WRONG TARGET OR FAILED DELIVERY? DIAGNOSING SCIENTIFIC CODING BEYOND PASS/FAIL

Anonymous authors
Paper under double-blind review

ABSTRACT

Strict scientific coding scores combine numerical computation, report construction, and final execution. A completion gap therefore need not represent an equally large difference in numerical content or program behavior. We examine this distinction through complete-reference matching, saved-content checks, and tests of frozen final programs linked to original execution records. In 43 historical starting-program pairs, strict completion differs by 65.1 percentage points, whereas the requested-vector match gap is bounded by 0–14.0 points. With reading rules fixed before model calls, GPT-5.4-mini and Qwen3.7 Plus show completion gaps of 66.7 and 75.0 percentage points, respectively, despite zero saved-content gaps. Haiku's completion gap exceeds its saved-content gap by at least 33.3 points; Gemini completes both starting conditions. A factorial design then varies computation and delivery support separately. Across six model configurations, the records distinguish correct numerical returns in invalid reports, numerical or return-type failures, and cases lacking qualifying final-source execution evidence. The prespecified Mini–Haiku replication yields an observed delivery-only versus computation-only completion difference of +20.8 points ($p = 0.125$). Joint evidence shows which completion differences accompany changes in tested computation and which coexist with matching numerical content and unfinished reporting or execution obligations.

1 INTRODUCTION

A scientific coding agent must compute the requested quantity, construct a usable report, and execute its final program. A strict success score records whether all requirements are met. To interpret a performance difference, however, we need to know *which requirements remain unfinished*. A failed trajectory may produce a well-defined alternative quantity, preserve the requested values in the wrong report structure, or leave a test-passing program without final-source execution evidence. These outcomes share a failed score but provide different evidence about computation and delivery.

Consider two weather stations with observations $\{0, 0\}$ and $\{12\}$. Averaging all observations gives 4; averaging the two station means gives 6. For an observation-equal request, a saved 6 identifies an alternative calculation, whereas 4 outside the required JSON object indicates matching content but invalid delivery. Both fail strict acceptance for different reasons.

Our layered analysis (Figure 1) combines complete-reference matching, saved-content checks outside the required report path, and tests of frozen final programs. Preservation, schema, and execution checks identify unfinished obligations. Together, these observations connect artifact content, tested program behavior, and actual execution.

Our first finding is completion–content separation: at least 51.2 points in 43 historical pairs, with analogous findings under pre-call reading rules for GPT-5.4-mini (Mini), Qwen3.7 Plus (Qwen), and Claude Haiku 4.5 (Haiku). Gemini completes both starts.

Our second finding concerns supplied support. A two-by-two design independently varies the availability of computation kernels and answer-free delivery constructors across six configurations. Joint records distinguish invalid reports, incorrect computations, and missing final-source execution evidence behind completion differences.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

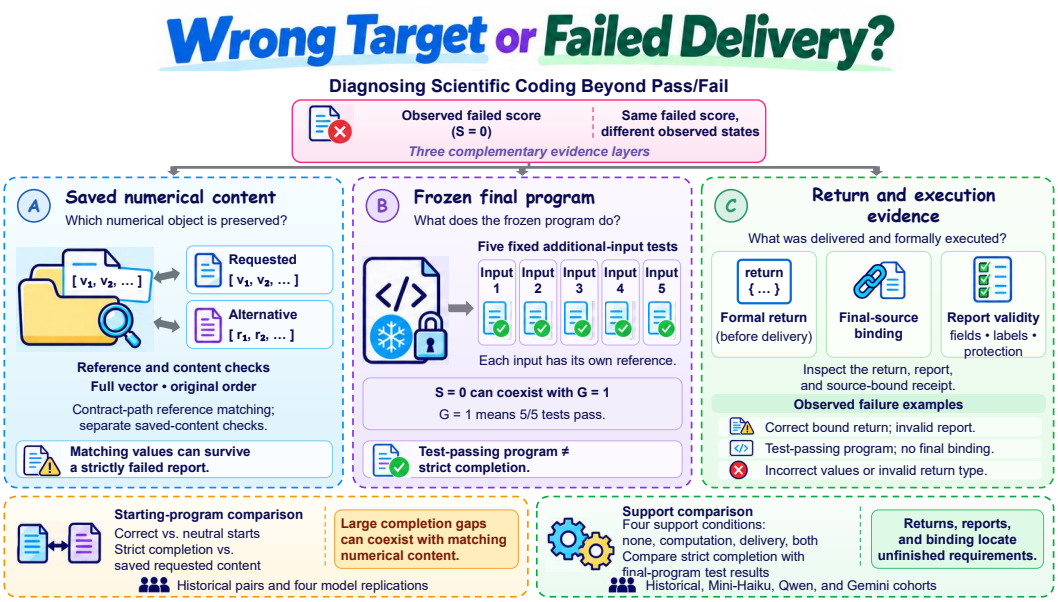

Figure 1: **Three evidence layers for scientific coding.** Saved numerical content, frozen-program tests, and original execution records provide complementary evidence. The two comparisons below apply these layers to starting programs and supplied support. S denotes strict completion; $G = 1$ requires all five fixed program tests to pass.

Prior work distinguishes output content from format compliance (Long et al., 2025; Xia et al., 2024), evaluates scientific programs and their execution (Chen et al., 2025; Tian et al., 2024), and audits benchmark specifications and scoring (Hu et al., 2026). Format-tolerant scoring alone identifies matching content, but does not establish program behavior on additional inputs or verify execution of the final submitted source. We combine content checks, frozen-program tests, and original execution records to distinguish correct, source-bound numerical returns in invalid reports, test-passing programs without qualifying final-source execution evidence, and incorrect values packaged by delivery support. Together, these records identify which requirements remain unfinished behind completion contrasts; Appendix E.2 specifies a reusable schema for these evidence states. We make two contributions:

- **Quantifying completion–content separation.** A better starting program can confer a large completion advantage without a comparable saved-content advantage. Complete-vector comparisons quantify this separation, and pre-call reading rules test its reach across model configurations (Sections 2–4).
- **Explaining completion contrasts beyond program-test differences.** Supplied computation can remain undelivered, while supplied delivery can package an incorrect computation. Factorial comparisons across six configurations connect these contrasting outcomes to program tests and original execution evidence (Sections 5–6).

2 Measurement Framework

2.1 The contract fixes the requested quantity

For input D, let $T^*(D)$ be the requested numerical object and $\mathcal{R}(D)$ a finite set of distinguishable references containing it. References, coordinate order, and numerical tolerances are fixed before model execution. Historical panels use either a public applicability rule to select between two definitions, or an explicitly named target with two alternatives; their condition codes and exact inputs are given in Appendices A.1 and B.1.

The public contract determines the target, independently of the starting implementation. Each target specifies a finite-input computation, including conventions such as quantile interpolation. In the station example, observation-equal and station-equal means are distinct targets; marginal summaries and counts are protected outputs that must remain unchanged.

2.2 Numerical identity and strict acceptance

A trajectory is *observed-valid* when its measurement is complete, even if the task fails or its output is unreadable. Complete-reference matching assigns *current* to a full requested match, *non-requested* to a full specified-alternative match, *other numerics* to a complete object matching no reference, and *undetermined* to an absent, malformed, or incomplete contract object. Matching requires the prescribed length, coordinate order, and tolerance.

Preservation checks protected values and structure; *schema* checks fields and labels, so a missing protected field can fail both. *Binding* links the final artifact to a formal run and its source, configuration, and observed dependencies. Changes to the source, relevant inputs, or output require a matching receipt; unrelated diagnostic files do not invalidate it (Appendix A.3.1). Strict acceptance is

$$\text{strict success} = \text{current numerics} \wedge \text{preservation} \wedge \text{schema} \wedge \text{binding}. \tag{1}$$

The contract path determines whether an object can be extracted; reference matching then determines its identity. A wrong label can coexist with a numerical match, whereas a missing path leaves identity undetermined. The separate saved-content rule in Section 4.3 examines such artifacts. A *bound substitution* requires an alternative match and binding in the two-target panels; the three-target endpoint additionally requires preservation and full schema.

2.3 Program and execution evidence

Saved content asks what is in the artifact; program tests ask what the final program does on fixed inputs; execution records ask what the original trajectory completed. Appendix A.2 documents independent implementation checks; Table 8 consolidates acceptance predicates. We record three outcomes:

Strict completion, S, requires all four checks in Equation 1. **Formal return, K,** records whether the computation-entry return from the last formal execution matches the requested reference, before delivery. Records distinguish incorrect returns, invalid types, execution exceptions, and absent qualifying returns; final-source binding is separate.

Program tests, G, evaluate the frozen computation entry on five inputs fixed before calls; $G = 1$ requires all five to pass. JSON tests use possibly overlapping within-collection subsets; native adaptations use task-specific inputs. Each input has its own reference, with no agent feedback. The suite numerically rejects 32 examined wrong JSON implementations with valid output shapes (Table 9); this measures selected-error discrimination, not domain-wide coverage. The five tests yield one outcome per program. Protocol-invalid trajectories retain null S, K, and G.

3 Experimental setup

A root is a fixed computation reused across conditions. Two distinct twelve-root collections answer our principal questions.

Does the starting-program advantage reflect saved content? Correct starts supply computation and report construction; neutral starts supply neither. Tasks cover F1 (Wine, WDBC, Glass), correlation (Abalone, red/white Wine Quality), and quantiles and fitting (Auto MPG, Airfoil, Yacht). We compare completion with saved-vector matches. Replications fix data-field background C; the historical 43 pairs come from the earlier study on these roots.

Which evidence changes with support? Four arms supply neither implementation, computation alone, delivery alone, or both. Eight JSON tasks cover F1, correlation, quantiles and fitting on Iris, Digits, Diabetes and Linnerud; four SciCode adaptations retain native returns (Tian et al., 2024). Outcomes are S, K, G and binding. The original wrong-target collection is separate (Section 4.1).

Table 1: **Evidence across the two principal comparisons.** Support entries give $(S \mid G)/n$ over valid sessions; ΔS is delivery-only minus computation-only over twelve planned pairs (percentage points). Three DeepSeek outcomes are unknown. Starting-program and support comparisons use distinct task collections. Starting-program columns give correct/neutral fractions; six historical and eight Haiku neutral contents are unclassified. Historical pairs are a selected DeepSeek/GLM subset. Mini–Haiku, Qwen, and Gemini belong to separate replication cohorts. Dashes denote comparisons absent from this summary.

Model / cohort	Support $S \mid G$	Support ΔS	Starting programs strict: correct / neutral	Saved-vector matches: correct / neutral
Historical pairs	—	—	41/43 / 13/43	$43/43$ / $\geq 37/43$
DeepSeek-v3.2	$(32 \mid 43)/45$	[0,16.7]	—	—
GLM-5.2	$(33 \mid 48)/48$	58.3	—	—
GPT-5.4-mini	$(31 \mid 43)/48$	41.7	8/12 / 0/12	12/12 / 12/12
Claude Haiku 4.5	$(48 \mid 48)/48$	0	12/12 / 0/12	$12/12$ / $\geq 4/12$
Qwen3.7 Plus	$(47 \mid 48)/48$	0	12/12 / 3/12	12/12 / 12/12
Gemini 3.1 Flash-Lite	$(46 \mid 47)/48$	0	12/12 / 12/12	12/12 / 12/12

Models and execution. We analyze DeepSeek-v3.2/GLM-5.2, Mini–Haiku, Qwen, and Gemini as separate cohorts (Table 1). Replications use fresh workspaces and one session per cell, with ceilings of 24 replies and 24 charged tool units; model-specific settings appear in Appendix A.4. Agents see neither private references nor test feedback. Each cohort description documents prior exposure and protocol changes.

4 Diagnosing scientific coding failures

Target-identity cases first show what failure scores conceal; saved-content evidence then clarifies starting-program gaps. Table 1 summarizes both principal comparisons.

4.1 Wrong-target outputs in the original setting

The original twelve roots cover weighted fitting, fixed-prediction risk, station aggregation, and observation support on Parkinsons Telemonitoring and Beijing Multi-Site Air Quality (Tsanas & Little, 2009; Chen, 2017). Both definitions are public; initial programs compute T_0. The snapshot has 359 valid observations, one unknown, and eighteen then-unstarted positions; the later order-panel continuation is separate (Tables 27 and 30).

Across this census, 60/359 valid trajectories deliver the complete bound alternative target-plus-protection object, accounting for 60/70 failures. In the 72-trajectory original-context subset, seventeen of nineteen failures have this identity. Applicable and inapplicable updates both yield 22/24 successes, yet only the latter failures deliver bound alternatives. Equal completion totals thus conceal different output identities. Rule presence, semantics, and length vary together in these presentations.

4.2 Variation across task interfaces

The corrected Palmer Penguins/NOAA CO-OPS panel yields 138/141 strict successes, three unknown outcomes, and no bound substitutions. Its six roots were exposed in a pilot before the public label correction; all three valid failures lack final binding (Appendix F.4). Initial-state, wording/layout, and menu comparisons appear in Appendices F.2, F.3, and F.5.

4.3 Requested content within failed deliveries

A three-target study varies background definitions and starting programs across 288 planned positions. The saved corpus contains 180 records: 174 valid and six protocol-invalid, leaving 108 positions unstarted. The original extractor identifies 111 current and 63 undetermined outputs, with

all undetermined outputs arising from neutral starts. Appendix B.1 documents the directional endpoint and stopping history.

Inspection of one undetermined output revealed the requested vector at top-level `values` rather than inside `target`. We then fixed a post-hoc reading rule and applied it to all 180 records. The object must have exactly the top-level keys `report`, `values`, and `protected`, the correct label, no `target` key, and a complete finite vector in the original order. The rule reads that vector without modifying it or searching other fields.

The check identifies 49 complete requested vectors: forty from GLM and nine from DeepSeek. All are execution-bound; 48 pass preservation and one fails it. Fourteen other undetermined outputs fall outside the rule and remain unclassified. These 49 content matches belong to artifacts that fail the original report contract (Appendix B.1.2).

4.4 Completion gaps and their numerical content

For n paired records, define $\Delta S = (s_c - s_n)/n$ from the correct- and neutral-start strict-success counts. If content classification gives $\Delta C \in [L_C, U_C]$, the separation satisfies $\Delta S - \Delta C \geq \Delta S - U_C$. Unclassified vectors determine the content bounds and are not scored as incorrect.

We retain 43 jointly valid, same-version pairs from 144 planned pairs: twenty DeepSeek and 23 GLM pairs across ten roots (Table 12). Correct starts yield 41 strict passes and 43 confirmed vectors; neutral starts yield 13 passes and at least 37 confirmed vectors, with six unclassified.

The completion gap is $28/43 = 65.1$ points, while the content gap lies in $[0, 6/43]$, or 0–14.0 points. Their separation is therefore at least $22/43 = 51.2$ points. Equivalently, neutral starts leave at least 24 vector-matching failures, versus two for correct starts (Appendix B.1.3).

We fix the content-reading rule before 48 Mini/Haiku sessions on the twelve original content roots, comparing correct and neutral starts under data-field background C.

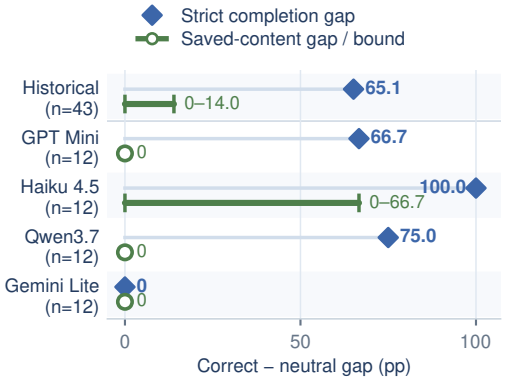

Figure 2: **Completion and saved-content gaps.** Diamonds: completion; circles and capped segments: content. Bounds resolve unclassified vectors. Units are percentage points; full configurations appear in Table 1.

Cross-developer starting-program replications. Mini yields 8/12 versus 0/12 strict successes, although all 24 vectors match: a 66.7-point completion gap with zero content gap. Haiku yields 12/12 versus 0/12 successes; twelve correct-start and four neutral-start vectors are confirmed, while eight neutral contents remain unclassified. Its 100-point completion gap exceeds its possible content gap by at least 33.3 points.

A separate 24-session Qwen replication yields 12/12 versus 3/12 strict successes, with every saved vector matching: a 75.0-point completion gap and zero content gap. Gemini's 24 sessions yield 12/12 successes and content matches under each start. Every trajectory in these four replications has final-source binding (Figure 2; Appendices B.2 and B.3). Because correct starters supply both implementations, the factorial comparison below separates their availability.

5 Separating computation and delivery support

5.1 Two implementations under a shared contract

Figure 3 illustrates why numerical computation and completed delivery must be distinguished. All arms share the task contract, answer-free output example, checking reminder, public metadata, and execution limits. The agent has access to neither private references nor program-test feedback.

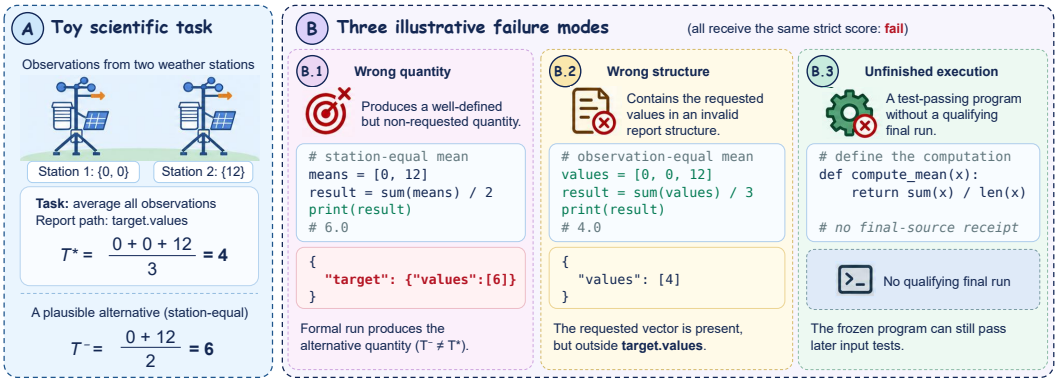

Figure 3: **The same failed score, three different evidence states.** A: a constructed weather-station task requests the observation-equal mean and the report path `target.values`. B: code and report excerpts distinguish a non-requested quantity, requested values in an invalid report structure, and a test-passing program without qualifying final-source execution evidence. These are illustrative examples, not sampled trajectories.

Computation support returns the requested numerical object. Delivery support packages the supplied values and public metadata for the common writer, without performing scientific calculations or correcting those values.

The four arms are **neither (00), computation only (10), delivery only (01), and both (11)**. Code use is optional, and the agent remains responsible for integration, execution, and submission. Each arm therefore varies which implementation is available to the agent. For the historical twelve-root, two-model study, the prespecified single-support contrast is

$$\Delta_{\text{swap}} = \frac{1}{12 \cdot 2} \sum_{r=1}^{12} \sum_{m=1}^{2} \{\mathbb{E}[S_{rm}^{\pi_0}(01)] - \mathbb{E}[S_{rm}^{\pi_0}(10)]\}, \tag{2}$$

where π_0 is the original execution protocol and expectations are over fresh sessions on the fixed roots and endpoints. Because the historical schedule spans protocol amendments, we report its realized completion contrast as a range over the missing outcomes. The common-protocol expectation remains a distinct target. The Mini–Haiku replication uses its own frozen assignments and slot-swap test (Section 6.4); Qwen and Gemini form separate later cohorts.

Development screen. A separate 32-position randomized development screen meets its prespecified investment rule; its design and complete results appear in Appendix F.6.

6 Completion under computation and delivery support

6.1 The historical twelve-root study

The historical support study crosses the twelve roots in Section 3 with DeepSeek-v3.2, GLM-5.2, and four arms: 96 positions with equal root and model weights. External-task identities were selected before the screen; final contracts and test packages were completed afterward, before model calls.

After 73 recorded positions, a continuation attempts the remaining 23 with unchanged scientific inputs, amended error isolation, and concurrent scheduling. The completed schedule contains 93 valid and three protocol-invalid outcomes. Appendix C.1 and Table 17 detail the execution stages and missing endpoints.

6.2 The same failed score, different unfinished obligations

The complete historical matrix appears in Figure 4A. Of 93 valid trajectories, 65 pass strict acceptance and 91 pass G. The formal-return record contains 81 correct returns; twelve trajectories have

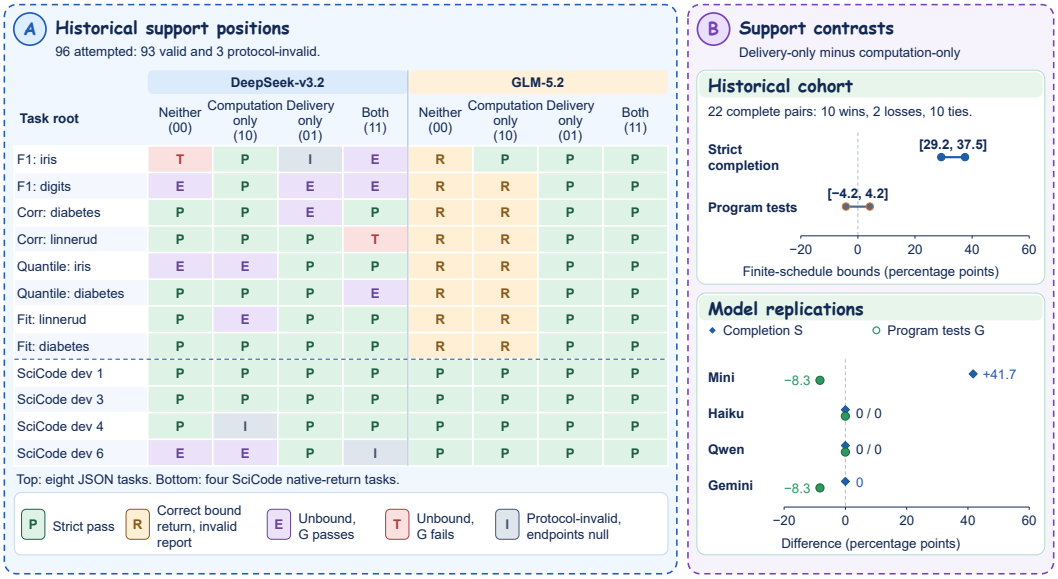

Figure 4: **Support contrasts and their underlying evidence.** A: all 96 historical positions, including three invalid outcomes. B, top: historical completion and program-test bounds for delivery-only minus computation-only. B, bottom: model-specific differences in the Mini–Haiku, Qwen, and Gemini replications. The Mini–Haiku aggregate is +20.8 points for S and -4.2 for G (exact $p = 0.125$ for S). Letter codes are defined in panel A; Table 23 gives arm-level counts, and Appendix C details cohort-specific failure states.

no qualifying return. Twenty-six of 28 strict failures pass G, including thirteen of fourteen without a supplied kernel. Portable replay covers all fourteen failures that began without a supplied kernel. Appendix C.1.4 documents the replay and measurement controls.

Correct returns, invalid reports. GLM has fifteen JSON failures without delivery support: seven with a kernel and eight without. All have correct final-source-bound returns and pass G, but fail schema; seven also fail preservation structure. These cases locate the unfinished requirement in report construction, including when the agent produces the computation.

Missing qualifying final-source execution evidence. DeepSeek's thirteen failures exhaust the tool budget without final binding. Twelve lack a qualifying return; one returns correctly from an earlier source version. Eleven programs pass G; two encounter execution exceptions on the test inputs. Binding records a formal receipt for the final source, separately from any shell execution.

6.3 Completion Contrasts and Their Program Evidence

GLM's seven delivery-only versus computation-only wins span all four JSON task families. Computation-only support leaves seven structural failures despite correct formal returns. DeepSeek's ten complete pairs contain three delivery wins, two losses, and five ties. The native-return stratum has 28/30 valid successes; its two valid failures pass G but lack qualifying final-source execution evidence.

Across both models, 22 complete pairs contain ten wins, two losses, and ten ties. Retaining all 24 planned pairs gives a strict-completion range of $[29.2, 37.5]$ percentage points, compared with $[-4.2, 4.2]$ for program tests (Figure 4B). Delivery-only versus neither support is $[37.5, 41.7]$ points. These finite-schedule comparisons quantify a larger completion difference than program-test difference. Missing primary endpoints and protocol amendments leave the original common-protocol confirmation unresolved (Appendix C.1).

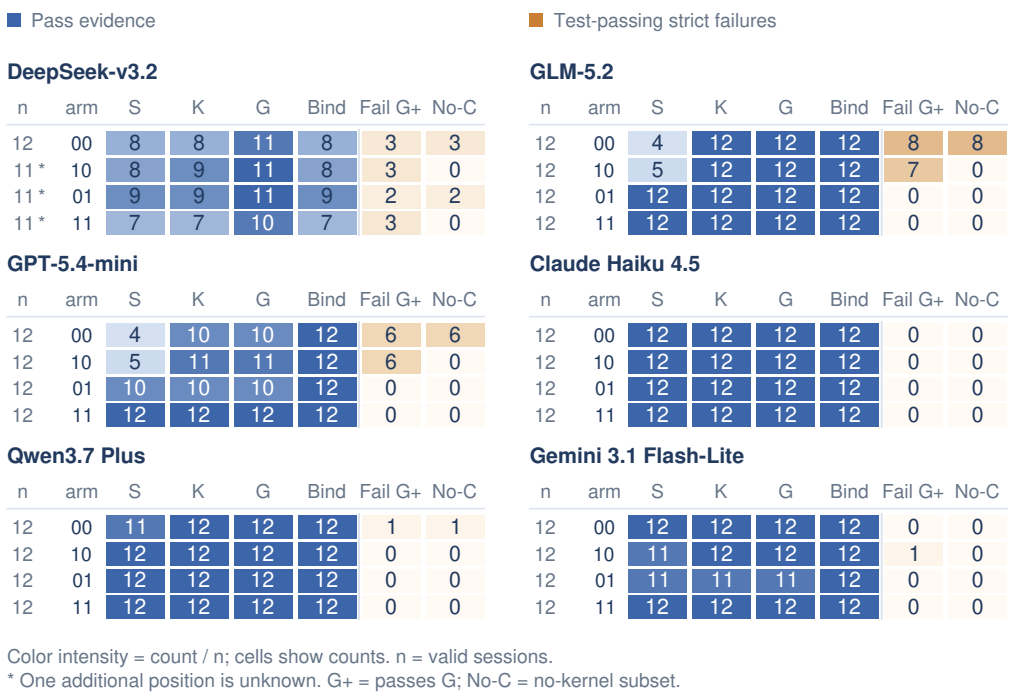

Figure 5: **Support evidence across six configurations.** Cells show counts; saturation is count divided by valid-session n. Blue: $S/K/G$/binding; orange: test-passing failures and their nested no-kernel subset. Arms 00/10/01/11 denote neither/computation/delivery/both. Asterisks mark historical unknowns. Cohorts and exact counts follow Tables 1 and 23.

6.4 Complete cross-developer replications

GPT-5.4-mini and Claude Haiku 4.5 repeat the twelve roots and four arms in 96 sessions, with packages, tests, analysis, and slots fixed before calls. They share gateway tool semantics and ceilings, with model-specific settings (Appendix C.2). The primary statistic equally weights these two models and twelve roots, independently of historical records.

All 96 positions are valid. Mini's twelve test-passing failures have correct bound returns but invalid reports; six start without a supplied kernel. Its five remaining failures comprise four incorrect F1 returns and one invalid native return type. Two numerical failures occur with delivery-only support: the constructor packages the agent's incorrect values without correcting them. Haiku completes all 48 support tasks (Figure 5). These tasks and program interfaces differ from its starting-program comparison.

Mini's delivery-minus-computation difference is $+41.7$ points for S and -8.3 for G; Haiku's differences are zero. The equal-weight means are $+20.8$ and -4.2 points. The prespecified two-sided slot-swap test of the sharp no-effect null gives $p = 0.125$, from six wins, one loss, and seventeen ties; it does not reject at 0.05. Appendix D.1 adds conservative intervals and paired task-composition sensitivity; the primary test is unchanged.

Qwen3.7 Plus yields 11/12 strict successes without support and 12/12 in each supported arm. All 48 trajectories have correct formal returns, passing program tests, and final binding. Its sole failure is an Iris-quantile report produced without a supplied kernel. The delivery-only minus computation-only contrasts are zero for both S and G (Appendix C.3).

Gemini 3.1 Flash-Lite completes 46/48 tasks, with 47 correct returns, 47 program-test passes, and 48 final bindings. A computation-only report fails schema despite a correct return and G pass; a delivery-only native result is incorrect and fails G. Its contrasts are zero for S and -8.3 points for G. The positive $\Delta S - \Delta G$ comes from lower program-test performance, with completion unchanged (Appendix C.4).

Across these configurations, correct numerical returns can fail reporting, while validly structured and bound deliveries can contain incorrect values. Program and execution evidence therefore help identify the unfinished requirements behind completion contrasts.

7 INTERPRETING SCIENTIFIC CODING FAILURES

7.1 IMPLICATIONS AND RELATION TO PRIOR WORK

The support records expose distinct combinations of computation, reporting, and execution evidence (Table 2).
Report requires a correct bound return and $G = 1$, but invalid schema. *Return* denotes a bound numerical or return-type failure; *Unbound* denotes missing final-source binding. The profiles partition observed failures, although individual checks may fail together.
These profiles distinguish correct returns in invalid reports from incorrect returns and missing final-source execution evidence. Program tests and original execution records answer different questions.

Table 2: **Support failure profiles.** Counts partition valid strict failures across the four arms. Three historical unknowns are excluded; configurations and cohorts follow Table 1.

Model	Fails	Report	Return	Unbound
DeepSeek	13	0	0	13
GLM	15	15	0	0
Mini	17	12	5	0
Haiku	0	0	0	0
Qwen	1	1	0	0
Gemini	2	1	1	0

Beyond content versus format. Data-science, scientific-code and repository benchmarks span complementary workflows (Lai et al., 2023; Sun et al., 2026; Siegel et al., 2024; Bogin et al., 2024; Xu et al., 2026). JSONSchemaBench studies constrained-decoding coverage, efficiency and quality; DeepJSONEval studies nested extraction (Geng et al., 2025; Zhou et al., 2025). Our added comparison quantifies how much a completion gap can exceed its saved-content counterpart, while retaining the original task score.

Beyond program tests alone. Agent interfaces shape execution (Yang et al., 2024; Ding et al., 2026); provenance research and ECP address execution histories and portable evaluation (Wang et al., 2026; Wattamwar et al., 2026). Our binding check is a specific source–run–artifact relation. Test augmentation and oracle-less incoherence address complementary correctness questions (Liu et al., 2023; Allamanis & Yin, 2025; Valentin et al., 2026). Here, original execution and program-source evidence distinguish supplied computation left undelivered from agent-produced computation that passes the tests but lacks a final receipt. Appendix E.2 specifies the evidence profile and candidate deployment checks.

7.2 SCOPE

The tasks are controlled and researcher-exposed, with shared sources and one session per cell in the model extensions. Historical content uses selected pairs and a post-hoc rule; replication rules precede calls, with eight Haiku artifacts unclassified. Historical support spans protocol amendments, and DeepSeek's pattern occurs at the stated tool limit. JSON and native tasks use different computations; G measures behavior on five fixed inputs. Broader workflows and repeated sessions are needed to assess transfer and stability. Appendix F.7 separately reports the contract study with a defective public example.

8 CONCLUSION

A failed score marks an unfinished task but does not identify which requirement remains unmet. Starting-program advantages can substantially exceed saved-content differences. Support comparisons distinguish correct returns in invalid reports, incorrect computations, and missing final-source execution evidence. Joint numerical, program-test, and execution records show which completion differences accompany changes in tested computation and which coexist with unfinished reporting or execution obligations.

AI USE STATEMENT

Generative AI tools assisted with literature searches, research and contract design, data preparation, implementation and debugging, reference calculations, experimental execution and analysis, and manuscript writing, figures, and typesetting. The computational checks reported here are author-side, AI-assisted checks. Responsibility for verifying the AI-assisted work and approving the submitted content rests with the authors.The authors have reviewed and verified the technical content, experimental results, and interpretations in the final manuscript, and take full responsibility for the submitted work.

REPRODUCIBILITY STATEMENT

Appendices A.2, F.6, and C.1 document reference checks, measurement, and execution protocols. The historical core bundle and five-program extension described in Appendix E.1 support saved-record arithmetic and computation-only replay of all fourteen unsupplied-kernel failed programs; the 96-position supplement reconstructs historical counts and runtime strata. Separate replication directories document the Mini–Haiku, Qwen, and Gemini analyses (Appendices B and C). Saved-workspace remeasurement agrees with all 48 Mini–Haiku starting-program records. The manuscript archive provides source-pinned plotting inputs, reproducible scripts, and vector exports, and supports paper compilation and figure regeneration. The separate experimental supplement contains the saved-result arithmetic and program-replay materials described in these appendices.

REFERENCES

Miltiadis Allamanis and Pengcheng Yin. Disproving program equivalence with LLMs. *arXiv preprint arXiv:2502.18473*, 2025. URL https://arxiv.org/abs/2502.18473v1.

Ben Bogin, Kejuan Yang, Shashank Gupta, Kyle Richardson, Erin Bransom, Peter Clark, Ashish Sabharwal, and Tushar Khot. SUPER: Evaluating agents on setting up and executing tasks from research repositories. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 12622–12645. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.702. URL https://aclanthology.org/2024.emnlp-main.702/.

Song Chen. Beijing Multi-Site Air Quality. UCI Machine Learning Repository, 2017. URL https://doi.org/10.24432/C5RK5G. Dataset, CC BY 4.0.

Xinyun Chen, Maxwell Lin, Nathanael Schaerli, and Denny Zhou. Teaching large language models to self-debug. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2460396f2d0d421885997dd1612ac56b-Abstract-Conference.html.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f12b4df26344f3be803c06b555252efe-Abstract-Conference.html.

Deming Ding, Shichun Liu, Enhui Yang, Jiahang Lin, Ziying Chen, Shihan Dou, Honglin Guo, Weiyu Cheng, Pengyu Zhao, Chengjun Xiao, Qunhong Zeng, Qi Zhang, Xuanjing Huang, Qidi Xu, and Tao Gui. OctoBench: Benchmarking scaffold-aware instruction following in repository-grounded agentic coding. arXiv preprint arXiv:2601.10343v2, 2026. URL https://arxiv.org/abs/2601.10343v2.

Saibo Geng, Hudson Cooper, Michał Moskal, Samuel Jenkins, Julian Berman, Nathan Ranchin, Robert West, Eric Horvitz, and Harsha Nori. JSONSchemaBench: A rigorous benchmark of structured outputs for language models. *arXiv preprint arXiv:2501.10868v3*, 2025. URL https://arxiv.org/html/2501.10868v3.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830. URL https://doi.org/10.1080/01621459.1963.10500830.

Allison Marie Horst, Alison Presmanes Hill, and Kristen B. Gorman. *palmerpenguins: Palmer Archipelago (Antarctica) penguin data*, 2020. URL https://doi.org/10.5281/zenodo.3960218. R package version 0.1.0.

Sihan Hu, Lyuhan Huang, Youjin Deng, and Kun Chen. SciCode-Verified: How benchmark defects underestimated the scientific-coding ability of language models. arXiv preprint arXiv:2608.04975v1, 2026. URL https://arxiv.org/abs/2608.04975v1.

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The UCI machine learning repository, n.d. URL https://archive.ics.uci.edu. Accessed September 26, 2026.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-Tau Yih, Daniel Fried, Sida Wang, and Tao Yu. DS-1000: A natural and reliable benchmark for data science code generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 18319–18345. PMLR, 2023. URL https://proceedings.mlr.press/v202/lai23b.html.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0943. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/43e9d647ccd3e4b7b5baab53f0368686-Abstract-Conference.html.

Do Xuan Long, Hai Nguyen Ngoc, Tiviatis Sim, Hieu Dao, Shafiq Joty, Kenji Kawaguchi, Nancy F. Chen, and Min-Yen Kan. LLMs are biased towards output formats! Systematically evaluating and mitigating output format bias of LLMs. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 299–330. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.15. URL https://aclanthology.org/2025.naacl-long.15/.

Luc Moreau and Paolo Missier. PROV-DM: The PROV data model. W3C recommendation, World Wide Web Consortium, 2013. URL https://www.w3.org/TR/2013/REC-prov-dm-20130430/. Edited by Luc Moreau and Paolo Missier.

NOAA CO-OPS. CO-OPS data retrieval API. National Oceanic and Atmospheric Administration, Center for Operational Oceanographic Products and Services, n.d. URL https://api.tidesandcurrents.noaa.gov/api/prod/. Accessed September 26, 2026.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. URL https://jmlr.org/papers/v12/pedregosa11a.html.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/6c0e99d736da621403018ca7b32b1a4d-Abstract-Conference.html.

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:2409.11363v2, 2024. URL https://arxiv.org/abs/2409.11363v2. First submitted September 17, 2024; version 2, June 22, 2026.

Weifeng Sun, Ye Fan, Yuchen Chen, Gou Tan, Jieke Shi, Yuan Yidi, Swee Liang Wong, Jonathan Pan, and David Lo. SciCodePile: A 128GB corpus and executable benchmark for challenging scientific code generation. *arXiv preprint arXiv:2607.19104v1*, 2026. URL https://arxiv.org/html/2607.19104v1.

Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, Shengyan Liu, Di Luo, Yutao Ma, Hao Tong, Kha Trinh, Chenyu Tian, Zihan Wang, Bohao Wu, Yanyu Xiong, Shengzhu Yin, Minhui Zhu, Kilian Lieret, Yanxin Lu, Genglin Liu, Yufeng Du, Tianhua Tao, Ofir Press, Jamie Callan, Eliu Huerta, and Hao Peng. SciCode: A research coding benchmark curated by scientists. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0963. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/36850592258c8c41cecdaa3dea5ff7de-Abstract-Datasets_and_Benchmarks_Track.html.

Athanasios Tsanas and Max Little. Parkinsons Telemonitoring. UCI Machine Learning Repository, 2009. URL https://doi.org/10.24432/C5ZS3N. Dataset, CC BY 4.0.

Thomas Jean-Michel Valentin, Ardi Madadi, Gaetano Sapia, and Marcel Böhme. Incoherence as oracle-less measure of error in LLM-based code generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(39):33305–33313, 2026. doi: 10.1609/aaai.v40i39.40616. URL https://ojs.aaai.org/index.php/AAAI/article/view/40616.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, 2020. doi: 10.1038/s41592-019-0686-2. URL https://doi.org/10.1038/s41592-019-0686-2.

Yiqi Wang, Jiaqi Zhang, Zhangkai Wu, Taotao Cai, Zirui Liu, Qingqiang Sun, Zequn Sun, Manqing Dong, Mingkai Zheng, Xuefei Yin, and Yanming Zhu. From agent traces to trust: A survey of evidence tracing and execution provenance in LLM agents. *arXiv preprint arXiv:2606.04990v4*, 2026. URL https://arxiv.org/html/2606.04990v4.

Aniket Wattamwar, Manav Anandani, and Mrunal Kakirwar. The evaluation context protocol (ECP): A portable contract for AI agent evaluation. *arXiv preprint arXiv:2608.19263v1*, 2026. URL https://arxiv.org/html/2608.19263v1.

xarray developers. xarray.Dataset.from_dict. Official API documentation, n.d.a. URL https://docs.xarray.dev/en/v2024.11.0/generated/xarray.Dataset.from_dict.html. Accessed September 26, 2026; experimental consumer pinned to xarray 2024.11.0.

xarray developers. xarray.open_dataset. Official API documentation, n.d.b. URL https://docs.xarray.dev/en/v2024.11.0/generated/xarray.open_dataset.html. Accessed September 26, 2026; experimental consumer pinned to xarray 2024.11.0.

Congying Xia, Chen Xing, Jiangshu Du, Xinyi Yang, Yihao Feng, Ran Xu, Wenpeng Yin, and Caiming Xiong. FOFO: A benchmark to evaluate LLMs' format-following capability. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 680–699. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.40. URL https://aclanthology.org/2024.acl-long.40/.

Zhipeng Xu, Jiahao Lu, Yining Zheng, Yuxin Wang, and Xipeng Qiu. SWE-bench Science: Can coding agents resolve engineering tasks in science? arXiv preprint arXiv:2608.19799v2, 2026. URL https://arxiv.org/abs/2608.19799v2. Version 2, September 1, 2026; first submitted August 20, 2026.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://papers.neurips.cc/paper_files/paper/2024/hash/5a7c947568c1b1328ccc5230172e1e7c-Abstract-Conference.html.

Zhicheng Zhou, Jing Li, Suming Qiu, Junjie Huang, Linyuan Qiu, and Zhijie Sun. DeepJSONEval: Benchmarking complex nested JSON data mining for large language models. *arXiv preprint arXiv:2509.25922v1*, 2025. URL https://arxiv.org/html/2509.25922v1.

APPENDIX CONTENTS

A Contracts, numerical checks, and execution evidence **15**

A.1 Task contracts and exact public inputs 15

A.2 Numerical references and execution verification 17

A.3 Execution binding and acceptance rules 18

A.4 Replication execution settings 20

B Starting programs: completion and saved numerical content **21**

B.1 Three-target study and saved-content analysis 21

B.2 Complete cross-developer starting-program replication 24

B.3 Qwen and Gemini: separation and a ceiling boundary 25

C Support packages: completion, program behavior, and failure states **26**

C.1 New-task computation-by-delivery study 26

C.2 Complete Mini–Haiku support replication 29

C.3 Qwen support replication and cohort provenance 30

C.4 Gemini support replication and cohort provenance 31

C.5 Exact counts and evidence profiles 31

D Statistical interpretation and sensitivity **34**

D.1 Bounds, sampling intervals, and task composition 34

E Reproducing the evidence **35**

E.1 Evidence map and executable core material 35

E.2 Reusable evidence records and possible uses 36

F Boundary conditions and development studies **37**

F.1 Original identities and context sensitivity 37

F.2 Crossing the starting implementation 39

F.3 Wording versus implementation layout 39

F.4 Corrected new sources: a non-reproduction boundary 40

F.5 Menus and component-score information 41

F.6 Computation-by-delivery development screen 42

F.7 Contract comparison: defect and unresolved correction 43

The appendices follow the argument: measurement, the two principal comparisons, statistical interpretation, reproducibility, and boundary studies. Entries link directly to their sections. Reused subsets remain part of their parent cohorts. Planned positions, valid trajectories, task roots, and test inputs are different counting units. Unknown outcomes remain unresolved; five test inputs yield one program-level G outcome. Content and schedule bounds describe incomplete finite records, not sampling confidence intervals.

A CONTRACTS, NUMERICAL CHECKS, AND EXECUTION EVIDENCE

This appendix defines the public contracts, complete numerical objects, and execution evidence, and summarizes their validation controls.

A.1 TASK CONTRACTS AND EXACT PUBLIC INPUTS

A.1.1 NOTATION AND COMPUTATIONAL RELATIONS

Table 3: **Experimental notation at a glance.** Codes identify the experimental conditions and interface interventions. The factorial study fixes the original-context baseline B and the initial target T_0.

Dimension	Code	Meaning
Fixed definitions	T_0, T_1	Original and revised numerical definitions.
Roles in one task	T^*, T^-	Requested and non-requested target, selected by public rules.
Applicability	O / A / I	No update / applies to this report / addressed elsewhere.
Context	B / F	Original context / model-written replacement context.
Context	H / R	Strong procedural reminder / public-rule-resolved contract.
Wording	U0 / U1	Original wording / same wording plus a fixed target-source instruction.
Program layout	K0 / K1	Original implementation / both implementations available through a switch.
Definition order	D0 / D1	Two orders of the same named definitions; archived as E0 / E1.
Failure count	q	Complete non-requested numerical target with original execution binding.

Table 4: What changes and what must not. The first two relations share clinical observations; the last two share station observations. The twelve roots instantiate these four constructions.

Relation (3 roots each)	Original T_0	Revised T_1	Protected outputs
Weighted fitting	Record-equal full/fold OLS, OOF predictions and MSE	Subject-equal weights in each fit and risk aggregation	Rows, subject counts, folds, raw z mean
Fixed-prediction risk	Record-equal MSE	Equal mean of subject MSEs; never refit	Same metadata and every supplied prediction
Station aggregation	Mean of observed x records	Equal mean of station x means	Marginal summaries and counts
Observation support	Mean of $x - y$ on joint support	Marginal mean(x) minus marginal mean(y)	Marginal summaries and joint count

Each relation has three original roots and two additional roots for the crossing study. Protected objects are required outputs; reference implementations and source-linked data slices are distinct from the public-message display copies.

A.1.2 PUBLIC-INPUT FILES AND CONDITION CONSTRUCTION

The thirteen unchanged `examples/` display files for `o3-station-june` are: `original-task.json.txt`, `original-config.json.txt`, `original-analysis.py.txt`; `original-evidence.json.txt`, `applicable-evidence.json.txt`, `inapplicable-evidence.json.txt`; `B-initial-0-system.txt`, `B-common-prefix.txt`, `B-common-suffix.txt`; `H-appended-reminder.txt`; `F-initial-0-system.txt`, `F-replacement-span.txt`; and `R-resolved-contract.json.txt`. They preserve indentation and soft wrapping, not a complete raw-message collection. Byte-identical display counterparts are in the separate supplement's `public-input-excerpts/` (Appendix E).

O has no rule; A has one applicable rule; I addresses the update to another report. Task, configuration, and starting program are otherwise identical. B concatenates the prefix, task under `TASK`, O

evidence under `CURRENT EVIDENCE`, and suffix; H appends its reminder. F preparation receives task/configuration/original evidence under file-name keys, with no numerical solution or code requested. Its returned code-fenced span replaces only the task/evidence text; public files remain accessible. Only fence extraction is recorded. The replacement retains no rules yet requests station-equal revision. R supplies applicability, source locations, and input hashes without an answer. B/F/R belong to the transfer block, H to the reminder block; canonical raw serializations are separate assets.

The immutable task and B/H/F/R inputs check the highest-version rule before report identity. O has zero rules and A/I one each, so competing-version precedence is not tested. Both targets use the same observations; raw-evidence manifests record CSV identity. Appendix F.2 later separates the contract from the starting implementation.

A.1.3 Complete representative contract and starting program

The complete public contract for `o3-station-june` is rendered here in compact form; the four `original-*` display files named above retain its original wording and serialization. The task names `data.csv`, `evidence.json`, and output `result.json`. Its fields are `row_id` (author-generated source-order identifier), `group` (original subject or station ID), and $x/y/z$ (O3/NO2/SO2). The initial `config.json` is `{"report_id":"o3-station-june","kind":"mean","version":1}`; O-condition `evidence.json` is `{"rules":[]}`.

The highest-version rule applies only if its `report_id` equals the task ID; otherwise the original contract remains. Originally, `target.value` is the mean of all non-NA x records and `target.estimand` is `observation_equal`. The applicable revision averages the non-NA x mean of each station with equal station weights and sets `target.estimand` to `station_equal`; every station has observed x, and no imputation is allowed. Protected values remain record-level marginal summaries/counts, with `joint_count` counting rows with both x and y observed.

The report must contain exactly `target` and `protected` at top level and the nested keys constructed below, with no field renaming, NaN, or infinity. Row order is preserved. Absolute and relative tolerances are 10^{-9} and 10^{-7}, respectively. Only public files and Python's standard library may be used; data changes, model fitting, network access, and installation are forbidden. The numerical results are authored finite-sample contracts, not scientific or clinical recommendations.

The complete supplied starting program follows, with only whitespace and line breaks reformatted. In particular, `station_counts` is a mapping from each original group ID to its record count, and each marginal mean excludes NA entries in its own column.

```
import csv,json,math
from collections import Counter
rows=list(csv.DictReader(open("data.csv")))
cfg=json.load(open("config.json"))
xs=[float(r["x"]) for r in rows if r["x"]!="NA"]
ys=[float(r["y"]) for r in rows if r["y"]!="NA"]
zs=[float(r["z"]) for r in rows if r["z"]!="NA"]
joint=[float(r["x"])-float(r["y"]) for r in rows
       if r["x"]!="NA" and r["y"]!="NA"]
if cfg["kind"]=="mean":
    target={"estimand":"observation_equal","value":math.fsum(xs)/len(xs)}
else:
    target={"estimand":"paired","value":math.fsum(joint)/len(joint)}
protected={
    "rows":len(rows),
    "station_counts":dict(sorted(
        Counter(r["group"] for r in rows).items())),
    "x_count":len(xs),"y_count":len(ys),"joint_count":len(joint),
    "x_mean":math.fsum(xs)/len(xs),"y_mean":math.fsum(ys)/len(ys),
    "z_mean":math.fsum(zs)/len(zs)}
with open("result.json","w") as f:
    json.dump({"target":target,"protected":protected},
              f,allow_nan=False)
```

A.2 NUMERICAL REFERENCES AND EXECUTION VERIFICATION

Table 5: **What the extra reference comparison adds.** Constructed station-mean examples, not agent observations. The requested mean is 4 and the alternative is 6. The shaded outputs 6 and 5 have identical four component bits but different numerical identities. Here 1/0 means satisfied/not satisfied; – means the target is absent. Preservation and schema can fail together. Measurement-unknown trajectories are outside this table.

Output	Current	Preserve	Schema	Bound	Strict	Identity / failed obligation
4	1	1	1	1	1	Current; all obligations met
6	0	1	1	1	0	Non-requested target
5	0	1	1	1	0	Other numerics
4	1	0	0	1	0	Current; protected field absent
4	1	1	1	0	0	Current; no final binding
–	0	1	0	0	0	Undetermined; target absent

A.2.1 NUMERICAL RECONSTRUCTION AND TOLERANCE

Each fit has an intercept and one feature. Inverse-count weights recalculate subject counts within each training complement; both full fits and out-of-fold (OOF) predictions use those weights. Subject-equal risk equally averages within-subject MSEs. Each fixed-prediction task supplies 5,875 record-equal predictions; all 17,625 serialized decimals are checked. The audit covers data slicing, canonical row order, and complete target/protected objects, not only a separating coordinate.

Original two-target acceptance uses

$$|u - v| \le \max\{a, r \max(|u|, |v|)\}, \qquad a = 10^{-9}, \quad r = 10^{-7}. \tag{3}$$

The three-target rule is separately frozen (Appendix B.1). A conservative radius for the symmetric predicate is

$$\rho(v) = \frac{\max(a, r|v|)}{1 - r} + 4[\mathrm{ulp}(v) + \mathrm{ulp}(a) + \mathrm{ulp}(r|v|)],$$

where ulp is a unit in the last place. `Decimal` starts from the exact binary64 representations; the four-ULP allowance covers outward rounding. Since $r \max(|u|, |v|) \le r(|v| + |u - v|)$ for accepted u, this radius encloses acceptance. Every root has a coordinate whose gap exceeds the sum of the two radii (minimum ratio 3,159), separating the complete objects. `acceptance_radius` in `audit_references_v1.py` and its input hashes are pinned in the report. The implementation uses the symmetric Python `math.isclose` predicate with these fixed tolerances; finite-input validation rejects NaN/infinity.

Table 6: Independent-implementation author CPU reference check on all twelve roots. Match counts complete T0/T1-plus-protection objects equal after conversion to binary64. Each ratio divides a pure numerical witness difference by the sum of conservative tolerance radii; values above one exclude dual matching. These are author-side checks of the reference implementations.

Root	Witness	$\lvert T_{0j} - T_{1j}\rvert$	Ratio	Match
Fit motor–PPE	Fold 4 slope	1.21449	645,351	2/2
Eval motor–PPE	MSE	0.0902368	6,771	2/2
Fit total–Shimmer(dB)	Fold 3 slope	0.6668	813,858	2/2
Eval total–Shimmer(dB)	MSE	0.144032	5,885	2/2
Fit motor–Jitter(Abs)	Fold 2 slope	6771.86	2,678,286	2/2
Eval motor–Jitter(Abs)	MSE	0.21267	15,646	2/2
O3 station mean	Mean	0.0474998	3,159	2/2
NO2 station mean	Mean	0.0742217	7,702	2/2
SO2 station mean	Mean	0.05853	21,300	2/2
NO2–SO2 support	Mean	0.278519	40,258	2/2
PM10–PM2.5 support	Mean	0.288138	105,584	2/2
O3–NO2 support	Mean	0.155875	28,800	2/2

Table 7: Observer, replay, and baseline checks from saved records; the author-side CPU checks are distinct from agent trajectories. Original grades are unchanged. Table 6 gives the complete per-root reference checks.

Author-side check	Recorded result
Observer controls	32 formal CPU executions plus two nonexecution stale-output checks: 34/34 prescribed behaviors pass; eight copied-trace fault checks are separate.
Final snapshots	180 replays; strict pass/pass 137, pass/fail 0, fail/pass 4, fail/fail 39; original/replay totals 137/141. Of 179 comparable outputs, 178 are byte-identical.
Original execution evidence	Three historical-input replays restore five files; 1,358 command receipts (322 formal) checked. Original E agrees for 180/180: 176 true, four false.
Transfer CPU gate	144 executions. Initial 24/36 (O/A/I: 12/0/12); float-equivalent 36/36; always-new 12/36; correct-target/wrong-protection 0/36.
Baseline comparison	DeepSeek B: 27/36 (O/A/I: 5/10/12); B minus initial: $+3$ overall ($-7/+10/0$). Descriptive, from saved records.

A.2.2 Observer validation and a constant-output counterexample

The v5 executor/v3 scorer controls cover stale output, code/input/output mutation, unrelated diagnostics, and legal output readback in ordinary/read-only modes. All 34 prescribed behavior checks pass; eight copied-trace fault checks are separate. An initial expectation that a constant emitter would omit `data.csv` from dependency membership was contradicted by trusted sandbox setup mentioning that path. The payload trace has no data-path read and its syntax tree opens only the output. A read-only assessment corrected the interpretation without rerunning the 32 executions or changing their report. The constant emitter passes fresh binding and strict matching: dependency membership and execution binding (archived as E) do not establish meaningful data use or formula execution. Missing/abnormal formal traces invalidate the trajectory.

A.2.3 Final-snapshot replay versus original-version replay

All 180 snapshots were replayed once in fresh CPU workspaces without saved results; private references/metadata were outside the candidate mount. Three use the original last execution's read-only mode; the case without an original formal run receives an ordinary-mode diagnostic. No replay is unknown or exits nonzero. The four replay-only passes are DeepSeek A cases: fit-motor-PPE B (no old output; later analysis change), NO2-station F (later analysis change), NO2–SO2-support B (configuration change), and PM10–PM2.5-support F (only a rejected planned call). The last case's static joint-support output becomes marginal-support output on replay. All four retain their original failure grades.

For the three changed versions, five historical input files were restored from saved writes/hashes. The fit reproduces its nonzero exit/missing output; the other two reproduce output bytes and return codes. This restores observed dependencies, not the operating-system or conversation history. Receipt reconstruction in Table 7 preserves original output-role/completeness facts.

A.2.4 Existing CPU controls

The 144 transfer controls cover twelve roots, O/A/I, and four controls (Table 7); all initial executions have fresh binding. Two earlier isolated CPU packages reconstruct six public selector contracts, reject four invalid inputs, and reaggregate 180 exported scores. These author-side controls, reference checks, and snapshot/historical-input replays are distinct from agent trajectories.

A.3 Execution binding and acceptance rules

A.3.1 What constitutes a qualifying execution?

A qualifying receipt identifies an actual formal invocation, execution/trace status, source/configuration, observed local dependencies, and output. Rejected planned calls and generic successful shell commands do not supply formal evidence. Ordinary/read-only modes retain their output-role and completeness requirements. For execution r, recorded manifest/output

D_r, O_r, final objects D_f, O_f, and byte digest H, binding requires

$$\text{receipt_valid}(r) \ \wedge \ H(D_r) = H(D_f) \ \wedge \ H(O_r) = H(O_f) \ \wedge \ \text{final_version}(r).$$

Manifest equality is entrywise over tracked paths. A relevant source/configuration/dependency/output change needs matching final-version evidence; independent diagnostic files do not invalidate it. Data/public task files are immutable. SHA256 uses the usual no-collision assumption; no adversarial-collision test was performed. Missing/abnormal traces are measurement-invalid, whereas validly observed absence of a matching final receipt fails binding.

This summarizes the existing version-consistency rule, without reconstructing absent traces. The portable requirement links versioned inputs, an execution, and its output (PROV entities and activity (Moreau & Missier, 2013)); notebook/scheduler adapters need separate validation and environment provenance. Binding alone does not prove formula execution, as the constant emitter shows. Statistics and record fields are in Appendices D and E.

A.3.2 Acceptance rules and the tested scope of G

Table 8: Acceptance predicates (v: reference), applied to every required coordinate. Numerical, schema, preservation, and binding checks remain separate.

Panel / object	Numerical predicate	Structure and preservation
Original two-target	$\lvert u-v\rvert \le \max(a, r\max(\lvert u\rvert, \lvert v\rvert))$; $a = 10^{-9}, r = 10^{-7}$	Complete finite objects, declared order; separate labels/protection checks.
Three-target content	Frozen complete-vector rule (Appendix B.1)	Fixed paths/labels; unclassified stays unclassified; separate schema/protection.
JSON support / tests	$\lvert u - v\rvert \le 10^{-9} + 10^{-7}\lvert v\rvert$	Exact-length ordered finite int/float list, not bool; declared nesting/protected metadata.
Native support / tests	$\lvert u - v\rvert \le 10^{-8} + 10^{-5}\lvert v\rvert$	Contract-specific normalized layouts/shapes, no broadcasting; frozen normalizer accepts numeric Booleans.

The development JSON suite uses five fixed SHA256-ranked subsets of $\lceil 0.65n \rceil$ rows, preserving source order (Appendix F.6). Later studies and replications retain their task-specific five-input packages. Shared-source inputs can overlap; the five tests yield one binary G per program, not five independent scientific problems. Table 9 measures selected-error discrimination, not domain-wide branch/boundary coverage or EvalPlus-style suite expansion (Liu et al., 2023).

Table 9: Fixed-test discrimination (Appendix C.1.4): two roots per JSON family, four wrong implementations per root. All 160 wrong-program/input outputs are legal finite vectors but fail numerically; all 40 correct-control tests pass. Dashes denote an unreported comparable census.

Task family	Relation-specific wrong implementations	Correct	Rejected
F1	Macro/weighted instead of micro F1	2/2	8/8
Correlation	Spearman/uncentered cosine instead of Pearson	2/2	8/8
Quantiles	Hyndman–Fan types 8/9 instead of type 7	2/2	8/8
Fitting	Through-origin/Theil–Sen instead of OLS	2/2	8/8
Total JSON	Each root also tests fixed original-input answers and reversed coordinates	8/8	32/32
Native tasks	Four SciCode adaptations; no matched semantic-mutant census	—	—

Native tasks cover conjugate gradient, Gauss–Seidel, incomplete Cholesky, and a spatial filter. The portable fourteen-program replay includes one native program, not a uniform native mutation audit. The unchanged-record strata below distinguish JSON/native tasks and the no-kernel subset of G-passing failures.

Table 10: Interface and computation-provenance strata from unchanged support records. F_G counts strict failures passing G; No-C is their subset without a supplied kernel, not a disjoint category. Unbound counts strict failures without final-source binding. JSON and native tasks have different computations; the rows describe evidence composition, not an interface ablation.

Model	Interface	n	Unk.	S	K	G	F_G	No-C	Unbound
DeepSeek	JSON	31	1	20	21	29	9	4	11
DeepSeek	native	14	2	12	12	14	2	1	2
GLM	JSON	32	0	17	32	32	15	8	0
GLM	native	16	0	16	16	16	0	0	0
Mini	JSON	32	0	16	28	28	12	6	0
Mini	native	16	0	15	15	15	0	0	0
Haiku	JSON	32	0	32	32	32	0	0	0
Haiku	native	16	0	16	16	16	0	0	0
Qwen	JSON	32	0	31	32	32	1	1	0
Qwen	native	16	0	16	16	16	0	0	0
Gemini	JSON	32	0	31	32	32	1	0	0
Gemini	native	16	0	15	15	15	0	0	0

A.4 Replication execution settings

All four adapters request nonstreaming chat completions, at most 24 replies and 24 charged tool units per trajectory, a reported 24,576-input-token ceiling, and 8,192 output tokens per call; temperature/top-p are omitted. Mini (AIHubMix) requests `reasoning_effort=none`; Haiku (AIHubMix) supplies no extended-thinking option; Qwen (Alibaba Cloud, Beijing) requests `enable_thinking=false`; Gemini (AIHubMix) requests `reasoning_effort=low`. These settings do not equalize compute or certify provider internals.

Full requested IDs are in Appendices C.2, C.3, and C.4. Mini's returned ID is `gpt-5.4-mini-2026-03-17`; the other service identifiers do not independently establish backend snapshots. Mini–Haiku is one frozen study; later Qwen/Gemini cohorts remain separate. Adapters provide no report repair, automatic final execution, or model fallback. Local tool errors can be handled within a trajectory; unresolved transport/measurement states are not replaced. The offline artifact contains inspection source, not a portable paid launcher.

B STARTING PROGRAMS: COMPLETION AND SAVED NUMERICAL CONTENT

This comparison asks whether a completion advantage is accompanied by an equally large saved-content advantage. Historical and replication cohorts retain separate records, reading rules, and denominators.

B.1 THREE-TARGET STUDY AND SAVED-CONTENT ANALYSIS

B.1.1 TASKS, REFERENCES, AND WHAT WAS HELD FIXED

Twelve author-constructed UCI roots (Kelly et al., n.d.) cover F1 (Wine, WDBC, Glass), correlation (red/white Wine Quality, Abalone), and quantiles/fitting (Auto MPG, Airfoil, Yacht). Shared sources are not independent domains. F1 predictions are deterministic author-generated, label-dependent perturbations, not trained-classifier outputs. Contract cards record input hashes, definitions/order, source links, shared sources, and development exposure. Selection/source order precede reference calculation; technical exclusions concern undefined/nonfinite outputs, unavailable metadata/resources, or nonseparable references, not model failures.

Requested/alternative roles rotate within each relation. Across backgrounds, the requested definition, data, and start are fixed: C describes fields; A/B describe alternatives a/b, explicitly not requested. These labels differ from earlier applicability/baseline codes. Complete vectors contain F1 by prediction column, ordered correlations, fixed quantiles, or slope/intercept/fixed-input predictions. Contracts fix missingness, ties, zero division, interpolation, and intercept conventions. Protected fields are CSV digest, row count, and header order; schema requires `target.report`, `target.values`, and `protected`. The report label does not encode numerical identity.

Before calls, library references (NumPy 2.3.3, SciPy 1.16.2, scikit-learn 1.7.2; Harris et al., 2020; Virtanen et al., 2020; Pedregosa et al., 2011) and independent formulas agree on complete vectors; all three references have pairwise disjoint acceptance regions under $|u - v| \leq 10^{-9} + 10^{-7}|v|$. These are saved author-side implementation checks.

B.1.2 POST-HOC READING OF SAVED NUMERICAL CONTENT

After the queue stopped, one GLM artifact motivated a fixed secondary rule, applied to all 180 records (174 valid):

1. Preserve original status, identity, strict/direction scores; invalid observations stay null.
2. Require an observed target with undetermined identity, exactly top-level `report`, `values`, `protected`, the required label, and no `target`.
3. Compare the entire finite vector, in original length/order, against all three frozen references. Reject Boolean/string entries, incomplete arrays, and ambiguous matches; do not search other paths, reorder, coerce, or insert values.
4. Annotate membership only: schema/strict failures remain; no rewriting or rerunning.

Exactly 49 qualify and match the request (GLM 40, DeepSeek nine), all from neutral starts with original binding; 48 preserve protected outputs and one does not. Fourteen further undetermined artifacts have extra top-level fields and remain unclassified.

In the motivating case the top-level array equals the weighted-F1 reference

$$(0.831586959146076,\ 0.18614244279729297,\ 0.6998431012312266),$$

distinct from micro/macro F1. Original current-numerics/preservation/schema/binding flags are false/true/false/true; identity remains undetermined. The extraction bit records the missing contract path despite matching values. Saved row/output/reference hashes and all 180 annotations are retained in the study archive, distinct from manuscript summary inputs.

B.1.3 HOW MUCH OF THE STARTING-STATE GAP IS A DEMONSTRATED CONTENT GAP?

For valid outputs, content intervals are $[1, 1]$ for original-current or secondary-rule confirmed requested vectors, $[0, 0]$ for established different-reference vectors, and $[0, 1]$ for unclassified undetermined targets. Invalid observations receive no interval. Averaging endpoints and subtracting

group intervals bounds the recorded content difference without independence or missing-at-random assumptions; it says nothing about unstarted positions or new inputs.

Table 11: Separate content coverage in the stopped three-target study. Confirmed content combines original current identity with the fixed secondary annotation; it is not strict success. All original grades are unchanged. The first two rows partition the sample, and the remaining rows subdivide them; they must not be summed together. Model/start coverage is unbalanced.

Subset	Valid	Strict	Original current	Post-hoc current	Confirmed content	Content unknown
Correct start	88	85	88	0	88	0
Neutral start	86	20	23	49	72	14
DeepSeek / correct	43	40	43	0	43	0
DeepSeek / neutral	32	20	23	9	32	0
GLM / correct	45	45	45	0	45	0
GLM / neutral	54	0	0	40	40	14

Correct-start content is $88/88$, neutral-start content $[72/86, 86/86]$, and their gap $[0, 14/86]$ (0–16.3 points), versus strict $85/88 - 20/86 = 73.3$ points. GLM has forty confirmed/fourteen unclassified neutral vectors, all 54 strict failures; DeepSeek has 32/32 confirmed neutral vectors, twenty strict passes. Coverage is unbalanced.

Of 144 planned pairs matched on root/model/background/repeat, 87 contain at least one unstarted member, two further pairs have an invalid member, and twelve further pairs cross runtime segments. The remaining 43 jointly valid same-segment pairs have strict transitions (both, correct-only, neutral-only, neither) of $(12, 29, 1, 1)$: difference $28/43 = 65.1$ points. All 43 correct vectors and 37 neutral vectors are confirmed; six neutral vectors remain unclassified, bounding the gap by $[0, 6/43]$ (0–14.0 points). The ten-root/two-model subset uses independent provider samples, shared roots, and post-outcome selection, not a randomized starting-state effect.

Table 12: Selection and root composition of the saved-content comparison: a selected 43-pair subset, not independent domains or a representative sample of all planned pairs.

Pair selection or composition			Count
Originally planned starter pairs			144
Excluded: a member unstarted			87
Excluded: a member protocol-invalid			2
Excluded: different execution-control versions			12
Included: jointly valid, same version			43
DeepSeek-v3.2 / GLM-5.2 pairs			20 / 23
Task root	Pairs	Task root	Pairs
correlation–abalone	2	fit–airfoil	3
correlation–wine–red	3	fit–auto–mpg	9
correlation–wine–white	3	fit–yacht	6
f1–wdbc	6	quantile–airfoil	3
f1–wine	3	quantile–yacht	5

All 36 public program contrasts (twelve roots, three backgrounds) share noncode inputs and parsing prefixes. Correct code supplies computation plus report construction; neutral code supplies neither. Section 5 separates these in a distinct modular-interface study. The intervals describe format/content separation (Long et al., 2025); requested-reference-only checking recovers matches, while alternatives identify non-requested quantities. They do not estimate repair effects.

B.1.4 Source study: directional endpoint and incomplete coverage

The plan has 96 root/start/model/repeat triples, with backgrounds assigned to slots. A complete a match with preservation/schema/binding scores $s = +1$, analogous b scores -1, and other valid outcomes score zero; invalid/unknown outcomes are null. The endpoint is $\sum_{i=1}^{96}(s_{i,A} - s_{i,B})/(2 \cdot 96)$;

both directional components must be positive for bidirectional following. Conditional randomization exchanges A/B while holding C's slot fixed.

An earlier 17-position version with hidden empty-search validation inconsistency is wholly excluded. The corrected runtime contributes 16 positions (15 valid, six strict); a prospective continuation isolates a bounded class of settled malformed-argument responses as job-local invalid, preserving inputs/old records and adding 164 positions without retry (159 valid, 99 strict). The retained 180 positions comprise 174 valid, 105 strict, six invalid; 108 planned positions are unstarted. A terminal two-call batch has malformed serialization and excessive remaining tool demand, executes no tools, and triggers the frozen protocol stop. It is neither mathematical failure nor efficacy stopping; readable static artifacts do not resolve invalid scientific fields.

Checkpoints 24/72/144 have zero qualified alternatives, but predeclared zero-event/worst-case-futility rules also require coverage. Missing outcomes therefore prevent early stopping; checkpoint 216 is not reached. Five A/B scores are missing in the first 60 triples and 77 in the full 96. For observed sums S_A, S_B and $m_A + m_B$ missing scores,

$$\Delta_N \in [(S_A - S_B - m_A - m_B)/(2N), (S_A - S_B + m_A + m_B)/(2N)].$$

Observed sums are zero: prefix bounds $[-5/120, 5/120]$, full-plan bounds $[-77/192, 77/192]$. These deterministic bounds retain the planned estimand, not confidence intervals; no confirmatory p-value is reported.

Table 13: Stopped three-target study. Original identities and strict scores are unchanged. The final column counts complete requested vectors under the narrow secondary rule; these remain failures. Each grouping partitions the same 180 records, and grouped rows must not be added together.

Subset	Rec.	Valid	Strict	Original current	Original undet.	Invalid	Post-hoc match
Correct start	90	88	85	88	0	2	0
Neutral start	90	86	20	23	63	4	49
DeepSeek	78	75	60	66	9	3	9
GLM	102	99	45	45	54	3	40
Load control	60	59	36	36	23	1	18
Display T_a	60	59	36	38	21	1	16
Display T_b	60	56	33	37	19	4	15

Table 14: All twelve prespecified roots, including unstarted positions. Root labels identify author-constructed computational tasks, not independent scientific domains. There are no original alternative identities among the observed rows. Post-hoc matches do not change this classification.

Root	Plan	Rec.	Valid	Strict	Undet.	Post-hoc match	Not started
correlation-abalone	24	15	13	9	4	3	9
correlation-wine-red	24	12	11	3	8	8	12
correlation-wine-white	24	9	8	3	5	2	15
f1-glass	24	15	15	8	6	6	9
f1-wdbc	24	15	15	9	4	4	9
f1-wine	24	15	15	8	6	6	9
fit-airfoil	24	15	14	7	6	6	9
fit-auto-mpg	24	21	21	17	4	2	3
fit-yacht	24	15	15	11	3	3	9
quantile-airfoil	24	18	18	12	6	4	6
quantile-auto-mpg	24	12	12	9	3	2	12
quantile-yacht	24	18	17	9	8	3	6

B.2 Complete cross-developer starting-program replication

B.2.1 Task coverage and pre-call specification

The same twelve content roots (distinct from support roots) use background C, correct/neutral starts, and two configurations: 48 fresh trajectories, 24 root–model pairs. Tasks were already exposed in research. Selection followed inspection of support-replication outcomes; before content calls, the matrix, original packages/references, reading rules/analysis, adapter, and budget policy were frozen. Inherited task/condition order is verified by HMAC reconstruction. Correct starts supply computation/report construction; neutral starts supply input reading. Packages are hash-checked against original conditions.

All 48 positions are valid, with no unknown, unstarted, retried, or replaced positions. Each endpoint runs serially with independent conversations/workspaces, using Appendix C.2's settings and original content tools/scorer. Returned IDs are `gpt-5.4-mini-2026-03-17` and `claude-haiku-4-5`. Settled local refusals permit within-trajectory recovery; uncertain measurement would stay unknown. No private-reference/content feedback is supplied.

B.2.2 Pre-call reading and finite-record analysis

Strict success retains numerical, preservation, label-inclusive schema, and final-binding checks. Contract-path classification identifies requested/non-requested vectors. The additional rule admits originally undetermined artifacts only with exactly `report`, `values`, `protected`, label `analysis_result`, and a complete ordered finite `values` vector. It reads only that field and leaves its values unchanged. It uses $10^{-9} + 10^{-7}|y|$ tolerance, rejects Booleans/nonfinite entries/duplicate JSON keys, and treats multiple hits as measurement errors.

Content is confirmed matching/nonmatching or unclassified. Strict success implies matching, not conversely. Each model uses all twelve planned pairs for strict differences, binary-resolution content bounds, and jointly constrained strict-minus-content bounds.

Table 15: All 48 starting-program positions. P: strict success and confirmed requested content. M: confirmed requested content but strict failure. U: strict failure with content unclassified by the pre-call reading rule, not a numerical-error label. All M/U records fail schema; * also fails preservation. Every position has final execution binding.

	GPT-5.4-mini		Claude Haiku 4.5	
Root	Correct	Neutral	Correct	Neutral
f1-wine	P	M	P	U
quantile-auto-mpg	P	M	P	U*
f1-wdbc	M	M	P	U
correlation-wine-white	P	M	P	U
fit-yacht	P	M	P	M
f1-glass	M	M	P	U
correlation-abalone	M	M	P	M
correlation-wine-red	P	M	P	U*
quantile-airfoil	M	M	P	M
fit-airfoil	P	M	P	U
quantile-yacht	P	M	P	M*
fit-auto-mpg	P	M	P	U

B.2.3 Results and interpretation

Mini strict success is 8/12 correct versus 0/12 neutral; all 24 vectors match. The strict/content gaps are 66.7/0 points; four correct and twelve neutral failures match numerically. Haiku is 12/12 versus 0/12; twelve correct and four neutral vectors match, eight neutral artifacts remain unclassified. Its strict gap is 100 points, content gap 0–66.7, and excess at least 33.3. Equal-model averages are 83.3, 0–33.3, and at least 50.0 points, separately from the historical 43 pairs.

Every output has final-source binding. All 28 strict failures fail schema, three Haiku neutral records also fail preservation, and twenty failures have confirmed requested vectors. Haiku's perfect support

completion versus zero neutral-start completion here compares different tasks/interfaces/starts; eight content objects here remain unclassified.

B.2.4 RESOURCES AND TRACEABILITY

Table 16: Starting-program scientific sessions only. Configured-rate cost estimates from returned usage, not invoices or wallet balances.

Model	Requests	Input tokens	Output tokens	USD
GPT-5.4-mini	162	541500	20723	0.499447
Claude Haiku 4.5	325	1900929	67244	2.461011

The 487 scientific requests cost an estimated $2.960458; technical usage and the earlier unresolved technical reservation are separate. Terminal checks cover descriptors, actual prompts, worker/record hashes, snapshots, and frozen code/inputs. Original-scorer/content remeasurement of saved workspaces agrees on all 48 results without new calls or program execution. The internal archive has 3,134 verified files, outside the compact supplement's scope. The separate supplement's `results/foreign-support-and-content/content48/` directory reproduces finite-record calculations, not an independent reader's raw-workspace remeasurement. The manuscript includes derived plotting summaries.

B.3 QWEN AND GEMINI: SEPARATION AND A CEILING BOUNDARY

These separate cohorts (Appendices C.3, C.4) use the same twelve content roots, C background, and pre-call rule; neither is pooled with historical pairs.

B.3.1 QWEN

Strict success is 12/12 correct versus 3/12 neutral; all 24 complete vectors match and all outputs have final binding. Nine neutral failures fail schema, one also preservation. Strict/content gaps are 75/0 points. Additional-input G belongs to the support study.

B.3.2 GEMINI

Both starts achieve 12/12 strict and 12/12 requested-vector matches; all 24 outputs have binding, schema, and preservation. Both gaps are zero, a ceiling boundary alongside Mini/Haiku/Qwen's observed separation. The complete Qwen/Gemini support/content root results appear together in Table 22.

C SUPPORT PACKAGES: COMPLETION, PROGRAM BEHAVIOR, AND FAILURE STATES

The support comparison independently varies a computation kernel and an answer-free delivery constructor. We retain the historical mixed-protocol cohort, the frozen Mini–Haiku replication, and the later Qwen and Gemini cohorts as separate analyses. Table 23 gives every model–arm denominator and endpoint count.

C.1 NEW-TASK COMPUTATION-BY-DELIVERY STUDY

C.1.1 TASK MATRIX AND EXECUTION STAGES

The historical plan crosses twelve roots, two models, and four arms: 96 trajectories and 24 root–model pairs for delivery-only minus computation-only. Eight author-constructed roots cover F1 (Iris, Digits), correlation (Diabetes, Linnerud), quantiles (Iris, Diabetes), and fitting (Linnerud, Diabetes). Four external roots adapt SciCode development tasks 1, 3, 4, and 6 (Tian et al., 2024): conjugate gradient, Gauss–Seidel, incomplete Cholesky, and a spatial filter. External identities were selected before the development screen; contracts, fixed inputs, and hashes were completed after that screen but before scientific calls. Shared datasets and computational families remain sources of dependence.

All arms receive the same specification, structural example or native-return contract, checking reminder, public metadata, and execution limits; either supplied implementation may be used optionally. Native adaptations add no JSON reporting obligation and do not produce official SciCode scores. Because computations and interfaces differ together across strata, their comparison does not isolate the interface. Arm slots were randomized within root–model blocks; each trajectory has a separate session and workspace. Historical model settings follow Appendix F.6, with no shared provider seed. The planned contrast weights all twelve roots and both models equally, including unfinished positions.

Table 17: **Coverage improves without equalizing running conditions.** Positions 1–73 and 74–96 differ in task/arm composition and execution control. Unbound counts use valid records. DeepSeek's 28/35 versus 4/10 is descriptive, not a concurrency effect. Table 18 resolves four control stages and their task composition.

Recorded segment	Model	Valid	Invalid	Strict S	Unbound
Initial records	DeepSeek	35	2	28	7
	GLM	36	0	24	0
September 21 suffix	DeepSeek	10	1	4	6
	GLM	12	0	9	0

Table 18: **Runtime-stratified support records.** Composition counts include invalid attempts; strict success and unbound counts use valid records. Original slots, inputs, code supports, scoring and per-trajectory limits are retained. Stage and task composition are not randomized against each other.

Stage (positions)	Model	Arms 00/10/01/11	JSON/ native	Valid/ invalid	S	Unbound
Initial (1–8)	DeepSeek	1/1/1/1	0/4	3/1	3	0
	GLM	1/1/1/1	0/4	4/0	4	0
First suffix (9–73)	DeepSeek	8/8/9/8	25/8	32/1	25	7
	GLM	8/8/8/8	24/8	32/0	20	0
Small continuation (74–81)	DeepSeek	1/1/1/1	3/1	4/0	2	2
	GLM	1/1/1/1	4/0	4/0	3	0
Expanded continuation (82–96)	DeepSeek	2/2/1/2	4/3	6/1	2	4
	GLM	2/2/2/2	4/4	8/0	6	0

The four stages in Table 18 retain the original slots, inputs, supports, numerical predicates, five program-test inputs, token ceilings, and 24-unit tool budget. Positions 1–8 ran in one serial worker and stopped at DeepSeek's computation-only SciCode 4 input-limit termination. An amendment isolated that termination and allowed only the never-started suffix, still stopping at the second cumulative invalid. A health/cost check at position 48 used no scientific effect estimate. The second stop, at position 73 (DeepSeek, Iris F1, delivery-only), followed rejection of a `write_file` batch; none of its operations ran. The reply reports 8,192 output tokens and a tool-call finish reason, which alone does not establish truncation. Both records retain null endpoints.

A separately frozen September 21 continuation attempted only positions 74–96, after initial results had been inspected. It isolated local invalid trajectories: positions 74–81 allowed one trajectory per endpoint, then an operational gate allowed at most two for positions 82–96. Transport unknowns retain reservations; two new unknown jobs pause an endpoint, and shared observation, environment, or accounting failures stop the queue. Concurrency decisions used no scientific effect estimates. Earlier records were never retried or replaced. Position 83 (DeepSeek, SciCode 6, both supports) ended with a tool-execution exception and null endpoints. All 96 positions are now recorded: 93 valid, three protocol-invalid, none unstarted.

Each root has eight attempts; Iris F1 and SciCode 4 and 6 each have one invalid. Stage composition differs: SciCode 4 occupies positions 1–8; Iris F1 contributes one position to 9–73 and seven to 74–81; Iris quantiles contributes eight to 82–96; SciCode 6 contributes one to 74–81 and seven to 82–96. The other eight roots each contribute eight to 9–73. Thus DeepSeek's 28/35 versus 4/10 strict successes, and GLM's 24/36 versus 9/12, do not identify effects of concurrency, date, or fault isolation. Table 17 also retains the unbound counts. Original task-by-stage rows remain in `tables/runtime_roots.tex`; Figure 4A shows all planned cells and unknowns.

Table 19: **Task strata of the same 96-position matrix.** Native adaptations yield 28/30 valid strict successes; both failures have passing final-program tests but lack qualifying final execution. Different computations occupy the two strata, so this is not a randomized interface comparison.

Support	Plan	Valid	Strict S	Correct return K	Tests G	Invalid
Author-constructed JSON tasks						
Neither (00)	16	16	5/16	13/16	15/16	0
Computation only (10)	16	16	7/16	15/16	16/16	0
Delivery only (01)	16	15	13/15	13/15	15/15	1
Both (11)	16	16	12/16	12/16	15/16	0
External native-return adaptations						
Neither (00)	8	8	7/8	7/8	8/8	0
Computation only (10)	8	7	6/7	6/7	7/7	1
Delivery only (01)	8	8	8/8	8/8	8/8	0
Both (11)	8	7	7/7	7/7	7/7	1

C.1.2 Planned contrasts and missing-outcome bounds

The primary comparison has 22 of 24 observed pairs: ten delivery wins, two losses, ten ties. Its observed mean $8/22$ does not identify the full planned contrast without missingness assumptions. Resolving each of the two missing primary endpoints as either binary value gives $[7/24, 9/24]$, or $[29.17, 37.50]$ percentage points. The third invalid belongs to arm 11 and does not enter this contrast. These are finite-schedule bounds, not confidence intervals; null records remain null.

Let π_{rma} denote the execution-control stage used for root r, model m, and arm a. The bounded statistic is

$$\widehat{\Delta}_{\text{schedule}} = \frac{1}{24} \sum_{r,m} \{ S_{rm}^{\pi_{rm,01}}(01) - S_{rm}^{\pi_{rm,10}}(10) \} .$$

Its missing values concern the realized schedule, not counterfactual sessions under a uniform protocol. Scientific arm slots were randomized; execution-control stages were not. The post-outcome continuation therefore does not identify the original common-protocol expectation in Equation 2.

Delivery-only versus neither has 23 observed pairs (ten wins, one loss, twelve ties), giving full-plan bounds $[9/24, 10/24]$, or $[37.50, 41.67]$ points. All 22 observed single-support pairs tie at $G = 1$; their full-plan bounds are $[-1/24, 1/24]$. GLM's twelve complete single-support pairs contain seven wins, no losses, five ties ($7/12$, or 58.33 points). DeepSeek's ten complete pairs contain three wins, two losses, five ties; its full-plan bounds are $[0, 2/12]$.

The original strong-claim rule requires a primary lower bound of at least 0.15, the specified randomization criterion, positive delivery-versus-baseline and model contrasts, support across families, and program-test coverage. The positive finite-plan lower bound and 23 passing delivery-only programs satisfy two numerical requirements, not the conjunction: missing primary endpoints prevent the prespecified complete-pair test, DeepSeek's lower bound is zero, and execution control changed after results were inspected. We substitute no complete-case significance test.

The seven JSON roots with all eight model–arm cells valid show DeepSeek/GLM strict counts of 5/7 versus 0/7 in arm 00 and 5/7 versus 7/7 in arm 01; all corresponding programs pass G. Iris F1 remains excluded only from this descriptive subset. The earlier six-root snapshot has 5/6 versus 0/6 in arm 00 and 4/6 versus 6/6 in arm 01, again with all programs passing G. These nested selections are not independent samples; their local completion ordering does not establish a stable ability ranking or test equal ability. Their unchanged source tables are `tables/common_seven.tex` and `tables/historical_comparison_evidence.tex`.

C.1.3 Failure stages and program evidence

Among 93 valid observations, 65 pass S, 81 have correct formal returns, and 91 pass all five program tests. The remaining twelve formal-return records have no qualifying return: there are zero observed wrong or exception/type returns. Three invalid trajectories are separate. Joint (S, G) counts are $(1, 1) : 65$, $(0, 1) : 26$, $(1, 0) : 0$, and $(0, 0) : 2$. The absence of $(1, 0)$ is empirical; S and G are independently defined. A test-passing failure requires original-return and binding evidence before it can be attributed to report construction.

Table 20: **Final-program evidence by initial computation provenance.** Of fourteen valid strict failures without a supplied kernel (00/01), thirteen pass all five tests. Thirteen of fourteen failures with a supplied kernel (10/11) also pass. Bound refers to original final-execution evidence, not a later test. This is a descriptive failure partition, not a treatment-effect estimate conditioned on failure.

Initial support	Failures	$G = 1$	$G = 1$, bound	$G = 1$, unbound
Neither (00)	12	11	8	3
Delivery only (01)	2	2	0	2
Computation only (10)	10	10	7	3
Both (11)	4	3	0	3

Without a supplied kernel (arms 00/01), 47 valid observations include fourteen strict failures, thirteen passing G: eight GLM failures have correct final-source-bound returns and invalid reports; five DeepSeek failures have only later test evidence; one further DeepSeek program fails G. With a supplied kernel (10/11), 46 valid observations include fourteen failures, thirteen passing G. Those passes establish tested behavior, not independent implementation by the agent. Table 20 is a descriptive failure partition, not a causal contrast conditioned on failure.

All fifteen GLM failures occur on JSON roots in arms 00/10. Each has $K = G = 1$ and final binding but fails schema; seven also fail preservation. On these roots, strict counts are 0/8, 1/8, 8/8, and 8/8 in arms 00/10/01/11, despite the same structural example and reminder. All thirteen DeepSeek failures exhaust the tool budget and lack final binding: twelve have no qualifying return, and one has a correct return followed by a source change. Eleven pass G. Linnerud correlation in arm 11 fails all five tests with import-or-input exceptions; Iris F1 in arm 00 fails all five with a computation-entry exception. These categories do not establish formula errors, and later tests do not complete the original execution obligation.

The native stratum has 30 valid observations: 28 with $S = K = G = 1$ and two DeepSeek SciCode 6 attempts with $S = K = 0, G = 1$ but no qualifying final execution. Two further native positions are invalid. Tests remain at ceiling while strict completion is imperfect (Table 19).

C.1.4 CPU REPLAY AND MEASUREMENT CONTROLS

Post-review CPU checks preserve original inputs, references, predicates, sources, and scores; they add no agent trajectories. Before the continuation, isolated replay of nine failed programs without supplied kernels reproduced all 45 fixed-input passes and their source/input hashes. The original three unbound DeepSeek trajectories remain unbound. A separately versioned extension covers the other five such failures, positions 75, 79, 84, 92, and 96: 25 fresh executions reproduce every saved per-input outcome. Four programs pass all five tests; position 75 raises `NotImplementedError` in its unchanged initial computation entry on all five. Two new GLM cases have original correct bound returns and invalid reports; the two passing DeepSeek cases have only later test evidence. The extension uses Linux/Python 3.11.15/NumPy 2.2.6 rather than the original Python 3.8.10/NumPy 1.24.4; it reproduces fixed-input outcomes, not the full observer or runtime identity.

For each of eight JSON roots, a separately written public-layout constructor passes the formal consumer with correct computation. Wrong delivered numbers pass structure but fail numerical acceptance; correct numbers outside `target` fail schema despite correct bound returns. All 24 formal controls behave as expected. Before these CPU executions, we fixed four shape-preserving wrong implementations per root: one emitting a constant original-input answer, one reversing coordinates, and two using family-specific alternatives (macro/weighted F1; Spearman/uncentered cosine; Hyndman–Fan 8/9 quantiles; through-origin/Theil–Sen fits). The latter uses median pairwise slope and median residual intercept.

Eight correct controls pass all five tests. All 32 wrong implementations fail G through numerical rejection of 160 finite, correctly shaped, execution-bound vectors. This distinguishes the selected alternatives, not arbitrary errors or untested boundaries. The original 269 CPU executions comprise 24 formal controls, 40 correct-control tests, 45 replays, and 160 mutation tests; the extension adds 25 checks. The formal controls cover examined JSON construction paths, not every specification interpretation or the four native adaptations.

C.1.5 SAVED RECORDS AND RESOURCES

Terminal tables retain all 96 rows in slot order and all original scientific/trajectory fields for positions 1–73. Appended S/K fields match raw records; G links to fixed-input reports and final snapshots. The generator checks score conjunctions, nulls, unique root–model–arm identities, and planned denominators without executing a model or scientific program. The 96 attempts contain 1,467 requests, including invalid attempts, with all scientific usage known and an estimated 32.602876 CNY cost. This excludes technical-validation calls and other studies; it is neither an invoice nor a balance.

C.2 COMPLETE MINI–HAIKU SUPPORT REPLICATION

C.2.1 COMMON REPLICATION PROTOCOL AND COHORT IDENTITY

All later support cohorts reuse the twelve researcher-exposed roots, public packages, numerical references, formal endpoints, and five-input tests. Each cell has a fresh conversation and workspace; author-side terminal-program tests provide no feedback. They record S, formal-return categories, preservation, schema, final-source binding, and G. Common adapter limits and requested reasoning settings are in Appendix A.4; no automatic final execution, numerical correction, report repair, or private-test feedback is supplied. Observable local tool errors retain frozen handling, while uncertain transport or measurement retains unknown endpoints. Cohorts use one session per cell and are analyzed separately.

Mini–Haiku crosses twelve roots, two configurations, and four arms in 96 observed-valid trajectories, with no unknown, unstarted, retried, or replaced scientific position. Packages, tests, execution policy, analysis, and assignments were frozen before September 22 calls. Seeded HMAC ordering is reproducible and randomizes conditions within root–model blocks. Each endpoint runs serially; endpoints may run concurrently. There is no contemporaneous GLM arm.

Requested IDs are `gpt-5.4-mini` and `claude-haiku-4-5` through AIHubMix nonstreaming chat completions. Receipts return `gpt-5.4-mini-2026-03-17` and `claude-haiku-4-5`; Haiku's backend snapshot is unobserved. The account holder reports gateway mapping, cross-model fallback, output repair, and recharge disabled; this is not an independent provider audit. Limits are

24 replies, 24 charged tool units, 24,576 reported input tokens, and 8,192 requested output tokens per call. Tool-argument constraints do not constrain the report. Per-request reservations and fixed wall-clock stops are separate resource controls; an earlier technical unknown remains in its technical ledger.

C.2.2 Outcomes and inference

The primary statistic equally weights 24 root–model delivery-only minus computation-only pairs, retaining the 8:4 JSON/native composition. Its difference is $5/24$, or 20.8 points: six wins, one loss, seventeen ties. The prespecified two-sided conditional slot-swap sharp-null test gives $p = 0.125$. Mini has six wins, one loss, five ties ($p = 0.125$); Haiku has twelve ties ($p = 1$). Holm-adjusted model-level values are 0.25 and 1. Neither aggregate nor adjusted model tests reject at 0.05.

The mean G contrast is $-1/24$, or -4.2 points. Thus $Q = \Delta S - \Delta G$ is 25.0 points (Mini: 50.0; Haiku: zero), reflecting increased completion and lower G in Mini's delivery-only arm. Joint counts are $(S, G) = (1, 1) : 79$, $(0, 1) : 12$, $(0, 0) : 5$, and $(1, 0) : 0$. All twelve test-passing failures have correct final-source-bound returns and invalid schema; six begin without a supplied kernel. All seventeen failures are Mini's: these twelve report failures, four incorrect F1 returns (including two delivery-only cases), and one invalid native return type in computation-only. The five computational failures also fail G. Every record has final binding; Haiku completes all 48 conditions.

Table 21: All new positions. P: strict pass and G pass; R: correct bound return, invalid report, G pass; N: incorrect return and G failure (schema may also fail); T: invalid return type and G failure. All positions have final binding. Row order follows the frozen root schedule.

	GPT-5.4-mini				Claude Haiku 4.5			
Root	00	10	01	11	00	10	01	11
quantile-diabetes-cd	R	R	P	P	P	P	P	P
scicode-dev-3	P	P	P	P	P	P	P	P
scicode-dev-1	P	P	P	P	P	P	P	P
scicode-dev-6	P	P	P	P	P	P	P	P
scicode-dev-4	P	T	P	P	P	P	P	P
f1-digits-cd	N	R	N	P	P	P	P	P
quantile-iris-cd	R	R	P	P	P	P	P	P
correlation-linnerud-cd	R	R	P	P	P	P	P	P
fit-linnerud-cd	R	P	P	P	P	P	P	P
fit-diabetes-cd	R	R	P	P	P	P	P	P
f1-iris-cd	N	P	N	P	P	P	P	P
correlation-diabetes-cd	R	R	P	P	P	P	P	P

The arithmetic supplement retains root contributions, source groups, and leave-one-root-out contrasts. One session per cell does not estimate repeated-session variation. Verification checked all 96 raw-to-derived hashes, final-source hashes, worker records, and 480 saved test outcomes; frozen analysis agrees with binary-pair arithmetic. These are saved-result checks, not independent replay of the 96 programs. The released `results/foreign-support-and-content/` contains rows, assignments, provenance, analysis, and generation inputs, not full raw terminal assets. Mini uses 228 scientific requests, 700733 input tokens, 30693 output tokens, 254 tool units, and an estimated USD 0.663749; Haiku uses 655, 4127856, 162408, 674, and USD 5.434178 respectively (total USD 6.097927). These configured-rate estimates are not invoices, and tokens/tool units do not equalize model compute.

C.3 Qwen support replication and cohort provenance

The later `qwen3.7-plus-2026-05-26` cohort contains 72 valid fresh sessions: 48 support cells and 24 cells on the distinct original content roots, with correct/neutral starts under background C. It uses Alibaba Cloud's Beijing compatible API, thinking disabled, and at most one trajectory at a time. The extension was chosen after earlier results were inspected and prepared before its own calls. Transport and accounting differ from the gateway cohort; it is separate from the Mini–Haiku primary test.

Strict counts in arms 00/10/01/11 are 11/12, 12/12, 12/12, and 12/12; all 48 returns are correct, all programs pass G, and all executions bind to final source. The sole failure, Iris quantiles in arm 00, passes preservation but fails report schema and contract-path extraction. It is a test-passing report failure without a supplied kernel. Both single-support contrasts are zero, with twelve ties. Table 22 retains every support and content root.

The 72 sessions use 731 settled scientific requests, estimated at CNY 10.598386, excluding technical admission and historical studies. Checks verify worker-to-record hashes, actual initial messages, final sources, content outputs, 721 runtime/input pins, six binaries, and agreement of saved G reports with derived records. They establish provenance and consistency, not new scientific execution. The original wrong-target, wording/layout, and interface-control panels keep their original model coverage.

C.4 Gemini support replication and cohort provenance

The `gemini-3.1-flash-lite` cohort likewise has 48 support and 24 distinct content sessions, all observed-valid and none unstarted. AIHubMix uses low reasoning effort and one trajectory at a time. Selection followed earlier results; packages and assignments were fixed before these calls. Receipt identity does not independently certify the backend. This cohort replaces no missing positions from the earlier interrupted Gemini configuration and is not pooled into the Mini–Haiku test.

Strict counts in arms 00/10/01/11 are 12/11/11/12 out of twelve; G counts are 12/12/11/12. All 48 records have final binding. Linnerud correlation in arm 10 has a correct bound return and passes G, but fails schema and contract-path extraction. SciCode 4 in arm 01 has a bound incorrect native return and fails G, while schema and preservation pass. Thus there are 47 correct returns, one incorrect return, and no missing final execution. Joint states are 46 strict/test passes, one test-passing strict failure, and one strict/test failure.

One delivery win, one loss, and ten ties give $\Delta S = 0$ and $\Delta G = -1/12$, or -8.3 points. Hence Q is 8.3 points entirely from lower G, with no completion improvement. Table 22 records all roots. Verification covers all 72 worker/record, message, source and content chains, 721 runtime/input pins, six binaries, and saved-test agreement. The 851 settled scientific requests cost an estimated USD 1.625479 at frozen rates, excluding earlier attempts and admission. This is arithmetic/provenance verification, not independent replay of 72 programs.

C.5 Exact counts and evidence profiles

Table 23 gives the counts behind Figure 5, retaining cohort identities and valid denominators. Table 2 partitions the same failures: first unbound records, then bound report failures with $K = G = 1$ and failed schema, then bound numerical/return-type failures with $K = G = 0$. Checks can fail together, but each failure belongs to one profile; the two unbound DeepSeek programs failing G stay in the unbound profile. `analysis/failure_profiles.py` verifies input hashes and memberships without running models or candidates.

The alternative connected-dot view of these same counts remains in the source archive as `figures/support_factorial.pdf`; Figure 5 and Table 23 provide the displayed views.

Table 22: Complete root matrices for the separate Qwen and Gemini cohorts, each with 72 fresh sessions. C/N denote correct/neutral starts; support and content roots differ. No historical or earlier Gemini position is replaced. Original per-model table sources remain in `tables/qwen_roots.tex` and `tables/gemini_roots.tex`.

A. Support: strict success / fixed-test pass

	Qwen				Gemini			
Root	00	10	01	11	00	10	01	11
correlation-diabetes-cd	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
correlation-linnerud-cd	1/1	1/1	1/1	1/1	1/1	0/1	1/1	1/1
f1-digits-cd	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
f1-iris-cd	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
fit-diabetes-cd	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
fit-linnerud-cd	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
quantile-diabetes-cd	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
quantile-iris-cd	0/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
scicode-dev-1	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
scicode-dev-3	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
scicode-dev-4	1/1	1/1	1/1	1/1	1/1	1/1	0/0	1/1
scicode-dev-6	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1

B. Starting programs: strict success and content match

	Qwen				Gemini			
	S		Match		S		Match	
Root	C	N	C	N	C	N	C	N
correlation-abalone	1	0	1	1	1	1	1	1
correlation-wine-red	1	1	1	1	1	1	1	1
correlation-wine-white	1	0	1	1	1	1	1	1
f1-glass	1	1	1	1	1	1	1	1
f1-wdbc	1	0	1	1	1	1	1	1
f1-wine	1	1	1	1	1	1	1	1
fit-airfoil	1	0	1	1	1	1	1	1
fit-auto-mpg	1	0	1	1	1	1	1	1
fit-yacht	1	0	1	1	1	1	1	1
quantile-airfoil	1	0	1	1	1	1	1	1
quantile-auto-mpg	1	0	1	1	1	1	1	1
quantile-yacht	1	0	1	1	1	1	1	1

Table 23: **Six-model support evidence by experimental arm.** Full model names and cohorts follow Table 1. Each arm plans twelve sessions; the Valid column gives the observed denominator. K counts correct last formal returns, and Bound counts final-source executions. The last two columns count strict failures passing G and their subset without a supplied kernel. Shading marks delivery-only (01).

Model	Arm	Valid	S	K	G	Bound	Fail, G pass	No kernel
DeepSeek	00	12	8	8	11	8	3	3
	10	11	8	9	11	8	3	0
	01	11	9	9	11	9	2	2
	11	11	7	7	10	7	3	0
GLM	00	12	4	12	12	12	8	8
	10	12	5	12	12	12	7	0
	01	12	12	12	12	12	0	0
	11	12	12	12	12	12	0	0
Mini	00	12	4	10	10	12	6	6
	10	12	5	11	11	12	6	0
	01	12	10	10	10	12	0	0
	11	12	12	12	12	12	0	0
Haiku	00	12	12	12	12	12	0	0
	10	12	12	12	12	12	0	0
	01	12	12	12	12	12	0	0
	11	12	12	12	12	12	0	0
Qwen	00	12	11	12	12	12	1	1
	10	12	12	12	12	12	0	0
	01	12	12	12	12	12	0	0
	11	12	12	12	12	12	0	0
Gemini	00	12	12	12	12	12	0	0
	10	12	11	12	12	12	1	0
	01	12	11	11	11	12	0	0
	11	12	12	12	12	12	0	0

D STATISTICAL INTERPRETATION AND SENSITIVITY

D.1 BOUNDS, SAMPLING INTERVALS, AND TASK COMPOSITION

Content bounds resolve unclassified saved vectors; historical schedule bounds resolve missing outcomes. Both describe finite records. Sampling intervals instead address an expected outcome under a stochastic execution or assignment model. The following uncertainty and sensitivity analyses were added during revision; primary tests and recorded outcomes are unchanged.

Prospective intervals. Conditioning on fixed tasks, packages, model settings, and design, assume independent root–model blocks and no interference across isolated sessions. Probabilities may vary across tasks/models, and arms or checks within a block may be dependent. These assumptions concern execution or assignment randomness, not sampled scientific domains. For $X_i \in [l, u]$, with $\bar{X} = n^{-1}\sum_i X_i$ and $\bar{\mu} = n^{-1}\sum_i \mathbb{E}X_i$, Hoeffding's inequality (Hoeffding, 1963) gives the pointwise interval

$$I_{.95} = \left[\bar{X} - (u-l)\sqrt{\frac{\log(40)}{2n}},\ \bar{X} + (u-l)\sqrt{\frac{\log(40)}{2n}}\right] \cap [l, u].$$

Fixed-arm S, K, and G use $[0, 1]$ and twelve blocks per model. Single-support contrasts use $X_i = Y_i(01) - Y_i(10) \in [-1, 1]$, with 24 Mini–Haiku blocks or twelve per model. These intervals concern expected fresh-session differences under fixed conditions, or finite assignment averages under randomized slots; they do not identify effects of protocol changes.

Table 24: Conservative, pointwise uncertainty intervals for the prospective single-support contrasts (percentage points). The independent unit is a root–model block, with both support outcomes retained inside each block. Mini–Haiku is the prespecified primary cohort; the two subsequent cohorts are separate. These post-review intervals do not replace the exact slot-swap test.

Cohort / model	Blocks	ΔS	95% interval	ΔG	95% interval
Mini–Haiku	24	20.8	[-34.6, 76.3]	-4.2	[-59.6, 51.3]
Mini	12	41.7	[-36.7, 100.0]	-8.3	[-86.7, 70.1]
Haiku	12	0.0	[-78.4, 78.4]	0.0	[-78.4, 78.4]
Qwen	12	0.0	[-78.4, 78.4]	0.0	[-78.4, 78.4]
Gemini	12	0.0	[-78.4, 78.4]	-8.3	[-86.7, 70.1]

Table 25: Conservative prospective arm-rate intervals. Apply this mapping to every Mini, Haiku, Qwen, or Gemini S, K, or G count in Table 23; each has twelve valid blocks. Intervals are pointwise, not simultaneous or task-population intervals; all-pass samples retain uncertainty. The expanded export remains in `analysis/clarification_exports/rate_intervals.tex`.

Count	4/12	5/12	10/12	11/12	12/12
95% interval (%)	[0.0, 72.5]	[2.5, 80.9]	[44.1, 100.0]	[52.5, 100.0]	[60.8, 100.0]

Intervals are conservative and pointwise, not simultaneous, and remain non-degenerate at ceiling. One session per cell does not estimate within-cell generation variance. Mini–Haiku's $[-34.6, 76.3]$-point completion interval indicates substantial uncertainty around $+20.8$; the prespecified sharp-null test remains $p = 0.125$ with its original threshold.

Paired root-composition sensitivity. Resampling eight JSON and four native roots with replacement within strata, carrying both models, all arms, and joint outcomes together, yields exact central 95% reweighting ranges of $[4.2, 37.5]$ points for ΔS and $[-16.7, 8.3]$ for ΔG. These empirical composition ranges are not fresh-session confidence intervals and do not override the randomization test. An all-tie sample stays at zero under reweighting without establishing equivalence. Deleting one complete root at a time, retaining both models and renormalizing eleven roots, gives $\Delta S \in [18.2, 27.3]$ and $\Delta G \in [-9.1, 0]$ points; this does not estimate performance on unseen tasks.

E REPRODUCING THE EVIDENCE

Two archives serve different purposes: the manuscript source compiles the paper and regenerates its summary analyses; `ICLR27_Supplement.zip` contains the separate `scientific-coding-experiments/` repository for result arithmetic, program replay, task packages, and runtime-source inspection. Table 26 connects the principal claims to their records. Reused subsets add no trajectories to their parent cohorts.

E.1 EVIDENCE MAP AND EXECUTABLE CORE MATERIAL

Table 26: Evidence and reproduction map. Summary arithmetic, candidate-program replay, and original execution evidence answer different questions.

Evidence	Records and scope	Paper location
Target identity and boundaries	359 original valid trajectories; corrected new-source panel: 141 valid. Different task interfaces remain separate.	Appendix F
Saved requested content	174 reused valid artifacts; 43 same-version pairs. Reading rules and pair selection determine the finite-content bounds.	Appendix B.1
Historical support	96 attempted positions, 93 valid and 3 unknown; runtime segments and missing-outcome bounds.	Appendix C.1
Model replications	Mini–Haiku: 96 support and 48 content sessions; Qwen and Gemini: 48 support and 24 content sessions each, all valid.	Appendices B, C
Program and measurement checks	Fourteen unsupplied-kernel failed programs, eight correct controls, and 32 semantic mutants; no new agent sessions.	Appendix A.3.2; commands below
Development and contract checks	Development screen and original/corrected contract panels retain their own coverage and defect status.	Appendix F

1. Recompute manuscript summaries. From the manuscript source root, with Python 3.11:

```
python3 analysis/clarifications.py --output-dir ../check-output
python3 analysis/failure_profiles.py --check
```

These standard-library commands verify pinned inputs and recompute contrasts, uncertainty and task-composition summaries, failure profiles, and the 288-record projection. They do not execute models or scientific programs. Figure regeneration uses `analysis/plot.py` and `analysis/requirements-used.txt`. The manuscript archive does not contain the full raw workspaces or program-replay packages.

2. Verify the separate experimental supplement. From its extracted repository root, with Python 3.11 and Node.js 22:

```
python3 scripts/check_repository.py
python3 scripts/reproduce.py --output-dir ../check-output
```

These check hashes and recompute the principal cohorts' saved results. `docs/PAPER_MAP.md` maps claims to files; `docs/COVERAGE.md` gives exact coverage. The historical 96-position records, runtime assignments, and arithmetic entry point are under `results/historical-support-96/`. Output directories must be new and outside the source package.

3. Replay the fourteen unsupplied-kernel failed programs. Inside the supplement's `reproduction/core-nine/`, run:

```
node saved_summary/reproduce.mjs
python3 run_programs.py --all --output /tmp/core-check-NEW.json
```

The first command reconstructs the initial 73-record snapshot (Node.js 18+). The second uses Linux/Python 3.9+ and the original five inputs and predicate to reproduce nine program passes, eight correct-control passes, and 32 semantic-mutant numerical rejections: 245 program–input checks.

Inside `reproduction/continuation-five/`, with Linux, Python 3.11, and NumPy 2.2.6:

```
python3 run_five.py --output /tmp/five-replay-NEW.json
```

This adds the five unsupplied-kernel failures from positions 74–96: twenty matching input outcomes and five computation-entry exceptions from the remaining program. Together the packages cover all fourteen cases, thirteen of which pass all five tests. They include final textual workspaces, original inputs/references, terminal reports, available receipts, predicates, dependencies, and byte manifests. Output filenames must be new; only the native spatial-filter case needs NumPy.

Replay tests computation, not original binding or strict completion. The supplement includes all four support packages for each JSON root, observer source and available saved receipts, but not the full original isolation runtime or all four native environments. Its one native replay uses different Python/NumPy versions. Host-path redactions affect reports, not scientific code or data. Run candidate code in a disposable environment; process limits alone are not a security sandbox. Author review and redistribution approval are separate from these automated checks.

E.2 Reusable evidence records and possible uses

`analysis/clarification_exports/evidence_records.jsonl` contains all 288 support positions: cohort, model, root, arm, measurement status, S, formal-return category, G, preservation, schema, binding, and computation provenance. Its companion `evidence_profile.schema.json` defines the projection; hashes and indices link it to source summaries. Missing receipt bodies, dependency manifests, and individual test outcomes remain null. Both files also appear under the supplement's `manuscript-analysis/`; they organize saved evidence rather than replace raw traces.

The profile can distinguish whether to inspect report construction, obtain final-source execution evidence, or investigate computation; repair benefits have not been measured. Provenance (Moreau & Missier, 2013; Wang et al., 2026), transport interfaces (Wattamwar et al., 2026), oracle-less checks (Valentin et al., 2026), and structured-generation evaluation (Geng et al., 2025; Zhou et al., 2025) offer complementary information, not substitutes for numerical references and source–execution correspondence.

F BOUNDARY CONDITIONS AND DEVELOPMENT STUDIES

These studies examine recurrence of target substitutions, the information retained by component scores, and the development decisions behind the main comparisons. Panels remain separate (Table 27). Original grades, unknown outcomes, protocol changes, and negative results are retained. Unchanged table sources provide additional root, condition, repeat, read-event, and resource summaries; the PDF consolidates repeated presentations of the same observations.

Table 27: **Historical identity-analysis coverage, with original scores retained.** The original census predates the order continuation in Table 30; its subset is not counted twice. Strict and alternative counts use valid observations. *Two-target panels count bound alternatives; the three-target panel also requires preservation and schema. Not scored denotes unknown or invalid, not a scientific failure. Panels are not pooled. The 49-vector post-hoc annotation does not enter these success counts.

Panel	Plan	Rec.	Valid	Strict	Alt.*	Not scored	Not started
Original revision census	378	360	359	289	60	1	18
Original-context subset	72	72	72	53	17	0	0
Initial-state crossing	288	288	288	280	2	0	0
Wording/layout factorial	96	96	96	75	13	0	0
Corrected new sources	144	144	141	138	0	3	0
Menu / starting program	144	144	142	135	4	2	0
Three-target (stopped)	288	180	174	105	0	6	108

F.1 ORIGINAL IDENTITIES AND CONTEXT SENSITIVITY

Question and design. The historical census combines 180 valid transfer trajectories, 54 valid fresh reminder trajectories (H54), and the original order snapshot: 125 valid, one measurement-unknown, and eighteen unstarted positions out of 144. Its 359 valid observations are dependent task cells; the common-six transfer subset reuses observations. B supplies original context, F a model rewrite, H a fixed reminder, and R deterministic selection. D0/D1 swap definition order. Table 28 keeps intervention contrasts within blocks.

Table 28: Strict successes / planned trajectories, separately by block. A dash denotes an absent arm. The historical order snapshot has 31 valid trajectories per arm under D0, and 31 B / 32 H under D1; all other cells are fully observed. Updated order coverage appears in Table 30. Full condition and component tables remain in the source package.

Setting	**Model**	**B: original**	**F: rewrite**	**H: remind**	**R: resolve**
Transfer	DeepSeek	27/36	15/36	–	35/36
Transfer	GLM	16/18	–	–	18/18
Transfer	Kimi	10/18	–	–	16/18
Reminder	DeepSeek	14/18	–	17/18	17/18
Order / D0	DeepSeek	23/36	–	30/36	–
Order / D1	DeepSeek	20/36	–	31/36	–
Crossing	DeepSeek	46/48	–	48/48	45/48
Crossing	GLM	47/48	–	47/48	47/48

Key results. The census contains 296 current targets (291 bound, five unbound), 62 alternatives (sixty bound, two unbound), and one undetermined output. Strict success holds in 289/359; 60/70 valid failures are bound alternatives: 37 transfer, three H54, and twenty order. All sixty also match the complete alternative target-plus-protection object. No complete output matches neither reference. Identity requires every predefined finite coordinate and required array length; malformed or

missing numerics are undetermined, while matching both references is a measurement error. Full-object checks additionally cover labels, structure, and protection.

Table 29: Original-context B subset of the transfer block: DeepSeek (36 records), GLM (18), and Kimi (18). The last three columns partition all nineteen strict failures; no other failure identity occurs in this subset. A and I have equal completion counts but different failure states. Counts reuse the saved terminal scores and existing numerical-identity analysis; no output is reclassified. These are 72 task–condition records, not 72 independent tasks or new trajectories.

			Strict failures: identity and execution		
Condition	Valid	Strict	Bound alternative	Current, unbound	Absent output
Original (O)	24	9	15	0	0
Applicable (A)	24	22	0	1	1
Inapplicable (I)	24	22	2	0	0
Total	72	53	17	1	1

Under applicable updates in this B subset, DeepSeek's fit-motor-PPE case has no saved output; its NO2–SO2-support case has matching target/protection and valid report structure but no final binding. The two inapplicable-update failures instead deliver bound alternatives, despite the equal strict totals in Table 29.

Two bound-current failures concern protected keys: Kimi's fit-motor-PPE A/R omits folds and renames `z_mean`; DeepSeek's fit-motor-Jitter(Abs) A/R renames `folds`. Available aliased values agree, but the schema failures remain. No existing same-name protected numerical coordinate disagrees; the absent output is separate.

Common-six B/R paired counts (both pass, R only, B only, both fail) are DeepSeek $(12, 6, 0, 0)$, GLM $(16, 2, 0, 0)$, and Kimi $(9, 7, 1, 1)$. Deleting any root leaves net gains of five, one–two, and five. These are descriptive omission ranges, not confidence intervals. A separate local Qwen B/H/R attempt remains incomplete: 31 rows, thirty valid with fourteen successes, one further attempt without a terminal row, and 22 unstarted positions.

Reminder and order limits. H54 has 48/54 successes. H and R each reach 17/18 but fail on different cells. Three bound alternatives are fit-motor-PPE O/B and O3-station O/B and O/R; three current-but-unbound outputs are O3-station I/B and I/H and NO2–SO2 support A/B. All six exhaust the tool budget; formal attempts alone do not establish final binding.

The original order snapshot has 104/125 valid successes. Binary completion of missing outcomes gives H-minus-B bounds $[8, 27]/72$ and D1-minus-D0 bounds $[-12, 7]/72$. On September 21, amended execution control attempted only the eighteen unstarted identities under original task packages and scoring: fifteen succeeded, two yielded non-valid rows, and one interrupted attempt had no row. The full schedule now has 143 raw records, 140 valid outcomes, 119 successes, and four unknowns (Table 30). Updated bounds are $[17, 21]/72$ and $[-5, -1]/72$. They measure completion sensitivity, not sampling uncertainty; post-outcome continuation does not independently confirm the finding or change the original census.

Table 30: **Definition-order schedule after its September 21 continuation.** D0/D1 are archived as E0/E1. All 144 positions were attempted. Four unknown scientific outcomes comprise three non-valid raw records and one interrupted attempt without a row. These counts update schedule coverage; they do not rerun the historical identity classification or change its denominator.

Order condition	Planned/attempted	Valid	Strict success	Unknown
B, D0	36	34	26	2
B, D1	36	34	23	2
H, D0	36	36	35	0
H, D1	36	36	35	0
Total	144	140	119	4

Rewrites, reads, and public consistency. Post-outcome AI-assisted annotation of all 36 F contexts finds seventeen correct specific applicability facts, twelve lacking a specific fact, four contradicted, one generic conditional, and two mixed/unknown. Four unambiguous cases retain a correct fact but request the alternative; two mixed cases combine a true fact with an incorrect applicability inference. This is not blinded semantic validation.

Among 21 wrong-query trajectories, thirteen directly read task and evidence: nine deliver alternatives and four current targets; the other eight deliver alternatives. The four current-output cases are Jitter(Abs) risk, NO2 station, NO2–SO2 support, and O3–NO2 support. Reads cover 1,400/2,006 task characters in the first and complete files in the others. All formal receipts retain initial code/configuration hashes, with no successful write to either file. The last case executes before reading task/evidence; a rejected plan-stage request in the support case is not execution. The 819-event ledger establishes access and order, not causal use.

A checker designed after the O3 example records two rules before scanning outputs: station-mean O/I requires $t = m_x$; marginal-support A requires $t = m_x - m_y$. Unsupported/ambiguous contracts are rejected. Reported values define conservative intervals with radius $\max(a, r|v|/(1-r))$ plus four ULPs per field; exact rational arithmetic flags only disjoint intervals. Controls cover rounding, overlap, missing/type errors, and unsupported forms.

Only 72/378 identities have covered contracts: 64 outputs and eight unstarted order cells. The 306 uncovered positions include ten other unstarted cells and the unknown. Twelve outputs contradict a rule; reference matching identifies all as alternatives, eleven bound. The 52 passes comprise 48 bound and four unbound current outputs. Coverage spans only six station/support roots from one source. This post-hoc necessary condition neither proves correctness nor generally replaces private references.

F.2 Crossing the starting implementation

Eight additional parameterized roots retain the original two sources and four relations while varying months, variables, and identifiers. Their 48 O/A/I-by-starting-state bundles pass independent-reference CPU checks, all returning the specified starting target, with 24/48 initially strict. The scientific matrix crosses eight roots, three conditions, two starts, B/H/R, and DeepSeek/GLM: all 288 positions are valid, with adjacent initial-state pairs in a fixed balanced order.

Before launch, the shared wrapper clarifies that the contract, not starting code, selects the target: adopt a revision only when applicable, and either retaining or changing the implementation can be correct. Both computations are exposed through a Boolean switch, and the output schema is explicit. All first-message hashes match frozen inputs; earlier prompts and scores are unchanged.

There are 280/288 strict successes. Of 144 initial-state pairs, 140 have current numerics on both sides and 136 have bound-current outputs on both; every remaining pair has one bound-current output. DeepSeek H-versus-B has two gains/no losses and R-versus-B two gains/three losses; GLM has one gain/one loss for either contrast. Eight failures comprise three alternatives (two bound), four current-but-unbound outputs, and one bound other-numerics output. The last repeats Dingling fourteen times in a 25-entry list for twelve stations, yielding 26.776080034941796 rather than 38.829540750342396 in saved-byte scalar reconstruction. No candidate is repaired or regraded. The two bound alternatives pass structure/preservation but fail labels. Near-ceiling performance differs from the original failure composition; wording, parameters, and layout changed together.

F.3 Wording versus implementation layout

Design. Following evidence of prompt-format sensitivity (Sclar et al., 2024), four exposed transfer roots cross O/A/I, two wordings, two layouts, and two fresh repeats: 96 DeepSeek-v3.2 B trajectories initially computing T_0. U0 retains the message; U1 appends:

> The target to execute is determined by the public task contract and the rules that apply to this report; neither a candidate recipe nor the supplied initial program determines the current target. If no update applies, follow the original target definition stated in the task.

K0 supplies the original program; K1 exposes both computations through a Boolean entry point defaulting to T_0, without selecting applicability. Messages are identical across layouts at fixed

U and differ only by the suffix at fixed K. CPU controls verify references, protection, and the switch before calls. Twenty-four balanced quartets support paired comparisons. U changes length/repetition/salience, and K changes executable information/integration work; these are concrete interface interventions.

Table 31: **The fixed-root instruction contrast includes gains and losses.** U0/U1 omit/add the prespecified target-source instruction; K0/K1 provide the original/switchable program. Strict and q use all 24 planned trajectories per arm; O (no update) has eight. Every removed q becomes strict success in its matched U1 cell, while other pairs regress. Each lower row summarizes 24 pairs from the same experiment: one model, four development-known roots, and two repeats.

Instruction	Layout	Strict ↑	q ↓	No-update q
Absent (U0)	Original (K0)	18/24	4/24	4/8
Present (U1)	Original (K0)	21/24	1/24	1/8
Absent (U0)	Switchable (K1)	16/24	6/24	6/8
Present (U1)	Switchable (K1)	20/24	2/24	2/8

U0 → U1 pairs	q → strict	New q	Strict gains	Strict losses
Original layout (K0)	3	0	5	2
Switchable layout (K1)	4	0	6	2

Results and interpretation. All 96 positions are valid: 75 strict successes, 79 current identities, sixteen alternatives, and one undetermined output. Thirteen alternatives are bound (q), all in O; each follows source modification and final-program execution, preserves the complete T_1 object, and passes structure but fails labels. Three further alternatives are unbound. Seven outputs lack binding, two fail protection, and eighteen fail full schema (two structural and seventeen label failures, one overlapping).

U1 removes three q events under K0 and four under K1, adds none, and converts every removed event to strict success. Strict pairs nevertheless contain five gains/two losses under K0 and six gains/two losses under K1. Regressions are a current fit with protection/schema failure, an undetermined unbound fit, and two unbound inapplicable risk outputs (one current, one alternative). K1 increases q by two under U0 and one under U1, while strict success falls by those counts. Layout therefore does not explain the U1 improvement.

The wording direction holds on three roots; fitting does not improve. K1 q falls 3/12 to 1/12 in each repeat; K0 stays 1/12 in repeat 1 and falls 3/12 to 0/12 in repeat 2. The planned schedule ends without missing positions or outcome-driven expansion; all 2,284 scientific requests settle. Saved-byte and independent arithmetic checks reproduce the grid and contrasts without candidate execution. These results retain sampling instability and concern one intervention, one model, and four development-known roots.

F.4 Corrected new sources: a non-reproduction boundary

Design and correction. Six roots introduce three Palmer Penguins species (Horst et al., 2020) for Pearson/Spearman correlation and three NOAA CO-OPS stations (NOAA CO-OPS, n.d.) for periodogram/Welch estimation. The 144-position matrix crosses roots, O/A/I, two starts, U0/U1, and DeepSeek/GLM. Both definitions and exact labels are public; starting code does not select the requested target. Source and computational relation change together.

Sources were retrieved on September 14, 2026. The Penguins CSV is pinned to repository commit `8957207b78d6ccd1b4654a9dd9c9041b657478ab`. NOAA requests use `water_level` for The Battery (8518750), Key West (8724580), and San Francisco (9414290), January 1–7, 2020, with MLLW datum, metric units, and GMT. Each station task retains the earliest 256 contiguous six-minute observations with quality fields `q=v` and `f=0,0,0,0`; all three selected windows span January 1, 2020, 00:00 to January 2, 01:30 GMT. The supplement's `docs/RESOURCE_SOURCES.md` links the pinned CSV and retained retrieval manifest; it does not add a full replay of this historical panel.

An earlier pilot stops after fifty records (49 observable, one unknown). Twelve label failures require strings absent from every public input/message; another required string appears only in starting code. Those grades remain development records. Correction publishes both labels uniformly without changing data/mathematics; no pilot observation is pooled. Prior root/outcome exposure precludes a held-out interpretation. Independent references, disjoint acceptance regions, 36 CPU bundle checks, delivery controls, and four fresh model technical tasks precede corrected science.

Results and scope. All 144 positions have terminal records: 141 valid, 138 strict successes, and three unknowns. Sequential v5/v6/v7 segments contain 29/20/95 positions; transport handling and execution windows change while scientific inputs/scoring remain fixed. Only unattempted positions continue. Two unknowns have pre-header disconnection evidence; one URL error lacks an identifiable underlying cause. Unknown requests retain reserves and are neither retried nor scored as failures or zero substitutions.

DeepSeek U0/U1 yield 32/34 and 34/35 strict successes; GLM yields 36/36 in each. There are 140 current identities and one alternative, with $q = 0/141$ overall and $0/3$ among valid failures. All valid outputs pass preservation/structure; 140 pass labels and 138 binding. All three valid failures are DeepSeek:

- Position 46, station 8518750, A/U0, initially T_1: alternative numerics, wrong label, no final binding despite two formal executions.
- Position 97, Chinstrap, I/U0, initially T_0: current numerics/protection/schema pass, but no formal execution or binding.
- Position 144, station 8724580, A/U1, initially T_1: the same passing components, with no formal execution or binding.

Binary completion allows zero–three bound alternatives and 138–141 successes over all 144 positions, not confidence intervals. O has 47/47 valid successes. Of 48 O/I pairs, 46 both succeed, one favors O, one is incomplete. DeepSeek wording pairs contain thirty common successes, two U1-only, one U0-only, and three incomplete; GLM has 36 common successes. Initially wrong programs yield 70/70 valid successes, initially correct ones 68/71. These overlapping decompositions do not reproduce the original O disadvantage or establish equivalence.

Changes in data, relations, labels, and presentation prevent single-factor attribution; the explicit dual-target interface does not estimate natural-repository prevalence. A pre-result feedback-pilot gate requires natural bound-alternative and other-numerics failures matched on components, source, and model. Neither occurs. Neither that pilot nor the conditional O-interface follow-up is triggered, leaving incremental repair usefulness untested.

F.5 Menus and component-score information

Design and results. This distinct 144-position study crosses six roots, two explicit requests, alternative prose present/absent, neutral/requested/alternative starting code, and DeepSeek/GLM. Four roots come from original development; Chinstrap correlation and station 8518750 are pilot-exposed. Labels are neutral. The 24 primary positions use requested-only prose and neutral code. Alternative code reveals its computation without alternative prose; replacing O/A/I with an explicit request changes more than menu presence.

Runtime segments record 6/89/49 positions, with 5/88/49 valid and 5/81/49 successful. Native-protocol termination and malformed arguments cause the two invalids; later handling accepts clean provider text termination without executing it or granting success. Earlier positions are never retried. Among 142 valid outputs, 135 are current, five alternative, and two undetermined. Four alternatives are bound, all with both definitions displayed, on fitting, risk, support, and correlation. Two start from requested code and two from alternative code. The primary subset has 23/24 successes and one undetermined failure, with no bound substitution.

Only 33 menu/no-menu pairs are valid within one runtime version: thirty common successes, two no-menu-only, one menu-only. Thirty-eight cross versions and one is incomplete; only one substitution has a same-version no-menu counterpart. Within-version neutral/requested pairs have eighteen common successes and two wins per side; neutral/alternative pairs have 23 common successes and

two neutral-only wins. Remaining pairs cross versions or are incomplete. Marginal differences do not identify causal menu or starting-program effects.

Information boundary. Grouping by (current numerics, preservation, schema including labels, binding) yields no mixed-identity group within any panel. The original/crossing/wording/corrected-source panels have 6/5/6/3 observed bit groups; the 88-valid first menu continuation has five, and its final 49 strict successes add no mixed group. Pooling the complete five two-target panels yields two collisions:

- $(0, 1, 1, 1)$: four menu alternatives and the crossing's other-numerics station output (26.776080034941796, versus requested 38.829540750342396 and alternative 39.453047530791785).
- $(0, 1, 0, 1)$: one undetermined menu output and bound alternatives from the original census (sixty), crossing (two), and wording/layout (thirteen).

Four bits alone therefore cannot recover every identity across these interfaces. Examples cross tasks/contracts; panel identity permits a lookup fitting the corpus, so this is no demonstrated advantage over a checker using task inputs. The bits themselves include private-reference checks. Table 5's constructed contrast and Appendix B.1.2's three-target annotation are separate. The menu panel has no other-numerics output, so the planned matched-failure repair experiment remains unperformed.

F.6 Computation-by-delivery development screen

Design and measurement. Four exposed roots (Wine F1, Abalone correlation, Auto MPG quantiles/fitting) cross DeepSeek/GLM and 00/10/01/11 supports: 32 slots, one sample per cell, with arm order randomized within root–model blocks. Computation support returns a vector; delivery support constructs the report without scientific data access or value changes. Availability is optional; contracts, examples, permissions, and ceilings are shared. Requested settings are nonthinking, temperature 0.6, top-p 0.95, no seed, and ceilings of 24 replies/charged tool units and 24,576 input/8,192 output tokens per request.

The evaluator records `compute(data, config)` before delivery; K uses its last formal return. All 26 K-positive records match final-source hashes; five valid trajectories have no qualifying execution ($K = 0$), and the invalid has null endpoints. Five pre-call inputs per root, hidden from agents, select the first $\lceil 0.65n \rceil$ rows in SHA256 order and restore source order; each has its own reference. Frozen final programs run in clean bounded CPU workspaces without agent feedback, unlike self-debugging (Chen et al., 2024). All 31 valid programs pass all five tests ($G = 1$), including the five unbound programs, without repairing their original execution requirement.

JSON matching requires exact list length/order, finite non-Boolean numbers, and $|a - b| \leq 10^{-9} + 10^{-7}|b|$. Native-return adaptations use $10^{-8} + 10^{-5}|b|$ after normalization with equal shapes/order; their normalizer accepts Boolean scalars. Neither searches alternative references. These predicates govern formal returns and additional-input tests.

Table 32: **Development results by model and support arm.** S: strict completion; K: requested computation return in formal execution; G: all five fixed additional-input tests pass. Fractions use valid trajectories. One of 32 positions is protocol-invalid (DeepSeek, both supports, quantiles) and remains unscored. All eight single-support pairs are observed: three favor delivery, none favor computation, and five tie. These are development results, separate from confirmation.

	DeepSeek-v3.2			GLM-5.2			All
Support	S	K	G	S	K	G	S
Neither (00)	4/4	4/4	4/4	0/4	4/4	4/4	4/8
Computation only (10)	2/4	2/4	4/4	1/4	4/4	4/4	3/8
Delivery only (01)	2/4	2/4	4/4	4/4	4/4	4/4	6/8
Both (11)	2/3	2/3	3/3	4/4	4/4	4/4	6/7

Table 33: All development strict-success outcomes by root and model. 1 and 0 denote observed success and failure; – is the original protocol-invalid position, not zero. No additional seeds, roots, or models were selected after the screen.

Root	Model	00	10	01	11
correlation abalone	DeepSeek	1	0	0	1
correlation abalone	GLM	0	0	1	1
f1 wine	DeepSeek	1	1	1	0
f1 wine	GLM	0	1	1	1
fit auto mpg	DeepSeek	1	0	0	1
fit auto mpg	GLM	0	0	1	1
quantile auto mpg	DeepSeek	1	1	1	–
quantile auto mpg	GLM	0	0	1	1

Results and stopping. Nineteen of 31 valid observations succeed. Seven GLM failures return correct computation but omit target nesting; three also flatten protected fields. Five DeepSeek failures exhaust the tool budget without qualifying execution, although saved numerics/schema/preservation and later tests pass. All eight 01–10 pairs are observed: three delivery wins, no computation wins, five ties. Against 00, delivery has four wins, two losses, two ties; both losses are DeepSeek correlation/fitting.

Malformed arguments at position 21 (DeepSeek quantiles/11) execute no tool and retain null endpoints. Continuation runs only eleven unstarted slots under unchanged science. Arm 11 has planned-position success bounds $[6/8, 7/8]$; segments are not randomized. All 493 scientific requests settle at estimated 10.395678 CNY; eight technical tasks add 90 requests/1.107507 CNY, with no unresolved reserve.

Prespecified investment rule. Gates require net gains at least two for 01–10 and one for 01–00; nonnegative model swap contrasts and at least two positive roots; six delivery-only G passes, three per model; baseline success below seven of eight; fewer than two invalids; and no admission fault. Worst-case completion evaluates gates while scientific nulls stay null. Observed values are three swap gains, two gains over baseline, model contrasts zero/three, three positive roots, eight delivery-only G passes, and four baseline successes. All gates pass, motivating confirmation investment rather than confirmatory inference. Added-JSON-interface results need not transfer to native returns requiring little delivery work.

F.7 Contract comparison: defect and unresolved correction

Design. Six exposed roots (Diabetes/Linnerud correlation and fitting, Digits F1, Iris quantiles) cross two models and JSON/NetCDF with computation/delivery support: 48 positions. Root selection uses fixed salted ordering with coverage constraints; conditions are randomized within root–model blocks. Requests, data, kernels, semantic obligations, and ceilings are shared. JSON uses `Dataset.from_dict` (xarray developers, n.d.a); NetCDF uses `open_dataset` with `netcdf4` (xarray developers, n.d.b). Consumers allow coordinate permutations and capped extra variables. Delivery modules share a Dataset constructor and differ only in writing. With no no-support arm, this compares supplied packages.

Original panel and preparation defect. All 48 positions are attempted once. Input-cap handling and observer file-limit failures leave positions 4 and 7 unknown; amended execution control continues only unstarted positions. There are 46 valid outcomes, 39 strict successes, forty correct formal returns, and 43 passing programs. Five valid failures end at observed rejected calls and two at the input ceiling. Four of seven failures pass all five program tests and start with correct kernels: three lack formal execution, one has a correct final-source-bound return and consumer-rejected artifact. The other three lack formal execution and fail tests; no observed $S = 1, G = 0$ occurs.

Let $p_{m,c,h}$ be the strict-completion proportion averaged over all six planned roots for model m, contract c, and support h, with C denoting computation support and D delivery support. The pre-

Table 34: **Recorded contract–support outcomes, with a defective public example.** Each row has six planned and recorded positions. Endpoint columns are passing counts, not rates; unknown positions are separate. There are 46 valid observations, 39 strict successes, 40 correct formal returns, and 43 passing final programs. The example defect described below prevents treating this panel as clean evidence of a delivery-contract effect.

Model	Contract	Support	Valid	S	K	G	Unknown
DeepSeek-v3.2	JSON	Compute	5	3	4	5	1
		Deliver	6	6	6	6	0
	NetCDF	Compute	6	6	6	6	0
		Deliver	6	5	5	5	0
GLM-5.2	JSON	Compute	6	5	5	6	0
		Deliver	5	4	4	4	1
	NetCDF	Compute	6	5	5	6	0
		Deliver	6	5	5	5	0

specified interaction, with equal model/root weights, is

$$\Theta_S = \tfrac{1}{2} \sum_m \{(p_{m,\mathrm{JSON},D} - p_{m,\mathrm{JSON},C}) - (p_{m,\mathrm{NetCDF},D} - p_{m,\mathrm{NetCDF},C})\}.$$

Unknowns retain signed weights, giving completion bounds $[16.7, 33.3]$ percentage points: DeepSeek $[50.0, 66.7]$, GLM $[-16.7, 0.0]$. The lower bound misses the predeclared 25-point discussion threshold; root heterogeneity and opposing model directions prevent a consistent effect claim. For known positions, $S = G - U + V$, with $U = \mathbf{1}(G = 1, S = 0)$ and $V = \mathbf{1}(S = 1, G = 0)$, yields weighted contributions $25.0 = 8.3 - (-16.7) + 0.0$: accounting, not mediation or a complete-plan estimate.

The public example puts bare lists in `coords`; JSON requires descriptors such as `{"dims":["coordinate"],"data":["a","b"]}`. The original raises `TypeError` in xarray 2024.11.0 with arbitrary zero data. All twelve JSON packages contain it; twelve NetCDF packages share the illustration. Delivery code bypasses it by serializing a Dataset correctly. DeepSeek's Diabetes-correlation JSON/computation artifact follows that example: computation/tests pass, but the consumer rejects it. This preparation defect confounds the support comparison; rejected-call failures have distinct causes. Arbitrary-value checks reproduce the defect; corrected examples pass zero/nonzero round-trips for all roots in both environments. These are technical checks, not model samples or revised grades.

Corrected example: incomplete evidence. Fresh sampling changes only the structural-example field in 24 packages, retaining roots, assignments, supports, consumers, references, limits, and scoring. The first ten records have eight valid and two response-timeout unknowns (positions 7/10). A frozen September 21 continuation permits concurrency after an operational gate, without retrying unknowns. An observation-chain failure at position 20 stops the queue; two other started positions are interrupted without terminal rows. Final coverage is 21 attempted, nineteen recorded, sixteen valid, five unknown, and 27 unstarted.

Table 35: **Corrected-example comparison, stopped after an observation-chain failure.** Tried includes two interrupted positions without terminal trajectory rows: 21 attempted, 19 recorded, 16 valid, five unknown, and 27 unstarted. All unknown outcomes remain null. This separately sampled panel is not pooled with the flawed-example comparison.

Model	Plan	Tried	Valid	S	K	G	Unk.	Not run
DeepSeek	24	11	8	7	8	8	3	13
GLM	24	10	8	8	8	8	2	14
Total	48	21	16	15	16	16	5	27

Fifteen of sixteen valid trajectories succeed; all sixteen have correct formal returns and passing final programs. The sole failure, DeepSeek JSON/computation on Linnerud correlation, passes

numerics/protection/consumer semantics but lacks final-source binding: its correct return belongs to earlier source, while the passing final program began with a supplied correct kernel. Full-plan Θ_S completion bounds are $[-141.7, 125.0]$ points, with no fully observed estimate. A difference of differences ranges over $[-200, 200]$; these are not rates or confidence intervals. The corrected prefix establishes neither an interaction nor its absence, and comparison with the flawed panel cannot identify the repair effect. Observation-chain failure limits evidence independently of the contract treatment.