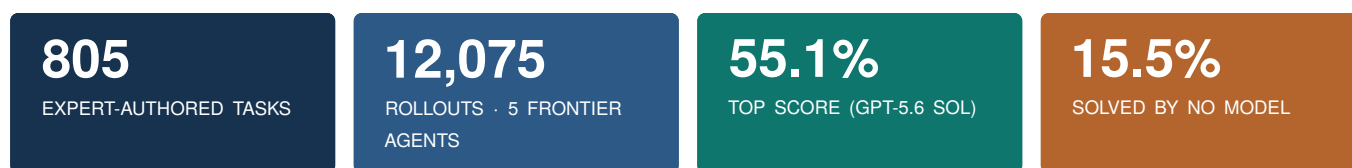


# Turing Frontier Agentic STEM (OTS)

805 expert-authored, verifier-graded terminal tasks in Physics, Mathematics, Chemistry and Bio&Medicine, for labs training and evaluating frontier research agents

Frontier Agentic STEM measures whether a frontier agent can *produce* a research-grade scientific result, not merely describe how one would be produced. Its 805 expert-authored tasks are packaged as sealed Harbor container environments spanning Physics, Mathematics, Chemistry and Bio&Medicine, each graded by a task-specific deterministic pytest verifier that recomputes the agent's outputs from raw arrays. Frontier Agentic STEM evaluates **whether an agent can satisfy a complete, machine-checked scientific output contract**, across professional workflows that take a human domain expert a mean of 58 hours.



All models run under the identical TERMINUS-2 harness, three runs per task, sealed containers. 805 tasks  $\times$  5 models  $\times$  3 runs. 12,075 rollouts were recorded; so every model is scored over all 805 tasks at  $k = 3$ . The 15.5% figure is the 125 tasks no model solved on any run.

## Introduction — what makes this data high-signal

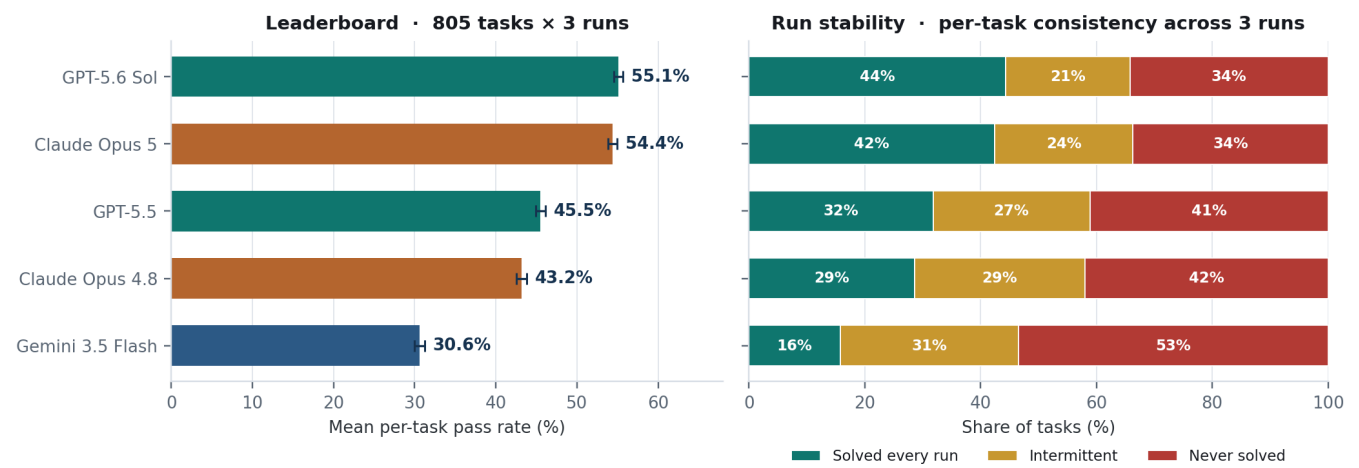
Frontier Agentic STEM is a expertise-heavy benchmark and training set for frontier agents in the natural sciences. Every task is derived from a specific piece of primary scientific literature, its governing equations, published datasets, reported numerical results, or reference algorithm, and then re-engineered by a subject-matter expert into a self-contained, terminal-native problem with a precise output contract and a deterministic verifier keyed to known-correct values. Its 805 tasks deliver four advances over existing public benchmarks.

- **Leakage-resistant construction.** The paper supplies ground truth; the expert supplies the instruction, the environment, the oracle solution and the machine-checkable contract. Nothing the agent reads is text a model may have memorized, and the verifier, oracle solution and expected values are sealed outside the agent-visible container.
- **Full-spectrum coverage.** 305 Physics, 192 Mathematics, 160 Chemistry and 148 Bio&Medicine tasks, spread across 94 distinct sub-domains from Computational and Theoretical Chemistry to Systems Biology to Inverse Problems and Uncertainty Quantification. Outside the four broad parent labels, no sub-domain exceeds 36 tasks.
- **Realistic interactive environments.** Each task ships an instruction, a Docker environment with a pinned scientific stack, raw input arrays, an oracle solution and a pytest suite. All 805 tasks disable internet access outright; timeouts are set per task. The agent must write and execute working scientific code against real data, not answer a question.
- **Robust behavioral verification.** Verifiers recompute outputs from the raw inputs rather than pattern-matching a submitted answer. The average task carries 80 assertions, with the deepest suites exceeding 4,000 parameterized checks.

Frontier Agentic STEM is a real research computation, executed end-to-end in a terminal, graded by a verifier that is deterministic, suitable for science-based tasks. Expert-time estimates are recorded on 805 tasks; those average 58 hours and total roughly 38,300 expert-hours, this is multi-day professional work.

## Leaderboard — mean per-task pass rate, 3 runs × 805 tasks

Two models separate from the field and are statistically indistinguishable from each other: GPT-5.6 Sol at **55.1%** and Claude Opus 5 at **54.4%**, a 0.7-point gap against pooled standard errors of roughly 0.6 points. The previous generation sits roughly ten points back, GPT-5.5 at 45.5% and Claude Opus 4.8 at 43.2%, and Gemini 3.5 Flash trails at 30.6%. No model is close to saturation: the leader fails 45% of its rollouts and never solves 275 tasks (34% of the suite) on any run.



**Left:** mean per-task pass rate with pooled within-task standard error. **Right:** each model's 805 tasks split by run stability, solved on all three runs, solved intermittently, or never solved. All models evaluated under the TERMINUS-2 harness for consistency.

Reliability, not accuracy, is where the field is thinnest. GPT-5.6 Sol solves 357 tasks on every run (44%) but touches 530 at least once; Claude Opus 5 solves 342 consistently and 534 at least once. That gap, 173 and 192 tasks respectively, is work the models can do but cannot reproduce. Gemini 3.5 Flash is the extreme case: 375 tasks solved at least once, but only 127 solved reliably, meaning two-thirds of its apparent capability is a coin flip.

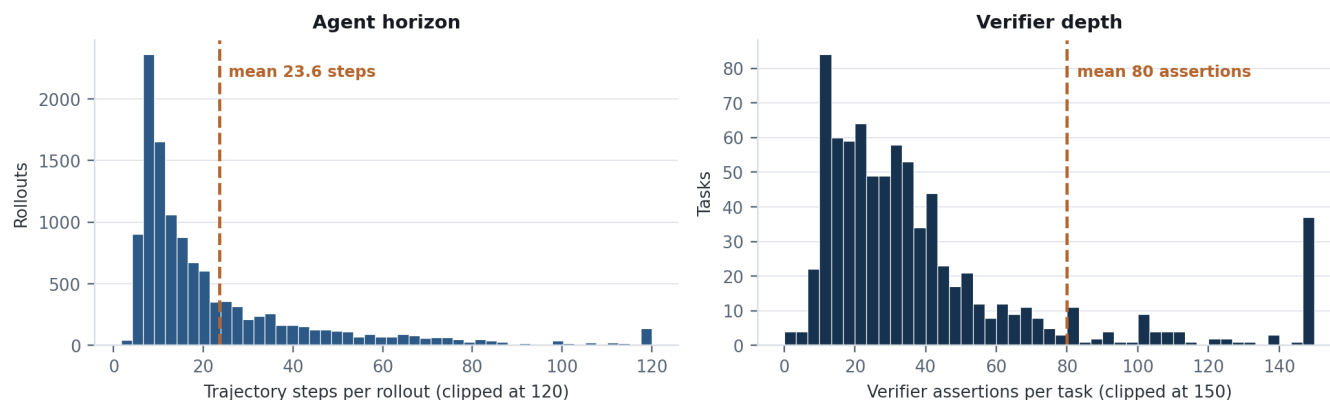
## The Output Contract — why the last assertion decides everything

Scientific capability is usually measured by whether a model can state a correct result. The work is not done that way. A research deliverable is a set of artifacts, tables with exact column names, arrays with specified dtypes, JSON certificates with specified keys, figures written to disk, computed by a deterministic pipeline from raw inputs, reproducible by a third party who reruns it. Frontier Agentic STEM grades that deliverable, and only that deliverable.

This is why agents look competent right up to the last mile, and it is what makes the set valuable. Across all failing attempts the average model still passes 82–87% of a task's assertions, and the two leaders clear 90% or more on two thirds of their failures. Even on the 125 tasks that no model solved, the average attempt cleared 77.6% of the verifier and 31% of those attempts cleared 90% or more.

## 1. Long-horizon execution grounded in observed trajectories

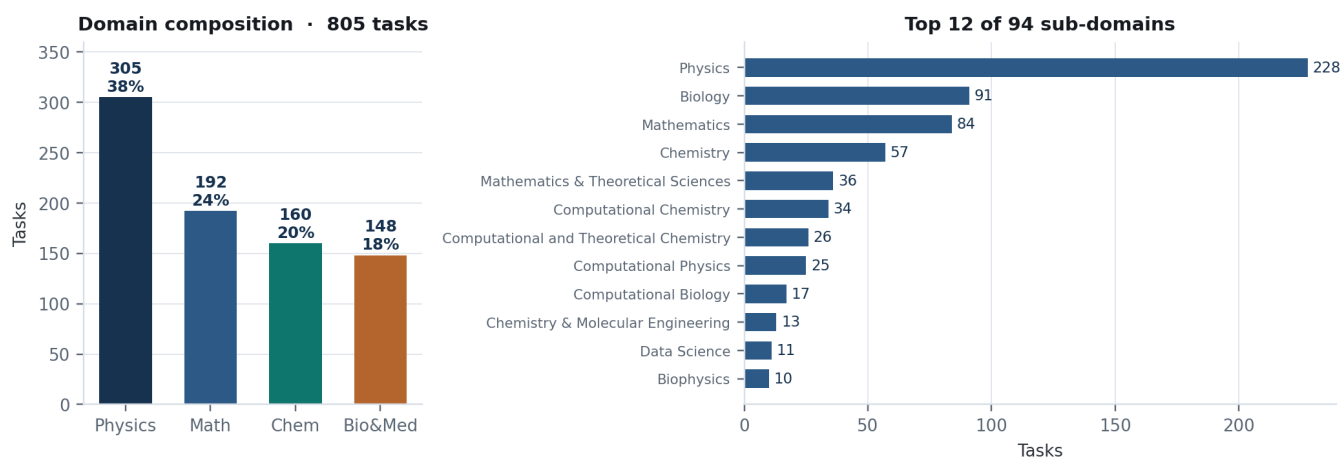
Horizon here is measured from recorded rollouts, not estimated human effort. The average rollout runs 23.6 trajectory steps over 10.2 minutes of wall-clock at \$1.45 of model spend; the distribution is right-skewed, with the 90th percentile at 53 steps and the longest single rollout at 936. Verifier depth scales with it: the average task carries 80 assertions.



**Left:** distribution of mean trajectory steps, one value per task–model–outcome group (5,090 values over 4,022 task–model cells), clipped at 120. **Right:** verifier assertions per task, clipped at 150; the tail runs to 4,202.

## 2. Distinct professional workflows across 94 sub-domains

The suite is not one template repeated. Physics contributes 305 tasks (38%), Mathematics 192 (24%), Chemistry 160 (20%) and Bio&Medicine 148 (18%), and those four parents fan out into 94 sub-domains.



**Left:** the four-way domain split of the 805 tasks. **Right:** the twelve largest sub-domains by task count; the remaining 82 hold ten tasks or fewer, and 70 of them hold three or fewer. Three sub-domains tie with Biophysics at ten tasks and fall outside the chart.

Each workflow ends in a different professional work product: a reconstructed inverse-problem solution, a certified numerical witness, an electronic-structure benchmark, an audit-grade bioinformatics pipeline. The verifier for each was written to that product, not to a shared rubric.

## 3. Leakage-resistant tasks in reproducible environments

The expert writes a fresh instruction, a fresh environment and a fresh oracle, and the verifier is keyed to values recomputed from raw inputs the agent must actually process. Held outside the agent-visible package: the pytest suite, the oracle solution, the expected numerical values, and any rubric. All containers run with internet disabled.

## 4. Verification is part of the contract

Grading is deterministic and binary. A rollout passes only when every assertion in its suite passes; there is no partial reward, so assertion counts are the only graded partial signal. Verifiers do not inspect implementation, they recompute.

## Methodology — how the data was built and validated

### Task and target selection

A candidate task had to satisfy three conditions: it had to be rooted in a specific piece of primary literature with checkable ground truth; it had to have a computationally substantive core that could be isolated and rebuilt as a self-contained problem; and it had to be expensive enough in expert time that no shortcut path exists. Published work is the starting point, never the answer key, the expert extracts a derivation to implement, a pipeline to reconstruct, or a result to reproduce within tolerance, and rebuilds it as an original problem. The resulting suite is 805 tasks: 305 Physics, 192 Mathematics, 160 Chemistry, 148 Bio&Medicine, across 94 sub-domains.

### Task construction

Every task ships four components: an `instruction.md` stating the scenario and the exact output contract; an `environment/` Docker definition with a pinned scientific stack and resource limits; a `solution/` oracle; and a `tests/` pytest suite. The agent-facing prompt withholds the oracle, the suite and every expected numerical value. A `task.toml` records the domain, tags, expert-time estimate and the agent and verifier timeouts.

| WORKFLOW                 | REQUIRED RESULT                                                                   | HOW IT IS VERIFIED                                                                          |
|--------------------------|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| Numerical reconstruction | A computed solution or reconstructed field written to disk in a specified format. | Recomputation against reference values within a stated tolerance.                           |
| Certified derivation     | A witness or certificate object whose fields encode the mathematical claim.       | Structural checks on the certificate plus independent re-derivation of its numeric entries. |
| Audit-grade pipeline     | A full artifact set, tables, JSON reports, archives, figures, from raw inputs.    | Per-artifact schema checks, cross-artifact consistency, and input-perturbation replay.      |

Test suites live outside the agent-visible tree and are mounted only at grading time.

### Quality assurance

Tasks are authored by subject-matter experts in the relevant field and validated end-to-end with the oracle solution before release. Every task was audited by reviewers independent of the author, examining:

- Prompt–verifier alignment:** the verifier tests what the instruction asks, no more and no less, every contractual string in the suite appears in the instruction.
- Acceptance breadth:** valid alternative implementations are accepted when they satisfy the same observable contract.
- Scientific realism:** the scenario represents meaningful professional research work; All 805 tasks additionally carry an expert-time estimate in `task.toml`.
- Leakage resistance:** answer keys, oracle artifacts and verifier shortcuts stay outside the agent-visible environment, and the container runs offline.
- Environment reliability:** the oracle passes deterministically, so failures reflect agent behavior rather than missing dependencies or verifier flakiness.

Grading in this release is fully deterministic. There is no judge or rubric layer in the scored path.

### Evaluation harness and scoring

We evaluated 5 frontier-agent configurations on all 805 tasks, running each 3 times per model for 2,415 trials per model. 12,075 rollouts appear in the export; Every model is scored over the same 805 tasks at  $k = 3$ . All models ran under the TERMINUS-2 harness with **high thinking mode**, except Gemini 3.5 flash and GPT 5.5 which are at medium thinking mode, against freshly built containers, with each task’s own agent and verifier timeouts. A task’s pass rate for a model is the fraction of its 3 runs that pass the full verifier. The leaderboard gives every task equal weight and averages these per-task pass rates:

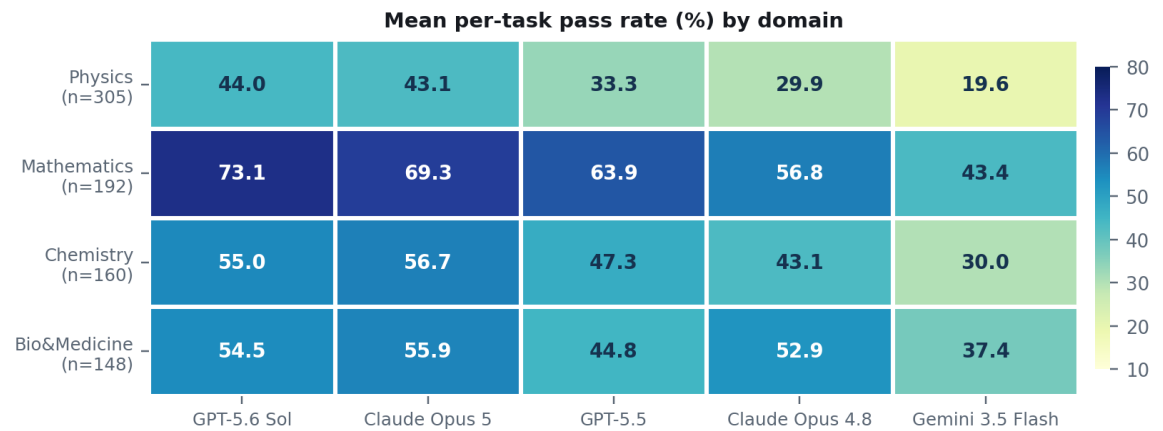
$$\text{Score}(m) = \frac{1}{n} \sum_{t=1}^n \hat{p}_{m,t}, \quad \hat{p}_{m,t} = \frac{1}{k} \sum_{r=1}^k y_{m,t,r}, \quad k = 3, \, n = 805,$$

which equals the model’s overall share of passing trials. A task is **consistently solved** when all three runs pass, **intermittently solved** when one or two pass, and **unsolved** when none do.

## Results — capability, specialization, and efficiency

### Performance by domain

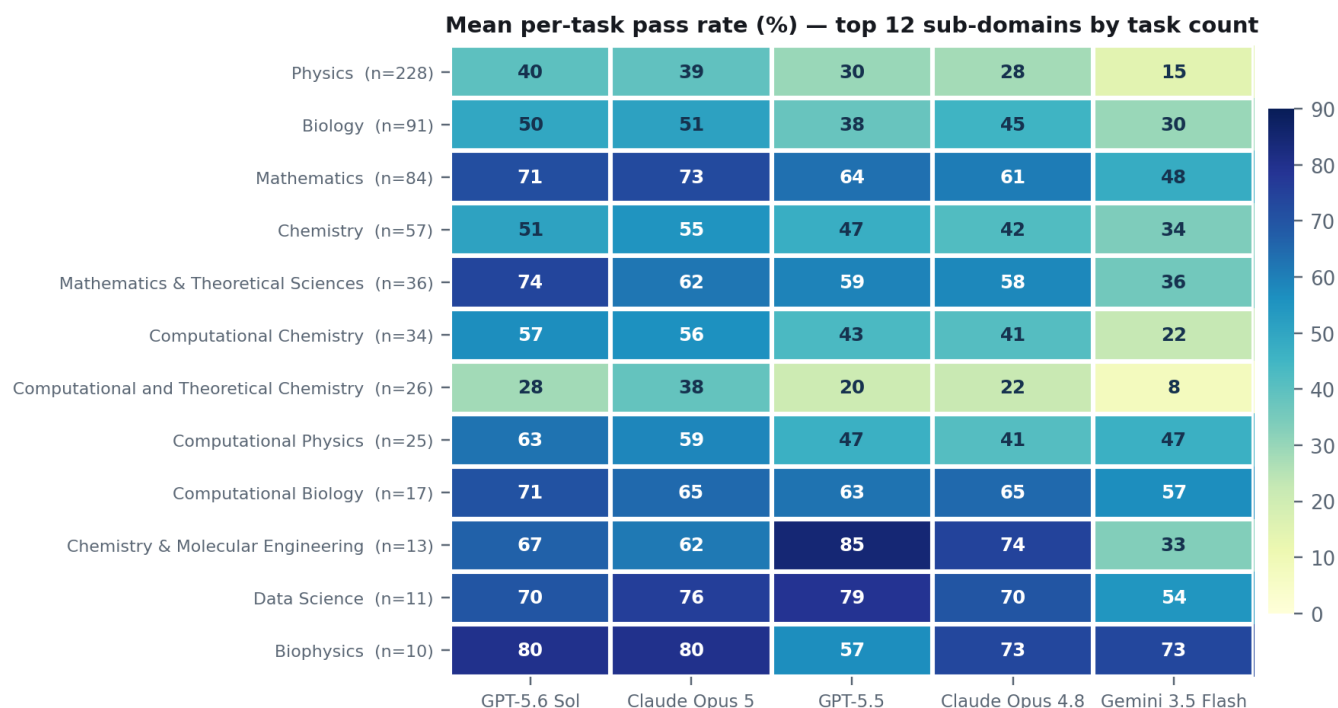
The aggregate leaderboard hides a 27-point spread across domains. Mathematics is the easiest domain benchmark-wide at 61.3% mean pass rate; Physics is the hardest at 34.0%, with Bio&Medicine (49.1%) and Chemistry (46.4%) between them. Mathematics-first and Physics-last hold for all five models, which means the difficulty at the extremes is a property of the domains as constructed rather than of any one model’s training mix; the middle two swap, with Chemistry ahead of Bio&Medicine for three of the five. GPT-5.6 Sol leads Mathematics (73.1%) and Physics (44.0%); Claude Opus 5 leads Chemistry (56.7%) and Bio&Medicine (55.9%), a clean split between the two leaders that the 0.7-point aggregate gap conceals entirely.



Each cell is the mean per-task pass rate for that model on that domain, over three runs per task. Shading runs from 10% to 80%.

### Performance by sub-domain

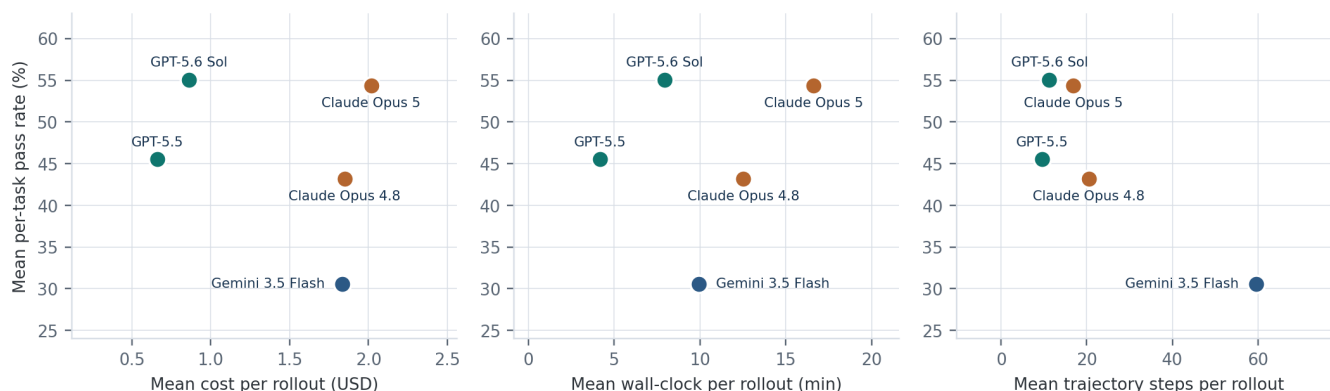
Splitting further separates broad capability from isolated success. Within the twelve largest sub-domains, the range runs from 8% (Gemini 3.5 Flash on Computational and Theoretical Chemistry) to 85% (GPT-5.5 on Chemistry & Molecular Engineering). The Physics sub-domain, 228 tasks, the single largest bucket in the suite, is the field-wide floor: no model exceeds 40% on it, and Gemini manages 14.6%. Computational and Theoretical Chemistry (26 tasks) is the second wall, topping out at 38%.



Mean per-task pass rate for the twelve largest sub-domains by task count; the full 94-row table is in Appendix A.2. Cells with small  $n$  are coarse, Biophysics (10 tasks) moves 10 points per task, and Chemistry & Molecular Engineering (13 tasks) 8 points per task.

## Accuracy–efficiency trade-offs

GPT-5.6 Sol sits on the efficient frontier: the highest score in the field at a mean \$0.86 and 11.3 steps per rollout, which is 43% of Claude Opus 5's cost and 67% of its steps for a statistically equivalent score. Claude Opus 5 is the most expensive configuration measured (\$2.02, 16.6 minutes) and buys nothing the leader does not already have on aggregate — its case rests on the 31 tasks it uniquely solves, not on its price. GPT-5.5 is the outright cheapest and fastest run in the study (\$0.66, 9.6 steps, 4.2 minutes) and gives up ten points of score for it, making it the best score-per-dollar option. Gemini 3.5 Flash is the outlier in the wrong direction: 59.5 steps and 55.7 tool calls per rollout, five times the leader's, at \$1.83, second-highest cost in the field, for the lowest score. Its cheapness per token does not survive contact with its step count.



Score against mean cost, wall-clock and trajectory steps per run; up and to the left is better. Points are colored by provider (● OpenAI, ● Anthropic, ● Google). These are means over all rollouts, not cost-per-success, on a cost-per-passing-run basis the spread widens further in GPT-5.6 Sol's favor.

## Overall finding

Two frontier models now clear roughly 55% of an 805-task expert-authored STEM suite, but neither can reproduce

half of what it solves. Of the 805 tasks, 125 (15.5%) defeat every model on every run and 226 (28.1%) are solved at least once by all five, but only 57 (7.1%) are solved by all five on all three runs. The models split on 454 tasks (56%), where at least one solves it and at least one never does. That combination, a wide unsolved frontier, low reproducibility, and dense partial reward that never converts, is what makes this set useful as training signal rather than as a scoreboard.

## Task Spotlight — ecb-astrocyte-neuron, five ways

We picked this task not because it is the hardest, but because it has the widest behavioral spread among tasks nobody solved. Five frontier agents attempted it fifteen times. Every attempt failed. The best of them failed by 1.67 assertions out of 90.

### The brief, and only the brief

instruction.md

The agent receives a raw-style adult mouse brain-slice dataset in a sealed offline container, messy experimental manifests, raw patch-clamp current traces, fluorescence imaging stacks and ROI masks, and must decide whether distinct endocannabinoids signal preferentially through neuronal or astrocytic mechanisms. It must build a deterministic, audit-grade analysis pipeline at `/workspace/solve.py` that canonicalizes the metadata, re-computes synaptic, calcium, ROI-topology, quality-control, pharmacology-contrast, permutation and bootstrap evidence from the raw arrays, and emits a mechanism certificate whose `global_verdict` must come out supported. Thirty named artifacts must be written. Sixteen named scientific libraries must be used substantively, each mapped to an artifact it produced.

Held back from the agent: the pytest suite, the oracle solution, every expected numeric value, and the perturbation replays. The agent works only from the briefing above. Expert-time estimate: 190 hours. Agent budget 3 hours, verifier budget 1 hour, internet disabled.

## Two stages, one sealed wall

**Stage 1, the science (every model cleared most of it).** Load the raw NPZ traces and imaging stacks, canonicalize unicode-variant genotype and drug aliases without dropping rows, extract baseline-subtracted peak deflections in fixed millisecond windows, compute  $\Delta F/F$  per ROI, run Welch tests and permutation contrasts, and build the directed mechanism graph. **Stage 2, the contract (where all fifteen rollouts died).** Emit all thirty artifacts with exactly the specified column names, exactly the specified dtypes and index conventions, exactly the specified seeds and RNG API, and make every artifact numerically consistent with every other.

## What the graders check

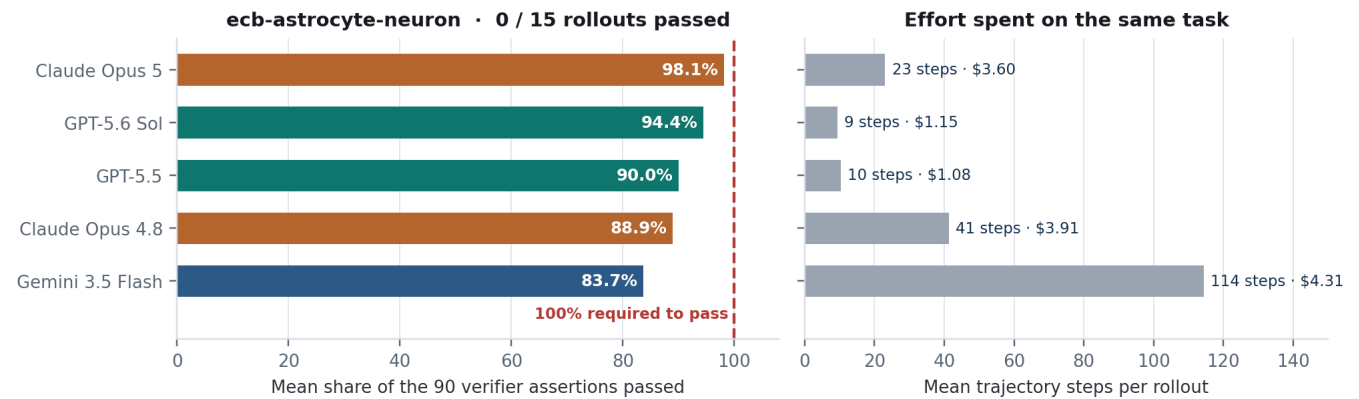
- **Recomputation, not inspection.** 90 assertions reload the agent's CSVs and JSONs and recompute the pulse amplitudes, noise, drift,  $\Delta F/F$ , AUC and Welch statistics from the raw arrays.
- **Input perturbation.** Two tests mutate the raw trace and image arrays and re-run the agent's pipeline; the metrics must move. A pipeline that inferred results from condition labels fails here even if every number was right.
- **Exact contractual strings.** `n_obs` not `observation_count`; `pca_component_1` not `pca_comp_0`; `xarray_dims` must be the string `roi, frame`, not a JSON list; `neo_quantities_units` must be `pA`, not `1.0 pA`.
- **Convention invariants.** `bbox_max_row` is an exclusive index per `skimage.regionprops`; `same_component` means direct pairwise overlap, not NetworkX connected-component membership; `response_auc` must be `np.trapezoid`, not a Riemann sum; Cohen's *d* must use the unweighted pooled convention, not the sample-size-weighted one.
- **Fixed randomness.** Exactly 1000 permutations and 250 bootstrap iterations, seed 1729, via `np.random.default_rng, RandomState` and global `np.random.seed` are explicitly invalid.

The wall

The obvious approach, read the condition names, which literally say `AM251` and `Dagla_K0`, and write the mechanistically expected verdict, cannot work, because the perturbation tests re-run the pipeline on mutated arrays and require the outputs to change. The strategic fork is therefore not scientific but architectural: an agent must decide, early, whether to build a single monolithic pass that computes everything and then serializes it, or thirty separately correct artifacts. The first is faster and gets the science right; the second is what the contract rewards. Every model chose the first.

| MODEL            | BEHAVIOR ON THE CONTRACT                                                                              | VERDICT | RUNS | ASSERTIONS |
|------------------|-------------------------------------------------------------------------------------------------------|---------|------|------------|
| Claude Opus 5    | Full pipeline, near-complete artifact set; misses on a handful of exact-string and convention checks. | ✗ FAIL  | 0/3  | 88.3 / 90  |
| GPT-5.6 Sol      | Terse 9-step solve; correct science, thinner artifact coverage.                                       | ✗ FAIL  | 0/3  | 85.0 / 90  |
| GPT-5.5          | Same shape as 5.6 Sol with weaker schema fidelity.                                                    | ✗ FAIL  | 0/3  | 81.0 / 90  |
| Claude Opus 4.8  | 41 steps of iteration; broader coverage, more convention errors.                                      | ✗ FAIL  | 0/3  | 80.0 / 90  |
| Gemini 3.5 Flash | 114 steps, \$4.31 per run, largest residual failure set.                                              | ✗ FAIL  | 0/3  | 75.3 / 90  |

Mean assertions passed of 90, averaged over three runs. No model found a passing path; the ordering of near-misses matches the overall leaderboard, and the spread between best and worst is 13 assertions, 14% of the suite.



**Left:** mean share of the 90 verifier assertions passed. **Right:** mean trajectory steps and cost per run. Effort and outcome are uncorrelated here, Gemini spends 12× GPT-5.6 Sol’s steps and 3.7× its cost to land 10 points lower.

The punchline

Claude Opus 5 passed 98.1% of this task’s verifier and earned nothing. That is not a scoring artifact; it is the point. In real research the pipeline that produces the right numbers under the wrong column names is not reproducible, and a pipeline that is not reproducible is not a result. Every one of the five failures here is a different, nameable weakness, schema drift, thin artifact coverage, convention substitution, over-iteration, which is exactly what makes a single task like this a usable training signal rather than one more zero on a leaderboard.

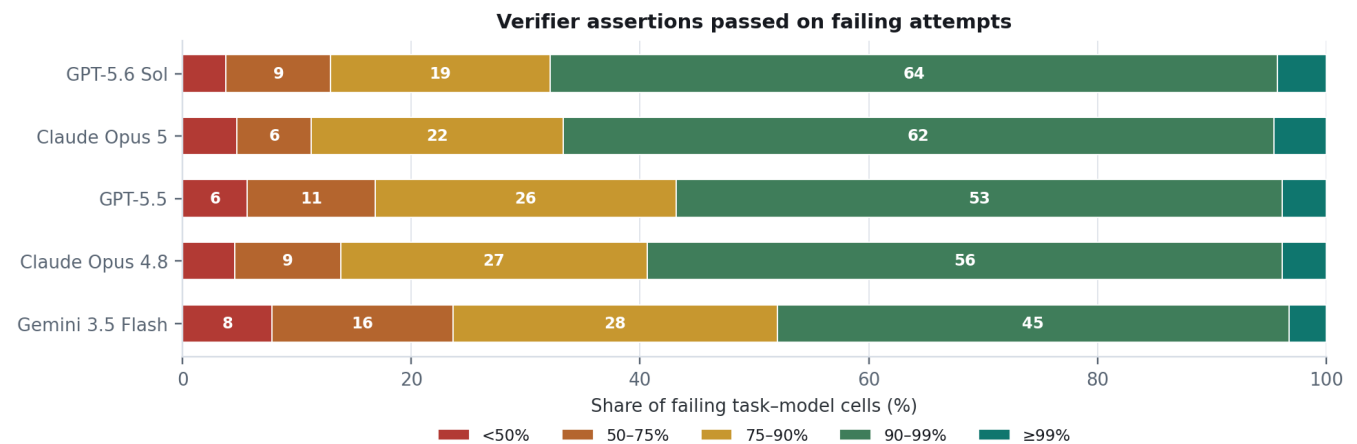
Failure Analysis — where the near-misses die

How to read the failure signal

This export records no failure taxonomy, and its continuous-reward column is identically zero. What it does record, for every failing rollout, is how many of the task’s assertions failed, which turns out to be the more informative signal, because it separates “did not understand the problem” from “did the science and missed the contract.” One caveat on the denominator: on 73 tasks the recorded assertion count differs between models, so the share below is computed per task–model cell against that cell’s own total rather than a single suite-wide figure.

| BAND   | MEANING IN FRONTIER AGENTIC STEM                                                                                                 | TYPICAL INSTANCE                                                                                                     |
|--------|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| ≥99%   | Contract-only failure. The science is complete; one or two exact-string, dtype or convention checks fail.                        | An exclusive-vs-inclusive bounding-box index, or a Riemann sum where the contract specifies trapezoidal integration. |
| 90–99% | Near-miss. A coherent, mostly-correct pipeline with a systematic error in one artifact family.                                   | All thirty artifacts written, but the bootstrap seeded through the wrong RNG API.                                    |
| 75–90% | Partial solve. A real attempt that omits or mis-specifies a whole workflow stage.                                                | The imaging branch computed, the ROI-topology branch skipped.                                                        |
| <50%   | Structural failure. The agent did not build a working pipeline, or built one that does not run to completion inside the timeout. | Timeouts, crashes and any refusals land here; the export does not distinguish them from graded verifier failures.    |

Failure profiles by model

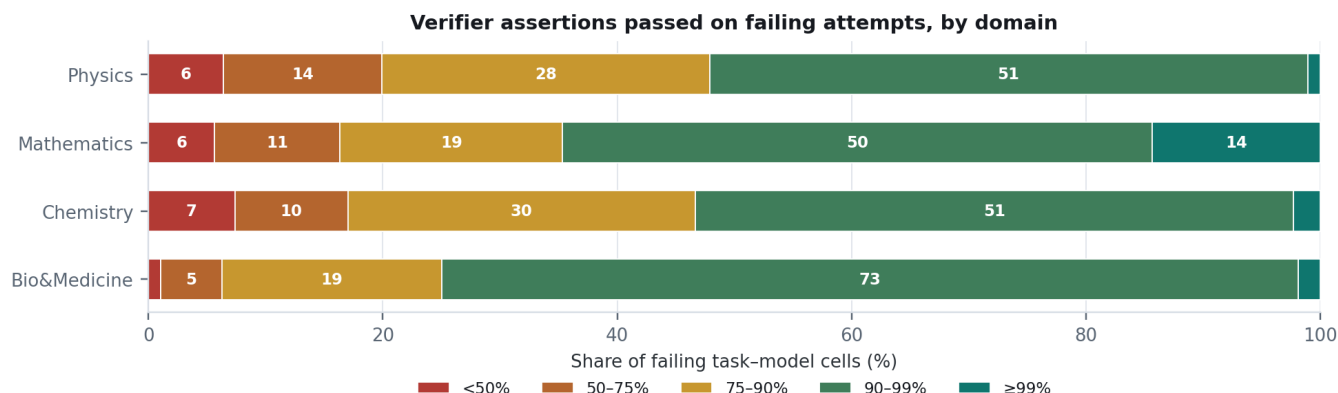


Each row is a percentage distribution over that model’s failing task-model cells, by the share of the task’s assertions the model passed.

The central pattern across all 12,075 rollouts is that “looks right” and “clears the verifier” are far apart, and the distance is small and consistent. Both leaders pass a mean 87.3% of assertions on the tasks they fail; GPT-5.6 Sol clears 90%-or-better on 67.9% of its failures and Claude Opus 5 on 66.7%. Even Gemini 3.5 Flash, the weakest model, clears half its verifier on 92.2% of its failures. Coverage tells the same story from the other side: 680 of the 805 tasks are solved at least once by at least one model and 125 by none, but only 57 are solved by all five models on all three runs. Only Gemini departs from the near-miss shape, 48.0% of its failures clear 90%, roughly twenty points below the leaders, and its 59.5-step average suggests it is iterating rather than converging.

The models are also not interchangeable at the margin. Claude Opus 5 uniquely solves 31 tasks no other model touches; GPT-5.6 Sol uniquely solves 21, GPT-5.5 15, Claude Opus 4.8 13, and even Gemini 3.5 Flash 10. Head to head, the 0.7-point gap between the two leaders sits on top of 188 tasks where exactly one of them succeeds, 23% of the suite decided in one direction or the other.

## Failure profiles by domain



Each row is a percentage distribution over failing task-model cells in that domain.

Physics fails hardest and fails earliest, it holds 64 of the 125 universally-unsolved tasks (21% of all Physics tasks), and its failures skew toward the lower bands, indicating attempts that break down before the contract stage. Chemistry loses 16.9% of its tasks to every model, concentrated in the Computational and Theoretical Chemistry sub-domain. Bio&Medicine (14.2%) fails late: its unsolved tasks are dominated by high-assertion audit pipelines like the spotlight task, where the science lands and the contract does not. Mathematics is the outlier at 6.8% universally unsolved, when the deliverable is a number or a certificate rather than a thirty-artifact directory, agents convert.

## Task-level failure mechanisms

**The zero-failure failure.** On `roh-f-spin-flip-matrices`, GPT-5.5's three failing runs record **zero failed assertions**, the attempts satisfied every check the suite counted and still did not register a pass, which means the run ended before the suite completed. Because the export pools non-graded terminations into "failed," this is the one place the collapse is directly visible. Nobody solved the task in fifteen rollouts.

**The one-assertion miss.** `involutive-doubled-cyclic-plumbing-complex` (258 assertions): best attempt 99.6%, one assertion short, zero passes across all five models.

**The near-miss cluster.** `rough-fracture-heat-transport` (98.7%), `exact-algebraic-self-similar-invariants` (98.6%), `microbe-atlas-benchmark` (98.5%), `hyperkahler-riemann-roch-census` (98.5%) and `matroid-chow-tutte-degrees` (98.3%), five more universally-unsolved tasks where a frontier agent got within two percent of a complete deliverable.

**The reproducibility failure.** Across the suite, 173 tasks are solved by GPT-5.6 Sol on some runs but not all, and 192 by Claude Opus 5. These are not hard tasks, they are tasks whose passing path the model can find but cannot re-find, which is the failure mode least visible on a single-run leaderboard.

**Off the near-miss axis.** Effort and success are decoupled. Gemini 3.5 Flash spends \$1.37 per run, more than either GPT configuration, and by far the most steps (59.5 steps, 55.7 tool calls) for the lowest score in the field; GPT-5.5 spends the least of both and beats it by fifteen points. Long trajectories in this suite are a symptom of not converging, not of working harder.

## Model Behavior — how each frontier model succeeds and fails

### GPT-5.6 Sol (high)

Leads at **55.1%** (1,330 of 2,415 rollouts passed), with run stability of consistent/intermittent/unsolved = 357/173/275. It sits on the efficient frontier — no model both scores higher and costs less — averaging 11.3 steps, 19.3 tool calls, 7.9 minutes and \$0.86 per rollout, though GPT-5.5 is cheaper and faster outright. Its solves concentrate in Mathematics (73.1%) and it holds the field's best Physics score (44.0%), the hardest domain. It uniquely solves 21 tasks. When it fails it fails narrowly — 67.9% of its failures clear 90% of the verifier, the best ratio measured — and its short trajectories mean the residual failures are contract errors made confidently rather than exploration that ran out of room.

## Claude Opus 5 (high)

Statistically tied at **54.4%** (1,313 passes), stability 342/192/271. It reaches the broadest coverage of any model — 534 tasks solved at least once, more than the leader — and uniquely solves 31 tasks, the most in the field. It leads Chemistry (56.7%) and Bio&Medicine (55.9%), the mirror image of GPT-5.6 Sol's specialization. The cost is efficiency: an average 16.8 steps, 24.2 tool calls, 16.6 minutes and \$2.02 per rollout — the most expensive configuration in the study, well over double the leader. Its failure profile is a near-tie with GPT-5.6 Sol (mean 87.3% of assertions, 66.7% of failures above 90%), and its 192 intermittent tasks are the largest reproducibility gap among the leaders.

## GPT-5.5 (medium)

**45.5%** (1,099 passes), stability 256/218/331. The cheapest and fastest configuration measured: an average 9.6 steps, 13.1 tool calls, 4.2 minutes and \$0.66 per rollout, which makes it the best score-per-dollar in the field even though it sits ten points off the lead. It solves concentrate in Mathematics (63.9%) and it posts the highest cell among sub-domains with at least ten tasks (84.6% on Chemistry & Molecular Engineering, 13 tasks). Failures are slightly wider than the leaders' — mean 85.0% of assertions, 56.9% above 90%.

## Claude Opus 4.8 (high)

**43.2%** (1,043 passes over 805 tasks), stability 230/237/338. It is the least reproducible of the four strong models: more tasks solved intermittently (237) than consistently (230). An average 20.5 steps and 24.9 tool calls at \$1.85 per rollout make it the second most expensive configuration for the second-lowest score. Strongest in Mathematics (56.8%), weakest in Physics (29.9%). Its failures average 86.5% of assertions with 59.4% above 90%, close to the leaders in shape, roughly ten points behind in conversion.

## Gemini 3.5 Flash (medium)

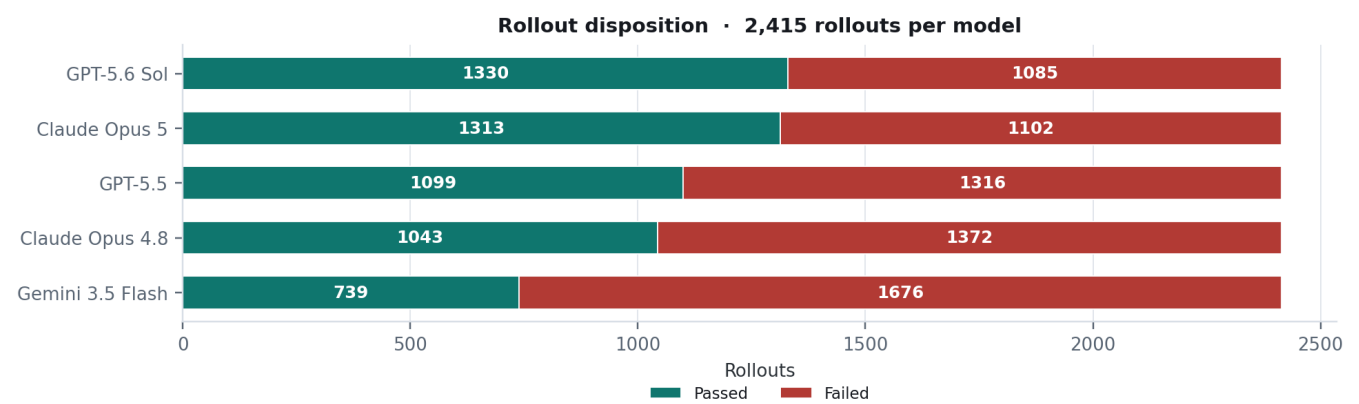
**30.6%** (739 passes), stability 127/248/430. The profile is distinctive: it solves 375 tasks at least once but only 127 reliably, so two-thirds of its apparent capability does not reproduce. It spends the most effort of any model — an average 59.5 steps and 55.7 tool calls, roughly five times the leader, at \$1.83 per rollout. Its Physics score (19.6%) is the field floor, and it posts the lowest cell among the twelve largest sub-domains (7.7% on Computational and Theoretical Chemistry). It is also the only model materially off the near-miss axis: mean 81.5% of assertions on failures and just 48.0% above 90%, meaning a substantial share of its failures are structural rather than contractual.

## Appendix A — per-model run statistics

This appendix reports the per-rollout telemetry behind the leaderboard: how the 12,075 rollouts disposed, the per-model means for steps, tool calls, wall-clock and cost, and the complete 94-row sub-domain breakdown that the main Results section shows only in part. All figures are drawn from the `TFRL-STEM-OTS_160_report.xlsx` run export, joined to `tfrl_stem_ots_domains.csv` for domain labels. Scoring is identical to the main sheet: every model is scored over all 805 tasks at  $k = 3$ .

### A.1 Where the 2,415 rollouts go

The leaderboard collapses each task to a per-task pass rate. This view does not: it counts raw rollouts, which separates a model that passes one task three times from a model that passes three tasks once. The export carries only two dispositions, so timeouts, crashes and any refusals are pooled into “failed” and the split is necessarily two-way.



Each model’s rollouts by disposition: passed or failed.

| MODEL                   | ROLLOUTS | PASSED | STEPS | TOOL CALLS | RUNTIME | COST   |
|-------------------------|----------|--------|-------|------------|---------|--------|
| GPT-5.6 Sol (high)      | 2,415    | 1,330  | 11.3  | 19.3       | 7.9m    | \$0.86 |
| Claude Opus 5 (high)    | 2,415    | 1,313  | 16.8  | 24.2       | 16.6m   | \$2.02 |
| GPT-5.5 (medium)        | 2,415    | 1,099  | 9.6   | 13.1       | 4.2m    | \$0.66 |
| Claude Opus 4.8 (high)  | 2,415    | 1,043  | 20.5  | 24.9       | 12.5m   | \$1.85 |
| Gemini 3.5 Flash (med.) | 2,415    | 739    | 59.5  | 55.7       | 9.9m    | \$1.83 |

Rollout-weighted means from the run export. Gemini’s step and tool-call means are five times the leader’s while its runtime is only 25% higher — it takes many short actions rather than few long ones. Rollout counts include Claude Opus 4.8’s 11 missing rollouts scored as failures; its means are over the 2,404 actually recorded. Total recorded model spend across all 12,064 rollouts is approximately \$17,400, a mean of \$1.45 per rollout.

| MODEL                   | CONSISTENT | INTERMITTENT | UNSOLVED | ANY PASS | UNIQUE | ASSERTIONS ON FAILS |
|-------------------------|------------|--------------|----------|----------|--------|---------------------|
| GPT-5.6 Sol (high)      | 357        | 173          | 275      | 530      | 21     | 87.3%               |
| Claude Opus 5 (high)    | 342        | 192          | 271      | 534      | 31     | 87.3%               |
| GPT-5.5 (medium)        | 256        | 218          | 331      | 474      | 15     | 85.0%               |
| Claude Opus 4.8 (high)  | 230        | 237          | 338      | 467      | 13     | 86.5%               |
| Gemini 3.5 Flash (med.) | 127        | 248          | 430      | 375      | 10     | 81.5%               |

**Consistent / Intermittent / Unsolved** are task counts by how many of the three runs passed (3, 1–2, and 0). **Any pass** is tasks solved at least once; **Unique** is tasks that model alone solves. **Assertions on fails** is the mean share of a task’s verifier assertions passed on rollouts that failed — the near-miss measure used throughout this sheet. The export has no refusal or termination-reason field and its continuous-reward column is identically zero, so neither is reported here.

### A.3 Complete sub-domain breakdown

All 94 sub-domains as labelled in the source file, ordered by task count. Values are mean per-task pass rate (%) over three runs. Sub-domains below roughly five tasks are shown for completeness but should be read as anecdotes rather than measurements, 70 of the 94 hold three tasks or fewer, where a single task moves a cell by 33 points. A handful of labels (HR & People Analytics, Supply Chain & Logistics, MLOps & Model Lifecycle Management and five others, 13 tasks in total) sit outside the four natural-science parents, and computational-biology appears as a duplicate slug of Computational Biology; both are reproduced as-labelled rather than silently remapped.

| SUB-DOMAIN                                                 | TASKS | G5.6S | OP5   | G5.5  | OP4.8 | GEM   | ALL   |
|------------------------------------------------------------|-------|-------|-------|-------|-------|-------|-------|
| Physics                                                    | 228   | 39.9  | 39.2  | 29.7  | 27.9  | 14.6  | 30.3  |
| Biology                                                    | 91    | 49.5  | 51.3  | 37.7  | 45.4  | 29.7  | 42.7  |
| Mathematics                                                | 84    | 71.4  | 72.6  | 63.5  | 60.7  | 48.0  | 63.3  |
| Chemistry                                                  | 57    | 50.9  | 55.0  | 47.4  | 42.1  | 33.9  | 45.8  |
| Mathematics & Theoretical Sciences                         | 36    | 74.1  | 62.0  | 59.3  | 58.3  | 36.1  | 58.0  |
| Computational Chemistry                                    | 34    | 56.9  | 55.9  | 43.1  | 41.2  | 22.5  | 43.9  |
| Computational and Theoretical Chemistry                    | 26    | 28.2  | 38.5  | 20.5  | 21.8  | 7.7   | 23.3  |
| Computational Physics                                      | 25    | 62.7  | 58.7  | 46.7  | 41.3  | 46.7  | 51.2  |
| Computational Biology                                      | 17    | 70.6  | 64.7  | 62.7  | 64.7  | 56.9  | 63.9  |
| Chemistry & Molecular Engineering                          | 13    | 66.7  | 61.5  | 84.6  | 74.4  | 33.3  | 64.1  |
| Data Science                                               | 11    | 69.7  | 75.8  | 78.8  | 69.7  | 54.5  | 69.7  |
| Biophysics                                                 | 10    | 80.0  | 80.0  | 56.7  | 73.3  | 73.3  | 72.7  |
| Bioinformatics                                             | 10    | 56.7  | 56.7  | 56.7  | 46.7  | 30.0  | 49.3  |
| Molecular Dynamics                                         | 10    | 70.0  | 53.3  | 43.3  | 43.3  | 46.7  | 51.3  |
| Quantum Physics                                            | 10    | 53.3  | 70.0  | 43.3  | 33.3  | 43.3  | 48.7  |
| Data Wrangling                                             | 9     | 22.2  | 48.1  | 25.9  | 44.4  | 29.6  | 34.1  |
| Materials Science                                          | 8     | 54.2  | 37.5  | 62.5  | 50.0  | 16.7  | 44.2  |
| Scientific Computing                                       | 8     | 66.7  | 70.8  | 54.2  | 37.5  | 8.3   | 47.5  |
| Data Science in STEM                                       | 7     | 100.0 | 73.3  | 86.7  | 60.0  | 40.0  | 72.0  |
| MLOps & Model Lifecycle Management                         | 5     | 100.0 | 86.7  | 93.3  | 33.3  | 33.3  | 69.3  |
| Machine Learning                                           | 5     | 40.0  | 40.0  | 33.3  | 60.0  | 53.3  | 45.3  |
| Machine Learning & Statistics                              | 5     | 80.0  | 80.0  | 86.7  | 73.3  | 40.0  | 72.0  |
| Signal Processing                                          | 4     | 50.0  | 50.0  | 66.7  | 83.3  | 58.3  | 61.7  |
| Fluid Dynamics                                             | 4     | 41.7  | 58.3  | 50.0  | 25.0  | 8.3   | 36.7  |
| physics                                                    | 3     | 33.3  | 33.3  | 33.3  | 22.2  | 22.2  | 28.9  |
| Systems neuroscience                                       | 3     | 77.8  | 44.4  | 22.2  | 77.8  | 33.3  | 51.1  |
| Bayesian Inference                                         | 3     | 100.0 | 77.8  | 11.1  | 44.4  | 55.6  | 57.8  |
| PDE Solver                                                 | 3     | 66.7  | 44.4  | 33.3  | 33.3  | 22.2  | 40.0  |
| Biomedical and Life Sciences                               | 3     | 66.7  | 77.8  | 33.3  | 22.2  | 88.9  | 57.8  |
| Uncertainty Quantification                                 | 3     | 66.7  | 66.7  | 33.3  | 55.6  | 55.6  | 55.6  |
| Numerical Optimization                                     | 2     | 16.7  | 50.0  | 50.0  | 50.0  | 16.7  | 36.7  |
| Business & Operational Analytics                           | 2     | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   |
| scientific_computing                                       | 2     | 33.3  | 16.7  | 0.0   | 16.7  | 0.0   | 13.3  |
| Inverse Problems                                           | 2     | 100.0 | 100.0 | 50.0  | 50.0  | 0.0   | 60.0  |
| Quantum Computing                                          | 2     | 50.0  | 0.0   | 66.7  | 0.0   | 33.3  | 30.0  |
| Biology-STEM                                               | 2     | 66.7  | 100.0 | 83.3  | 100.0 | 83.3  | 86.7  |
| Data Auditing                                              | 2     | 100.0 | 50.0  | 100.0 | 0.0   | 33.3  | 56.7  |
| Convex Optimization                                        | 2     | 83.3  | 100.0 | 50.0  | 0.0   | 0.0   | 46.7  |
| Plasma Physics                                             | 1     | 100.0 | 100.0 | 100.0 | 66.7  | 66.7  | 86.7  |
| Computational Mathematics                                  | 1     | 0.0   | 100.0 | 100.0 | 100.0 | 66.7  | 73.3  |
| Climate & Atmospheric Science / Spatiotemporal Forecasting | 1     | 100.0 | 100.0 | 100.0 | 0.0   | 0.0   | 60.0  |
| Calculus Of Variations                                     | 1     | 100.0 | 100.0 | 100.0 | 66.7  | 0.0   | 73.3  |
| Adsorption / Surface Chemistry                             | 1     | 100.0 | 100.0 | 66.7  | 100.0 | 100.0 | 93.3  |
| Computational Geometry                                     | 1     | 100.0 | 100.0 | 100.0 | 0.0   | 0.0   | 60.0  |
| Computational Biology + Perinatal Microbiome Analysis      | 1     | 0.0   | 33.3  | 100.0 | 66.7  | 66.7  | 53.3  |
| Supply Chain & Logistics Analytics                         | 1     | 100.0 | 66.7  | 0.0   | 0.0   | 0.0   | 33.3  |
| Finite Element Analysis                                    | 1     | 66.7  | 66.7  | 66.7  | 0.0   | 33.3  | 46.7  |
| HR & People Analytics                                      | 1     | 100.0 | 100.0 | 0.0   | 100.0 | 100.0 | 80.0  |
| Algebraic Combinatorics                                    | 1     | 100.0 | 100.0 | 66.7  | 0.0   | 0.0   | 53.3  |
| Marketing & Commercial Strategy                            | 1     | 100.0 | 66.7  | 100.0 | 0.0   | 0.0   | 53.3  |
| Multi-Agent Systems                                        | 1     | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Molecular Biology                                          | 1     | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |

| SUB-DOMAIN (CONT.)                                      | TASKS | G5.6S | OP5   | G5.5  | OP4.8 | GEM   | ALL   |
|---------------------------------------------------------|-------|-------|-------|-------|-------|-------|-------|
| Computational-biology                                   | 2     | 100.0 | 100.0 | 100.0 | 100.0 | 66.7  | 93.3  |
| Graph Theory                                            | 1     | 100.0 | 0.0   | 100.0 | 66.7  | 0.0   | 53.3  |
| Quantum Chemistry                                       | 1     | 100.0 | 100.0 | 66.7  | 100.0 | 100.0 | 93.3  |
| Variational Inference                                   | 1     | 100.0 | 100.0 | 66.7  | 100.0 | 66.7  | 86.7  |
| Robotics                                                | 1     | 100.0 | 100.0 | 100.0 | 33.3  | 33.3  | 73.3  |
| Scientific Computing / PDEs / Optimal Control           | 1     | 66.7  | 0.0   | 0.0   | 66.7  | 0.0   | 26.7  |
| Inverse Problems / Imaging / Uncertainty Quantification | 1     | 100.0 | 66.7  | 100.0 | 33.3  | 100.0 | 80.0  |
| Biomedical Data Analysis                                | 1     | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| LLM Evaluation / Guardrailng / Safety Benchmarking      | 1     | 33.3  | 100.0 | 66.7  | 0.0   | 66.7  | 53.3  |
| Spatiotemporal Modeling                                 | 1     | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   |
| Machine Learning Engineering & Mathematics              | 1     | 100.0 | 100.0 | 33.3  | 100.0 | 0.0   | 66.7  |
| Operations Analytics                                    | 1     | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Science simulation                                      | 1     | 100.0 | 100.0 | 33.3  | 0.0   | 0.0   | 46.7  |
| Numerical Simulation                                    | 1     | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| proteomics-transcriptomics-integration                  | 1     | 0.0   | 66.7  | 0.0   | 33.3  | 0.0   | 20.0  |
| Computational Fluid Dynamics                            | 1     | 66.7  | 33.3  | 0.0   | 100.0 | 0.0   | 40.0  |
| Operations Research                                     | 1     | 100.0 | 100.0 | 100.0 | 66.7  | 66.7  | 86.7  |
| Quantum Science and Technology - Quantum Circuits       | 1     | 0.0   | 100.0 | 0.0   | 0.0   | 0.0   | 20.0  |
| Astrophysics                                            | 1     | 33.3  | 0.0   | 33.3  | 0.0   | 0.0   | 13.3  |
| Cheminformatics                                         | 1     | 100.0 | 100.0 | 100.0 | 0.0   | 0.0   | 60.0  |
| scientific computing                                    | 1     | 100.0 | 33.3  | 33.3  | 0.0   | 0.0   | 33.3  |
| Particle Tracking                                       | 1     | 100.0 | 100.0 | 100.0 | 100.0 | 66.7  | 93.3  |
| quantum-many-body-physics                               | 1     | 100.0 | 66.7  | 100.0 | 66.7  | 100.0 | 86.7  |
| Computed Tomography                                     | 1     | 100.0 | 100.0 | 66.7  | 33.3  | 33.3  | 66.7  |
| Motion Planning                                         | 1     | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Machine Learning & Artificial Intelligence              | 1     | 0.0   | 0.0   | 66.7  | 100.0 | 100.0 | 53.3  |
| Control Theory                                          | 1     | 100.0 | 100.0 | 100.0 | 66.7  | 33.3  | 80.0  |
| Algebraic Number Theory                                 | 1     | 100.0 | 100.0 | 33.3  | 66.7  | 66.7  | 73.3  |
| Polymer Physics                                         | 1     | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Macroeconomics & Specialized Policy                     | 1     | 100.0 | 100.0 | 33.3  | 0.0   | 0.0   | 46.7  |
| Robotics Simulation                                     | 1     | 0.0   | 0.0   | 0.0   | 0.0   | 100.0 | 20.0  |
| Statistics                                              | 1     | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   | 0.0   |
| Computational Mechanics                                 | 1     | 100.0 | 100.0 | 100.0 | 100.0 | 66.7  | 93.3  |
| Supply Chain & Logistics                                | 1     | 100.0 | 100.0 | 66.7  | 100.0 | 0.0   | 73.3  |
| Epidemiology                                            | 1     | 33.3  | 0.0   | 0.0   | 100.0 | 0.0   | 26.7  |
| Linear Algebra                                          | 1     | 66.7  | 100.0 | 100.0 | 66.7  | 33.3  | 73.3  |
| quantitative-finance                                    | 1     | 0.0   | 66.7  | 0.0   | 0.0   | 0.0   | 13.3  |
| Computational Materials                                 | 1     | 0.0   | 0.0   | 0.0   | 0.0   | 33.3  | 6.7   |
| Systems Biology                                         | 1     | 100.0 | 0.0   | 100.0 | 0.0   | 66.7  | 53.3  |

Columns: GPT-5.6 Sol, Claude Opus 5, GPT-5.5, Claude Opus 4.8, Gemini 3.5 Flash, and the mean over all five models.

## What's Included — delivery and how to engage

- **The task package.** 805 tasks in native Harbor format. Each ships `instruction.md`, an `environment/` Dockerfile with a pinned scientific stack and declared CPU, memory and storage limits, a `solution/` oracle, a `tests/` pytest suite, and a `task.toml` carrying domain, tags, expert-time estimate and timeouts.
- **The evaluation artifacts.** 12,075 rollouts across 5 frontier-agent configurations under the TERMINUS-2 harness, three runs per task, with per-rollout trajectory steps, tool calls, agent execution time, model cost, total assertions and assertions failed, released as the passed-run and failed-run exports behind this sheet.
- **The analysis layer.** Domain and 94-way sub-domain labels for every task, per-task run-stability segmentation (consistent / intermittent / unsolved), near-miss assertion-share analysis, and the per-model efficiency telemetry reproduced in Appendix A.

Every task is proven solvable by its oracle solution before release, and every verifier is deterministic, so a failure is a capability signal rather than infrastructure noise. Oracle solutions, verifier suites and expected values are held outside the agent-visible environment and are not distributed with the agent-facing package. Individual task contents are deliberately excluded from this public sheet beyond the single spotlight, which is reproduced with its solution path redacted.

### Interested in task samples or a deeper briefing?

A walkthrough covers sample tasks in your domains of interest, the full run export, and how the set can be extended or specialized for your training and evaluation needs.

[labs@turing.com](mailto:labs@turing.com) · Turing Frontier Research Lab