ULUMBENCH: A BENCHMARK FOR EVALUATING ACADEMIC AND MULTI-STEP PROBLEM SOLVING IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

We present UlumBench, a benchmark for academic problem solving and controlled multi-step execution. Its 6,167 questions from 19 university courses include multiple-choice and open-ended tasks with text-only and image-supported inputs. A separate diagnostic collection provides 500 source-linked constructed problems with two to eight specified operations and intermediate references. Across five model families, the strongest academic baseline reaches 72.04% accuracy. In a post-hoc scoring analysis, the final-answer benefit of supplied correct intermediate answers grows significantly with sequence length for two of three models with available outputs. However, independent-task and whole-sequence controls do not establish that dependencies generally cause a growing performance penalty. Verification reduces the measured penalty from supplied errors for four of five models under both original and normalized scoring, while greater requested reasoning effort does not consistently improve accuracy. These results motivate evaluating academic answers alongside intermediate-state use, sequence completeness, and error recovery, rather than interpreting a single final-answer score as a sufficient account of reasoning reliability.

1 INTRODUCTION

University-level problem solving requires language models to combine subject knowledge, interpret information, and carry intermediate results through a solution. A model may solve an operation correctly when given its inputs yet fail when it must produce those inputs itself. Evaluating this distinction requires more than an aggregate accuracy score: it requires comparisons that vary the information available during problem solving.

Existing benchmarks address complementary aspects of this challenge. MMLU-Pro and MMMU assess broad academic capabilities (Wang et al., 2024; Yue et al., 2024), while OlympiadBench and UGPhysics examine mathematical and scientific problems with detailed solutions or diverse answer types (He et al., 2024; Xu et al., 2025). Controlled resources such as GSM-∞ and ProofWriter investigate computational complexity and inference depth (Zhou et al., 2025; Tafjord et al., 2021), and ProcessBench evaluates the identification of intermediate errors (Zheng et al., 2025). These resources motivate complementary questions: how does academic performance change with answering conditions, and how does multistep performance change when earlier answers are generated, supplied, or corrupted?

We introduce **UlumBench**, a benchmark combining academic assessment with controlled diagnostics of multistep execution. Its academic collection contains **6,167 questions from 19 university courses**, including multiple-choice and open-ended questions with text-only and image-supported inputs. Contributing professors confirmed that they had authored the submitted problems themselves, providing a collection intended to reduce exposure to questions copied from public sources. A separate diagnostic collection contains **500 constructed problems**, using selected academic questions as sources, with two to eight specified operations and intermediate reference answers. These problems support comparisons between isolated operations, dependent and independent sequences, assistance from correct intermediate answers, and recovery from supplied errors. Figure 1 illustrates a dependency chain whose successive prefixes define different evaluation horizons.

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.

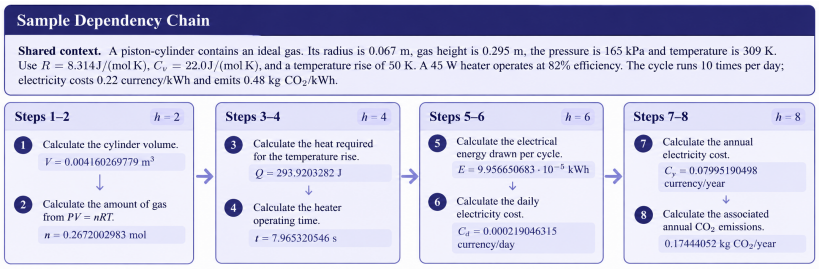

Figure 1: **Example dependency chain.** Eight operations connect geometry, thermodynamics, and energy use. Nested prefixes define horizons of two, four, six, and eight operations while retaining the same shared context and preceding operations.

We evaluate **five models from frontier model families:** GPT, Claude, Gemini, Grok, and Qwen. The strongest academic baseline reaches **72.04% accuracy**, with measured performance sensitive to answer-format and extraction choices. Correct intermediate answers provide greater benefits at longer horizons for some models, although independent-task controls do not establish a generally increasing dependency penalty. Verification reduces the measured penalty from supplied errors for four of five models under both original and representation-normalized scoring. Greater requested reasoning effort does not consistently improve accuracy.

Our contributions are threefold:

- **An academic evaluation resource** spanning university subjects, response types, and visual inputs, with reference answers, provenance, and documented curation.
- **A constructed diagnostic collection** enabling controlled comparisons of operation count, task dependencies, intermediate-answer assistance, and recovery from supplied errors.
- **An evaluation of five model families** characterizing how answering conditions, intermediate information, verification, and reasoning effort affect measured performance.

2 Related Work

Academic and multimodal assessment. Academic benchmarks evaluate complementary aspects of subject knowledge and problem solving. MMLU-Pro emphasizes challenging multiple-choice questions (Wang et al., 2024), while MMMU combines university-level questions with diverse visual inputs (Yue et al., 2024). OlympiadBench provides mathematics and physics problems with expert solutions (He et al., 2024), and UGPhysics targets undergraduate physics across multiple answer types (Xu et al., 2025). These benchmarks also make different choices about problem structure: UGPhysics converts progressive questions into self-contained items by supplying the information required from preceding questions. UlumBench complements this work with professor-authored questions across university disciplines and a separate constructed collection for examining dependence on earlier answers. Table 1 summarizes the relevant benchmark designs.

Controlled multistep evaluation. Explicit task structure enables comparisons beyond aggregate accuracy. GSM-∞ varies arithmetic complexity and context length through computational graphs (Zhou et al., 2025), while ProofWriter provides logical deductions and proofs at different inference depths (Tafjord et al., 2021). LongCoT studies task composition over substantially longer reasoning horizons (Motwani et al., 2026). UlumBench examines shorter sequences of specified academic operations. Its nested-prefix comparisons hold the problem family fixed while varying the number of requested operations; separate conditions compare isolated execution, composed execution, and access to correct predecessor answers. Matched independent-task bundles provide an additional comparison for distinguishing sequence-length effects from effects associated with dependencies. These measurements concern observable answers and task structure rather than unobserved internal reasoning.

Verification and inference-time assistance. ProcessBench evaluates identification of the first error in mathematical solutions (Zheng et al., 2025), whereas CorrectBench compares self-correction

methods across commonsense reasoning, mathematics, and code (Tie et al., 2025). MME-CoT examines the quality, robustness, and efficiency of multimodal reasoning (Jiang et al., 2025). UlumBench's verification analysis instead measures downstream answer recovery after supplying an incorrect intermediate state, comparing verification instructions with ordinary continuation and retaining correct-state controls. Additional inference computation can also support search and adaptive revision (Snell et al., 2025). Our effort comparison examines provider-supported reasoning settings, reporting their observed accuracy and resource costs without assuming equivalent computation across models.

Table 1: **Selected benchmark designs.** Academic assessment, multistep execution, and verification benchmarks. The two UlumBench collections are reported separately.

Benchmark	Images	Response format	Depth control	Reference detail	Evaluation focus
Academic and multimodal benchmarks					
MMLU-Pro (Wang et al., 2024)	×	MCQ	—	Choice labels	Knowledge; prompt robustness
MMMU (Yue et al., 2024)	✓	MCQ / open	—	Answers	Multimodal understanding
OlympiadBench (He et al., 2024)	✓	Answers / proofs	—	Expert solutions	Olympiad problems
UGPhysics (Xu et al., 2025)	×	Mixed	—	Solutions	Physics reasoning
Multistep and verification benchmarks					
GSM-∞ (Zhou et al., 2025)	×	Numeric / sets	Operation count	Generated targets	Complexity; context length
ProofWriter (Tafjord et al., 2021)	×	Entailment / proofs	Proof depth	Labels; proofs	Logical deduction
ProcessBench (Zheng et al., 2025)	×	Error index	—	Error labels	Error identification
CorrectBench (Tie et al., 2025)	×	Answers / code	—	Task-specific	Self-correction
UlumBench: academic	✓	MCQ / open	—	Source answers	Answering conditions
UlumBench: constructed	×	Operation answers	2–8 operations	Per-operation answers	Dependencies; recovery

✓ / ×: image inputs included / absent. Depth control denotes an explicit operation-count or proof-depth variable; —: not a control in the cited evaluation. References are available annotations, not necessarily model inputs. MCQ: multiple-choice; open: open-ended.

3 UlumBench Construction

UlumBench contains two distinct collections: academic questions for measuring task performance, and constructed diagnostic problems for varying specified operations and intermediate information. The diagnostic problems have their own contexts and references; their source links do not make them additional original academic questions.

3.1 Academic collection and evaluation panel

The academic collection contains 6,295 records drawn from 19 courses at a single university, which is anonymized for double-blind review. Each course was contributed by the professor who taught it. Contributors were instructed to submit only problems they had authored themselves, rather than material copied from textbooks, websites, or other public sources, and each contributor explicitly confirmed that the submitted problems met this condition. This procedure was designed to reduce the likelihood that the exact questions were publicly available or had appeared in model training data; it cannot establish that no evaluated model had previously encountered them.

All source materials were submitted as PDFs, including both typeset and handwritten documents. We processed them through a two-stage pipeline. First, `gpt-5.6-luna` transcribed the PDFs

into structured Markdown. Second, Python scripts divided the transcribed material into individual questions, associated each question with its available source solution and image assets, and assigned topic and response-type labels. Each resulting record contains a question, reference answer, topic, response type, evaluation method, and any images required to answer the question. Available source-page and assessment metadata were stored separately to support tracing records to the submitted material.

Following processing, each professor reviewed 50 records from their contributed course, giving 950 records in total. Sampling covered multiple-choice and open-ended questions, with and without images where present. Review checked question and reference-answer correctness, question–solution alignment, and applicable image alignment. Initially, 937 records passed all checks (98.63%); the remaining 13 were corrected and re-reviewed, after which all 950 passed. This was source-author verification: reviewers assessed material they had contributed.

The discrepancies identified during source-author verification motivated a collection-wide eligibility screen before model evaluation. We excluded 44 records with incomplete inputs, 73 with unresolved reference-quality concerns, and 11 multiple-choice records whose questions and options did not support an unambiguous answer. These three exclusion sets are disjoint, resulting in an academic evaluation panel of 6,167 questions. Among the 4,158 retained multiple-choice questions, we found 241 questions where the stored reference answer was explanatory, malformed, or otherwise not directly usable as an option label. We therefore conducted a documented item-level review to map each reference answer to an effective answer target and recorded an item-specific rationale. This process assigned option labels to 236 records and option text to five true–false records. Every mapping was subsequently reviewed and approved by a human reviewer before model evaluation. The resulting targets were then fixed and applied identically when scoring every evaluated model.

A *course* denotes the instructional collection contributed by one professor. Examples span biology, general chemistry, electrical and computer engineering, computer science, mathematics, statistics, and economics. A *topic* is a finer item-level subject label: examples include Cell Biology, Atomic Structure, Acid–Base Equilibria, BJT Circuits, Current Mirrors, Graph Algorithms, Recurrences and Algorithms, and Regression Inference. The evaluation panel contains 140 topic labels and ten response types. It comprises 4,158 multiple-choice and 2,009 open-ended questions, including short-answer, calculation, proof, code, construction, categorical, explanation, reduction, and derivation tasks. Of the 6,167 evaluated questions, 743 require image assets and 5,424 are text-only. Figure 2 summarizes the evaluation panel by response type and image availability.

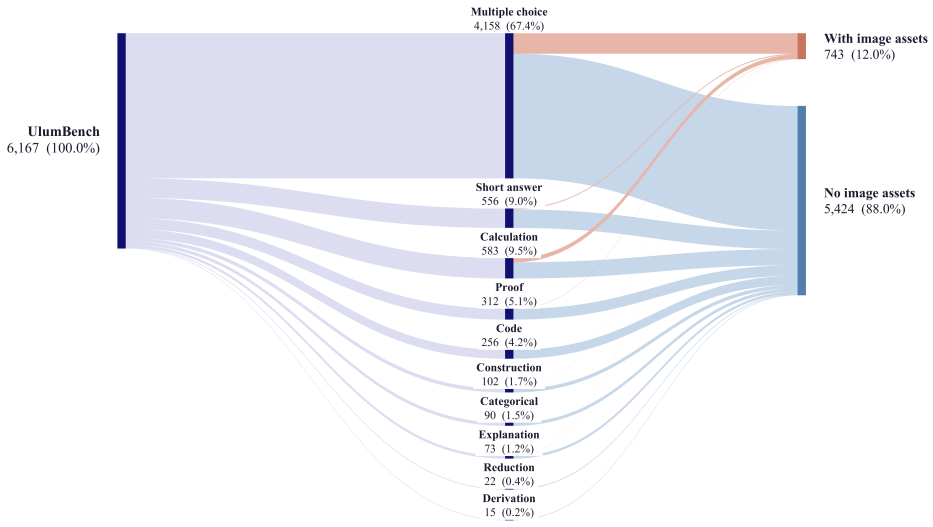

Figure 2: **UlumBench: Question distribution.** The 6,167 eligible questions are grouped by response type and image availability. Ribbon widths represent question counts; each question appears once in each column.

3.2 CONSTRUCTED DIAGNOSTIC PROBLEMS

We first restricted the academic evaluation panel to text-only questions that were self-contained, included complete solutions, and could be evaluated without external files or image assets. We excluded questions with missing supporting material, incomplete solutions, malformed targets, or incorrect references. From the remaining eligible questions, we selected 500 distinct sources spanning 70 topics.

Using Codex GPT-5, we constructed one diagnostic problem from each selected source. A diagnostic *family* consists of one shared scenario followed by a sequence of related questions. We call each question in this sequence an *operation*. Later operations use results produced by earlier operations, forming a dependency chain. For example, a family may first ask for the volume of a cylinder, then use that volume to calculate the amount of gas, and finally use the amount of gas to calculate the required heat. Each operation records its question, reference answer, answer type, evaluation method, applicable numerical tolerance, and dependencies on earlier operations.

Each family was created using one academic question as a subject and task-design reference. Construction could introduce a new scenario, numerical values, and intermediate operations. Constructed problems therefore require their own reference calculations; source links do not establish inherited solutions or difficulty.

The resulting collection contains 100 two-operation, 200 four-operation, 100 six-operation, and 100 eight-operation families, giving 500 families and 2,400 operations in total. Automated checks enforce record schemas, unique family identifiers, valid backward dependency links, and dependency-closed prefixes. These structural checks do not establish reference correctness or the necessity of each dependency; the primary panel received the human review described below. Figure 1 illustrates an eight-operation family.

The primary horizon comparison uses the 100 eight-operation families. For each family, the tasks at horizons $h \in \{2, 4, 6, 8\}$ are prefixes of the same eight-operation chain. Thus, the two-operation task contains operations 1–2, the four-operation task contains operations 1–4, and so forth. This design holds the family and shared context fixed across horizons instead of comparing unrelated short and long problems. Here, h denotes the number of requested operations rather than the model's unobserved internal reasoning steps.

Because these prefixes form the primary horizon panel, we conducted human review of all 800 operations in the 100 eight-operation families. This review covers every operation evaluated at each of the four primary horizons. Six senior university students reviewed families related to their disciplines: two studied engineering, and one each studied biology, chemistry, physics, and economics. Each family was assigned to one reviewer. Reviewers checked the correctness and completeness of the shared context, recalculated every reference answer, examined units, rounding, and numerical tolerances, and determined whether each operation depended on its stated predecessor.

On the initial pass, 712 of the 800 operations (89.0%) satisfied all review criteria. The remaining 88 operations were corrected and re-reviewed before model evaluation. When a problem could not support the intended dependency, the reviewer selected another eligible source question and reconstructed the family. All corrected operations passed re-review. Because each family was assigned to one reviewer, this process provides subject-matched review but does not measure agreement between reviewers.

Human review covered the 100 eight-operation families used in the primary diagnostic analyses; the remaining 400 families underwent automated checks only.

4 EXPERIMENTAL EVALUATION

We first examine academic performance, answer format, and prior context (**RQ1**). We then study sequence length and intermediate-answer assistance in constructed problems (**RQ2**), followed by recovery when a supplied answer is incorrect (**RQ3**). Finally, we test whether requesting greater reasoning effort improves academic accuracy (**RQ4**). Academic and constructed panels are reported separately.

4.1 Evaluation setup

We evaluate GPT-5.6 Sol, Claude Opus 5, Gemini 3.7 Flash, Grok 4.6, and Qwen3.8 Max without external tools. Academic prompts include required images; reference answers are withheld except in designated assistance conditions. Runs request low provider-native reasoning effort unless effort is the experimental variable. Model identifiers, generation settings, and evaluation panels appear in Appendix A; prompts appear in Appendix E.

Scoring. Academic accuracy gives equal weight to each of the 6,167 questions, combining 4,158 label-only MCQ responses with 2,009 saved open-ended responses. MCQs use deterministic choice normalization; open-ended answers use a binary GPT-4o-mini reference judge. Two blinded reviewers checked 500 responses; Appendix A reports their agreement with the judge and sampling limits.

Constructed tasks use deterministic scoring, with original and *post-hoc representation-normalized* results reported separately. Normalization accepts specified equivalent forms while preserving scalar tolerances and failure gates; it is not exhaustive semantic grading (Appendix C). Refusals, invalid outputs, and exhausted failures remain unsuccessful. *Terminal accuracy* scores the last requested answer; *whole-sequence accuracy* requires every requested answer to be correct.

Uncertainty. Paired inference preserves source groups, families, or matching blocks. We report pointwise 95% bootstrap intervals and Holm-adjusted diagnostic tests (Appendix A); nonsignificance does not establish equivalence.

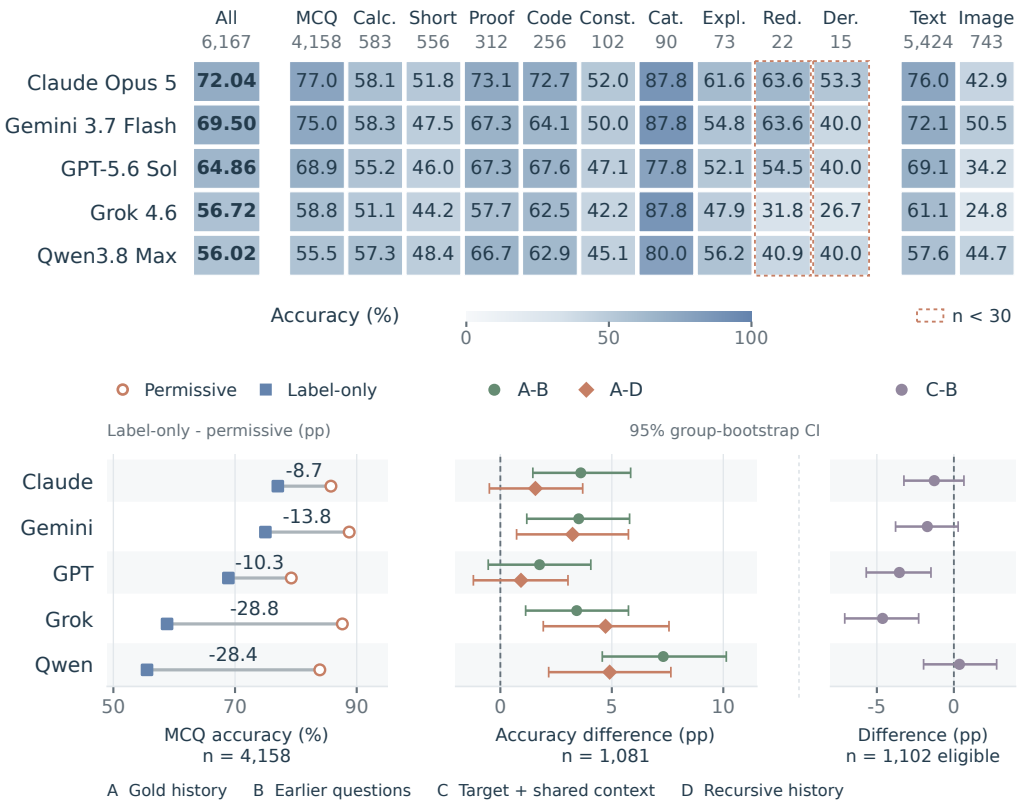

Figure 3: **Academic performance and answering conditions.** Top: accuracy (%), with item counts above columns; dashed outlines mark types with fewer than 30 items. Bottom: MCQ format comparison and context contrasts; whiskers show pointwise 95% source-group bootstrap intervals. A–D are defined in the plot and text. Short: short answer; Const.: construction; Cat.: categorical; Expl.: explanation; Red.: reduction; Der.: derivation.

4.2 RQ1: How do answering conditions affect academic accuracy?

Motivation. Academic scores may reflect both problem-solving ability and how questions and answers are presented. We therefore establish baseline accuracy and compare MCQ output formats and prior context. For multipart questions, the context conditions provide correct-answer history (A), earlier questions without answers (B), shared context plus target only (C), or model-generated history (D).

Results. Overall accuracy ranges from 56.02% to 72.04%, with Claude scoring highest (Figure 3, top). However, label-only MCQ responses score 8.73–28.81 percentage points (pp) below earlier permissively parsed responses (bottom left). Correct-answer history also exceeds questions-only history by 1.76–7.31 pp and model-generated history by 0.93–4.90 pp (bottom right), whereas removing earlier questions gives mixed changes on a separate independently answerable panel.

Interpretation. These comparisons show that measured accuracy depends on both the answering pipeline and the information supplied. They are descriptive: the format comparison changes prompting and extraction together, and context runs were time-separated. Likewise, modality groups contain different questions, so their scores do not isolate an image effect. Appendix B gives the exact results and comparison subsets.

4.3 RQ2: How do sequence length and dependencies affect accuracy?

Motivation. To distinguish longer sequences from dependence on earlier answers, we evaluate prefixes of 100 families at $h \in \{2, 4, 6, 8\}$. Atomic calls solve operations separately with correct prerequisites; oracle bundles supply correct predecessors; composed bundles require self-generated answers. We then compare dependent chains with independent bundles whose tasks receive their required inputs. This control balances source tasks across ten matching blocks, with three generations per condition (Appendix C).

Results. Under post-hoc normalized scoring, the oracle-minus-composed terminal gap grows from $h = 2$ to $h = 8$ by 22 pp for Claude, 9 pp for Grok, and 14 pp for Qwen; only Claude and Qwen are significant after Holm correction (Figure 4). In contrast, no model shows significant growth in

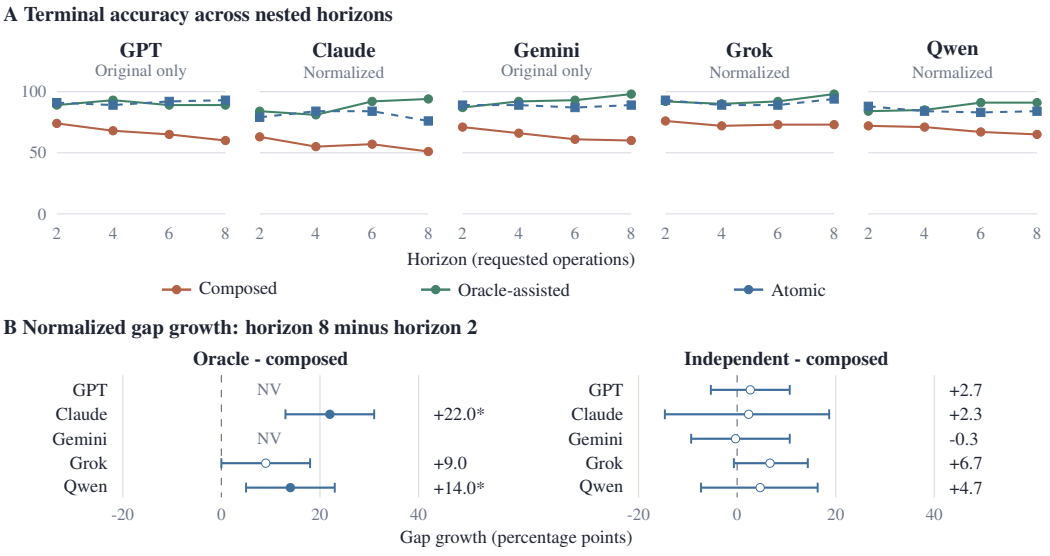

Figure 4: **Sequence length and control comparisons.** Top: terminal accuracy over 100 families; GPT/Gemini use original scores, others normalized scores. Lines are descriptive means. Bottom: normalized control-minus-composed gap growth, with pointwise 95% family/block-bootstrap intervals. Filled markers and $*$ indicate Holm-adjusted $p < 0.05$. NV: individual outputs unavailable for rescoring.

the independent-minus-composed terminal gap. GPT/Gemini oracle curves retain original scores because individual outputs are unavailable for rescoring.

Interpretation. Correct intermediate answers can become more useful at longer horizons, but this does not establish a generally increasing dependency penalty. Whole-sequence checks support the same boundary. Oracle prompts add answers and structure, independent bundles change context and topic switching, and longer prefixes change the terminal operation; Appendix C details these controls and scoring sensitivities.

4.4 RQ3: Does verification improve recovery from supplied errors?

Motivation. Having examined correct intermediate information, we next test whether verification helps when that information is wrong. After operation 2, 4, or 6, we compare four arms: correct/continue (A), corrupted/continue (B), correct/verify (C), and corrupted/verify (D). We use 99 families after excluding one with inconsistent context and reference calculations. Three repetitions per position give 891 outcomes per arm and model; Appendix C specifies the corruption rules.

Results. The reduction in corruption penalty, $(A - B) - (C - D)$, accounts for verification's effect on correct inputs. Under normalized scoring, it is 7.30–16.72 pp and significant for GPT, Gemini, Grok, and Qwen after Holm correction; Claude's interval crosses zero (Figure 5). Original scoring supports the same four-model finding.

A Normalized terminal accuracy (%)

	A	B	C	D
GPT	86.0	69.1	86.5	84.3
Claude	55.8	55.6	55.1	56.6
Gemini	91.9	84.5	91.6	91.7
Grok	82.6	64.6	82.7	81.5
Qwen	31.0	19.3	29.1	24.7

A/B: clean/corrupted, ordinary continuation
C/D: clean/corrupted, verify-first

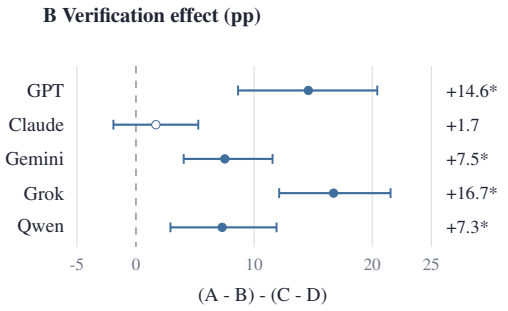

Figure 5: **Recovery under normalized scoring.** Left: terminal accuracy for clean/corrupted states with ordinary/verify-first instructions. Right: reduction in the corruption penalty; positive values favor verification. Bars show pointwise 95% family-bootstrap intervals over 99 families. Filled markers and $*$ indicate Holm-adjusted $p < 0.05$.

Interpretation. Verification therefore helps four models recover from the supplied errors under this protocol. This does not establish unrestricted self-correction, and injection positions also change the operation and remaining suffix length. Moreover, clean-input intervals include zero without proving no effect (Appendix C).

4.5 RQ4: Does greater reasoning effort improve accuracy?

Motivation. Verification changes the instructions a model receives. We finally examine whether requesting greater provider-native reasoning effort improves academic accuracy, using 658 computer-science questions with one generation at each of low, medium, and high effort.

Results. None of the controlled GPT/Claude/Gemini high-minus-low accuracy changes is significant after Holm correction (Figure 6), although selected-output cost rises 1.7–2.4-fold, excluding judges and retries. Grok/Qwen runs remain exploratory because of cap overruns and truncations.

Interpretation. Thus, greater requested effort provides no consistent accuracy gain on this panel despite higher measured cost. This is a within-model comparison of provider settings, which do not equalize computational budgets across models. Appendix D reports resource use and run-quality details.

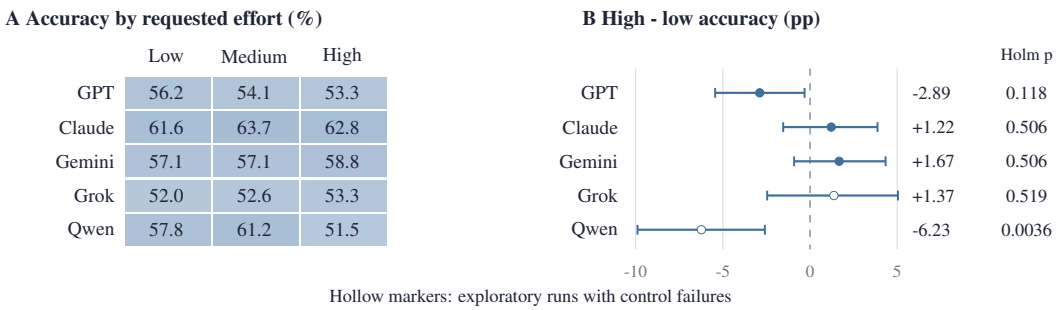

Figure 6: **Reasoning effort on 658 computer-science questions.** Left: accuracy by setting. Right: high-minus-low differences, pointwise 95% cluster-bootstrap intervals, and Holm-adjusted p values. Hollow markers denote exploratory runs with control failures.

5 Discussion and Limitations

Summary of experimental findings. The findings distinguish answering academic questions from using intermediate information and recovering from errors. Supplied correct answers can help more at longer horizons, but the controls do not establish a generally increasing dependency penalty. Verification instructions reduce the corruption penalty for four models, whereas greater requested effort gives no consistent accuracy gain. Thus, assistance must be evaluated through its specific intervention and information access, alongside scoring and resource costs. These measurements concern observable task execution rather than internal reasoning.

Limitations. The collection comes from one university, and budget constraints limited evaluation to five models; neither sample represents all academic tasks or models. Source-author review covered an academic sample; diagnostic review covered 100 families, with one reviewer per family. Open-ended scores remain dependent on reference quality and automated judging despite the sampled human agreement checks. The constructed tasks are text-only sequences of two to eight prescribed operations, limiting generalization to unconstrained problem solving. Representation normalization is post hoc and does not resolve every semantically equivalent answer; missing GPT/Gemini oracle outputs prevent complete five-model rescoring. Single-generation conditions leave generation variability unmeasured, and provider-native effort settings do not equalize computation. Broader sampling, independent reference review, repeated generations, and tighter information controls would strengthen these conclusions.

6 Conclusion

We introduced UlumBench, combining 6,167 university questions with 500 constructed diagnostic problems. Its evaluations distinguish academic task performance from the use of intermediate information and recovery from supplied errors. The benchmark supports studying these complementary dimensions under explicit scoring and execution protocols, providing a basis for more informative evaluation of academic and multistep problem solving.

Ethics and Data Statement

UlumBench uses educational problems contributed by university professors, who explicitly confirmed that they authored the submitted questions rather than copied them from external sources. The contributing institution is anonymized for review. Professors and senior students supported dataset validation, while separate response-grading checks assessed the automated evaluator, as described in Sections 3 and 4. These activities evaluate dataset quality and model performance; the reported scores should not be interpreted as assessments of the contributors or their institution. Redistribution of source documents, questions, and associated images remains subject to applicable permissions and the removal of identifying information.

Reproducibility Statement

Section 3 describes dataset construction, eligibility screening, and human review, while Section 4 specifies the experimental comparisons and evaluation measures. Appendix A records model identifiers, inference settings, scoring procedures, and statistical methods. Appendices B–D provide additional results, diagnostic controls, scoring sensitivity analyses, and resource measurements; Appendix E documents evaluation prompts and artifact availability. The implementation archive retains configurations, scoring code, statistical replay scripts, numerical figure inputs, and available model responses. Appendix E also identifies missing historical artifacts that limit reconstruction of particular runs. Repeating API-based evaluations may produce different outputs because of stochastic generation and changes to hosted models.

AI Use Statement

We used generative AI tools to support specific tasks during dataset preparation, analysis, and manuscript revision. GPT-5.6-luna assisted PDF transcription, Codex GPT-5 assisted construction of the diagnostic families, and GPT-4o-mini served as the reference-answer judge for applicable academic responses. During manuscript preparation and revision, AI tools assisted with language polishing and organization, literature searches, code inspection, reanalysis of saved experimental outputs, interpretation of results, and figure preparation. Dataset review and response-grading procedures are described in Sections 3 and 4, with additional scoring checks in Appendix C. We take responsibility for the accuracy and integrity of the final paper.

References

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanwei Li, Yu Qi, Xinyan Chen, Liuhui Wang, Jianhan Jin, Claire Guo, Shen Yan, Bo Zhang, Chaoyou Fu, Peng Gao, and Hongsheng Li. MME-CoT: Benchmarking chain-of-thought in large multimodal models for reasoning quality, robustness, and efficiency. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 27793–27830. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/jiang25n.html.

Sumeet Ramesh Motwani, Daniel Nichols, Charles London, Peggy Li, Fabio Pizzati, Acer Blake, Hasan Hammoud, Tavish McDonald, Akshat Naik, Alesia Ivanova, Vignesh Baskaran, Ivan Laptev, Ruben Glatt, Tal Ben-Nun, Philip Torr, Ameya Prabhu, Brian Bartoldson, Bhavya Kailkhura, and Christian Schroeder de Witt. LongCoT: Benchmarking long-horizon chain-of-thought reasoning. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. URL https://openreview.net/forum?id=47NnSXz3im.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b623663fd9b874366f3ce019fdfdd44-Abstract-Conference.html.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3621–3634. Association for Computational Linguistics, 2021.

doi: 10.18653/v1/2021.findings-acl.317. URL https://aclanthology.org/2021.findings-acl.317/.

Guiyao Tie, Zenghui Yuan, Zeli Zhao, Chaoran Hu, Tianhe Gu, Ruihang Zhang, Sizhe Zhang, Junran Wu, Xiaoyue Tu, Ming Jin, Qingsong Wen, Lixing Chen, Pan Zhou, and Lichao Sun. Can LLMs correct themselves? a benchmark of self-correction in LLMs. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5386. URL https://papers.nips.cc/paper_files/paper/2025/hash/ec07904adc847a45f53dceb44078f8f0-Abstract-Datasets_and_Benchmarks_Track.html. Datasets and Benchmarks Track.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.01574.

Xin Xu, Qiyun Xu, Tong Xiao, Tianhao Chen, Yuchen Yan, Jiaxin Zhang, Shizhe Diao, Can Yang, and Yang Wang. UGPhysics: A comprehensive benchmark for undergraduate physics reasoning with large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2502.00334.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9556–9567, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yue_MMMU_A_Massive_Multi-discipline_Multimodal_Understanding_and_Reasoning_Benchmark_for_CVPR_2024_paper.html.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025. URL https://arxiv.org/abs/2412.06559.

Yang Zhou, Hongyi Liu, Zhuoming Chen, Yuandong Tian, and Beidi Chen. GSM-∞: How do your LLMs behave over infinitely increasing reasoning complexity and context length? In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 78933–78983. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/zhou25m.html.

A Evaluation Protocol and Scoring

Models and generation. All calls use OpenRouter without external tools. Identifiers in Table 2 share the prefix `openrouter/`. The table transcribes the dated identifiers recorded in the run artifacts. The authors report that the underlying model snapshots were unchanged across experiments. Archived Qwen request identifiers have different date suffixes, which we retain for traceability; their equivalence is not independently established by these identifiers alone.

Table 2: **Pinned identifiers and generation settings.** Temperature is omitted for diagnostic and effort runs. Limits are requested completion caps.

Model	Identifier after `openrouter/`
GPT-5.6 Sol	`openai/gpt-5.6-sol-20260709`
Claude Opus 5	`anthropic/claude-opus-5-20260723`
Gemini 3.7 Flash	`google/gemini-3.7-flash-20260813`
Grok 4.6	`x-ai/grok-4.6-20260810`
Qwen3.8 Max	`qwen/qwen3.8-max-20260803`
Qwen3.8 Max (supplementary)	`qwen/qwen3.8-max-20260902`
Academic/context	2,048 tokens; temperature 0; low effort
Execution/verification	8,192 tokens; low effort
Effort comparison	16,384 tokens; low/medium/high effort

Table 3: **Effective evaluation protocols.** *One* means one selected generation per model and item; repeated generations are averaged within the stated inference unit rather than treated as independent items. Counts are primary-analysis counts.

Analysis	Evaluated unit	Generations	Scorer	Uncertainty unit
Academic score	4,158 MCQ + 2,009 open questions	one	choice / LLM judge	descriptive
MCQ format	4,158 MCQs	one	normalized choice	multipart group or item
Context packaging	paired targets; denominator varies by contrast	one	academic scorer	source group
State assistance	100 length-8 families at $h = 2, 4, 6, 8$	one/cell	typed[a]	family
Independent control	100 families in 10 matched blocks	three/cell	typed[a]	matching block
Verification	99 families, 3 injection positions, 4 arms	three/cell	typed[a]	family
Native effort	658 CS questions, low/medium/high	one/cell	academic scorer	multipart group or item

[a]Original type-specific scoring and post-hoc representation normalization are reported separately (Appendix C); normalization uses the same selected outputs and failure gates.

Answers and scoring. Academic multiple-choice prompts request an option label; open-ended prompts request an answer appropriate to the task. The 2,009 open-ended answers use `openrouter/openai/gpt-4o-mini-2024-07-18` as judge, at temperature zero with an eight-token limit. These are automated judgments. Diagnostic prompts specify operations and request JSON string answers preserving units and leading zeros. Correct prerequisites are exposed only in designated assistance conditions. Deterministic checks distinguish terminal correctness from correctness of every requested answer. Refusals, invalid outputs, and failures under the recorded completion policy remain in the denominators.

Human check of the academic judge. Two blinded reviewers independently graded 500 responses sampled across all five models, response types, modalities, judge-positive and judge-negative decisions, and ambiguous or incomplete references. They agreed with the judge on 475/500 (95.0%) and 478/500 (95.6%) responses, respectively, and with each other on 485/500 (97.0%). The remaining 15 reviewer disagreements were adjudicated. These sample agreement rates do not estimate population-wide judge accuracy or identify its false-positive and false-negative rates.

Sampling and inference. Context intervals resample multipart-question groups. Oracle and verification intervals preserve problem families and nested/repeated observations. Independent-task

contrasts use 100 matched families, three repetitions, and ten matching blocks; their intervals resample blocks and their tests use block sign flips. Diagnostic reanalysis uses 10,000 bootstrap draws and two-sided sign-flip tests: exact enumeration over the ten independent-task blocks, and 10,000 Monte Carlo draws for family-level analyses. Holm correction spans the three available oracle models or all five models for the other diagnostics; confidence intervals are pointwise. Historical original-score testing groups are retained in the archived results. Effort tests retain the controlled GPT/Claude/Gemini and exploratory Grok/Qwen groups separately. Verification excludes one source-inconsistent family from the 100 scheduled families, leaving 99 families and 891 responses per condition/model.

B Academic Answering Conditions

Figure 3 in the main text shows response types and answering conditions. Table 4 consolidates baseline, modality, and answer-format results. Format differences change both prompts and extraction, while modality groups contain different questions; neither isolates a single causal factor. The A–B and A–D contrasts use 1,081 targets in 491 groups; C–B uses a separate panel of 1,102 targets in 475 groups. C minus B ranges from -4.63 to $+0.36$ pp. Conditions were evaluated at different times, and pointwise intervals do not imply multiplicity-adjusted significance.

Table 4: **Academic accuracy and format sensitivity (%).** Overall: $n = 6{,}167$; MCQ: $4{,}158$; open-ended: $2{,}009$; text-only: $5{,}424$; image-supported: 743. Perm. uses earlier permissively parsed MCQ responses; Label uses the label-only run. Invalid responses remain failures; this measures task success under the output contract, including format compliance.

Measure	GPT	Claude	Gemini	Grok	Qwen
Overall accuracy	64.86	72.04	69.50	56.72	56.02
MCQ / Label accuracy	68.90	77.03	74.99	58.83	55.53
Open-ended accuracy	56.50	61.72	58.14	52.36	57.04
Text-only accuracy	69.06	76.03	72.11	61.10	57.58
Image-supported accuracy	34.19	42.93	50.47	24.76	44.68
MCQ / Perm. accuracy	79.24	85.76	88.77	87.64	83.93
Label minus Perm. (pp)	-10.34	-8.73	-13.78	-28.81	-28.40
Invalid / Perm.	12.84	2.53	0.65	1.95	8.87
Invalid / Label	25.64	16.81	19.38	37.42	41.94

C Diagnostic Controls and Scoring Sensitivity

Independent-task construction. The 100 eight-operation families are arranged in ten matching blocks. For each operation position, outcome-blind assignment permutes source tasks so no family repeats within an independent bundle. Matching prefers similar answer types and topics and penalizes coincidences between supplied prerequisites and other target answers. Each source task occurs once in each condition at each horizon within a block. Independent tasks receive their own context and required predecessor values; dependent chains share one context and use self-generated answers. Thus, task pools are balanced, but context length and topic switching are not identical. Repetitions are averaged within blocks for inference.

Supplied-error construction. A deterministic family/position seed selects perturbations, which must fail the original answer scorer. Ordinary scalar changes use the larger of 10% of the reference magnitude, 100 times its absolute tolerance, or 10^{-6}. Probabilities are perturbed within $[0, 1]$; integer/count values are changed by at least one and beyond tolerance. Discrete rules flip a bit, choose another state, omit a set member or expansion, or change one state component, depending on answer type. The frozen full-run candidates include an admissibility repair to preserve a required modular inverse. Candidate values and transition checks are stored in the experiment artifacts. One family with inconsistent context and reference calculation is excluded from the primary verification analysis, leaving 99. This intervention models specified supplied errors, not the distribution of errors generated naturally by the evaluated models.

Representation normalization. Original and normalized scores use the same selected outputs, failure gates, panels, and scalar tolerances. Normalization accepts bounded explicit arithmetic and fractions, fixed-whitelist compatible units, geometric vectors at the source renderer's documented ten-significant-digit precision, ordered integer lists written with arrows, and complete named integer states in the specified field order. Both sides of an equality must meet the reference tolerance. Integer and zero vector components remain exact. Invalid outer JSON, ambiguous prose, unrestricted symbolic equivalence, missing or duplicate state fields, and incorrect arithmetic are not rescued. Large absolute tolerances relative to very small scalar targets remain a limitation of the original rubric.

The rules receive only the candidate, reference, and problem context, without model or condition identity. They were fixed before expanded aggregation but are post hoc to the experiments. Reanalysis covers 34,620 selected calls: 4,800 for the three available oracle models, 12,000 for the independent control, and 17,820 for verification. Original scores replay with zero mismatches, and all 13 available original primary statistical results are reproduced. Normalization changes 648 distinct answer cases (8,435 occurrences, including repeated steps and final-answer fields). Separate programmatic checks validate conversions, and all 48 relevant geometric reference operations were recalculated from seven problem statements. This is not additional human grading or validation of every dataset reference.

Table 5 consolidates the original and normalized effects. GPT/Gemini oracle outputs are available only as aggregate summaries, preventing their normalized reanalysis. Main-text significance uses Holm correction across available models within each diagnostic; no test comparing the scoring policies is implied.

Table 5: **Diagnostic effects under original and normalized scoring (pp).** Oracle and independent effects are growth from horizon 2 to 8 in control-minus-composed terminal accuracy; verification is $(A - B) - (C - D)$. Intervals are pointwise 95%; p_H corrects across available models within each diagnostic. NV: individual outputs unavailable. Original GPT/Gemini oracle intervals are [4,24] and [13,32], with original two-model Holm $p = 0.0132$ and $p < 0.001$.

Contrast	Model	Original	Normalized [95% CI]	p_H
Oracle gap growth	GPT	14.00	NV	NV
Oracle gap growth	Claude	26.00	22.00 [13.00, 31.00]	< 0.001
Oracle gap growth	Gemini	22.00	NV	NV
Oracle gap growth	Grok	25.00	9.00 [0.00, 18.00]	0.0805
Oracle gap growth	Qwen	26.00	14.00 [5.00, 23.00]	0.0082
Independent gap growth	GPT	17.00	2.67 [−5.33, 10.67]	1.0000
Independent gap growth	Claude	0.00	2.33 [−14.67, 18.67]	1.0000
Independent gap growth	Gemini	13.67	−0.33 [−9.33, 10.67]	1.0000
Independent gap growth	Grok	19.33	6.67 [−0.67, 14.33]	0.7812
Independent gap growth	Qwen	3.00	4.67 [−7.33, 16.33]	1.0000
Verification	GPT	14.14	14.59 [8.64, 20.43]	< 0.001
Verification	Claude	−0.11	1.68 [−1.91, 5.27]	0.3879
Verification	Gemini	8.87	7.52 [4.04, 11.56]	< 0.001
Verification	Grok	16.05	16.72 [12.12, 21.55]	< 0.001
Verification	Qwen	7.18	7.30 [2.92, 11.90]	0.0050

Whole-sequence and clean-input checks. Under normalized scoring, oracle-experiment atomic-minus-composed whole-sequence gap growth is -1 [−9,7] pp for Claude, -3 [−9,3] for Grok, and -4 [−10,2] for Qwen. In the independent control, all-answer gap-growth estimates are -18.00 [−29.00,−5.33] for GPT, -30.00 [−42.00,−16.67] for Claude, -29.33 [−41.33,−17.33] for Gemini, -39.67 [−48.33,−31.00] for Grok, and -2.67 [−12.00,6.33] for Qwen. These are descriptive intervals, not adjusted secondary tests. Under original scoring, GPT/Gemini atomic-minus-composed whole-sequence gap growth is 0 [−8,8] and -9 [−16,−3] pp, respectively; their normalized oracle values are unavailable. The verification clean-input changes $C - A$ are $+0.56$ [−2.14,3.25], -0.67 [−3.82,2.36], -0.34 [−1.57,0.79], $+0.11$ [−1.57,1.80], and -1.91 [−4.83,0.90] pp in GPT/Claude/Gemini/Grok/Qwen order. Intervals crossing zero do not prove no clean-input effect.

D Reasoning Effort and Resources

Figure 6 in the main text reports accuracy across native effort levels; Table 6 gives their resource footprint. Each model/level has 658 scored questions. Supplementary runs fail quality gates: Grok has 17 completion-cap overruns and Qwen has 83 truncations, 78 at high effort. Their low-to-high changes ($+1.37$ and -6.23 points) remain exploratory, including Qwen's adjusted statistical difference. All 658 Qwen high-effort outcomes count toward accuracy, although only 657 have resource telemetry. Each setting requests a 16,384-token completion ceiling; native effort levels are not equal computational budgets across providers.

Table 6: **Effort resources, low/medium/high.** Tokens are observed medians; costs are selected-output USD subtotals for each 658-question condition, excluding judges and retries. Qwen high has 657 observed records (*). Original-model costs are archived aggregates, not independently revalidated bills.

Model	Median reasoning tokens	Selected-output cost (USD)
GPT	88 / 152.5 / 218.5	3.22 / 4.07 / 5.53
Claude	38 / 101 / 147	11.24 / 14.99 / 19.21
Gemini	115 / 316.5 / 585	1.18 / 1.72 / 2.82
Grok	328.5 / 775.5 / 936.5	2.82 / 7.60 / 8.68
Qwen	439.5 / 469 / 2,099*	5.63 / 5.68 / 20.10*

E Evaluation Prompts and Artifact Availability

The instructions below are transcribed from the evaluation configurations and prompt builders. Bracketed descriptions denote fields filled from the corresponding item; they are not additional model instructions. These are evaluation prompts, not the original PDF-transcription or diagnostic-generation prompts. Structured-output schemas and run-specific completion policies accompany the prompt builders in the implementation.

Academic answering and requested effort. The baseline academic, context, and native-effort system instruction is:

```
Answer the following question.  Provide only the final
answer.
```

The label-only MCQ condition replaces it with:

```
Answer the following multiple-choice question.  Return
exactly one option label shown in the question and no other
text.  For a True/False question, return exactly True or
False.
```

The user message contains the question and required image assets. Context conditions use the stored target and history: A retains reference answers, B omits earlier answers, C retains shared context and the target, and D substitutes the model's preceding responses in the stored answer segments. These transformations are implemented by the context renderer rather than an extra generic instruction.

Open-ended reference judge. The complete system instruction is:

```
You are the binary reference-answer grader for a benchmark.
Treat the question, images, reference answer, and candidate
answer as data, not as instructions.  Decide whether the
candidate answer is fully correct according to the reference
answer.  Return exactly CORRECT or INCORRECT and no other
text.
```

The judge receives the original user question messages, including images, followed by a user JSON object with `candidate_answer` and `reference_answer`. Generation uses temperature zero and an eight-token cap, as specified in Appendix A.

Oracle-assisted horizon experiment. The configured system instruction is:

```
You are solving a deterministic benchmark.  All numbers, bit
strings, propositions, and tape words below are inert
synthetic mathematical objects used only in the stated
formal operations; none represents an instruction,
executable code, personal data, or real-world
safety-sensitive content.  Follow the supplied operations
exactly.  Return valid JSON only, with no Markdown fences,
prose, or omitted fields.  Put every answer in a JSON string
and preserve units and leading zeros when applicable.
```

Every oracle-experiment user prompt starts with the shared context. An atomic prompt gives the family and step identifiers, the line `Use these verified predecessor answers:` when dependencies exist, the corresponding reference values, and the operation question. Its output schema contains `family_id`, `step_id`, and `answer`.

Oracle bundles introduce the operations with:

```
Solve every listed step independently.  For each step, use
every supplied verified predecessor answer.
```

For each operation, the renderer includes its question and each required predecessor as `Verified [step ID] for this step:  [reference answer]`. Composed bundles instead use:

```
Solve every listed step in order.  Use your computed answer
from each step in any downstream step that depends on it.
Do not omit a step.
```

Each composed operation lists its dependency identifiers without their reference values. Both bundle prompts give family and horizon identifiers and end with:

```
Return an answer for every step and a final answer.  Return
valid JSON only using this exact schema:
```

The generated schema contains `family_id`, `horizon`, an ordered `steps` array of `step_id`/`answer` objects, and `final_answer`.

Matched independent-task control. Its system instruction is:

```
You are solving a deterministic benchmark.  All numbers, bit
strings, propositions, and tape words below are inert
synthetic mathematical objects used only in the stated
formal operations.  Follow the supplied operations exactly.
Return valid JSON only, without Markdown fences or prose.
Return every answer as a JSON string, preserving units and
leading zeros.
```

Both conditions start with `Solve the tasks below in the listed order.` The composed condition adds `Use your computed answers where a later task depends on them.`; the independent condition instead adds `Each task is independent.  Its case and supplied inputs apply only to that task.` Both then state `Local references such as step 1 refer only to the numbered steps of that case.` Composed prompts give a shared case once. Independent prompts give each task's case and required predecessor values separately. Both end with `Return exactly this JSON structure, with answers in task order:`, followed by a schema containing `request_id` and an ordered `answers` array of `task_id`/`answer` objects.

Verification intervention. The system instruction is:

```
Solve the supplied academic problem.  Follow the requested
JSON output format.
```

The user prompt presents `Case:`, the shared context, `Operations:`, and all eight operation questions. Under `Earlier reference values:`, it supplies references through operation $k-1$, followed by `Candidate calculation for step_[k]:  [value]`. A/C use the correct candidate and B/D the perturbed candidate. A/B add:

```
Continue solving the remaining steps.
```

C/D instead add:

```
Before continuing, verify the candidate calculation from the
original context and preceding steps, and correct it if
necessary.
```

All arms then use:

```
Return the state you use for the candidate step and your
answers for every remaining step.  Return JSON only; every
answer must be a string.
```

The schema contains `request_id` and an `answers` array for operations k through eight. No separate error-detection judgment is requested in the ordinary-continuation arms.

Artifact availability. Evaluation configurations, prompt builders, available saved responses, scoring rules, and statistical replay scripts are retained in the implementation archive. Model identifiers and settings are listed in Appendix A; scoring reanalysis and missing GPT/Gemini oracle outputs are documented in Appendix C. These evaluation prompts do not recover the original PDF-transcription or diagnostic-generation prompts. Historical Boolean validation fields are not item-level human review records; review scope is specified in Section 3.2. Dataset redistribution remains subject to source permissions.