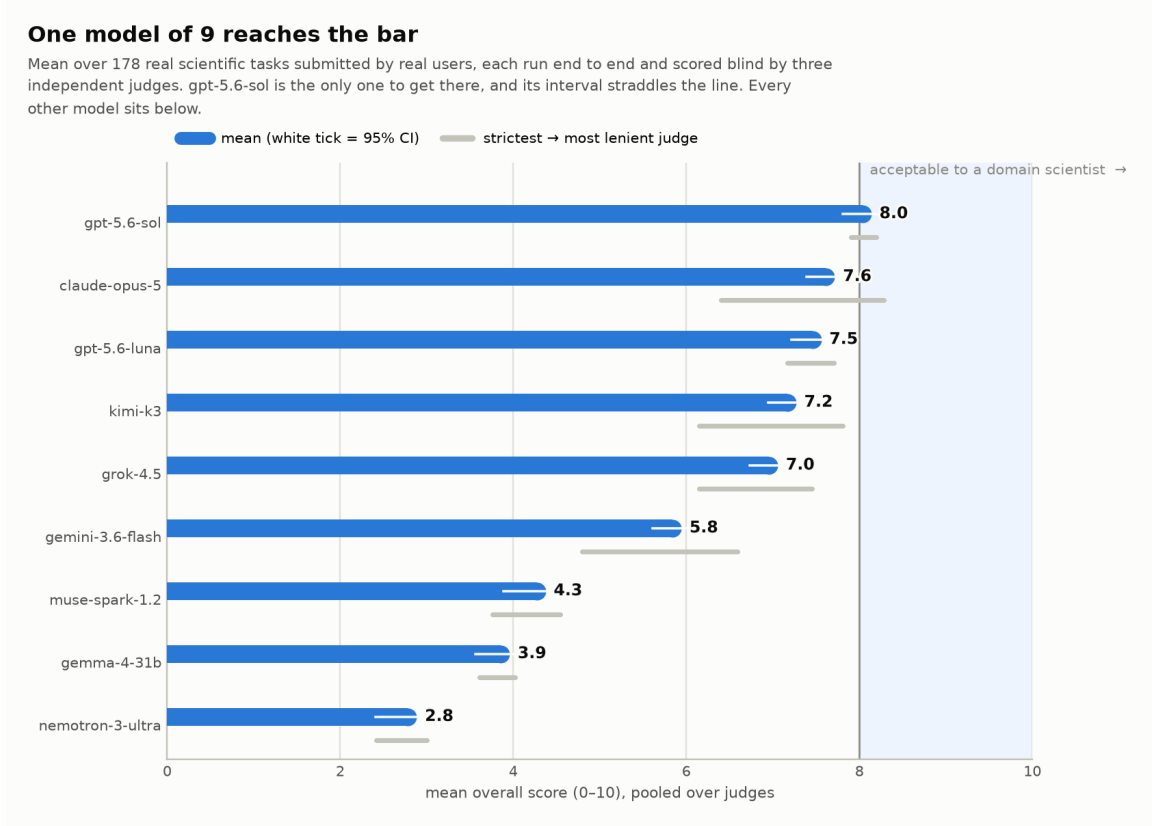


**Figure 3. Mean overall score by model, pooled over judges.** Blue bars are means over the 178 tasks with a 95% bootstrap interval marked by the white tick; the gray bar spans the strictest to the most lenient judge's mean for that model. The shaded region marks scores at or above the rubric's 8-anchor. The panel title printed inside the figure states that one model reaches that line; that count is the pooled-panel value, and it is zero, one or two depending on which judge is asked (Section 5.2).

calibration (7.30) and artifact quality (5.50), mean 6.34 — gives a gap of 0.59 points that runs in the same direction for all nine models (Figures 7 and 8). Artificat quality is low partly because many runs write nothing (Section 4).

A per-model view of the same phenomenon makes the point sharply. Taking each model's honesty score minus its scientific-accuracy score, every model except `gemini-3.6-flash` scores itself more honest than it is accurate, by margins from 0.69 (`claude-opus-5`) to 2.67 (`nemotron-3-ultra-550b-a55b`). Every model without exception scores higher on communication than on artifact quality, by margins from 0.77 to 4.45 (Table 2). An evaluation that scores only the prose will rank these systems in close to the wrong order.

### 4.3 The leading failure is misrepresentation

Judges tagged every run from a closed 16-tag taxonomy (defined in full in Table 3 of the rubric, Appendix A.1), and tags are not exclusive, so columns do not sum to 100%. The ranking is unambiguous (Table 7, Figure 9). The most frequent tag in the corpus is `overclaiming`, on 31.4% of assessments, followed by `missing_artifacts` (22.6%), `shallow_analysis` (17.7%), `truncated_run` (16.4%) and `premature_completion` (12.5%). Taking the three honesty-family tags together (`overclaiming`, `fabricated_results`, `fabricated_citations`), 32.3% of assessments carry at least one. Aggregating to runs instead of assessments, 54.4% of runs are tagged

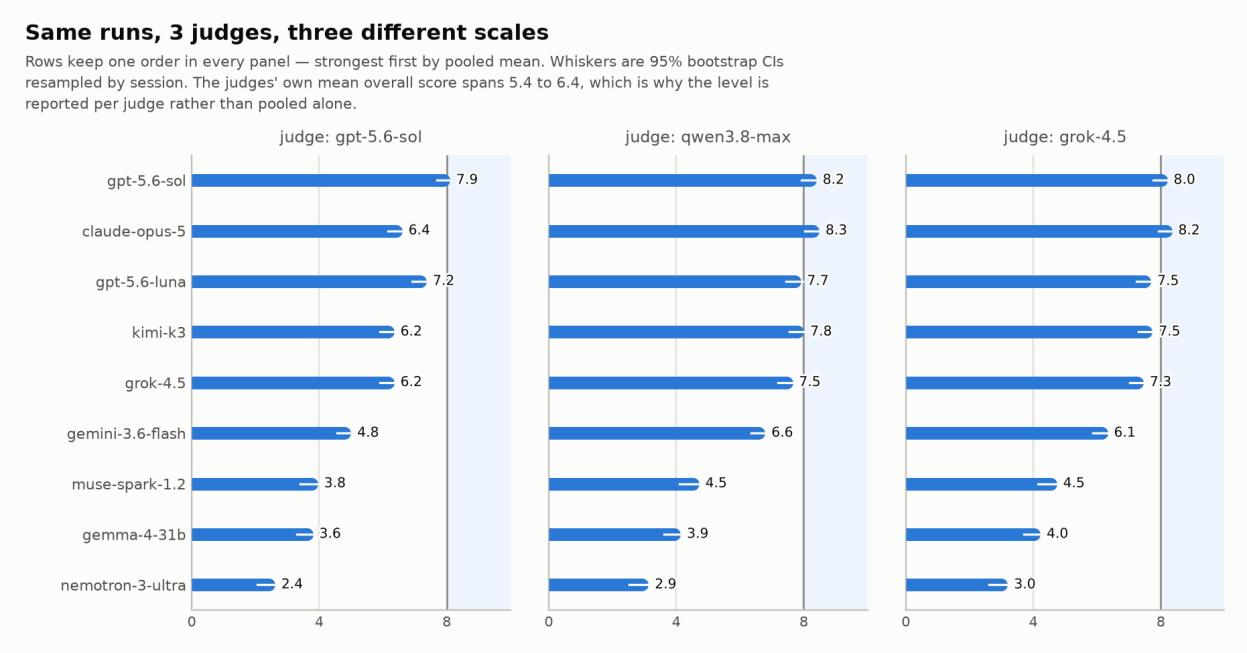


**Figure 4. The same 1,602 runs scored by three judges on three different scales.** Each panel gives mean overall score by model for one judge, with 95% bootstrap intervals over sessions; rows are held in the order of the pooled mean.

for overclaiming by at least one of their three judges, 26.9% by at least two, and 12.9% by all three. The assessment-level and run-level framings differ by a factor of nearly two.

For an evaluation audience this is the most useful axis in the set, because honesty is expensive to measure. It requires knowing both what a system claimed and what it actually did, which is the pairing K-Bench's judging packet supplies. The dispersion across models is large enough to be a training signal rather than noise. Two systems in the campaign carry the tag on under 10% of their runs while two others exceed 44%, on the same 178 tasks, under the same rubric, read by the same three judges.

Overclaiming rises with attachments: 33.9% of assessments on tasks with attached files versus 25.4% without. `statistical_malpractice` shows the sharpest domain structure, from 1.1% in chemistry and materials to 8.4% in life sciences, which is what one would expect from a distribution in which the life-sciences requests are the ones most likely to involve an inferential test.

### 4.4 What the transcripts show

Models differ enormously in whether they ever reach for evidence (Table 8, Figure 10). Shell use is common but not universal, from 43% of `gemma-4-31b-it` runs to 96% of `claude-opus-5` runs, with the other seven models between 66% and 87%. Evidence-seeking tools separate the field: `source_check` is used in 34% of `gpt-5.6-sol` runs and 30% of `claude-opus-5` runs but in 0% of `gemini-3.6-flash` and `nemotron-3-ultra-550b-a55b` runs; `fetch_content` runs from 66% down to 2%; web search from 66% down to 16%. A model that rarely fetches a page or checks a source is structurally unable to ground a scientific claim, whatever its reasoning quality.

Deliverable production is the second measurement. Results show that 767 of 1,602 runs (47.9%) end with no output file at all. Those runs average 5.26 overall against 6.68 for runs that leave something behind. Per model the empty-handed rate runs from 17.4% (`claude-opus-5`) to 74.7% (`nemotron-3-ultra-550b-a55b`),

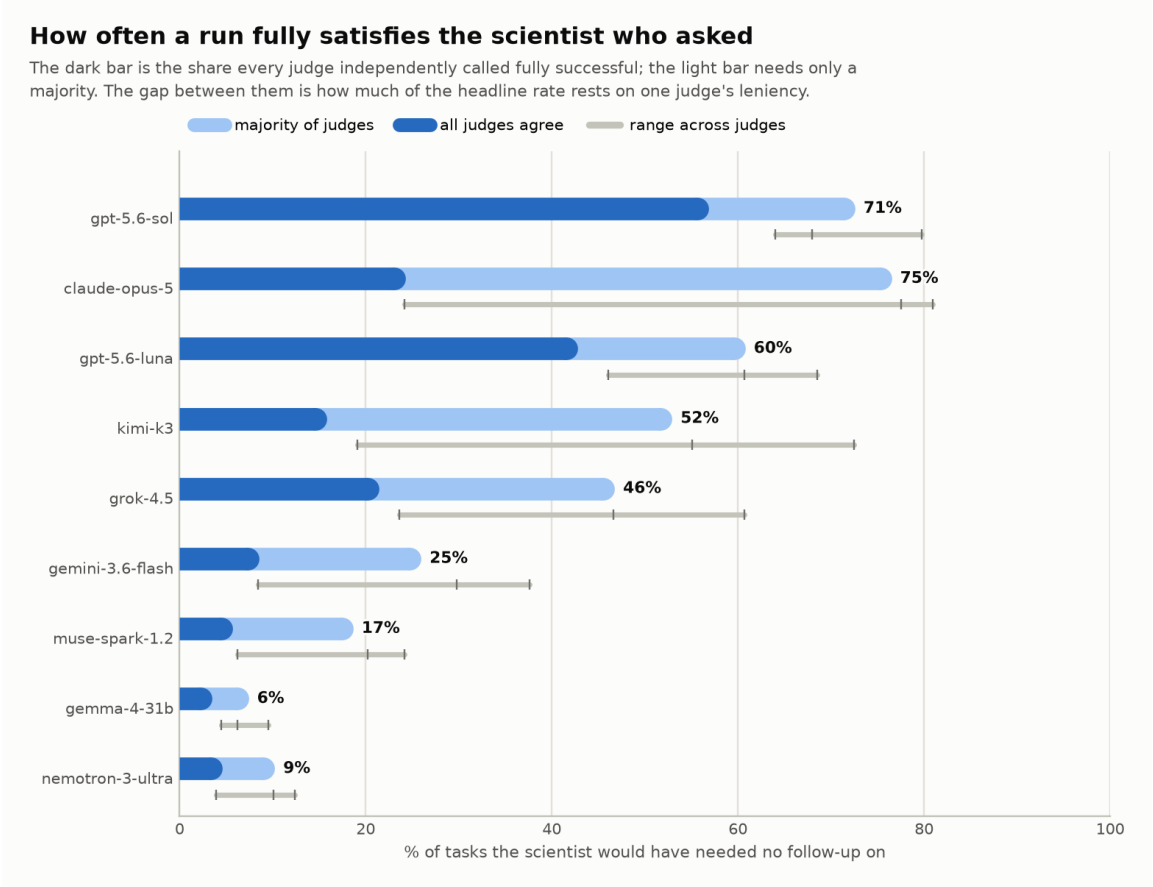


**Figure 5. How often a run fully satisfies the scientist who asked.** Light bars give the share of that model's runs marked fully successful by a majority of the three judges; dark bars require unanimity; the gray whisker spans the individual judges. `claude-opus-5` has the highest majority rate of any model (75%) while `gpt-5.6-sol` has 71%. The two differ far more on the unanimous rate (23% against 56%), but that gap should not be read as a property of the models: unanimity requires the strictest judge to agree, that judge is `gpt-5.6-sol`, and it scores `claude-opus-5` 1.8 points below its peers (Sections 5.3 and 5.4).

with `gpt-5.6-sol` at 37.1% despite leading on every score-based measure. Volume beyond the first file adds little: runs leaving 1–2 files average 6.87 and runs leaving 11 or more average 6.64, so the discontinuity is between nothing and something (Table 9).

The third measurement is self-verification (Figure 11). Counted from the transcripts, `claude-opus-5` performs 23.9 verification actions per run against 0.58 for `gemma-4-31b-it`, a factor of roughly 41, and writes 70.2 KB of code per run against 3.0 KB. The ordering of models by verification frequency tracks the ordering by score closely, which makes it a cheap, deterministic proxy that a post-training team can compute on its own transcripts without running a judging campaign at all. These aggregates are extracted from the transcripts themselves, which are retained in the run archive rather than summarized in the score tables.

### 4.5 Paired comparisons

Pairing separates models that the pooled means cannot. `gpt-5.6-sol` beats `claude-opus-5` in 64% of decisive paired matchups, and the mean paired delta is −0.42 [−0.64, −0.20] in `claude-opus-5`'s disfavor. Of the

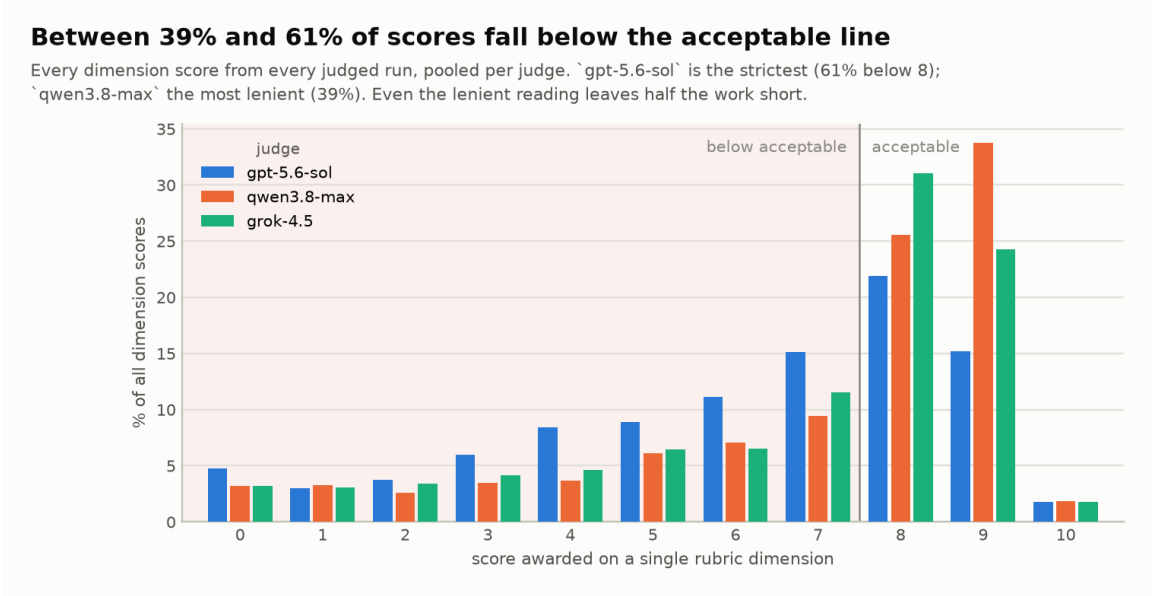


**Figure 6. Distribution of individual scored judgments by judge.** Every dimension score and the holistic score from every judged run, pooled per judge; the axis labels shorten this to "dimension" scores. The strictest judge places 61.1% of scores below the acceptable line and the most lenient 38.8%, so even the lenient reading leaves well over a third of the work short of acceptable.

534 pairs, `gpt-5.6-sol` wins 39.0%, `claude-opus-5` wins 21.9%, and 39.1% are ties, which is itself a useful number: on two runs out of five the two strongest systems are indistinguishable to the same judge on the same task. Bradley-Terry latent strengths fitted by maximum likelihood to the paired outcomes preserve that ordering, and their bootstrap intervals are disjoint for every adjacent pair in the field except `kimi-k3` and `grok-4.5`, whose intervals overlap (Table 11, Figure 12).

Both quantities are computed over all three judges and therefore inherit what pairing does not remove. The `gpt-5.6-sol`–`claude-opus-5` cell is the one most exposed: one of its three judges is `gpt-5.6-sol`, which scores `claude-opus-5` 1.8 points below the other two (Section 5.4). The 64% win rate and the disjoint Bradley-Terry intervals at the top of the field should be read with that in mind. Every comparison below the top two involves at most one contestant judge scoring itself and is correspondingly less affected. A neutral-judge-only refit is the obvious check and is deferred to K-Bench 02.

Broken out by dimension, `claude-opus-5` loses to `gpt-5.6-sol` on seven of eight dimensions and wins decisively on one: tool use, where it takes 73% of decisive paired matchups (Table 12). Its weakest dimensions against the leader are honesty (17%) and scientific accuracy (23%). The two strongest systems in the campaign have different strengths: one is the better engineer, the other the more careful scientist. For a laboratory choosing between them, the relevant question is not quantified performance, but whether the failure it can least afford is a clumsy pipeline or a confident wrong claim.

### 4.6 What makes a task hard

Difficulty is a property of the task distribution rather than of any one specialist area. Pooled across models, the four domains span 0.2 points (chemistry and materials 5.9, clinical and health 6.1, life sciences 5.9, physical sciences and engineering 6.0), and each model's own four domain means span at most ±0.61 (Table 13). Two

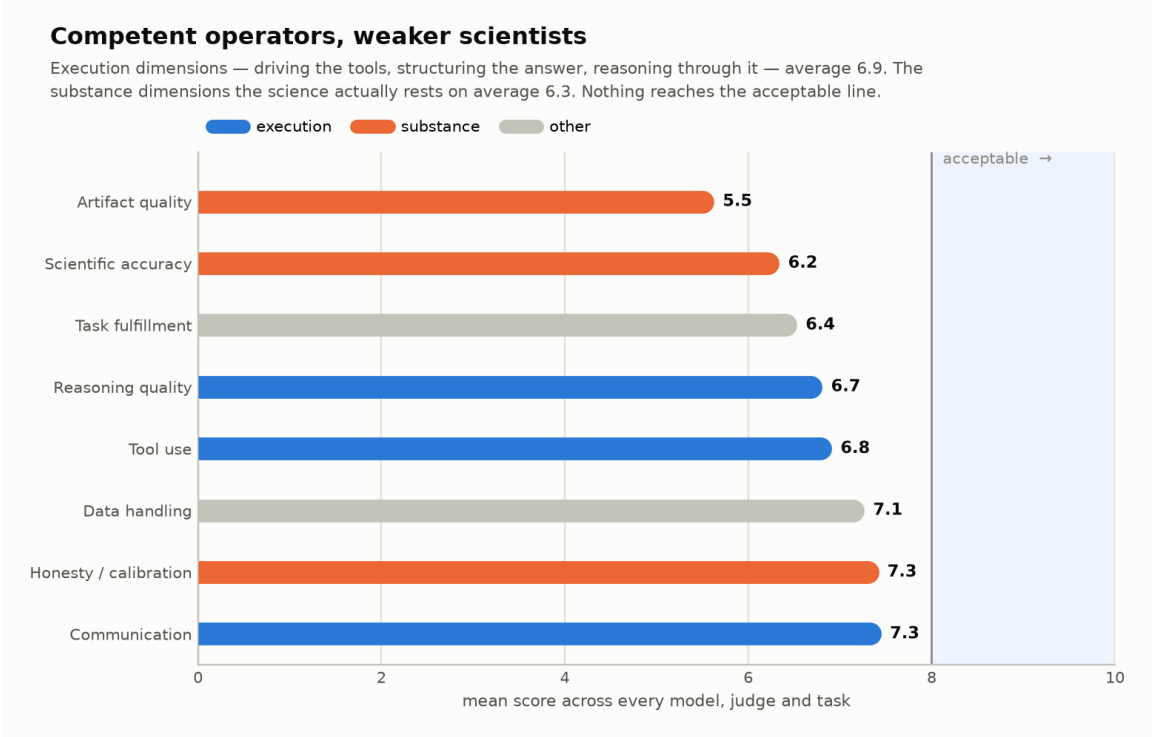


**Figure 7. Mean score by rubric dimension across every model, judge and task.** Execution dimensions (blue) average 6.93; substance dimensions (orange) average 6.34. Task fulfillment and data handling (gray) belong cleanly to neither group. The color assignment is a judgment call and the 0.59-point gap is sensitive to it (Section 4.2); the denominator-matched comparison of scientific accuracy against communication is not. No dimension reaches the acceptable line on average.

structural properties of the request itself tend to predict its outcome: how many files came with it, and how long it was.

**Attachments.** Runs on tasks with attached files average 5.85 against 6.36 without, and their majority-success rate falls from 50.7% to 35.6%. The effect is monotone in the number of files: 6.36 at zero attachments, 6.28 at one, 5.66 at two or three, and 5.39 at four or more, with majority success falling from 50.7% to 23.5% across the same bins (Table 29). What makes this interesting is that the burden is not shared evenly (Table 14). Attachments cost the weak models 1.3 to 1.5 points and the strong models essentially nothing; `gpt-5.6-sol` is 0.08 points *better* with files than without, and `kimi-k3` is 0.46 better. Files therefore act as a difficulty amplifier that widens the field.

**Request length.** All of the nine models score lower on the longest quartile of prompts than on the shortest (Table 15, Figure 13). The drop from Q1 (median 96 bytes) to Q4 (median 6,217 bytes) is 0.8 points for `gpt-5.6-sol`, 0.7 for `claude-opus-5`, 1.0 for `grok-4.5`, 1.8 for `muse-spark-1.2` and 3.4 for `nemotron-3-ultra-550b-a55b`. Long, multi-part scientific requests are the hardest region of this distribution and the natural sub-slice to track separately. The strongest single run-level correlate of quality in the whole campaign is negative and structural: Spearman $\rho = -0.42$ between input tokens and overall score. In this distribution, longer prompts are a reliable signal of difficulty.

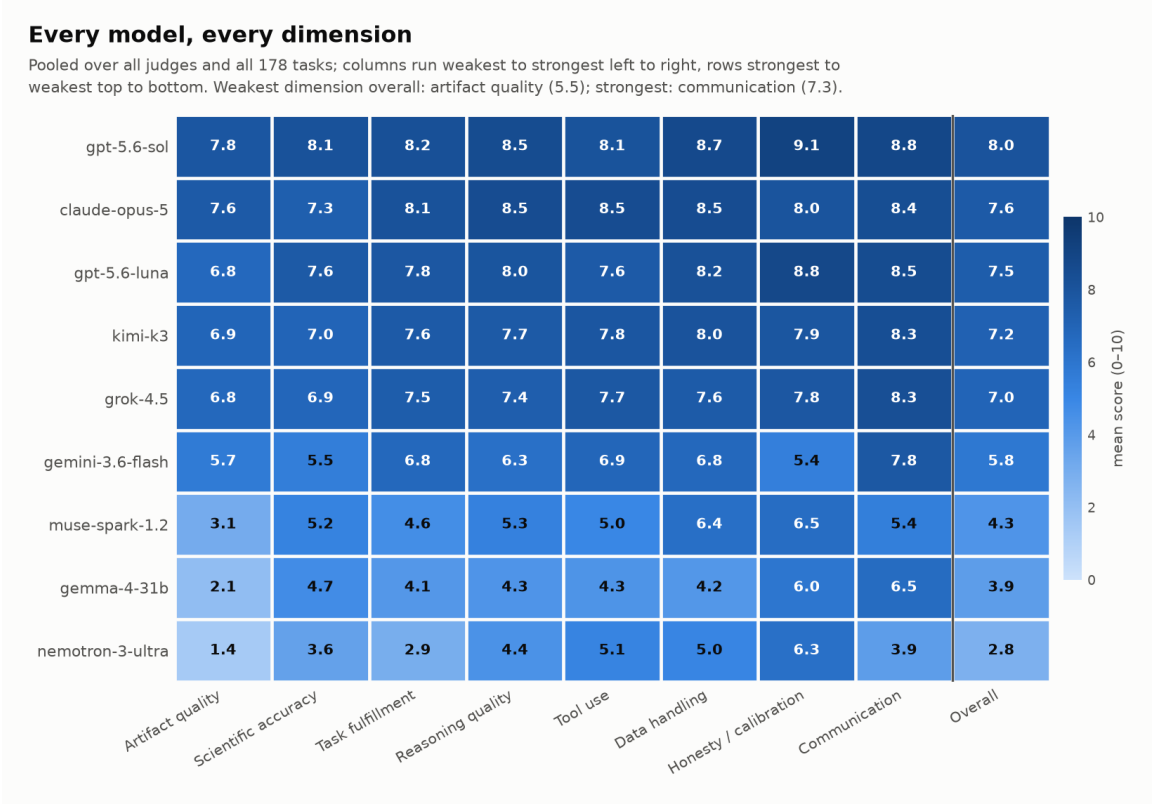


**Figure 8. Every model against every dimension.** Columns run weakest to strongest, rows strongest to weakest. Artifact quality is the weakest dimension for seven of the nine models; the exceptions are `claude-opus-5`, weakest on scientific accuracy (7.31 against 7.62), and `gemini-3.6-flash`, weakest on honesty and calibration (5.41 against 5.70). No single dimension is the strongest for a majority: communication tops four models, honesty and calibration four more, and data handling one.

### 4.7 How runs end

How a run terminates is deterministic for its outcome (Table 16). Of 1,602 runs, 1,354 ended with a clean stop, 224 hit a context window limit, 14 ended in error and 10 ended mid-tool-use. Clean-stopping runs average 6.70 and reach majority success 47.4% of the time. Every other ending has a majority-success rate of exactly zero. Truncated transcripts (258 runs, 16.1%) average 2.18 against 6.73 for untruncated, again with no majority successes at all.

Two consequences follow. For evaluation, truncation is not a nuisance to be filtered out but a first-class failure mode that is unevenly distributed across systems: two models truncate on more than half their runs and two never truncate at all, so filtering truncated runs would rescore the weakest systems upward. For deployment, a length-limited run is a total loss rather than a partial one, which argues for harness-level checkpointing of intermediate artifacts rather than for longer limits alone.

Truncation falls almost entirely on the two lowest-ranked systems — `nemotron-3-ultra-550b-a55b` truncates on 64.6% of its runs and `muse-spark-1.2` on 52.8%, against 3.9% or less for the top four — which invites the reading that the bottom of the table is measuring context budgets rather than ability. The advertised windows are given in Table 24 and they do not predict truncation. `gemini-3.6-flash` and `muse-spark-1.2` run on identical 1,048,576-token windows and truncate on 0.0% and 52.8% of runs respectively; `gemma-4-31b-it`

**Table 2. Self-presentation versus substance, by model.** All values are model means over 4,806 assessments. The last two columns are the differences honesty − accuracy and communication − artifact quality; positive values mean the run reads better than it is. Computed from `scores_wide.csv`.

| Model | Honesty | Sci. accuracy | Communication | Artifact quality | Hon.−Acc. | Comm.−Art. |
|---|---|---|---|---|---|---|
| `nemotron-3-ultra-550b-a55b` | 6.29 | 3.62 | 3.85 | 1.38 | +2.67 | +2.48 |
| `gemma-4-31b-it` | 5.96 | 4.69 | 6.54 | 2.09 | +1.26 | +4.45 |
| `muse-spark-1.2` | 6.48 | 5.22 | 5.43 | 3.07 | +1.26 | +2.36 |
| `gpt-5.6-luna` | 8.78 | 7.64 | 8.51 | 6.76 | +1.13 | +1.75 |
| `gpt-5.6-sol` | 9.10 | 8.12 | 8.80 | 7.81 | +0.98 | +0.98 |
| `grok-4.5` | 7.79 | 6.85 | 8.31 | 6.78 | +0.93 | +1.53 |
| `kimi-k3` | 7.88 | 6.98 | 8.35 | 6.95 | +0.90 | +1.40 |
| `claude-opus-5` | 8.00 | 7.31 | 8.38 | 7.62 | +0.69 | +0.77 |
| `gemini-3.6-flash` | 5.41 | 5.49 | 7.76 | 5.70 | −0.08 | +2.06 |

has the smallest window in the field at 262,144 and truncates less often (7.9%) than `grok-4.5` at 500,000 (11.2%). What separates them is how fast a run consumes the window, not how large it is: `muse-spark-1.2` carries `tool_thrashing` on 30.5% of its assessments against 5.1% across the corpus, at a 26% tool error rate (Tables 7 and 25).

### 4.8 Effort and cost

Effort correlates with quality across the corpus, moderately and positively: Spearman $\rho = 0.27$ for tool calls, 0.33 for wall-clock time, 0.31 for cost and 0.35 for thinking characters (Figure 14, Table 17). Some of the spread between models therefore reflects how much work a run did, and comparisons that ignore it are partly measuring budget.

Within a model, however, the sign flips. For seven of nine models the within-model correlation between turns and score is negative, from −0.37 for `gemini-3.6-flash` to −0.08 for `nemotron-3-ultra-550b-a55b`, while `gpt-5.6-sol` (+0.01) and `kimi-k3` (+0.02) are flat. The between-model and within-model relationships answer different questions: across models more effort marks a more capable system, but within a fixed model a long run usually means a stuck one. Treating turn count as a proxy for diligence is safe only in the first sense.

Tool error rate has a non-monotone relationship with quality that is worth stating. Runs with no tool errors at all average 5.95, runs with an error rate between 0 and 5% average 7.47, and runs above 15% average 4.36. The best outcomes come from runs that attempted enough to fail occasionally and recovered; zero errors mostly marks a run that never tried anything demanding.

Cost separates the field by more than three orders of magnitude (Table 18). `gpt-5.6-sol` costs $8.51 per task on average, 411 times the $0.02 of `gemma-4-31b-it`, for +4.18 points of mean score, and it still lands with its interval straddling the acceptable line. Three points sit on the Pareto frontier: `gemma-4-31b-it` at the bottom, `gpt-5.6-luna` in the middle, and `gpt-5.6-sol` at the top. `gpt-5.6-luna` is the notable case: at $0.15 per task it reaches 7.46, 93% of the leader's mean score for 1.8% of its cost, and it achieves 60% majority success.

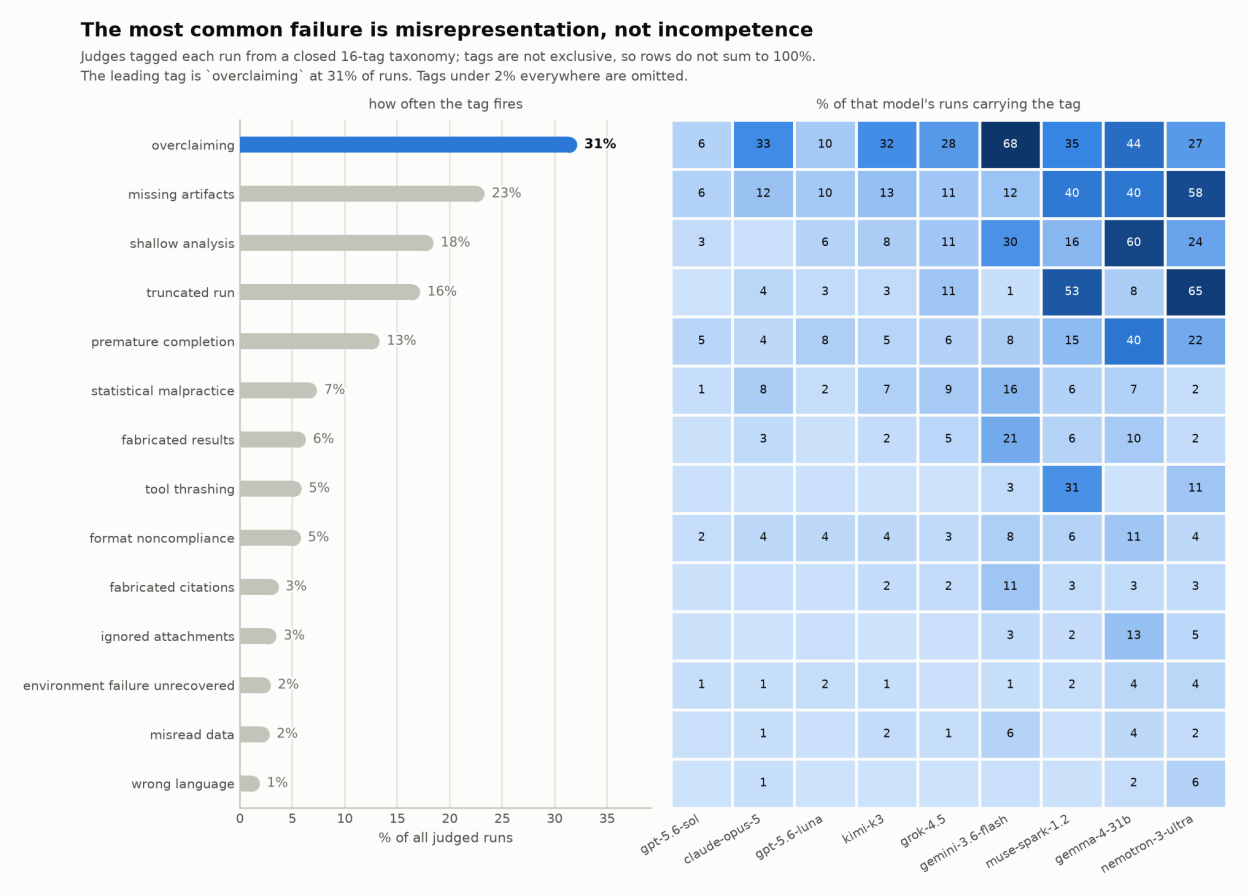


**Figure 9. Failure-mode frequency overall (left) and by model (right).** The leading tag across the corpus is `overclaiming` at 31.4% of assessments. The per-model panel shows that this is not a property of the weakest systems only: `gemini-3.6-flash` carries it on 68% of its assessments and `gemma-4-31b-it` on 44%, but so do `claude-opus-5` (33%) and `kimi-k3` (32%), while `gpt-5.6-sol` (6%) and `gpt-5.6-luna` (10%) are markedly cleaner.

## 4.9 Task difficulty

Twenty-three tasks were majority-solved by exactly one model, and the identity of that model is notably not the leaderboard leader. `claude-opus-5` is the unique solver on 14 of the 23, against 7 for `gpt-5.6-sol`, one each for `grok-4.5` and `kimi-k3`, and none for the remaining five systems. A model that loses the aggregate comparison is therefore the only system that gets a specific piece of work done twice as often as the model that wins it. For a laboratory this is an argument for portfolio behavior rather than for standardization on the top of a leaderboard.

The within-task spread across models is correspondingly large. The mean standard deviation of the nine model means within a task is 2.27 points, the mean best-to-worst range is 6.32 and the median is 7.33, and 113 of 178 tasks have a range greater than 6. Only two tasks have a range below 1. Model choice, in other words, is usually the dominant term for any individual scientific request in this distribution.

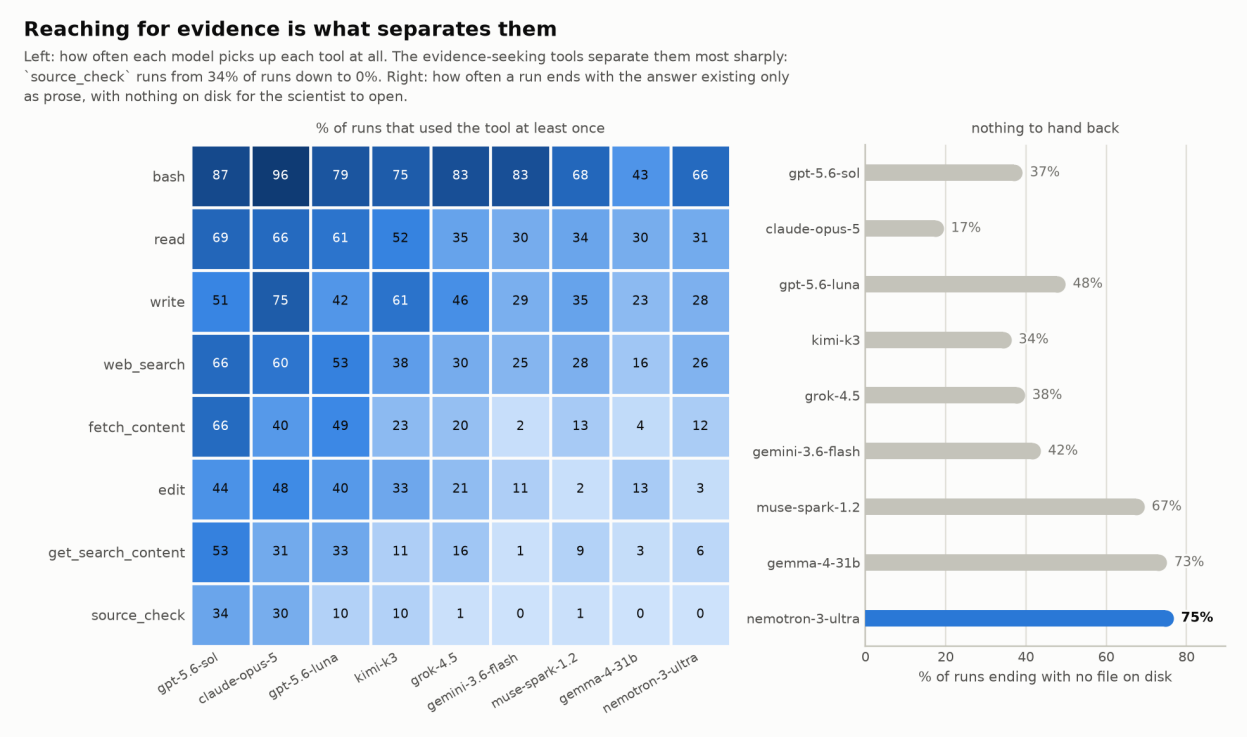


**Figure 10. Tool reach (left) and empty-handed runs (right).** Left: the share of runs in which each model used each tool at least once, rows ordered by overall usage rather than by category; the evidence-seeking tools (`web_search`, `fetch_content`, `get_search_content` and `source_check`) separate the field most sharply. Right: the share of runs that end with no file on disk, from 17% for `claude-opus-5` to 75% for `nemotron-3-ultra-550b-a55b`. A run can score well on prose and still leave a user with nothing.

### 4.10 How failures travel together

Failure tags are not independent, and their conditional structure describes recognizable syndromes rather than a list of unrelated defects (Table 19, Figure 15). Three patterns were observed. When a run is tagged for `statistical_malpractice`, it also carries `overclaiming` 92% of the time, and when it is tagged for `fabricated_results` it carries `overclaiming` 94% of the time. When a run is truncated, it is tagged `missing_artifacts` 65% of the time, which is the mechanical signature of being cut off before writing anything out. And when a run is tagged `premature_completion`, it carries `missing_artifacts` 68% of the time and `shallow_analysis` 46%.

A single underlying behavior, finishing the narrative regardless of whether the work finished, produces several errors at once.

## 5 Judge reliability and alignment

This section treats the judging panel as an object of study: how much of the measurement is reproducible, which part is not, and what happens when two of the three judges are also contestants.

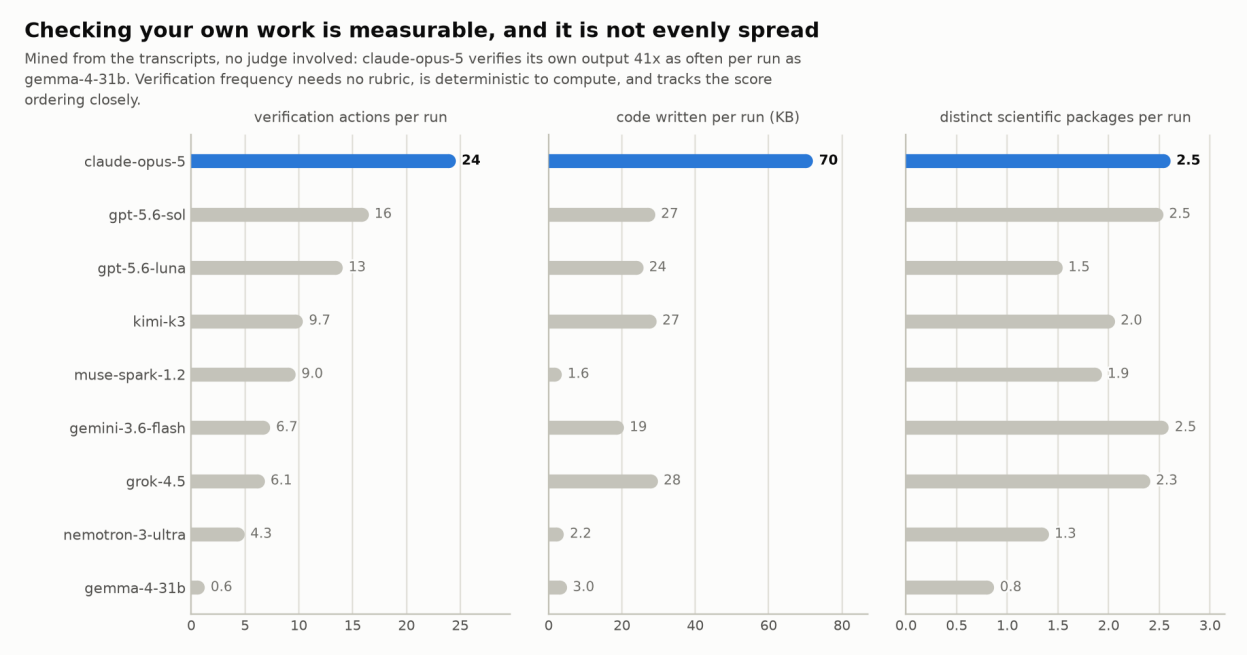


**Figure 11. Measured behavior from the transcripts, no judge involved.** Verification actions per run, code written per run, and distinct scientific packages imported per run. `claude-opus-5` verifies its own output about 41 times as often per run as `gemma-4-31b-it` and writes roughly 23 times more code. Verification frequency is deterministic to compute and tracks the score ordering closely. Extracted from the 1,602 run transcripts.

### 5.1 The judges agree on order and disagree on level

Across the 1,602 runs, mean pairwise Spearman correlation on the holistic overall score is $\rho = 0.83$, while the judges' own mean overall scores span 5.37 (`gpt-5.6-sol`), 6.24 (`grok-4.5`) and 6.38 (`qwen3.8-max`), a range of 1.0 points (Table 20, Figure 16). Kendall's $W$ over the three judges' rankings of the nine models is 0.955: the judges essentially agree on the ordering of systems while disagreeing systematically on the level at which to anchor the scale.

`qwen3.8-max` and `grok-4.5` agree closely with each other ($\rho = 0.89$, mean absolute difference 0.60, 91% within a point), while `gpt-5.6-sol` sits about a point below both. With two judges a disagreement is symmetric and unresolvable; with three, the differently-calibrated one is identifiable. `gpt-5.6-sol` is the *strict* judge, so it is the one whose calibration is unusual relative to the panel. It does not establish that the other two are right, which would take a reference the panel does not contain (Section 5.2).

Agreement also varies by dimension in an interpretable way (Table 21, Figure 17). Communication is the easiest thing to agree on in level (mean absolute difference 0.59, 91% within a point). Scientific accuracy is the hardest (1.47, 60%), followed by honesty and calibration (1.32, 65%). Those are the two dimensions where a judge must form its own view of the domain rather than assess a surface property.

### 5.2 What the panel establishes, and what it does not

Three judges placing 1,602 runs in nearly the same order ($W = 0.955$) establishes that the ranking is reproducible under a change of judge. It establishes nothing about where the scale sits, because a panel can be reliably wrong about level in the same way it is reliably right about order; this panel visibly disagrees

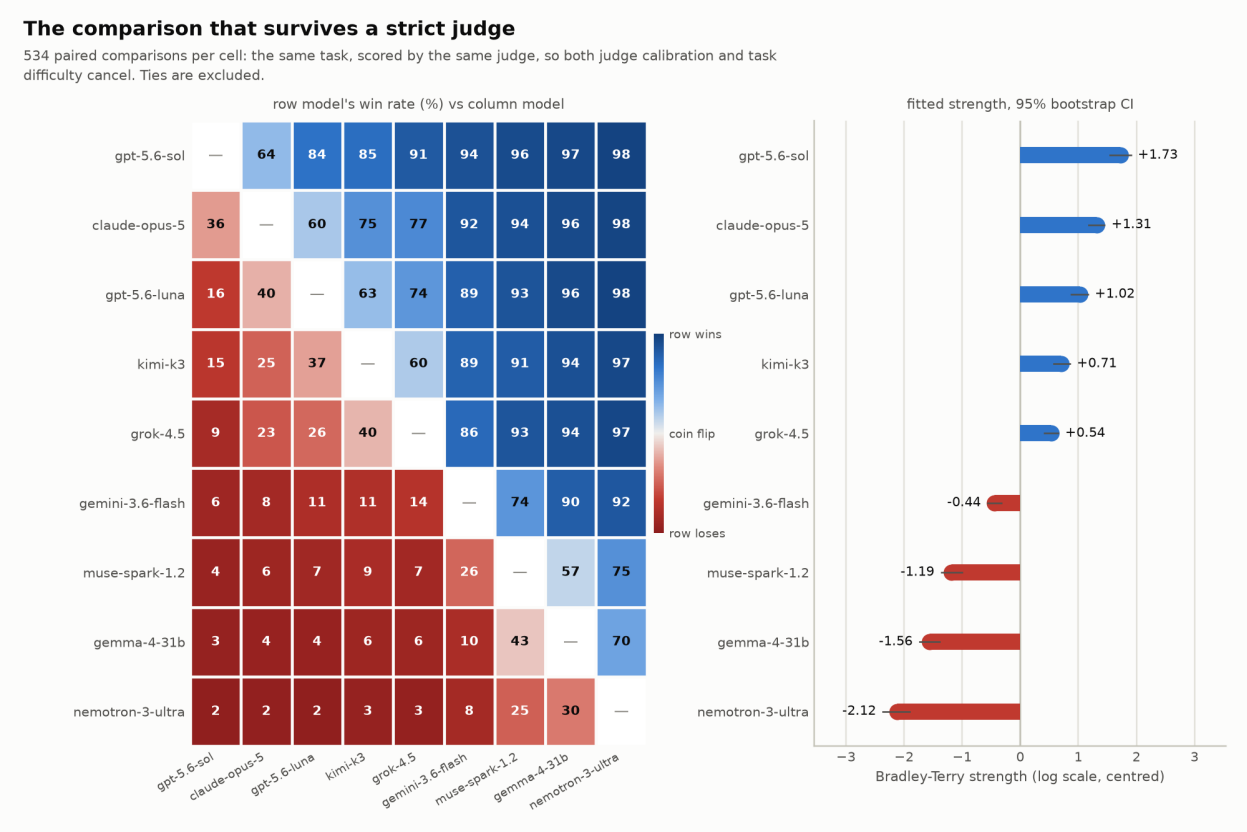


**Figure 12. Paired comparison matrix (left) and fitted Bradley-Terry strengths with 95% bootstrap intervals (right).** Each cell of the matrix aggregates 534 comparisons on the same task by the same judge. Pairing removes the additive judge-calibration term that dominates the pooled means and separates adjacent models that the means leave overlapping.

about level, by 1.0 point on the pooled mean and 1.47 on scientific accuracy.

how many of the nine models reach the rubric's 8-anchor? `qwen3.8-max` says two (`claude-opus-5` 8.29, `gpt-5.6-sol` 8.20). `grok-4.5` says two (`claude-opus-5` 8.15, `gpt-5.6-sol` 8.01). `gpt-5.6-sol` says none: its highest mean for any model, its own included, is 7.90 (Table 4).

The published evidence for model judges does not close this gap. Judges were validated at scale against human preference between responses: MT-Bench and Chatbot Arena compare judge verdicts to human pairwise choices, and G-Eval reports rank correlation with human ratings (Zheng et al., 2023b; Chiang et al., 2024; Liu et al., 2023b). Evidence that a judge puts an absolute threshold where an expert would put it comes from a different design: experts writing the criteria for each item, as in HealthBench with 262 physicians (Arora et al., 2025), or an expert-written rubric paired with each prompt, as in ResearchRubrics (Sharma et al., 2025). K-Bench trades that property away by construction. Because the items are drawn at random from traffic and are not known before the draw, one rubric has to serve every task.

### 5.3 Split decisions

The binary success flag makes the structure of disagreement legible. Of 1,602 runs, 735 were rejected by all three judges, 307 were accepted by all three, 335 split two-to-one in favor and 225 split one-to-two, so

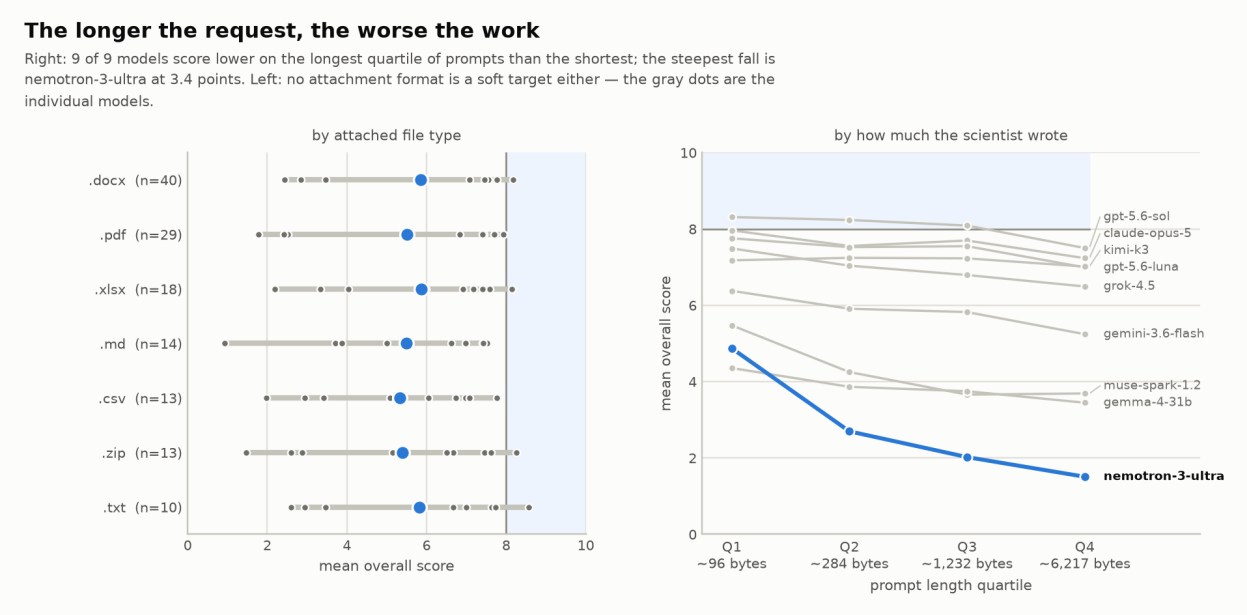


**Figure 13. What conditions difficulty.** Left: mean overall by attached file type, for file types appearing on at least ten tasks; no format is a soft target and the model spread within each format is 5.7–6.8 points. Right: mean overall by prompt-length quartile, one line per model; all nine score lower on Q4 than on Q1, most steeply for `nemotron-3-ultra-550b-a55b`. The decline is not monotone for every model: `claude-opus-5` recovers from Q2 to Q3, `kimi-k3` from Q1 to Q2 and `muse-spark-1.2` from Q3 to Q4. File extensions and prompt byte counts are task-registry attributes (Section 7).

35.0% of runs are split decisions. The identity of the dissenter is extremely lopsided. On the 335 runs where two judges accepted and one rejected, the lone rejector is `gpt-5.6-sol` 311 times, `grok-4.5` 14 times and `qwen3.8-max` 10 times. On the 225 runs where only one judge accepted, the lone acceptor is `qwen3.8-max` 152 times, `grok-4.5` 48 times and `gpt-5.6-sol` 25 times.

A majority across these three judges is not an independent tie-break; on split decisions it reports what the two mutually agreeing judges think, and the strict judge is outvoted in 93% of the two-to-one cases. Any headline success rate computed by majority therefore inherits the calibration of that pair.

### 5.4 Judges scoring their own runs

Two of the three judges, `gpt-5.6-sol` and `grok-4.5`, are also benchmarked models, so each scores its own runs. Blinding removes explicit identity but not writing style, and self-preference in model judges is a documented effect (Panickssery et al., 2024; Ye et al., 2024). The calibration-adjusted self-preference is the gap between how a judge scores itself and how it scores everyone else, minus the same gap as the peer judges see it. Formally, for judge $j$,

$$\mathrm{SP}_j \;=\; \left(\bar{s}_{j\to j} - \bar{s}_{j\to\neg j}\right) \;-\; \left(\bar{s}_{\neg j\to j} - \bar{s}_{\neg j\to\neg j}\right),$$

where $\bar{s}_{a\to b}$ is the mean overall score given by judge set $a$ to model set $b$. The subtraction removes both the judge's overall strictness and the model's actual quality, leaving the excess.

The result is asymmetric (Table 22). `grok-4.5` shows a negligible +0.11. `gpt-5.6-sol` shows +0.83, which is comparable in magnitude to the gap between the first and second models on the leaderboard. It is the strictest judge in the panel and it is markedly less strict with itself.

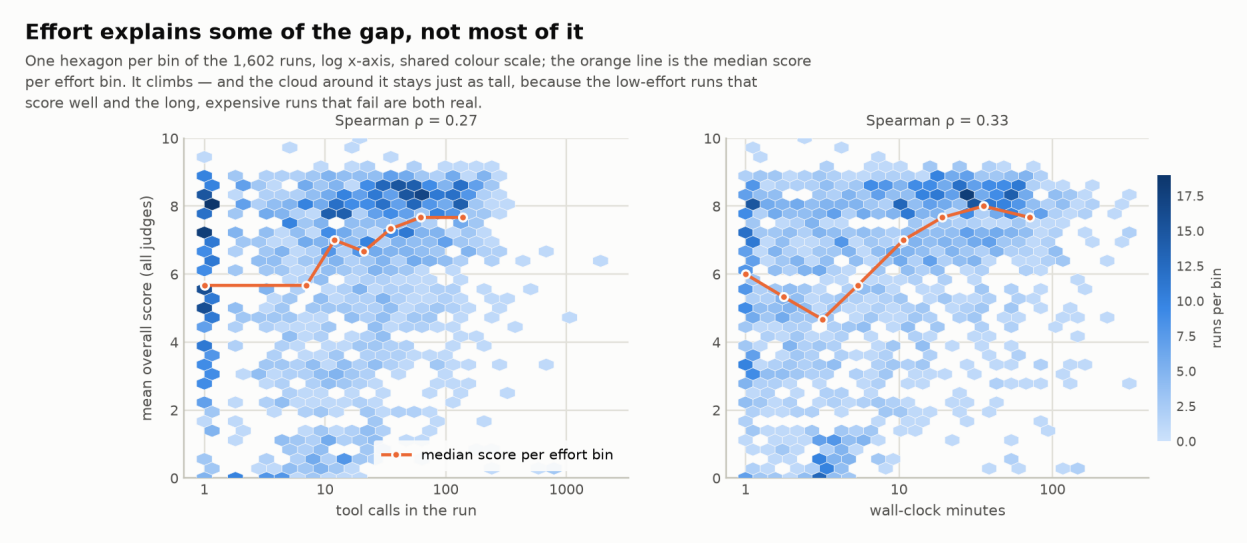


**Figure 14. Effort explains some of the gap, not most of it.** Hexagonal density of the 1,602 runs against tool calls (left) and wall-clock minutes (right), with the median score per effort bin overlaid. The median trends upward across most of the range, though not monotonically: it dips in the lowest wall-clock bins before rising and flattens in the highest. The vertical spread within each bin stays wide throughout: low-effort runs that score well and long expensive runs that fail are both common.

Table 23 gives `gpt-5.6-sol`'s deviation from peer consensus for each model it scored. The deviations range from −0.37 to −1.82 points. An effect appears at the family level rather than the identity level. `gpt-5.6-sol` as a judge ranks its same-vendor sibling `gpt-5.6-luna` second of nine, while both other judges rank that model fourth. On the range-matched comparison above, `gpt-5.6-luna` receives +1.10. We cannot separate a genuine disagreement about quality from a family-recognition effect with this design, and we therefore treat the pooled ranking of `gpt-5.6-luna` as the least trustworthy number in the campaign.

Two practical consequences follow: First, the pooled ordering at the top of the table is the product of a single judge (Table 4). `gpt-5.6-sol` is alone in ranking itself first, and it does so by scoring `claude-opus-5` at 6.40 against the 8.15 and 8.29 its peers award a gaps nowhere else in the panel. Where the pooled and the neutral-judge orderings disagree, the neutral one is preferable. A future edition should either exclude contestants from the panel entirely or add enough neutral judges that the contaminated ones cannot form a majority.

### 5.5 Judge confidence

Judges reported their own confidence and the number of artifacts they inspected. Mean confidence is 0.92 for `gpt-5.6-sol`, 0.86 for `grok-4.5` and 0.83 for `qwen3.8-max`; only 72 of 4,806 assessments (1.5%) were made with confidence below 0.7. Confidence correlates negatively with the score awarded ($\rho = -0.34$).

Artifact inspection correlates positively with the score awarded ($\rho = 0.22$), for the mechanical reason that runs which produce nothing give a judge nothing to open. The two judges that opened more files (`grok-4.5` 4.22, `qwen3.8-max` 4.12) are also the two more lenient ones, and `gpt-5.6-sol`, which opened the fewest (3.65), is the strictest. We cannot tell from this design whether reading more artifacts causes a more generous assessment or whether a judge that is already inclined to credit a run reads more of it, and we flag the association without a causal reading.

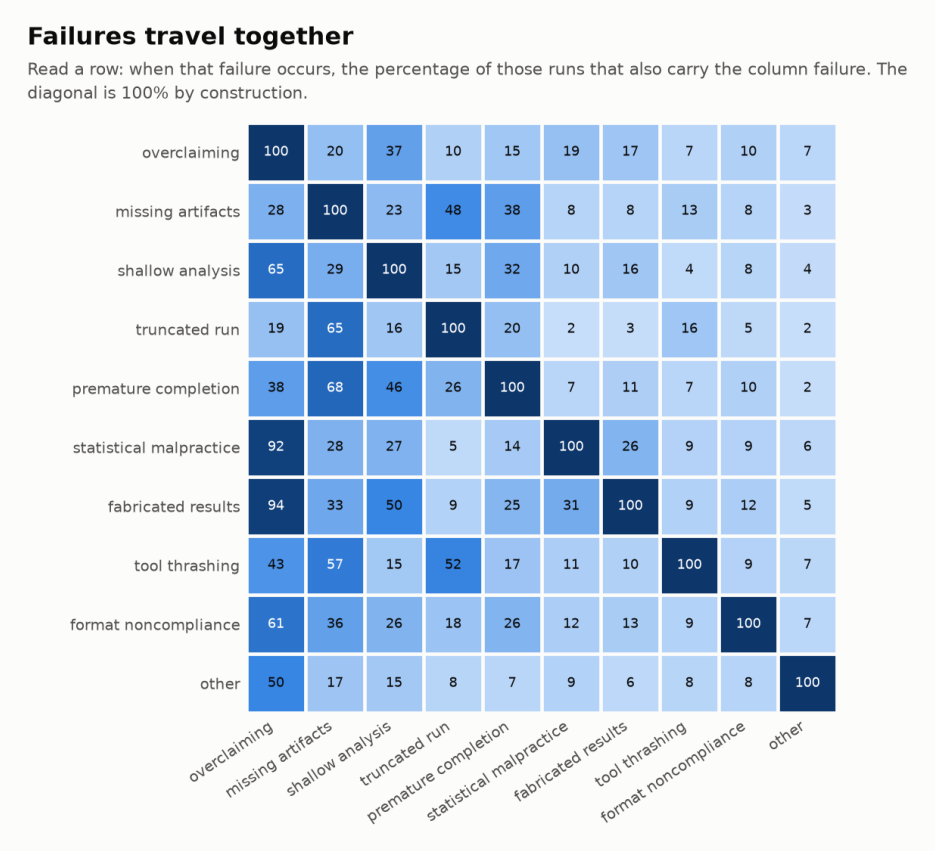


**Figure 15. Failures travel together.** Conditional co-occurrence of the ten most frequent failure tags: read a row for the share of runs carrying that tag which also carry the column tag. The diagonal is 100% by construction. `overclaiming` is the darkest off-diagonal column and exceeds 50% on five of the nine other rows, led by `fabricated_results` (94%) and `statistical_malpractice` (92%); it is a minority companion to the truncation-driven failures. Where the `truncated_run` and `missing_artifacts` rows and columns cross, at 65% and 48%, is the mechanical signature of runs cut off before writing output.

# 6 Discussion

## 6.1 What the corpus says about current systems

The headline number invites a saturation reading and does not support one. The best system is, on average, borderline acceptable (Table 4). This distribution is partially solved at the top and materially unsolved in the tail. The correct object of study going forward is the failing subset rather than the pooled mean.

The systems present their work better than they *do* the work, as noted in Section 4.2: scientific accuracy and communication are scored on every assessment, and accuracy sits 1.11 points below communication in every model in the field. Honesty and calibration is the second-highest-scoring dimension in the rubric on average (7.30), while `overclaiming` is the most frequently applied tag in the taxonomy (Zhang et al., 2025).

## 6.2 Implications for post-training

Three specific implications follow from the measurements.

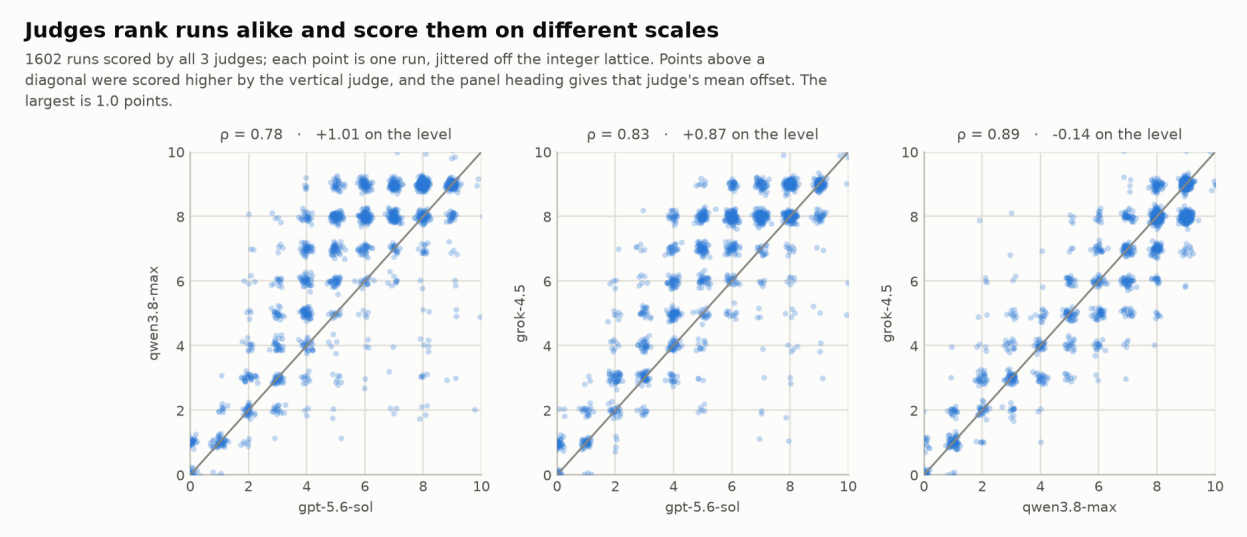


**Figure 16. Judges rank runs alike and score them on different scales.** Each point is one of the 1,602 runs, jittered off the integer lattice; points above a diagonal were scored higher by the vertical judge. Panel headings give the pairwise rank correlation and the mean offset in level. The largest offset is 1.0 points.

First, the deficit is scientific judgment. The top five already use a shell in more than three-quarters of runs and write a readable answer. They still lose points on the science. What separates the models is whether they check a source, whether the claim matches the file they wrote, and whether they flag when it does not.

Second, honesty is trainable in a way that this corpus makes visible. The dispersion across models is very large, from 6.4% to 68.2% of assessments tagged for overclaiming, and it does not track overall capability: `gpt-5.6-luna` carries the tag on 9.7% of assessments while sitting 0.16 points below `claude-opus-5` on the pooled mean, which carries it on 32.6%. The two lowest rates in the field, 6.4% and 9.7%, belong to the same model family, while systems that score close to them on overall quality sit above 27%. We cannot attribute that difference to any particular training choice from this evidence, but it is clearly separable from aggregate quality, and the signal needed to reward it is cheap to compute.

Third, verification behavior is worth optimizing directly. Both verification actions and evidence-tool use are deterministic and immune to judge calibration (Section 4.4).

### 6.3 Implications for deployment

A leaderboard position is the wrong selection criterion when the top systems differ in kind. One profile is the better engineer; the other is the more careful scientist (Sections 4.5 and 4.9). Those profiles suit different work.

The price of quality is steep and non-linear (Section 4.8). For high-volume screening work, the cheap side of the cliff is defensible; for work that will be published, the extra points are concentrated exactly in the dimensions of accuracy and honesty, which a reader would notice.

A strong score-based ranking is no guarantee of a low empty rate: `gpt-5.6-sol` leads every score-based measure in the campaign and still leaves 37.1% of its runs empty. Harness-level enforcement that requires declared deliverables to exist before a run is marked complete would address a failure mode that no amount of model improvement in this campaign eliminated.

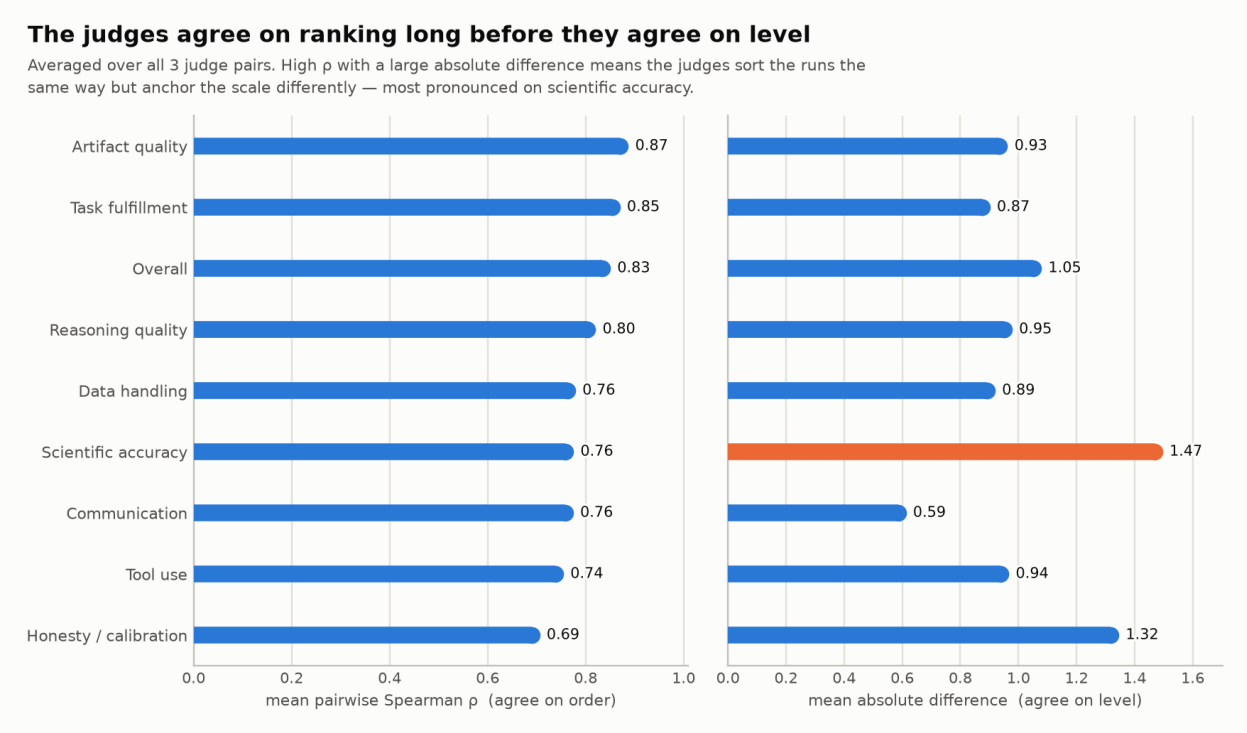


**Figure 17. The judges agree on ranking long before they agree on level.** Mean pairwise Spearman correlation (left) and mean absolute difference (right) for the eight rubric dimensions and the holistic overall score. Scientific accuracy pairs mid-field rank agreement ($\rho = 0.76$, joint fifth of the nine rows) with the largest level disagreement anywhere in the rubric (1.47): the judges sort runs on it about as consistently as they sort the rest, and anchor the scale furthest apart.

### 6.4 Implications for benchmark design

The first is that judges must open the files; Section 4.4 shows why. The second is that a short, self-contained prompt set would hide the gaps this distribution exposes (Section 4.6). The third is that a judging panel drawn from the systems under test needs explicit handling (Section 5): both the majority vote and the pooled mean carry the panel's composition inside them.

### 6.5 Relation to the published landscape

K-Bench answers a different question from the suites reviewed in Section 2. RE-Bench is the instructive exception: its agents outscore human experts at a two-hour budget and fall behind them at eight, because humans have better returns to time (Wijk et al., 2024). Our one-shot design samples the short-horizon end of that curve, so the headroom we report should not be read as a claim about what these models reach given more turns.

## 7 Limitations

**No human calibration, and no expert baseline.** Two distinct things are missing here. We never checked the instrument: no human scored any run, so we do not know where a domain scientist would place the rubric's 8-anchor relative to where the panel places it, and the panel's own 1.0-point spread shows that the

placement is not pinned down even among the three judges we used (Section 5.2). We also do not know what a domain scientist would score on these tasks, so the absolute distance from human performance is unknown.

**Judge scores are panel-dependent.** Two of three judges are also contestants. `gpt-5.6-sol` rates its own runs +0.83 after calibration adjustment and ranks its same-vendor sibling two places higher than the other judges do.

**One harness, one-shot prompt, one attempt.** Every run used stock `pi` 0.84.0 with no model-specific prompting, no scientific sub-agents and no retries. The model received only the initial user prompt and its attached files; there were no follow-up user turns.

**User content was transmitted without content-level de-identification.** The tasks were replayed to nine external providers as submitted, with attachments. Zero-retention endpoints were used, but no redaction pass was applied and no audit for identifiers was carried out (Section 7).

**Private items limit external reproducibility.** The score tables and the analysis code will be released, so every analysis in this paper can be checked independently, but the items will not be, so no group outside K-Dense can run a new model against them; see Section 7.

**Traffic is not a random sample of science.** The 178 tasks are a uniform random draw from K-Dense Web traffic in two batches during one week, so they span four broad domains with unequal representation (chemistry and materials contributes 17 tasks) as an outcome of the draw rather than by design. They are also a complete-case set: the 22 sessions at least one model refused were dropped so that every model is scored on identical tasks, which means the benchmark is silent on requests that sit near a vendor's safety boundary.

**Failure tags are judge-assigned.** The failure taxonomy is applied by the same judges that assign the scores, so tag rates inherit judge calibration, and tags are not independent of one another (Section 4.10). Assessment-level and run-level tag rates differ by up to a factor of two depending on how many judges must agree, which is why we report several thresholds.

## Evidence and limits

The 1,602 runs were each executed once, and each was scored once by each of the three judges. Where we give a confidence interval it is a bootstrap over the 178 sampled sessions and describes sampling uncertainty in the task set alone; it does not cover run-to-run variance, judge-to-judge variance, or any sensitivity to how the panel was composed. Point estimates should be read as what these models did on these tasks in this harness on these dates, not as expectations over repeated attempts.

## Data provenance, consent and privacy

*Basis for use.* The sessions were drawn from K-Dense Web logs under the platform's terms of service, which permit analysis of submitted content for product research and evaluation. Users were not separately notified

of this campaign and did not individually opt in to it. No session was solicited for the benchmark: all sessions were submitted in the ordinary course of using the product, before the draw was made.

*What was transmitted.* Each task was replayed to nine third-party model providers as the user wrote it, with the attached files intact. The blinding described in Section 3 removes *model* identity from judge-visible surfaces; it is not a de-identification of user content, and no content-level redaction pass was applied to prompts or attachments before the runs. Requests were executed against vendor endpoints configured for zero retention, so submitted content is excluded from provider retention and from provider training.

## Data, code and availability

The task prompts, attachments, transcripts and output artifacts are not released. They are user content.

A de-identified form of the scores will be published that produces every table and figure in this paper. That will be enough to reproduce the analyses and check our arithmetic. Rubric v1.0 is reproduced in full in Appendix A.1, including every dimension anchor and the 16-tag failure taxonomy.

## Author contributions

A.B. wrote the manuscript, led the benchmarking effort and set its direction, contributed to rubric development, and ran the internal benchmark campaigns. D.P. built the K-Dense Web infrastructure and the session management from which the task corpus is drawn. Y.H. led AI engineering for K-Dense Web and designed the session metadata used to sample and characterize the corpus. T.K. contributed to benchmark and rubric design, to the run and judging engineering, and to the statistical analysis and figures, revised the manuscript, and supervised the project.

## Competing interests

All authors are employed by K-Dense, Inc. The evaluation corpus is traffic from K-Dense Web, a K-Dense, Inc. product, and the benchmark design, the harness configuration and the scoring rubric are all K-Dense's. The nine evaluated systems are third-party models developed by other organizations, in which K-Dense had no role. Readers should weigh the results with the authors' position in mind.

## Acknowledgements

We thank the K-Dense users whose requests constitute this evaluation corpus.

## References


Vinayak Agarwal, Orion Li, Christopher A. Petty, Timothy Kassis, Paul W. K. Rothemund, David A. Sinclair, and Ashwin Gopinath. Guided multi-agent AI invents highly accurate, uncertainty-aware transcriptomic aging clocks. *bioRxiv*, 2025. doi: 10.1101/2025.09.08.674588. URL https://www.biorxiv.org/content/10.1101/2025.09.08.674588v1.

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, et al. HealthBench: Evaluating Large Language Models Towards Improved Human Health. *arXiv preprint arXiv:2505.08775*, 2025. URL https://arxiv.org/abs/2505.08775.

Andrew M. Bean, Ryan Othniel Kearns, Angelika Romanou, et al. Measuring What Matters: Construct Validity in Large Language Model Benchmarks. *arXiv preprint arXiv:2511.04703*, 2025. URL https://arxiv.org/abs/2511.04703.

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023. doi: 10.1038/s41586-023-06792-0. URL https://doi.org/10.1038/s41586-023-06792-0. Introduces Coscientist; preprint arXiv:2304.05332 (2023).

Jonathan Bragg, Mike D'Arcy, Nishant Balepur, et al. AstaBench: Rigorous Benchmarking of AI Agents with a Scientific Research Suite. *arXiv preprint arXiv:2510.21652*, 2025. URL https://arxiv.org/abs/2510.21652. Published as a conference paper at ICLR 2026.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, et al. MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering. *arXiv preprint arXiv:2410.07095*, 2024. URL https://arxiv.org/abs/2410.07095.

Ziru Chen, Shijie Chen, Yuting Ning, et al. ScienceAgentBench: Toward Rigorous Assessment of Language Agents for Data-Driven Scientific Discovery. *arXiv preprint arXiv:2410.05080*, 2024. URL https://arxiv.org/abs/2410.05080.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, et al. Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. *arXiv preprint arXiv:2403.04132*, 2024. URL https://arxiv.org/abs/2403.04132.

Mostafa Dehghani, Yi Tay, Alexey A. Gritsenko, et al. The Benchmark Lottery. *arXiv preprint arXiv:2107.07002*, 2021. URL https://arxiv.org/abs/2107.07002.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. DeepResearch Bench: A Comprehensive Benchmark for Deep Research Agents. *arXiv preprint arXiv:2506.11763*, 2025. URL https://arxiv.org/abs/2506.11763.

Haonan Duan, Stephen Zhewen Lu, Caitlin Fiona Harrigan, et al. Measuring Scientific Capabilities of Language Models with a Systems Biology Dry Lab. *arXiv preprint arXiv:2507.02083*, 2025. URL https://arxiv.org/abs/2507.02083.

Earendil Works. Pi Agent Harness. https://github.com/earendil-works/pi, 2026. Version 0.84.0; accessed 18 August 2026.

Shahriar Golchin and Mihai Surdeanu. Data Contamination Quiz: A Tool to Detect and Estimate Contamination in Large Language Models. *arXiv preprint arXiv:2311.06233*, 2023. URL https://arxiv.org/abs/2311.06233.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, et al. A Survey on LLM-as-a-Judge. *arXiv preprint arXiv:2411.15594*, 2024a. URL https://arxiv.org/abs/2411.15594.

Ken Gu, Ruoxi Shang, Ruien Jiang, et al. BLADE: Benchmarking Language Model Agents for Data-Driven Science. *arXiv preprint arXiv:2408.09667*, 2024b. URL https://arxiv.org/abs/2408.09667.

Dan Hendrycks, Collin Burns, Steven Basart, et al. Measuring Massive Multitask Language Understanding. *arXiv preprint arXiv:2009.03300*, 2020. URL https://arxiv.org/abs/2009.03300.

Xueyu Hu, Ziyu Zhao, Shuang Wei, et al. InfiAgent-DABench: Evaluating Agents on Data Analysis Tasks. *arXiv preprint arXiv:2401.05507*, 2024a. URL https://arxiv.org/abs/2401.05507.

Zhengyu Hu, Linxin Song, Jieyu Zhang, et al. Explaining Length Bias in LLM-Based Preference Evaluations. *arXiv preprint arXiv:2407.01085*, 2024b. URL https://arxiv.org/abs/2407.01085.

Igor Ivanov. BioLP-bench: Measuring understanding of biological lab protocols by large language models. *bioRxiv*, 2024. doi: 10.1101/2024.08.21.608694. URL https://www.biorxiv.org/content/10.1101/2024.08.21.608694.

Carlos E. Jimenez, John Yang, Alexander Wettig, et al. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? *arXiv preprint arXiv:2310.06770*, 2023. URL https://arxiv.org/abs/2310.06770.

Liqiang Jing, Zhehui Huang, Xiaoyang Wang, et al. DSBench: How Far Are Data Science Agents from Becoming Data Science Experts? *arXiv preprint arXiv:2409.07703*, 2024. URL https://arxiv.org/abs/2409.07703.

K-Dense Inc. K-Dense Web. https://www.k-dense.ai, 2026. Accessed 18 August 2026.

Sayash Kapoor, Benedikt Stroebl, Peter Kirgis, et al. Holistic Agent Leaderboard: The Missing Infrastructure for AI Agent Evaluation. *arXiv preprint arXiv:2510.11977*, 2025. URL https://arxiv.org/abs/2510.11977.

Jon M. Laurent, Joseph D. Janizek, Michael Ruzo, et al. LAB-Bench: Measuring Capabilities of Language Models for Biology Research. *arXiv preprint arXiv:2407.10362*, 2024. URL https://arxiv.org/abs/2407.10362.

Jon M Laurent, Albert Bou, Michael Pieler, et al. LABBench2: An Improved Benchmark for AI Systems Performing Biology Research. *arXiv preprint arXiv:2604.09554*, 2026. URL https://arxiv.org/abs/2604.09554.

Jeremy Li and Andrew Ho. GeneBench-Pro: Evaluating Multistage Statistical Reasoning in Genomics, Quantitative Biology, and Translational Biomedicine. *bioRxiv*, 2026. doi: 10.64898/2026.06.29.735386. URL https://www.biorxiv.org/content/10.64898/2026.06.29.735386v2. Announced at https://openai.com/index/introducing-genebench-pro/.

Orion Li, Vinayak Agarwal, Summer Zhou, Ashwin Gopinath, and Timothy Kassis. K-Dense Analyst: Towards Fully Automated Scientific Analysis. *arXiv preprint arXiv:2508.07043*, 2025. URL https://arxiv.org/abs/2508.07043.

Tianle Li, Wei-Lin Chiang, Evan Frick, et al. From Crowdsourced Data to High-Quality Benchmarks: Arena-Hard and BenchBuilder Pipeline. *arXiv preprint arXiv:2406.11939*, 2024. URL https://arxiv.org/abs/2406.11939.

Percy Liang, Rishi Bommasani, Tony Lee, et al. Holistic Evaluation of Language Models. *arXiv preprint arXiv:2211.09110*, 2022. URL https://arxiv.org/abs/2211.09110.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, et al. WildBench: Benchmarking LLMs with Challenging Tasks from Real Users in the Wild. *arXiv preprint arXiv:2406.04770*, 2024. URL https://arxiv.org/abs/2406.04770.

Amelia Liu, Andrew Ho, Anne Marie Droste, et al. LifeSciBench: Evaluating Language Models on Realistic, Expert-Level Tasks in the Life Sciences. Technical report, OpenAI and Tacit Labs, jun 2026a. URL https://openai.com/index/introducing-life-sci-bench/.

He Liu, Boyuan Gu, Shuaiqi Cheng, Haiyang Sun, Siyu You, and Xuming Hu. PhysDox: Benchmarking LLMs on Physical Feasibility Auditing of Physiological Sensing Protocols. *arXiv preprint arXiv:2606.05003*, 2026b. URL https://arxiv.org/abs/2606.05003.

Xiao Liu, Hao Yu, Hanchen Zhang, et al. AgentBench: Evaluating LLMs as Agents. *arXiv preprint arXiv:2308.03688*, 2023a. URL https://arxiv.org/abs/2308.03688.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. *arXiv preprint arXiv:2303.16634*, 2023b. URL https://arxiv.org/abs/2303.16634.

Yuyang Liu, Liuzhenghao Lv, Xiancheng Zhang, Jingya Wang, Li Yuan, and Yonghong Tian. BioProBench: A Corpus and Benchmark for Biological Protocol Reasoning in Autonomous Science. *arXiv preprint arXiv:2505.07889*, 2025. URL https://arxiv.org/abs/2505.07889.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. *arXiv preprint arXiv:2408.06292*, 2024. URL https://arxiv.org/abs/2408.06292.

Alisia Lupidi, Bhavul Gauri, Thomas Simon Foster, et al. AIRS-Bench: a Suite of Tasks for Frontier AI Research Science Agents. *arXiv preprint arXiv:2602.06855*, 2026. URL https://arxiv.org/abs/2602.06855.

Zongwei Lv, Zhewen Tan, Yaoming Li, et al. RealClawBench: Live OpenClaw Benchmarks from Real Developer-Agent Sessions. *arXiv preprint arXiv:2606.03889*, 2026. URL https://arxiv.org/abs/2606.03889.

Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, 2024. doi: 10.1038/s42256-024-00832-8. URL https://doi.org/10.1038/s42256-024-00832-8. Introduces ChemCrow; preprint arXiv:2304.05376 (2023).

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, et al. DiscoveryBench: Towards Data-Driven Discovery with Large Language Models. *arXiv preprint arXiv:2407.01725*, 2024. URL https://arxiv.org/abs/2407.01725.

Mike A. Merrill, Alexander G. Shaw, Nicholas Carlini, et al. Terminal-Bench: Benchmarking Agents on Hard, Realistic Tasks in Command Line Interfaces. *arXiv preprint arXiv:2601.11868*, 2026. URL https://arxiv.org/abs/2601.11868.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for General AI Assistants. *arXiv preprint arXiv:2311.12983*, 2023. URL https://arxiv.org/abs/2311.12983.

Ludovico Mitchener, Jon M Laurent, Alex Andonian, et al. BixBench: a Comprehensive Benchmark for LLM-based Agents in Computational Biology. *arXiv preprint arXiv:2503.00096*, 2025. URL https://arxiv.org/abs/2503.00096.

Deepak Nathani, Lovish Madaan, Nicholas Roberts, et al. MLGym: A New Framework and Benchmark for Advancing AI Research Agents. *arXiv preprint arXiv:2502.14499*, 2025. URL https://arxiv.org/abs/2502.14499.

Odhran O'Donoghue, Aleksandar Shtedritski, John Ginger, et al. BioPlanner: Automatic Evaluation of LLMs on Protocol Planning in Biology. *arXiv preprint arXiv:2310.10632*, 2023. URL https://arxiv.org/abs/2310.10632.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM Evaluators Recognize and Favor Their Own Generations. *arXiv preprint arXiv:2404.13076*, 2024. URL https://arxiv.org/abs/2404.13076.

Long Phan, Alice Gatti, Nathaniel Li, et al. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649(8099):1139–1146, 2026. doi: 10.1038/s41586-025-09962-4. URL https://doi.org/10.1038/s41586-025-09962-4. Introduces Humanity's Last Exam (HLE); preprint arXiv:2501.14249 (2025).

Inioluwa Deborah Raji, Emily M. Bender, Amandalynne Paullada, Emily Denton, and Alex Hanna. AI and the Everything in the Whole Wide World Benchmark. *arXiv preprint arXiv:2111.15366*, 2021. URL https://arxiv.org/abs/2111.15366.

David Rein, Betty Li Hou, Asa Cooper Stickland, et al. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. *arXiv preprint arXiv:2311.12022*, 2023. URL https://arxiv.org/abs/2311.12022.

Manasi Sharma, Chen Bo Calvin Zhang, Chaithanya Bandi, et al. ResearchRubrics: A Benchmark of Prompts and Rubrics for Evaluating Deep Research Agents. *arXiv preprint arXiv:2511.07685*, 2025. URL https://arxiv.org/abs/2511.07685.

Chuhan Shi, Xiaoquan Ren, Sicheng Song, Haobo Li, Rui Sheng, and Yushi Sun. Are LLMs Ready for Scientific Discovery? A Capability-Oriented Benchmark for AI Scientists. *arXiv preprint arXiv:2607.11079*, 2026. URL https://arxiv.org/abs/2607.11079.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the Judges: A Systematic Study of Position Bias in LLM-as-a-Judge. *arXiv preprint arXiv:2406.07791*, 2024. URL https://arxiv.org/abs/2406.07791.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can LLMs Generate Novel Research Ideas? A Large-Scale Human Study with 100+ NLP Researchers. *arXiv preprint arXiv:2409.04109*, 2024. URL https://arxiv.org/abs/2409.04109.

Chenglei Si, Tatsunori Hashimoto, and Diyi Yang. The Ideation-Execution Gap: Execution Outcomes of LLM-Generated versus Human Research Ideas. *arXiv preprint arXiv:2506.20803*, 2025. URL https://arxiv.org/abs/2506.20803.

Zachary S. Siegel, Sayash Kapoor, Nitya Nadgir, Benedikt Stroebl, and Arvind Narayanan. CORE-Bench: Fostering the Credibility of Published Research Through a Computational Reproducibility Agent Benchmark. *arXiv preprint arXiv:2409.11363*, 2024. URL https://arxiv.org/abs/2409.11363.

Shivalika Singh, Yiyang Nan, Alex Wang, et al. The Leaderboard Illusion. *arXiv preprint arXiv:2504.20879*, 2025. URL https://arxiv.org/abs/2504.20879.

Aditya Sivakumar, Ashu Singhal, Nicholas Larus-Stone, and Nithin Parsan. BenchBench-Protocol: Evaluating Real-World Wet-Lab Protocol Reasoning and Modification. Technical report, Benchling, aug 2026. URL https://www.benchling.com/blog/can-llms-work-in-the-wet-lab.

Zhangde Song, Jieyu Lu, Yuanqi Du, et al. Evaluating Large Language Models in Scientific Discovery. *arXiv preprint arXiv:2512.15567*, 2025. URL https://arxiv.org/abs/2512.15567.

Rebecca Soskin Hicks, Mikhail Trofimov, Dominick Lim, et al. HealthBench Professional: Evaluating Large Language Models on Real Clinician Chats. *arXiv preprint arXiv:2604.27470*, 2026. URL https://arxiv.org/abs/2604.27470.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, et al. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022. URL https://arxiv.org/abs/2206.04615.

Giulio Starace, Oliver Jaffe, Dane Sherburn, et al. PaperBench: Evaluating AI's Ability to Replicate AI Research. *arXiv preprint arXiv:2504.01848*, 2025. URL https://arxiv.org/abs/2504.01848.

Liangtai Sun, Yang Han, Zihan Zhao, et al. SciEval: A Multi-Level Large Language Model Evaluation Benchmark for Scientific Research. *arXiv preprint arXiv:2308.13149*, 2023. URL https://arxiv.org/abs/2308.13149.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, et al. Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models. *arXiv preprint arXiv:2404.18796*, 2024. URL https://arxiv.org/abs/2404.18796.

Miles Wang, Robi Lin, Kat Hu, et al. FrontierScience: Evaluating AI's Ability to Perform Expert-Level Scientific Tasks. *arXiv preprint arXiv:2601.21165*, 2026. URL https://arxiv.org/abs/2601.21165.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, et al. SciBench: Evaluating College-Level Scientific Problem-Solving Abilities of Large Language Models. *arXiv preprint arXiv:2307.10635*, 2023. URL https://arxiv.org/abs/2307.10635.

Hjalmar Wijk, Tao Lin, Joel Becker, et al. RE-Bench: Evaluating frontier AI R&D capabilities of language model agents against human experts. *arXiv preprint arXiv:2411.15114*, 2024. URL https://arxiv.org/abs/2411.15114.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. $\tau$-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. *arXiv preprint arXiv:2406.12045*, 2024. URL https://arxiv.org/abs/2406.12045.

Jiayi Ye, Yanbo Wang, Yue Huang, et al. Justice or Prejudice? Quantifying Biases in LLM-as-a-Judge. *arXiv preprint arXiv:2410.02736*, 2024. URL https://arxiv.org/abs/2410.02736.

Jinwei Zhang, Xucheng Liang, Yu Zhang, Ruijie Yu, Xiaokang Yang, Yaohui Jin, and Yanyan Xu. ChemReason-Bench: Benchmarking Large Language Models for Procedural Reasoning in Experimental Chemistry. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33211–33248, San Diego, California, United States, jul 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1535. URL https://aclanthology.org/2026.acl-long.1535/.

Weichen Zhang, Yiyou Sun, Pohao Huang, Jiayue Pu, Heyue Lin, and Dawn Song. MIRAGE-Bench: LLM Agent is Hallucinating and Where to Find Them. *arXiv preprint arXiv:2507.21017*, 2025. URL https://arxiv.org/abs/2507.21017.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. WildChat: 1M ChatGPT Interaction Logs in the Wild. *arXiv preprint arXiv:2405.01470*, 2024. URL https://arxiv.org/abs/2405.01470.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. LMSYS-Chat-1M: A Large-Scale Real-World LLM Conversation Dataset. *arXiv preprint arXiv:2309.11998*, 2023a. URL https://arxiv.org/abs/2309.11998.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *arXiv preprint arXiv:2306.05685*, 2023b. URL https://arxiv.org/abs/2306.05685.

# A Appendix

## A.1 Scoring rubric v1.0

The judge-facing rubric is reproduced below in full, matching version 1.0 (2026-08-07).

**Scoring Rubric — Agentic Scientific Task Benchmark**

**Version: 1.0** (2026-08-07)

You are evaluating the work of an anonymous AI research agent on a real scientific task submitted by a real user. The agent ran in a sandbox with shell access, file tools, and web search, and produced a final answer plus (possibly) output files.

*How to score*

- Every dimension is an **integer 0–10**. Written anchors are given at 0, 3, 5, 8, 10; interpolate for in-between scores.
- **Scores must discriminate.** The purpose of this evaluation is to measure how far current models are from expert-level scientific work. A score of **8 means a domain scientist would accept the work with minor edits**. Reserve 9–10 for genuinely publishable, expert-grade output. Do not cluster scores in the 6–7 comfort zone: if the work has real gaps, score it in the 3–5 range; if it is superficial or wrong, score lower still.
- Judge what was **actually delivered**, not what was promised. A plan to do an analysis is not an analysis. A script that was never run produces no results.
- Ground every score in evidence: the task prompt, the final answer, the transcript digest, and the artifacts you inspect. Spot-check claims against artifacts wherever possible (e.g., does the number quoted in the answer appear in the results file?).
- If the run was cut off mid-execution (a truncation banner will say so), score the work that exists — do not extrapolate credit for what might have followed — and tag `truncated_run`.

*Dimensions*

**1. `task_fulfillment` — Did it do what was asked?** Coverage of every explicit request and every reasonable implicit requirement, at the depth the user asked for.

- **10** — Every explicit and reasonable implicit requirement fully met at the requested depth; nothing the user would need to ask again for.
- **8** — All major requirements met; at most one minor sub-request shallow or missing.
- **5** — The core question is addressed, but sub-requests are dropped, depth is below what was asked, or a deliverable (e.g., "make me a PPT/figure/table") is missing.
- **3** — Only a fraction of the request addressed; major deliverables absent.
- **0** — Off-task, no substantive answer, or answered a different question.

**2. `scientific_accuracy` — Is the science right?** Correctness of scientific claims, methods, statistics, units, formulas, and citations.

- **10** — Methodologically defensible throughout; claims accurate; statistics appropriate and correctly executed; citations real and relevant.
- **8** — Sound overall; minor imprecision that would not change conclusions.
- **5** — Broadly plausible but with unchecked assumptions, questionable method choices, or minor errors that a reviewer would flag.
- **3** — Material scientific errors: wrong method for the question, misused statistics, incorrect units/conversions, or misinterpreted results.
- **0** — Fabricated results or citations, pseudo-science, or fundamentally wrong.

**3. `reasoning_quality` — Was the approach intelligent?** Planning, problem decomposition, hypothesis-driven exploration, and error recovery, as visible in the transcript digest.

- **10** — Clear plan, sensible decomposition, adapts intelligently to what it finds, verifies intermediate results before building on them.
- **8** — Good plan and adaptation with occasional inefficiency.
- **5** — Some structure but linear/mechanical; misses obvious checks; recovers from errors slowly or by trial-and-error.
- **3** — Little visible planning; flails between approaches; builds on unverified intermediate results.
- **0** — Incoherent; no discernible strategy.

**4. `tool_use` — Were the tools used competently?** Effective use of shell, file tools, and web search: right tool for the job, efficient sequences, graceful recovery from failures.

- **10** — Fluent: efficient commands, sensible environment setup, quick diagnosis and recovery from failures, no wasted cycles.
- **8** — Competent with minor waste (redundant reads, an avoidable dead end).
- **5** — Gets there but inefficiently: repeated failed commands with small tweaks, clumsy environment management, ignores informative error messages.
- **3** — Substantial thrashing: loops of near-identical failing commands, abandons tools that would have worked, fights the environment instead of adapting.
- **0** — Tool use actively counterproductive or essentially absent when clearly needed.

**5. `data_handling` — Were the user's files used correctly?** Whether attached files were actually loaded, parsed correctly, sanity-checked, and faithfully represented. **N/A if the task had no attachments and needed no data.**

- **10** — All attachments loaded and parsed correctly; contents sanity-checked (shapes, ranges, missingness); analysis faithful to the actual data.
- **8** — Data used correctly; light on sanity checks.
- **5** — Data loaded but partially used, or used without verification; minor misreadings that don't invalidate the main result.
- **3** — Attachments ignored, misparsed, or replaced with assumed/simulated stand-ins without telling the user.
- **0** — Claims to have analyzed data it never opened; invents columns or values.

**6. `artifact_quality` — Are the output files any good?** Completeness, correctness, and usefulness of files produced (reports, figures, tables, code, processed data). **N/A if a prose answer is the natural deliverable and none were expected.**

- **10** — Artifacts are complete, correct, well-organized, and directly usable; code runs as-is; figures/tables are publication-quality.
- **8** — Usable with minor cleanup; code essentially correct.
- **5** — Artifacts exist but are incomplete, partially broken, or need real work to use; or key promised artifacts are missing while others exist.
- **3** — Mostly broken, placeholder, or trivial artifacts; promised deliverables absent.
- **0** — No artifacts despite the task clearly requiring them, or artifacts are misleading (e.g., a "results" file of fabricated numbers).

**7. `communication` — Is the answer well communicated?** Structure, clarity, appropriate length, and **language match: the response must be in the language of the user's prompt** (you must report this explicitly).

- **10** — Clear, well-structured, right length and register for the request; in the prompt's language; a scientist could act on it immediately.
- **8** — Clear and complete with minor verbosity or structural noise.
- **5** — Understandable but bloated, poorly organized, or padded with boilerplate; or partially in the wrong language.
- **3** — Hard to follow; buries the answer; wrong language for substantial portions.
- **0** — Incoherent or entirely in the wrong language.

**8. `honesty_calibration` — Is the agent honest about what it did?** Absence of hallucination and overclaiming; failures acknowledged rather than papered over; limitations stated.

- **10** — Every claim traceable to work actually done; failures and limitations stated plainly; uncertainty appropriately flagged.

- **8** — Honest overall; minor unflagged uncertainty.
- **5** — Overstates polish or completeness; glosses over steps that failed; presents assumptions as facts.
- **3** — Presents placeholder or simulated numbers as computed results; claims success on visibly failed steps.
- **0** — Systematic fabrication: invented results, citations, or a false narrative of what was done.

*Overall score and success flag*

- `overall` (0–10) — Holistic quality of this run as a response to this user's request. **Not an average** of the dimensions: weight what mattered most for this particular task.
- `fully_successful` (boolean) — Would the user who submitted this task be satisfied with this response **without needing any follow-up**? Apply a demanding standard: this is the bar of a paying scientist-user, not a benevolent grader.

*Failure-mode taxonomy*

Tag every failure mode that applies (empty list if none). Use only these tags:

**Table 3. Failure-mode taxonomy** from rubric v1.0. Judges apply every tag that fits and may apply none.

| Tag | Definition |
|---|---|
| `premature_completion` | Stopped and declared done while major work remained. |
| `fabricated_results` | Presented numbers/findings that were never computed (placeholders, invented values, simulated data passed off as real). |
| `fabricated_citations` | Cited papers, datasets, or sources that don't exist or don't support the claim. |
| `ignored_attachments` | User-provided files were not opened or not used when the task required them. |
| `misread_data` | Files were opened but parsed or interpreted incorrectly (wrong columns, wrong units, wrong sheet). |
| `tool_thrashing` | Extended loops of near-identical failing commands with no strategy change. |
| `environment_failure_unrecovered` | A missing package/dependency/resource blocked progress and the agent never found a workaround. |
| `wrong_language` | Response not in the language of the user's prompt (substantially). |
| `truncated_run` | The run was cut off before the agent finished (use with the truncation banner). |
| `scope_drift` | Did substantial work the user didn't ask for while neglecting what they did ask for. |
| `missing_artifacts` | Promised or clearly-required output files were not produced. |
| `statistical_malpractice` | Wrong test, p-hacking, invalid multiple-comparison handling, misused models, uninterpretable statistics presented as valid. |
| `shallow_analysis` | Superficial treatment where the task demanded depth (e.g., generic textbook answer to a specific data question). |
| `overclaiming` | Final answer overstates quality, completeness, or certainty of what was done. |
| `format_noncompliance` | Ignored an explicit format request (file type, structure, template, length). |
| `other` | Anything else — must be explained in the summary. |

*Confidence*

Report `confidence` in $[0, 1]$: how confident you are in your own scores given what you could inspect. Lower it when artifacts were too large to verify, the run was truncated, or the domain is outside what you could check.

## A.2 Results and judge tables

The tables below support Sections 4 and 5. They are numbered in the order they are cited in the main text.

**Table 4. Headline results by model.** Overall is the mean across 178 tasks of the three-judge mean, with a 95% percentile bootstrap interval over sessions. The three judge columns give the same quantity computed from that judge alone. Majority success requires more than half of the three judges to independently mark the run fully successful; unanimous requires all three. "Scores ≥ 8" is the share of that model's individual scored judgments at or above the acceptable line. Computed from `scores_wide.csv`.

| Model | Overall (95% CI) | gpt-5.6-sol | qwen3.8-max | grok-4.5 | Majority | Unanimous | Scores ≥ 8 |
|---|---|---|---|---|---|---|---|
| gpt-5.6-sol | 8.04 [7.80, 8.23] | 7.90 | 8.20 | 8.01 | 71% | 56% | 89% |
| claude-opus-5 | 7.61 [7.40, 7.82] | 6.40 | 8.29 | 8.15 | 75% | 23% | 79% |
| gpt-5.6-luna | 7.46 [7.23, 7.68] | 7.17 | 7.71 | 7.49 | 60% | 42% | 77% |
| kimi-k3 | 7.17 [6.95, 7.38] | 6.15 | 7.81 | 7.54 | 52% | 15% | 69% |
| grok-4.5 | 6.96 [6.74, 7.17] | 6.15 | 7.46 | 7.26 | 46% | 20% | 64% |
| gemini-3.6-flash | 5.84 [5.60, 6.06] | 4.80 | 6.59 | 6.13 | 25% | 7% | 36% |
| muse-spark-1.2 | 4.27 [3.91, 4.65] | 3.76 | 4.52 | 4.54 | 17% | 4% | 25% |
| gemma-4-31b-it | 3.86 [3.57, 4.15] | 3.61 | 3.94 | 4.02 | 6% | 2% | 16% |
| nemotron-3-ultra-550b-a55b | 2.78 [2.38, 3.14] | 2.42 | 2.93 | 3.00 | 9% | 3% | 14% |

**Table 5. Score distribution by judge**, counts over the eight rubric dimensions plus the holistic overall, N/A excluded. Computed from `scores_wide.csv`.

| Judge | n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | ≥ 8 | < 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| gpt-5.6-sol | 13,271 | 631 | 400 | 500 | 792 | 1,117 | 1,179 | 1,481 | 2,010 | 2,908 | 2,014 | 239 | 38.9% | 25.9% |
| grok-4.5 | 13,240 | 424 | 410 | 448 | 548 | 611 | 855 | 859 | 1,527 | 4,109 | 3,215 | 234 | 57.1% | 18.4% |
| qwen3.8-max | 13,423 | 432 | 442 | 345 | 468 | 495 | 818 | 947 | 1,266 | 3,434 | 4,531 | 245 | 61.2% | 16.3% |

## A.3 Supporting tables

**Table 6. Rubric dimensions, hardest to easiest.** Mean is pooled over all models, judges and tasks. Percentages use non-N/A denominators, so **n** is the applicable count for that dimension and the two conditional dimensions are scored only where they applied. Per-model columns use abbreviated names, left to right in leaderboard order. Computed from `scores_wide.csv`.

| Dimension | Mean | ≥ 8 | < 5 | n | sol | opus | luna | kimi | grok | gemini | muse | gemma | nemotron |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Artifact quality | 5.50 | 40.6% | 32.4% | 3,094 | 7.81 | 7.62 | 6.76 | 6.95 | 6.78 | 5.70 | 3.07 | 2.09 | 1.38 |
| Scientific accuracy | 6.22 | 42.7% | 23.5% | 4,806 | 8.12 | 7.31 | 7.64 | 6.98 | 6.85 | 5.49 | 5.22 | 4.69 | 3.62 |
| Task fulfillment | 6.41 | 51.2% | 25.2% | 4,806 | 8.23 | 8.12 | 7.78 | 7.62 | 7.45 | 6.77 | 4.64 | 4.12 | 2.94 |
| Reasoning quality | 6.68 | 48.8% | 19.5% | 4,806 | 8.48 | 8.45 | 7.97 | 7.65 | 7.38 | 6.27 | 5.28 | 4.29 | 4.39 |
| Tool use | 6.79 | 53.8% | 16.2% | 4,806 | 8.07 | 8.46 | 7.61 | 7.84 | 7.75 | 6.90 | 5.00 | 4.32 | 5.13 |
| Data handling | 7.14 | 59.8% | 13.5% | 3,198 | 8.66 | 8.47 | 8.18 | 8.02 | 7.57 | 6.80 | 6.36 | 4.17 | 5.04 |
| Honesty / calibration | 7.30 | 59.7% | 13.1% | 4,806 | 9.10 | 8.00 | 8.78 | 7.88 | 7.79 | 5.41 | 6.48 | 5.96 | 6.29 |
| Communication | 7.33 | 70.7% | 11.9% | 4,806 | 8.80 | 8.38 | 8.51 | 8.35 | 8.31 | 7.76 | 5.43 | 6.54 | 3.85 |

**Table 7. Failure-mode frequencies, as a percentage of judged runs (assessments).** Tags are not exclusive. Computed from the `failure_modes` field of `scores_wide.csv`.

| Failure mode | All | gpt-5.6-sol | claude-opus-5 | gpt-5.6-luna | kimi-k3 | grok-4.5 | gemini-3.6-flash | muse-spark-1.2 | gemma-4-31b | nemotron-3 |
|---|---|---|---|---|---|---|---|---|---|---|
| overclaiming | 31.4 | 6.4 | 32.6 | 9.7 | 31.8 | 27.5 | 68.2 | 34.6 | 44.2 | 27.3 |
| missing_artifacts | 22.6 | 6.4 | 12.4 | 9.7 | 13.5 | 11.0 | 12.2 | 39.5 | 40.3 | 58.1 |
| shallow_analysis | 17.7 | 2.6 | 0.9 | 5.6 | 8.4 | 11.0 | 30.0 | 16.3 | 60.5 | 23.8 |
| truncated_run | 16.4 | 0.0 | 4.5 | 2.8 | 2.8 | 11.2 | 1.1 | 52.8 | 7.9 | 64.6 |
| premature_completion | 12.5 | 4.9 | 3.6 | 7.7 | 5.2 | 6.0 | 8.4 | 15.2 | 40.4 | 21.5 |
| statistical_malpractice | 6.6 | 1.5 | 7.9 | 1.9 | 6.9 | 9.2 | 16.3 | 6.4 | 7.1 | 2.1 |
| fabricated_results | 5.5 | 0.4 | 3.0 | 0.4 | 2.2 | 4.7 | 21.3 | 5.8 | 9.6 | 2.2 |
| tool_thrashing | 5.1 | 0.6 | 0.0 | 0.6 | 0.0 | 0.0 | 2.8 | 30.5 | 0.2 | 11.4 |
| format_noncompliance | 5.0 | 2.1 | 4.1 | 3.9 | 3.9 | 3.2 | 7.7 | 5.8 | 10.9 | 3.7 |
| other | 4.4 | 1.9 | 4.7 | 5.2 | 4.7 | 3.4 | 6.7 | 2.8 | 5.2 | 4.7 |
| fabricated_citations | 2.9 | 0.4 | 0.9 | 0.6 | 2.2 | 2.4 | 11.2 | 2.8 | 2.8 | 2.8 |
| ignored_attachments | 2.7 | 0.4 | 0.0 | 0.7 | 0.2 | 0.6 | 3.0 | 2.2 | 12.7 | 4.7 |
| environment_failure_unrecovered | 2.2 | 1.5 | 1.3 | 2.1 | 1.5 | 0.9 | 1.1 | 2.1 | 4.5 | 4.5 |
| misread_data | 2.1 | 0.9 | 1.1 | 0.6 | 2.2 | 1.5 | 5.6 | 0.7 | 3.6 | 2.2 |
| wrong_language | 1.1 | 0.0 | 1.1 | 0.2 | 0.0 | 0.0 | 0.7 | 0.4 | 1.7 | 6.0 |
| scope_drift | 0.4 | 0.6 | 0.2 | 0.6 | 0.0 | 0.0 | 0.7 | 0.0 | 0.6 | 1.1 |

**Table 8. Share of runs that used each tool at least once (%).** Computed from the `tool_calls_by_tool` field of `run_metrics.csv` over all 1,602 runs.

| Tool | All | sol | opus | luna | kimi | grok | gemini | muse | gemma | nemotron |
|---|---|---|---|---|---|---|---|---|---|---|
| bash | 75 | 87 | 96 | 79 | 75 | 83 | 83 | 68 | 43 | 66 |
| read | 45 | 69 | 66 | 61 | 52 | 35 | 30 | 34 | 30 | 31 |
| write | 43 | 51 | 75 | 42 | 61 | 46 | 29 | 35 | 23 | 28 |
| web_search | 38 | 66 | 60 | 53 | 38 | 30 | 25 | 28 | 16 | 26 |
| fetch_content | 26 | 66 | 40 | 49 | 23 | 20 | 2 | 13 | 4 | 12 |
| edit | 24 | 44 | 48 | 40 | 33 | 21 | 11 | 2 | 13 | 3 |
| get_search_content | 18 | 53 | 31 | 33 | 11 | 16 | 1 | 9 | 3 | 6 |
| source_check | 9 | 34 | 30 | 10 | 10 | 1 | 0 | 1 | 0 | 0 |

**Table 9. Deliverables on disk versus outcome.** Left: runs grouped by whether any output file exists. Right: runs grouped by file count. Computed by joining `run_metrics.csv` to run-level means from `scores_wide.csv`.

| Files? | n runs | Overall | Majority | File count | n runs | Overall | Majority |
|---|---|---|---|---|---|---|---|
| No | 767 | 5.26 | 37.5% | 0 | 767 | 5.26 | 37.5% |
| Yes | 835 | 6.68 | 42.4% | 1–2 | 259 | 6.87 | 45.2% |
| | | | | 3–10 | 202 | 6.49 | 40.1% |
| | | | | 11+ | 374 | 6.64 | 41.7% |

**Table 10. Paired win rate of the row model against the column model (%), same task and same judge, ties excluded.** 534 paired comparisons per cell. Computed from `scores_wide.csv`.

| | gpt-5.6-sol | claude-opus-5 | gpt-5.6-luna | kimi-k3 | grok-4.5 | gemini-3.6-flash | muse-spark-1.2 | gemma-4-31b | nemotron-3 |
|---|---|---|---|---|---|---|---|---|---|
| gpt-5.6-sol | — | 64 | 84 | 85 | 91 | 94 | 96 | 97 | 98 |
| claude-opus-5 | 36 | — | 60 | 75 | 77 | 92 | 94 | 96 | 98 |
| gpt-5.6-luna | 16 | 40 | — | 63 | 74 | 89 | 93 | 96 | 98 |
| kimi-k3 | 15 | 25 | 37 | — | 60 | 89 | 91 | 94 | 97 |
| grok-4.5 | 9 | 23 | 26 | 40 | — | 86 | 93 | 94 | 97 |
| gemini-3.6-flash | 6 | 8 | 11 | 11 | 14 | — | 74 | 90 | 92 |
| muse-spark-1.2 | 4 | 6 | 7 | 9 | 7 | 26 | — | 57 | 75 |
| gemma-4-31b-it | 3 | 4 | 4 | 6 | 6 | 10 | 43 | — | 70 |
| nemotron-3-ultra-550b-a55b | 2 | 2 | 2 | 3 | 3 | 8 | 25 | 30 | — |

**Table 11. Bradley-Terry strengths fitted from the paired outcomes** (log scale, centered, 95% bootstrap CI over sessions), and mean paired score delta against the leader with win and loss shares. Strengths are identified up to an additive constant, so only differences are interpretable.

| | Bradley-Terry | | | Versus `gpt-5.6-sol` | | | |
|---|---|---|---|---|---|---|---|
| **Model** | **Strength** | **CI low** | **CI high** | **Mean Δ** | **CI** | **Wins** | **Losses** |
| `gpt-5.6-sol` | +1.73 | +1.54 | +1.90 | — | — | — | — |
| `claude-opus-5` | +1.31 | +1.17 | +1.45 | −0.42 | [−0.64, −0.20] | 0.22 | 0.39 |
| `gpt-5.6-luna` | +1.02 | +0.87 | +1.16 | −0.58 | [−0.78, −0.38] | 0.10 | 0.50 |
| `kimi-k3` | +0.71 | +0.58 | +0.86 | −0.87 | [−1.07, −0.66] | 0.11 | 0.61 |
| `grok-4.5` | +0.54 | +0.42 | +0.65 | −1.08 | [−1.30, −0.87] | 0.06 | 0.67 |
| `gemini-3.6-flash` | −0.44 | −0.56 | −0.32 | −2.20 | [−2.43, −1.96] | 0.05 | 0.83 |
| `muse-spark-1.2` | −1.19 | −1.36 | −1.00 | −3.76 | [−4.15, −3.37] | 0.04 | 0.89 |
| `gemma-4-31b-it` | −1.56 | −1.73 | −1.38 | −4.18 | [−4.49, −3.85] | 0.03 | 0.93 |
| `nemotron-3-ultra-550b-a55b` | −2.12 | −2.36 | −1.90 | −5.25 | [−5.65, −4.85] | 0.02 | 0.94 |

**Table 12. Paired win rate against `gpt-5.6-sol` by rubric dimension (%), same task and same judge, ties excluded.** For the two conditional dimensions, pairs in which either run was marked not-applicable are dropped, so those rows rest on fewer decisive pairs than the other six. Computed from `scores_wide.csv`.

| **Dimension** | `opus` | `luna` | `kimi` | `grok` | `gemini` | `muse` | `gemma` | `nemotron` |
|---|---|---|---|---|---|---|---|---|
| Tool use | **73.2** | 26.2 | 38.4 | 28.1 | 12.9 | 8.2 | 3.3 | 4.5 |
| Reasoning quality | 49.1 | 17.1 | 13.7 | 6.7 | 4.7 | 3.4 | 2.3 | 2.0 |
| Task fulfillment | 46.1 | 18.8 | 18.6 | 15.0 | 11.4 | 6.4 | 2.8 | 2.0 |
| Data handling | 44.6 | 12.1 | 11.9 | 9.0 | 5.8 | 2.1 | 0.8 | 0.6 |
| Artifact quality | 40.3 | 13.2 | 17.6 | 12.3 | 8.2 | 2.5 | 1.6 | 1.3 |
| Communication | 30.0 | 17.1 | 20.6 | 13.2 | 5.3 | 2.9 | 2.1 | 2.5 |
| Scientific accuracy | 22.5 | 18.6 | 11.1 | 5.9 | 3.0 | 2.1 | 1.8 | 1.6 |
| Honesty / calibration | 17.2 | 22.2 | 7.5 | 6.8 | 0.6 | 1.9 | 4.6 | 2.3 |

**Table 13. Mean overall by domain and model.** Computed from `scores_wide.csv` joined to the domain field of `run_metrics.csv`.

| **Domain** | **n tasks** | `sol` | `opus` | `luna` | `kimi` | `grok` | `gemini` | `muse` | `gemma` | `nemotron` |
|---|---|---|---|---|---|---|---|---|---|---|
| Chemistry, drug, materials | 17 | 8.33 | 7.75 | 7.69 | 7.02 | 6.94 | 5.57 | 4.00 | 3.18 | 2.75 |
| Clinical and health | 59 | 7.80 | 7.61 | 7.57 | 7.20 | 7.14 | 6.08 | 4.40 | 3.95 | 3.33 |
| Life sciences | 59 | 8.21 | 7.81 | 7.42 | 7.14 | 6.79 | 5.52 | 4.14 | 3.56 | 2.72 |
| Physical sciences, eng., CS | 43 | 8.00 | 7.30 | 7.26 | 7.22 | 6.94 | 6.05 | 4.39 | 4.40 | 2.13 |

**Table 14. Effect of attachments on mean overall, by model.** 125 of 178 sessions carry at least one file. Computed by joining `run_metrics.csv` to run-level means.

| Model | No attachments | Has attachments | Δ |
|---|---|---|---|
| gemma-4-31b-it | 4.93 | 3.40 | −1.52 |
| muse-spark-1.2 | 5.29 | 3.84 | −1.45 |
| nemotron-3-ultra-550b-a55b | 3.69 | 2.40 | −1.29 |
| gemini-3.6-flash | 6.23 | 5.68 | −0.56 |
| claude-opus-5 | 7.73 | 7.57 | −0.16 |
| gpt-5.6-luna | 7.52 | 7.43 | −0.09 |
| grok-4.5 | 7.00 | 6.94 | −0.06 |
| gpt-5.6-sol | 7.98 | 8.06 | +0.08 |
| kimi-k3 | 6.84 | 7.30 | +0.46 |

**Table 15. Mean overall by prompt-length quartile.** Quartiles are formed over the 178 tasks by the byte length of the first user message, which is held in the task registry rather than in the score tables (Section 7). Eight of the nine models are tabulated here for width; `gemma-4-31b-it` is plotted alongside them in Figure 13 and also scores lower on Q4 than on Q1.

| Bin | Median bytes | n | sol | opus | luna | kimi | grok | gemini | muse | nemotron |
|---|---|---|---|---|---|---|---|---|---|---|
| Q1 | 96 | 45 | 8.31 | 7.96 | 7.76 | 7.18 | 7.49 | 6.38 | 5.47 | 4.87 |
| Q2 | 284 | 44 | 8.23 | 7.55 | 7.52 | 7.24 | 7.04 | 5.91 | 4.25 | 2.70 |
| Q3 | 1,232 | 45 | 8.09 | 7.70 | 7.55 | 7.23 | 6.79 | 5.82 | 3.66 | 2.02 |
| Q4 | 6,217 | 44 | 7.50 | 7.23 | 6.99 | 7.02 | 6.49 | 5.24 | 3.69 | 1.51 |
| Q1 → Q4 drop | | | 0.8 | 0.7 | 0.8 | 0.2 | 1.0 | 1.1 | 1.8 | 3.4 |

**Table 16. Run endings and outcomes.** Left: mean overall score and majority-success rate by final stop reason. Right: truncation rate by model, with that model's mean overall for reference. Computed from `run_metrics.csv` joined to run-level means.

| Ending | n | Overall | Majority | Model | Truncated | Overall |
|---|---|---|---|---|---|---|
| stop | 1,354 | 6.70 | 47.4% | nemotron-3-ultra-550b-a55b | 64.6% | 2.78 |
| length | 224 | 2.22 | 0.0% | muse-spark-1.2 | 52.8% | 4.27 |
| toolUse | 10 | 3.60 | 0.0% | grok-4.5 | 11.2% | 6.96 |
| error | 14 | 0.00 | 0.0% | gemma-4-31b-it | 7.9% | 3.86 |
| Not truncated | 1,344 | 6.73 | 47.8% | claude-opus-5 | 3.9% | 7.61 |
| Truncated | 258 | 2.18 | 0.0% | kimi-k3 | 2.8% | 7.17 |
| | | | | gpt-5.6-luna | 1.7% | 7.46 |
| | | | | gpt-5.6-sol | 0.0% | 8.04 |
| | | | | gemini-3.6-flash | 0.0% | 5.84 |

**Table 17. Run-level Spearman correlates of overall score** ($n$ = 1,602). Computed from `run_metrics.csv` joined to run-level means.

| Variable | $\rho$ | Variable | $\rho$ |
|---|---|---|---|
| Input tokens | −0.42 | Turns | +0.23 |
| Thinking characters | +0.35 | Output files | +0.21 |
| Wall-clock seconds | +0.33 | Tool error rate | −0.16 |
| Cost (USD) | +0.31 | Attachment count | −0.16 |
| Tool calls | +0.27 | | |

**Table 18. Cost and efficiency.** Mean and median inference cost per task, total campaign cost, and two efficiency ratios. Computed from `run_metrics.csv`; total generation cost across all 1,602 runs was $3,649.18.

| Model | Overall | Mean $ | Median $ | Total $ | Majority | Score/$ | Maj. pts/$ |
|---|---|---|---|---|---|---|---|
| `gpt-5.6-sol` | 8.04 | 8.51 | 4.72 | 1,514.92 | 71% | 0.9 | 8.4 |
| `claude-opus-5` | 7.61 | 6.69 | 4.04 | 1,190.15 | 75% | 1.1 | 11.3 |
| `gpt-5.6-luna` | 7.46 | 0.15 | 0.06 | 26.80 | 60% | 49.5 | 395.6 |
| `kimi-k3` | 7.17 | 1.11 | 0.40 | 197.51 | 52% | 6.5 | 46.6 |
| `grok-4.5` | 6.96 | 0.38 | 0.23 | 67.96 | 46% | 18.2 | 119.2 |
| `gemini-3.6-flash` | 5.84 | 0.82 | 0.50 | 146.45 | 25% | 7.1 | 30.0 |
| `muse-spark-1.2` | 4.27 | 2.42 | 0.27 | 430.57 | 17% | 1.8 | 7.2 |
| `gemma-4-31b-it` | 3.86 | 0.02 | 0.00 | 3.68 | 6% | 186.4 | 298.7 |
| `nemotron-3-ultra-550b-a55b` | 2.78 | 0.40 | 0.08 | 71.14 | 9% | 7.0 | 22.5 |

**Table 19. Failure co-occurrence, $P$(column | row) in percent.** Read a row as: when this failure occurs, how often the column failure also occurs. Computed from the `failure_modes` field of `scores_wide.csv`.

| Given | `overclaim.` | `missing_art.` | `shallow` | `truncated` | `premature` | `stat. malp.` |
|---|---|---|---|---|---|---|
| `overclaiming` | 100 | 20 | 37 | 10 | 15 | 19 |
| `missing_artifacts` | 28 | 100 | 23 | 48 | 38 | 8 |
| `shallow_analysis` | 65 | 29 | 100 | 15 | 33 | 10 |
| `truncated_run` | 19 | 65 | 16 | 100 | 20 | 2 |
| `premature_completion` | 38 | 68 | 46 | 26 | 100 | 7 |
| `statistical_malpractice` | 92 | 28 | 27 | 5 | 14 | 100 |

**Table 20. Pairwise judge agreement on the holistic overall score** ($n$ = 1,602 runs). $A - B$ is the mean difference in level. Computed from `scores_wide.csv`.

| Judge A | Judge B | Spearman $\rho$ | Mean $\lvert A - B \rvert$ | Within ±1 | $A - B$ |
|---|---|---|---|---|---|
| `gpt-5.6-sol` | `qwen3.8-max` | 0.78 | 1.37 | 62.5% | −1.01 |
| `gpt-5.6-sol` | `grok-4.5` | 0.83 | 1.18 | 69.0% | −0.87 |
| `qwen3.8-max` | `grok-4.5` | 0.89 | 0.60 | 91.3% | +0.14 |

**Table 21. Judge agreement by rubric dimension.** Means over the three judge pairs, computed on runs where all three judges scored the dimension. Computed from `scores_wide.csv`; *n* is smaller for the two conditional dimensions because all three judges must have marked them applicable.

| Dimension | n runs | Mean pairwise $\rho$ | Mean \|diff\| | Within ±1 |
|---|---|---|---|---|
| Artifact quality | 971 | 0.86 | 0.92 | 77% |
| Task fulfillment | 1,602 | 0.85 | 0.87 | 81% |
| Overall | 1,602 | 0.83 | 1.05 | 74% |
| Reasoning quality | 1,602 | 0.80 | 0.95 | 78% |
| Data handling | 1,004 | 0.76 | 0.88 | 81% |
| Scientific accuracy | 1,602 | 0.76 | 1.47 | 60% |
| Communication | 1,602 | 0.76 | 0.59 | 91% |
| Tool use | 1,602 | 0.74 | 0.94 | 81% |
| Honesty / calibration | 1,602 | 0.69 | 1.32 | 65% |

**Table 22. Calibration-adjusted self-preference for the two judges that are also contestants.** All values are mean overall scores. Computed from `scores_wide.csv`.

| Judge | Scores itself | Scores others | Peers score it | Peers score others | Adjusted SP |
|---|---|---|---|---|---|
| `gpt-5.6-sol` | 7.90 | 5.06 | 8.10 | 6.09 | **+0.83** |
| `grok-4.5` | 7.26 | 6.11 | 6.80 | 5.76 | **+0.11** |

**Table 23. `gpt-5.6-sol`'s deviation from peer consensus, by model judged.** "Peer mean" is the mean of the two other judges' means for that model. A single additive strictness term would make the deviation column constant; it is not. The two smallest magnitudes belong to models the panel already places near 3, where downward deviation is bounded by the floor of the scale, which biases the pooled baseline — and hence the self-preference estimate of Table 22 — toward zero. Computed from the per-judge columns of Table 4.

| Model judged | `gpt-5.6-sol` | Peer mean | Deviation |
|---|---|---|---|
| `claude-opus-5` | 6.40 | 8.22 | −1.82 |
| `gemini-3.6-flash` | 4.80 | 6.36 | −1.56 |
| `kimi-k3` | 6.15 | 7.68 | −1.53 |
| `grok-4.5` | 6.15 | 7.36 | −1.21 |
| `muse-spark-1.2` | 3.76 | 4.53 | −0.77 |
| `nemotron-3-ultra-550b-a55b` | 2.42 | 2.97 | −0.55 |
| `gpt-5.6-luna` (same family) | 7.17 | 7.60 | **−0.43** |
| `gemma-4-31b-it` | 3.61 | 3.98 | −0.37 |
| `gpt-5.6-sol` (itself) | 7.90 | 8.11 | **−0.21** |

**Table 24. Exact systems evaluated.** All models were accessed through OpenRouter; the identifier column is the slug passed to the API and the date is the model's OpenRouter listing date, not a vendor snapshot date. Context length is the window advertised by the endpoint. All nine benchmarked models ran under the stock `pi` 0.84.0 harness in Modal sandboxes with the same tool set, with thinking level `max` requested where the model exposed one, no model-specific prompting, no sub-agents and no retries, against zero-retention endpoints (Section 7). The campaign ran between 6 and 12 August 2026. Context length does not predict truncation (Section 4.7).

| Model as named here | Provider | OpenRouter identifier | Listed | Context |
|---|---|---|---|---|
| `gpt-5.6-sol` | OpenAI | `openai/gpt-5.6-sol` | 9 Jul 2026 | 1,050,000 |
| `claude-opus-5` | Anthropic | `anthropic/claude-opus-5` | 24 Jul 2026 | 1,000,000 |
| `gpt-5.6-luna` | OpenAI | `openai/gpt-5.6-luna` | 9 Jul 2026 | 1,050,000 |
| `kimi-k3` | Moonshot AI | `moonshotai/kimi-k3` | 16 Jul 2026 | 1,048,576 |
| `grok-4.5` | xAI | `x-ai/grok-4.5` | 8 Jul 2026 | 500,000 |
| `gemini-3.6-flash` | Google | `google/gemini-3.6-flash` | 21 Jul 2026 | 1,048,576 |
| `muse-spark-1.2` | Meta | `meta/muse-spark-1.2` | 5 Aug 2026 | 1,048,576 |
| `gemma-4-31b-it` | Google | `google/gemma-4-31b-it` | 2 Apr 2026 | 262,144 |
| `nemotron-3-ultra-550b-a55b` | NVIDIA | `nvidia/nemotron-3-ultra-550b-a55b` | 4 Jun 2026 | 512,288 |
| `qwen3.8-max` (judge only) | Alibaba | `qwen/qwen3.8-max` | 3 Aug 2026 | 1,000,000 |

**Table 25. Objective run metrics by model.** Medians over 178 runs except where noted. "No output files" is the share of runs ending with an empty output tree. Computed from `run_metrics.csv`.

| Model | Med. turns | Med. tool calls | Tool error rate | Med. minutes | Med. $/task | Total $ | No output files |
|---|---|---|---|---|---|---|---|
| `gpt-5.6-sol` | 40.5 | 59.5 | 0.04 | 17.9 | 4.72 | 1,514.92 | 37.1% |
| `claude-opus-5` | 38.0 | 47.0 | 0.03 | 29.0 | 4.04 | 1,190.15 | 17.4% |
| `gpt-5.6-luna` | 30.0 | 44.0 | 0.05 | 28.1 | 0.06 | 26.80 | 47.8% |
| `kimi-k3` | 13.0 | 13.5 | 0.05 | 20.3 | 0.40 | 197.51 | 34.3% |
| `grok-4.5` | 9.0 | 12.0 | 0.09 | 4.6 | 0.23 | 67.96 | 37.6% |
| `gemini-3.6-flash` | 16.0 | 15.0 | 0.10 | 4.2 | 0.50 | 146.45 | 41.6% |
| `muse-spark-1.2` | 11.0 | 11.0 | 0.26 | 2.1 | 0.27 | 430.57 | 67.4% |
| `gemma-4-31b-it` | 2.0 | 1.0 | 0.06 | 2.1 | 0.00 | 3.68 | 73.0% |
| `nemotron-3-ultra-550b-a55b` | 7.0 | 7.0 | 0.20 | 3.3 | 0.08 | 71.14 | 74.7% |

**Table 26. Score consistency by model**, over all 4,806 assessment-level holistic scores. No model is a high-variance gambler: the standard deviations are similar across the top five, so the ranking reflects level rather than luck. Computed from `scores_wide.csv`.

| Model | Mean | SD | p10 | Median | p90 | Share $\geq 8$ | Share $\leq 4$ |
|---|---|---|---|---|---|---|---|
| `gpt-5.6-sol` | 8.04 | 1.61 | 6 | 9 | 9 | 82% | 6% |
| `claude-opus-5` | 7.61 | 1.80 | 5 | 8 | 9 | 67% | 7% |
| `gpt-5.6-luna` | 7.46 | 1.71 | 5 | 8 | 9 | 69% | 8% |
| `kimi-k3` | 7.17 | 1.79 | 5 | 8 | 9 | 60% | 9% |
| `grok-4.5` | 6.96 | 1.76 | 4 | 8 | 9 | 52% | 11% |
| `gemini-3.6-flash` | 5.84 | 1.84 | 3 | 6 | 8 | 24% | 27% |
| `muse-spark-1.2` | 4.27 | 2.65 | 1 | 4 | 8 | 15% | 53% |
| `gemma-4-31b-it` | 3.86 | 2.18 | 1 | 4 | 7 | 7% | 64% |
| `nemotron-3-ultra-550b-a55b` | 2.78 | 2.66 | 0 | 2 | 7 | 7% | 72% |

**Table 27. Conditional-dimension applicability and language compliance.** N/A shares are the proportion of assessments on which the judge marked the dimension inapplicable; language compliance is the share of assessments in which the judge recorded that the answer was in the language of the prompt. Computed from `scores_wide.csv`.

| Model | Data hand. N/A | Artifact q. N/A | Lang. OK | Condition | Data hand. N/A | Artifact q. N/A |
|---|---|---|---|---|---|---|
| `claude-opus-5` | 24.7% | 15.0% | 97.8% | No attachments | 87.4% | 58.7% |
| `gemini-3.6-flash` | 28.5% | 41.2% | 99.3% | Has attachments | 10.6% | 25.8% |
| `gemma-4-31b-it` | 49.1% | 48.5% | 95.1% | **All** | **33.5%** | **35.6%** |
| `gpt-5.6-luna` | 32.6% | 43.1% | 99.8% | | | |
| `gpt-5.6-sol` | 32.0% | 34.5% | 100.0% | | | |
| `grok-4.5` | 29.0% | 34.5% | 100.0% | | | |
| `kimi-k3` | 33.1% | 30.3% | 99.8% | | | |
| `muse-spark-1.2` | 37.1% | 41.8% | 88.4% | | | |
| `nemotron-3-ultra-550b-a55b` | 35.0% | 31.8% | 89.1% | | | |

**Table 28. Unsolved tasks and unique solvers by domain.** A task is unsolved if no model's run was called fully successful by a majority of judges; it has a unique solver if exactly one of the nine models cleared that bar. Computed from `scores_wide.csv` and `run_metrics.csv`.

| Domain | n tasks | Unsolved | % unsolved | Unique-solver tasks |
|---|---|---|---|---|
| Physical sciences, engineering, CS | 43 | 7 | 16.3% | 8 |
| Life sciences | 59 | 8 | 13.6% | 7 |
| Clinical and health | 59 | 6 | 10.2% | 6 |
| Chemistry, drug, materials | 17 | 1 | 5.9% | 2 |
| **All** | **178** | **22** | **12.4%** | **23** |

**Table 29. Outcome by tool error rate and by attachment count.** Both are run-level bins. The tool-error relationship is non-monotone: the best outcomes come from runs that attempted enough to fail occasionally and recovered. Computed from `run_metrics.csv` joined to run-level means.

| Tool error rate | n | Overall | Majority | Attachments | n | Overall | Majority |
|---|---|---|---|---|---|---|---|
| 0 | 664 | 5.95 | 42.0% | 0 | 477 | 6.36 | 50.7% |
| 0–5% | 268 | 7.47 | 59.3% | 1 | 477 | 6.28 | 45.7% |
| 5–15% | 357 | 6.42 | 40.6% | 2–3 | 333 | 5.66 | 32.4% |
| >15% | 313 | 4.36 | 18.8% | 4+ | 315 | 5.39 | 23.5% |

**Table 30. Mean overall by attached file type**, for file types appearing on at least ten tasks. "Spread" is best model minus worst model. File extensions are task-registry attributes rather than fields of the score tables (Section 7). A task attaching several formats contributes to each, so the rows are not disjoint.

| Type | n | sol | opus | luna | kimi | grok | gemini | muse | gemma | nemotron | Spread |
|---|---|---|---|---|---|---|---|---|---|---|---|
| `.docx` | 40 | 8.18 | 7.77 | 7.55 | 7.45 | 7.08 | 5.83 | 3.46 | 2.85 | 2.44 | 5.73 |
| `.pdf` | 29 | 7.93 | 7.70 | 7.44 | 7.40 | 6.84 | 5.52 | 2.52 | 2.41 | 1.78 | 6.15 |
| `.xlsx` | 18 | 8.15 | 7.59 | 7.41 | 7.19 | 6.91 | 5.98 | 4.04 | 3.33 | 2.19 | 5.96 |
| `.md` | 14 | 7.52 | 6.98 | 7.40 | 7.43 | 6.62 | 5.00 | 3.71 | 3.88 | 0.93 | 6.60 |
| `.csv` | 13 | 7.77 | 6.97 | 6.74 | 7.08 | 6.05 | 5.08 | 3.41 | 2.95 | 1.97 | 5.79 |
| `.zip` | 13 | 8.26 | 7.46 | 6.67 | 7.62 | 6.51 | 5.15 | 2.87 | 2.59 | 1.46 | 6.79 |
| `.txt` | 10 | 8.57 | 7.63 | 7.00 | 7.73 | 6.67 | 5.80 | 3.47 | 2.93 | 2.60 | 5.97 |

**Table 31. Mean overall by sampling batch.** The two batches were drawn one day apart and sampled independently; model ordering is stable across them. Computed from `run_metrics.csv` joined to run-level means.

| Model | 2026-08-06-full (93 tasks) | 2026-08-07-batch2-cpu (85 tasks) |
|---|---|---|
| gpt-5.6-sol | 8.00 | 8.08 |
| claude-opus-5 | 7.51 | 7.73 |
| gpt-5.6-luna | 7.50 | 7.41 |
| kimi-k3 | 7.02 | 7.33 |
| grok-4.5 | 6.98 | 6.93 |
| gemini-3.6-flash | 5.91 | 5.76 |
| muse-spark-1.2 | 4.79 | 3.71 |
| gemma-4-31b-it | 4.33 | 3.34 |
| nemotron-3-ultra-550b-a55b | 3.18 | 2.35 |

**Table 32. Judging operations.** All 4,806 assessments completed; 4,798 produced schema-valid scores on the first attempt, 6 required two attempts and 2 required three. Computed from `scores_wide.csv`.

| Judge | Assessments | Mean artifacts opened | Mean confidence | Mean overall awarded |
|---|---|---|---|---|
| gpt-5.6-sol | 1,602 | 3.65 | 0.92 | 5.37 |
| grok-4.5 | 1,602 | 4.22 | 0.86 | 6.24 |
| qwen3.8-max | 1,602 | 4.12 | 0.83 | 6.38 |
| **All** | **4,806** | **4.00** | **0.87** | **6.00** |

Total judging cost $996.99 over 151.1 hours of judge wall-clock time; 72 assessments (1.5%) were made with confidence below 0.7.