The Endless Exam: Mathematical Constructions from Today's Models toward Superintelligence

Anonymous authors
Paper under double-blind review

Abstract

We introduce the Endless Exam, a benchmark spanning fourteen parameterised families of mathematical construction problems, with verifiable scores that distinguish progress before and beyond published mathematical frontiers. Each submitted object is checked automatically for validity and assigned a relative quality score against a published frontier or construction baseline, without capping improvements at 1. The benchmark draws long-term challenges from open mathematical problems and generates larger instances by varying their parameters. Compact certificates allow large constructions to be verified without listing every element. Across nine models evaluated on 69 distinct instances, continuous quality scores distinguish performance even though none of the 30 published-frontier references is surpassed. Size–quality curves show how construction quality changes as problem size increases. We release the generators, verifiers, references, model responses and analysis to support continued measurement before and beyond human frontiers.

1 Introduction

Measuring progress toward artificial superintelligence (ASI) requires both scores that distinguish advances beyond the best human answers and tasks that continue to offer room for improvement. Mathematical construction problems meet these needs because finding a better object can demand a new idea, while its validity can be checked and its quality measured automatically. Varying the problem's parameters then provides a continuing supply of harder instances.

Benchmark saturation makes this need immediate: a recent study finds that 29 of 60 language-model benchmarks exhibit high or very high saturation, limiting their ability to distinguish leading models (Akhtar et al., 2026). ARC Prize recently reported 99.9% for GPT-6 Astra on ARC-AGI-3 Semi-Private with its Provider Adapter harness (Kamradt, 2026).[1] Once models reach benchmark ceilings, evaluation must continue to distinguish their progress and offer harder instances.

We introduce the *Endless Exam*, a benchmark built from fourteen parameterised families of mathematical construction problems, with 69 evaluation instances. Consider a progression-free set: a collection of points in $\mathbb{F}_5^n$ containing no three distinct points x, y, z with $x + z = 2y$ (Figure 1). Given a proposed set, the evaluator checks for forbidden triples and counts its points, rewarding larger valid sets even when they exceed every published construction. Changing the dimension produces a new task. We apply the same approach to graphs and codes, checking each construction and scoring its quality.

In many of these families, the best published constructions still leave a substantial gap to the proven upper bounds. For seven-colour Schur partitions and progression-free sets in $\mathbb{F}_5^6$, these bounds are 8.1 and 11.6 times the best published construction sizes, respectively, and the optimal sizes remain unknown.

To assess current model performance on these construction problems, we evaluate nine models in fourteen tool-free configurations: GPT-6 Astra, GPT-5.6 Luna, Claude Fable 5.1, Claude Opus 5.5, DeepSeek V4.1 Flash, Qwen3.8-27B and Qwen3.5 at 4B, 9B and 27B. To examine the effect of tool

[1] The 99.9% result uses high reasoning effort. With the Standard evaluation setup, the same report gives 62.7% at max effort.

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.

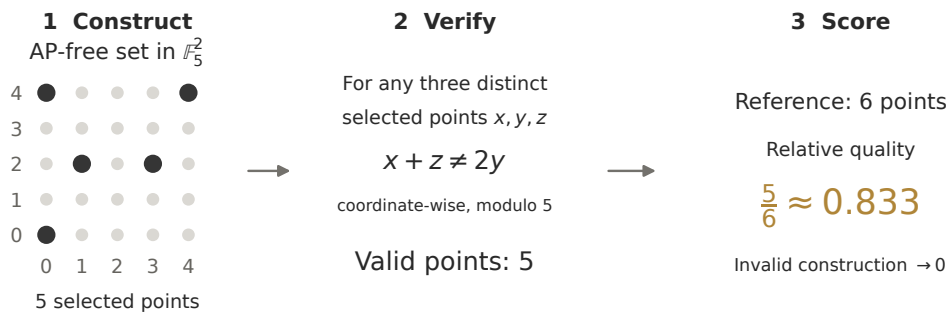


Figure 1: An illustrative AP-free set in $\mathbb{F}_5^2$. Here x, y, z denote distinct selected points, with coordinate-wise arithmetic modulo 5. Dividing the five valid points by the six-point optimum gives a relative quality of $5/6 \approx 0.833$.

access, we additionally evaluate Astra, Luna and Opus 5.5 at high effort with code and web access on the 69 instances.

Our contributions are:

1. **An uncapped measure of mathematical progress.** Relative quality puts diverse construction objectives on a common scale and preserves the magnitude of improvements beyond a reference.
2. **An expandable suite of construction problems.** Fourteen families generate new instances by varying parameters such as dimension, graph degree and code length; we evaluate 69 instances.
3. **Automatic verification at large scales.** We verify constructions through their mathematical properties, even when the optimum is unknown. Compact descriptions and algebraic certificates support evaluation of objects too large to enumerate, including submitted codes with over a trillion codewords (Section 3.4).

The Endless Exam is an open-source benchmark, and we welcome community contributions of new constructions, task families and verifier improvements.

2 Related work

Mathematical reasoning benchmarks. Humanity's Last Exam (Phan et al., 2025) and FrontierMath (Glazer et al., 2024; Epoch AI, 2026a) assess expert-level problem solving. IMProof-Bench (Schmitt et al., 2025) and First Proof (Abouzaid et al., 2026) evaluate research-level proofs, IMO-AnswerBench (Luong et al., 2025) evaluates Olympiad answers, and ARC-AGI (Chollet, 2019; Chollet et al., 2025) tests abstract reasoning. Table 1 compares representative mathematical benchmarks and problem collections.

Verifiable constructions and problem generation. MathConstruct (Balunović et al., 2025) checks constructed objects and varies problem parameters to assess robustness, while MathConstraint (Pati et al., 2026) generates constraint problems, uses solvers to determine whether solutions exist, and checks submitted constructions. The Endless Exam assigns graded scores to valid constructions: for the same instance, more set elements, more graph vertices or fewer scalar multiplications receive higher scores. This lets us measure how much constructions improve at a given size and how their quality changes at larger sizes.

Open mathematical problems. FrontierMath: Open Problems and the Erdős Problems and Optimization Constants collections track unsolved questions and mathematical bounds (Epoch AI, 2026b; Bloom, 2021; Davis et al., 2026). HorizonMath (Wang et al., 2026) evaluates open problems with automated verification and reports the proportion solved. EternalMath (Ma et al., 2026) derives

Dataset	Problems	Difficulty	Unsolved	Non-proof	Open eval.	Auto-check	Uncapped score
FrontierMath	338	Univ.–Res.	×	✓	×	✓	×
FrontierMath: Open Problems	50	Research	✓	✓	×	✓	×
IMProofBench	77	Research	✓	×	×	×*	×
First Proof	10	Research	×	×	×	×	×
Erdős Problems	1,217	Research	✓	×	×	×	×
IMO-AnswerBench	400	Olympiad	×	✓	✓	✓	×
Opt. Constants	115	Research	✓	✓	×	×	×
MathConstruct	127	Olympiad	×	✓	✓	✓	×
MathConstraint	329	Tunable	×	✓	✓	✓	×
HorizonMath	113	Research	✓	✓	✓	✓	×
Endless Exam	**69**	Research, scalable	✓	✓	✓	✓	✓

Table 1: Comparison of mathematical benchmarks and problem collections, using the cited releases. Unsolved indicates inclusion of open research targets. Our 69 instances span 14 families. Scalable denotes larger instances generated by increasing parameters under the same rules. *IMProofBench's main proofs are human-graded; follow-up answers are automatically checked.

executable, parameterised tasks from mathematical literature, while Formal Conjectures (Firsching et al., 2026) collects research problems formalised in Lean for proof discovery and autoformalisation. The Endless Exam scores construction quality on an *uncapped scale*, distinguishing the magnitude of improvements even after a published frontier is surpassed.

Mathematical discovery systems. FunSearch (Romera-Paredes et al., 2024), AlphaEvolve (Novikov et al., 2025) and AlphaTensor (Fawzi et al., 2022) have improved mathematical constructions. AlphaEvolve evolves programs using numerical quality feedback, and its mathematical study spans 67 problems with generalisation across parameters (Georgiev et al., 2025). To compare such progress across systems, families and problem sizes, the Endless Exam combines relative quality, an uncapped overall score and size–quality curves within a common evaluation framework.

3 The exam

3.1 Design principles

We select construction families that support continued measurement through improving construction quality and increasing problem size. Each task specifies the conditions a construction must satisfy and an objective to optimise, such as maximising the number of graph vertices or minimising the number of scalar multiplications in a matrix multiplication algorithm. Among valid constructions, every improvement in the objective raises the score, including improvements beyond the reference. The evaluator checks each construction's mathematical properties and computes its objective value directly, without a model judge. These checks do not require knowing the optimum, allowing progress to be measured even on open problems. Basing scores on the submitted object also makes them resistant to reward hacking through persuasive prose or unsupported quality claims. Compact descriptions and certificates extend this approach to objects too large to enumerate. Varying the parameters creates new instances under the same mathematical rules; larger instances admit more possible constructions and can make high-quality ones harder to find, providing more demanding tasks as models advance.

3.2 Families and instances

The fourteen families span forbidden-pattern sets, geometry, graphs, codes and algebraic designs (Table 3). We evaluate the 69 instances of benchmark release `bench-v1.0`. To our knowledge, the mathematical optimum is unknown for each of the 69 main-evaluation instances. For 30 instances, *published frontiers* provide a reference for measuring how closely models approach existing mathe-

matical results and whether they improve on them. For the remaining 39, we use verified constructions generated by the algorithms described in Appendix E as reproducible baselines. These baselines may be weaker than published constructions. Every reference is supported by a verified construction and remains unchanged within `bench-v1.0`. Both groups use the same relative-quality score, which increases as constructions improve, including beyond the reference (Section 4).

3.3 Representative tasks

These three examples are drawn from the 69 evaluation instances and illustrate graphs, geometry and compactly represented codes. Reference values appear here to explain scoring; model prompts contain only the task and answer requirements.

Degree–diameter graphs: degree 4, diameter 5

Task. Construct an undirected graph with as many vertices as possible, subject to two constraints: each vertex has at most four neighbours, and any two vertices can be connected by a path of at most five edges. Self-loops and repeated edges are not allowed.
Answer and verification. An edge list specifies the graph. The verifier checks vertex degrees and shortest-path distances, then counts the vertices of a valid graph. GPT-6 Astra at high effort without tools submits a graph with 203 vertices, compared with the published frontier of 364. Its relative quality is therefore $203/364 \approx 0.558$ (Section 4).

Heilbronn triangles: 46 points in a square

Task. Place 46 distinct points in the unit square so that the smallest triangle formed by any three points has the largest possible area. Each coordinate must be a multiple of 10^{-4}.
Answer and verification. Points are submitted as pairs of integers between 0 and 10,000, scaled by 10^{-4}. The verifier checks their count, distinctness and range, then computes the minimum triangle area using exact integer arithmetic and the coordinate scale. A larger minimum area scores higher. This instance is evaluated against a construction baseline, illustrating how verified geometric quality provides a continuous score.

Trifference codes: ternary strings of length 64

Task. Find as many distinct length-64 strings over $\{0, 1, 2\}$ as possible. For every three distinct strings, there must be at least one position where their symbols are all different: one is 0, one is 1 and one is 2. The objective is the number of strings satisfying this condition together.
Answer and verification. A generator matrix describes all linear combinations of its rows modulo 3. The verifier checks the three-string condition algebraically and counts codewords from the number of independent rows, without enumerating the full code. For this instance, GPT-6 Astra at high effort with tools submits an 11×64 matrix with 11 independent rows. Choosing a coefficient of 0, 1 or 2 for each row gives $3^{11} = 177{,}147$ distinct strings. This is 9 times the construction baseline, so its relative quality is 9. Appendix D.4 describes the certificate check.

3.4 Answer formats and verification

The examples in Section 3.3 illustrate two accepted forms of answer: an explicit object and a compact description. Across the suite, compact forms include products, lattice orbits, digit sets, difference sets, Cayley generators and algebraic certificates. The verifier either expands the description and checks the resulting object, or verifies the components and algebraic conditions that establish its validity and size. In our evaluations, checking the finite factors of a submitted Shannon code certifies over a trillion codewords (Appendix B.2).

The primary verifier checks whether each submitted construction satisfies the mathematical constraints and computes its objective value. For every construction it judges valid, a second verifier checks the same constraints and recomputes the objective value using a separate implementation of the mathematical checks. The two verifiers agree on validity and objective value for all these model constructions and all 69 reference constructions. We also test the verifiers on deliberately invalid constructions and compare their decisions with exhaustive checks on small Shannon and trifference instances. They reject all invalid test cases and agree with the exhaustive checks (Appendix D.1). Appendix D gives the certificate checks and their size and time limits.

3.5 Evaluation protocol

All configurations receive the same instances, with prompts specifying the problem, its parameters and the required answer format (Appendix F). In the tool-free track, models produce one response per instance within a budget of 128k output tokens, including reasoning tokens. Invalid responses and generation-budget exhaustion score zero. Each effort level is evaluated as a separate configuration (Appendix G).

The tool-assisted track gives GPT-6 Astra, GPT-5.6 Luna and Claude Opus 5.5, all at high effort, code execution and web access while retaining the mathematical prompts and answer formats. Each evaluation has four CPU threads, 16 GiB of memory and a two-hour wall-clock limit, with no limit on total output tokens. When an evaluation finishes or reaches a resource limit, we assess the construction saved before the deadline. Valid constructions receive their quality scores; missing or invalid submissions receive zero (Appendix H).

4 Scoring

4.1 Relative quality and overall score

We measure *relative quality* with the same formula for both reference groups. For parameters p, verified objective a and reference $h(p)$,

$$r(a,p) = \begin{cases} a/h(p), & \text{valid maximisation answer,} \\ h(p)/a, & \text{valid minimisation answer,} \\ 0, & \text{invalid or empty response.} \end{cases}$$

Relative quality 1 matches the reference, and every further improvement increases the ratio without clipping it at 1. References remain unchanged within each benchmark version, so successive improvements remain comparable.

For instances with a published frontier, $h(p)$ is the objective value of the cited construction. For the remaining instances, $h(p)$ is the best objective value among the verified constructions produced by the algorithms in Appendix E. Relative quality above 1 indicates improvement over the reference; surpassing a published frontier yields a candidate mathematical record. Appendix E documents the references and baseline procedures.

Published eight-colour Schur constructions illustrate how scores distinguish successive improvements. Against the historical reference of 5,041 elements derived from Fredricksen & Sweet (2000), constructions of sizes 5,286 (Rowley, 2021) and 5,362 (Bengone et al., 2026) receive scores of 1.049 and 1.064, respectively. Both surpass the reference, but the scores distinguish their quality. The current benchmark uses 5,362 as its eight-colour reference (Appendix B.4).

The **overall score** is $S = (100/69)\sum_{i=1}^{69} r_i$, where r_i is the relative quality for instance i. Invalid submissions receive zero, and a score of 100 matches the references on average. Table 2 also reports mean relative quality for each reference group, with 95% confidence intervals obtained by bootstrapping instances within that group.

Each construction instance provides a richer picture of performance than a binary correct/incorrect label. Verification establishes whether the answer is valid, and relative quality measures how well the construction performs against its published frontier or construction baseline. For example, valid constructions with relative quality scores of 0.5, 0.9 and 1.2 would all pass a validity check, but receive different quality scores. Our 69 instances therefore reveal more than 69 success-or-failure outcomes: they show how close each construction comes to its reference and how far it improves beyond it.

4.2 Gap closed

Gap closed measures progress from the reference h to a proven bound b. For maximisation it is $\max(0, \ln(a/h)/\ln(b/h))$. For minimisation, the ratios a/h and b/h are replaced by h/a and h/b. Constructions that do not improve on the reference score 0, while reaching the bound scores 1; invalid answers also score 0. We average gap closed over the 64 instances with proven bounds: 30

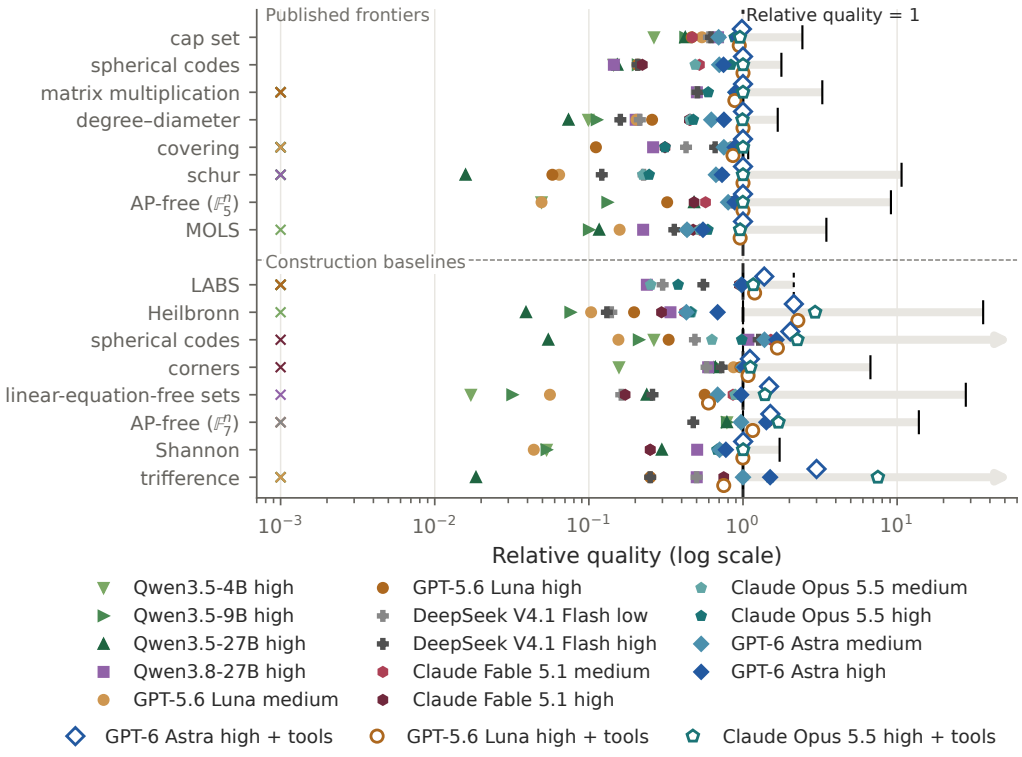


Figure 2: Mean relative quality across fourteen families. Model points and bar endpoints are normalised to the reference and averaged over the same instances in each row; model means include zero scores. Upper rows use published frontiers; lower rows use construction baselines. A ratio of 1 matches the reference. Grey bars extend from the reference level of 1 to mathematical bounds or conjectured targets. A solid mark at the right end indicates a proven bound; a dashed mark indicates a conjectured target. An arrow means the endpoint lies beyond the plotted range. Filled markers indicate tool-free evaluations and hollow markers tool-assisted evaluations. Crosses mark means below 0.001. AP-free: arithmetic-progression-free; LABS: low-autocorrelation binary sequences; MOLS: mutually orthogonal Latin squares.

use published frontiers and 34 use construction baselines. The 5 LABS instances use a conjectured target and are reported separately (Appendix C.15).

5 Results

5.1 Distinguishing today's models

Tool-free overall scores span 7.55–91.90 (Table 2). On the 30 instances with published frontiers, mean relative quality spans 0.07–0.76 despite a 0% breakthrough rate for all fourteen configurations. GPT-6 Astra at high effort reaches 0.76 (95% CI 0.69–0.83). Mean relative quality on the 39 instances with construction baselines ranges from 0.08 to 1.04. Astra high also achieves the largest tool-free mean gap closed, at 1.6% across the 64 instances with proven bounds. Figure 2 and Appendix C show the differences across families.

Both model size and reasoning effort affect quality. On instances with published frontiers, Qwen3.5's mean relative quality rises from 0.07 to 0.12 to 0.13 at 4B, 9B and 27B (Figure 7). Raising effort from medium to high increases Astra's mean from 0.66 to 0.76 and GPT-5.6 Luna's from 0.16 to 0.25. Claude Fable 5.1's overall score instead falls from 71.90 at medium effort to 43.73 at high effort, where 32 of the 69 responses exhaust the 128k output budget. Additional reasoning effort can therefore consume the budget without producing a completed construction. Opus 5.5 scores 66.70 at

configuration	Overall score	Mean relative quality: Published frontiers (30)	Mean relative quality: Construction baselines (39)	Valid fraction	Gap closed
Qwen3.5-4B high	7.55	0.07 [0.03, 0.11]	0.08 [0.03, 0.15]	0.38	0.000
Qwen3.5-9B high	14.52	0.12 [0.07, 0.17]	0.17 [0.07, 0.27]	0.43	0.000
Qwen3.5-27B high	16.18	0.13 [0.06, 0.22]	0.18 [0.09, 0.29]	0.52	0.000
Qwen3.8-27B high	38.07	0.31 [0.20, 0.42]	0.44 [0.28, 0.59]	0.62	0.004
GPT-5.6 Luna medium	16.70	0.16 [0.10, 0.24]	0.17 [0.08, 0.27]	0.57	0.000
GPT-5.6 Luna high	34.85	0.25 [0.16, 0.35]	0.42 [0.29, 0.55]	0.68	0.002
DeepSeek V4.1 Flash low	37.11	0.39 [0.29, 0.49]	0.36 [0.23, 0.48]	0.78	0.000
DeepSeek V4.1 Flash high	44.90	0.38 [0.26, 0.50]	0.50 [0.34, 0.67]	0.75	0.006
Claude Fable 5.1 medium	71.90	0.59 [0.48, 0.69]	0.82 [0.65, 0.99]	0.86	0.012
Claude Fable 5.1 high	43.73	0.54 [0.40, 0.68]	0.36 [0.20, 0.51]	0.52	0.003
Claude Opus 5.5 medium	66.70	0.58 [0.47, 0.68]	0.74 [0.57, 0.93]	0.88	0.006
Claude Opus 5.5 high	69.81	0.57 [0.46, 0.69]	0.79 [0.61, 1.01]	0.83	0.009
GPT-6 Astra medium	75.34	0.66 [0.58, 0.73]	0.83 [0.71, 0.95]	0.99	0.006
GPT-6 Astra high	**91.90**	0.76 [0.69, 0.83]	1.04 [0.89, 1.22]	1.00	0.016
GPT-6 Astra high + tools	143.16	0.998 [0.995, 1.000]	1.76 [1.43, 2.30]	1.00	0.063
GPT-5.6 Luna high + tools	119.35	0.961 [0.931, 0.987]	1.37 [1.09, 1.68]	0.96	0.047
Claude Opus 5.5 high + tools	**180.06**	0.985 [0.965, 0.998]	2.43 [1.52, 4.10]	1.00	0.075

Table 2: Results on 69 instances. The final three rows use code and web access (Section 3.5). Overall score is uncapped. Brackets show 95% confidence intervals. Validity uses all 69 responses; gap closed uses 64 instances with proven bounds (Section 4.2).

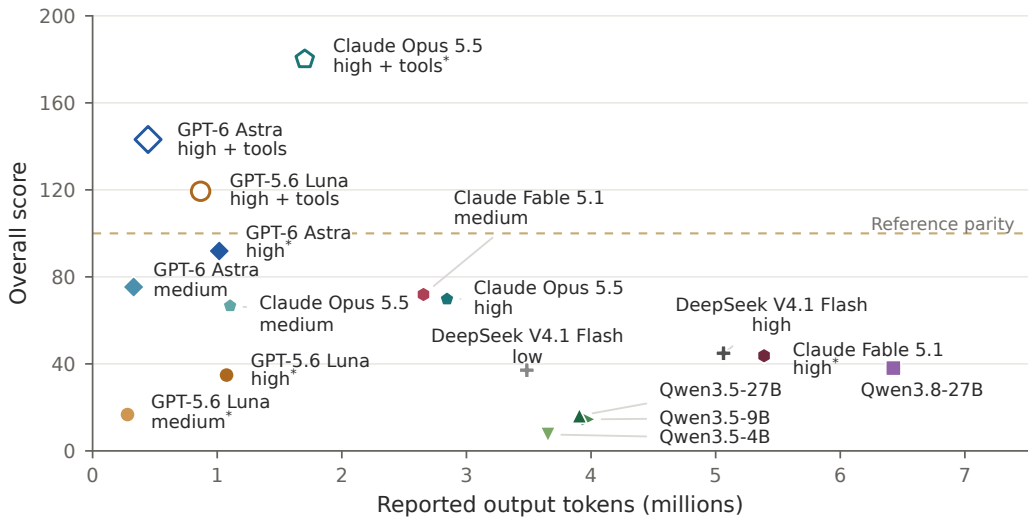


Figure 3: Overall score and output tokens for 69 instances, including reasoning and all model turns. For models marked with an asterisk, some output-token counts are unavailable, so the plotted totals are lower bounds. All Qwen models use high effort; tokenisation varies by model.

medium effort and 69.81 at high effort, with 61 and 57 valid answers, respectively. Figure 3 relates overall scores to output tokens: Astra medium outscores Fable 5.1 medium with fewer tokens, while Qwen3.8-27B uses the most reported output tokens but scores below both.

Code and web access. GPT-6 Astra at high effort produces valid constructions on all 69 instances without tools. With code and web access, it improves their average quality, raising its overall score from 91.90 to 143.16 while maintaining 100% validity. GPT-5.6 Luna's score at high effort rises from 34.85 to 119.35 as its valid answers increase from 47 to 66. On the 45 instances with valid answers in both tracks, its mean relative quality also rises from 0.50 to 1.32. Luna therefore improves both

its ability to produce valid answers and the quality of those answers. Claude Opus 5.5 rises from 69.81 to 180.06, with validity increasing from 57 to 69 instances.

On the 30 instances with published frontiers, tool-assisted Astra, Luna and Opus 5.5 match 29, 23 and 23 references, respectively, without surpassing any. The evaluated models already match many published frontiers in this group, but have yet to surpass one. On the 39 instances with construction baselines, Astra, Luna and Opus 5.5 improve on the construction baselines for 33, 25 and 33 instances (Appendix H). Opus 5.5 achieves the highest overall score, with its advantage concentrated in Heilbronn and the length-64 trifference instance, while Astra achieves the highest mean relative quality on published-frontier instances. These improvements already demonstrate scoring beyond 1; future improvements over published frontiers would be measured in the same way.

5.2 Quality as problem size increases

With tools at high effort, mean gap closed remains small: 6.3% for GPT-6 Astra, 4.7% for GPT-5.6 Luna and 7.5% for Claude Opus 5.5 across the 64 instances with proven bounds. Despite improvements over some references, the models close only a small fraction of the logarithmic gap to the proven bounds, leaving substantial room for further improvement under this metric.

If future models close a substantial fraction of these gaps and the existing instances become easy, larger parameter settings can provide more demanding tasks under the same mathematical rules. The name *Endless Exam* reflects these two routes for continued measurement: improving construction quality within an instance and extending the task to larger dimensions, graph diameters or codeword lengths as models advance.

To examine how current construction methods scale, we evaluate four configurations: DeepSeek V4.1 Flash at low effort, and Qwen3.8-27B, GPT-6 Astra and GPT-5.6 Luna at high effort. Each configuration is tested at four sizes in five benchmark families and an additional integer progression-free task, with two independent responses per size and the same token budget. Size–quality curves show whether construction quality keeps pace with the reference as the task grows. Figure 4 shows three families; Appendix C gives all six tasks and an additional graph series with maximum degree four.

Larger graph and spherical-code instances expose widening gaps to the reference constructions. For graphs of maximum degree three, all four models have lower mean relative quality at diameter 10 than at diameter 4. At diameter 10, the means are 0.30 for Astra, 0.17 for Luna, 0.09 for Qwen3.8-27B and 0.00 for DeepSeek V4.1 Flash. Astra, Luna and Qwen3.8-27B still return valid graphs at this size, but their constructions fall well below the reference. All four models' spherical-code scores also fall between dimensions 10 and 16. On the larger instances, some models fail to produce valid constructions, while others return valid constructions that fall further below the reference.

The gap to a proven bound can also widen while a model stays close to the reference. For cap sets in six dimensions, 112 points meet both the reference and the proven upper bound. In twelve dimensions, both Astra answers combine two such caps to obtain 12,544 points, reaching relative quality 0.97 against the reference of 12,928 points. The proven upper bound is 50,571, so these answers reach only 24.8% of that bound. The gap between the reference and the proven upper bound also grows across the graph and Schur size series (Appendix C). These bounds need not be attainable; their separation from known constructions marks an unresolved mathematical gap at larger sizes.

5.3 Sensitivity to computational search budget

The reference searches also let us examine how much construction quality improves with additional computation. For these procedures, eight families are classified as *search-resistant*, with strong constructions based on products, lattices, digit sets or recursions, while five may benefit from direct search. Across all eight search-resistant families, increasing the search budget from 10 to 600 seconds improves the baseline by less than 10% on every measured instance, while Heilbronn benefits more (Figure 13; Appendix E.5). Trifference is excluded because its baseline procedure selects and combines existing codes without using the search-time budget.

These measurements describe the tested search algorithms. Tool-assisted models can use broader methods, and discovery systems such as AlphaEvolve can develop new algorithms (Georgiev et al.,

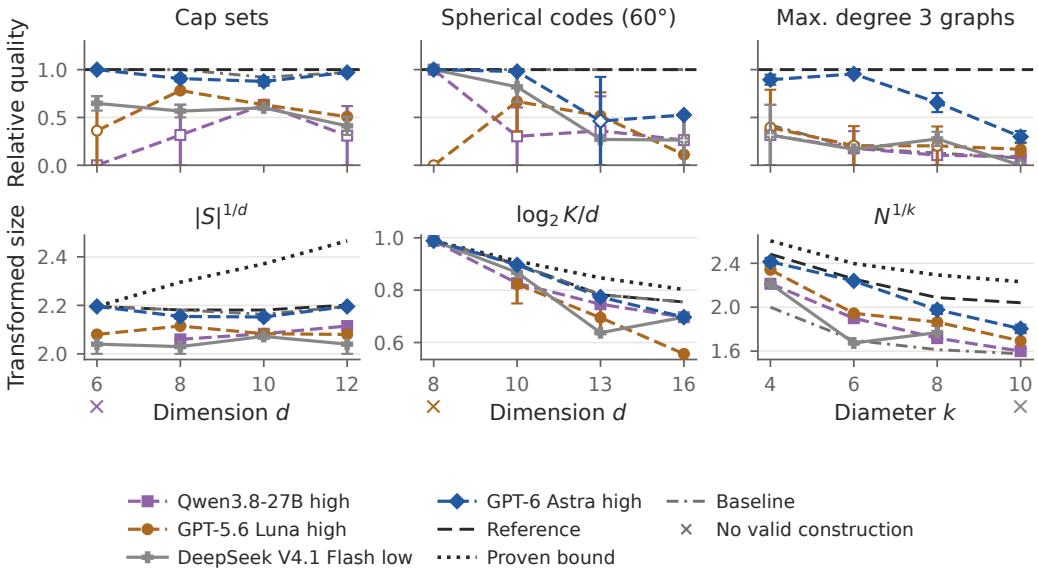


Figure 4: Size–quality curves with 128k output tokens. Upper panels show mean relative quality over two independent responses per size, including zeroes; whiskers show their range and hollow markers indicate an invalid response. Grey dash-dot lines show baseline/reference ratios. Lower panels rescale valid construction sizes; whiskers span valid responses and crosses indicate none. Here d is dimension, k graph diameter, $|S|$ cap-set size, K spherical-code size and N graph order; $|S|^{1/d}$ gives the exponential growth base. Appendix C.4 gives the full comparison.

2025). As stronger methods narrow the gap to mathematical bounds, future benchmark versions can add larger instances. Each released version retains its original instances and references so that scores remain comparable.

6 Limitations

Within the available evaluation budget, we test selected models and reasoning-effort levels, using one evaluation per instance and configuration in the main comparison. Repeated evaluations would quantify run-to-run variability, which the instance-bootstrap intervals do not measure. Answer formats and verification budgets also limit the constructions that can be evaluated (Appendix D.5); future benchmark versions can extend certificate formats alongside instance sizes, while preserving earlier releases for comparison.

7 Conclusion

The Endless Exam measures mathematical progress before and beyond published frontiers by combining uncapped scores with parameterised families and compact certificates. At high effort, tools raise the scores of GPT-6 Astra, GPT-5.6 Luna and Claude Opus 5.5 to 143.16, 119.35 and 180.06, with gains beyond construction baselines while none of the 30 published-frontier references is surpassed. Historical Schur constructions show how scores distinguish subsequent breakthroughs, and size–quality curves extend the comparison to larger tasks, providing ways to continue measurement toward superintelligence.

Reproducibility statement

The supplementary material provides anonymised generators, verifiers, reference constructions, model responses, inference settings and analysis scripts to reproduce every score, table and figure. Instances are specified by family, parameters and random seed. Published constructions link to their sources; independent verifiers and regression tests accompany the release.

AI use statement

The author led the study and is responsible for the final content. AI tools assisted with literature search, research design, implementation, mathematical derivations, analysis and writing.

References

R. Julian R. Abel. Existence of five MOLS of orders 18 and 60. *Journal of Combinatorial Designs*, 23(4):135–139, 2015.

Mohammed Abouzaid, Andrew J. Blumberg, Martin Hairer, Joe Kileel, Tamara G. Kolda, Paul D. Nelson, Daniel Spielman, Nikhil Srivastava, Rachel Ward, Shmuel Weinberger, and Lauren Williams. First proof. *arXiv preprint arXiv:2602.05192*, 2026.

Mubashara Akhtar, Anka Reuel, Prajna Soni, Sanchit Ahuja, Pawan Sasanka Ammanamanchi, et al. When AI benchmarks plateau: A systematic study of benchmark saturation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. arXiv:2602.16763.

Mislav Balunović, Jasper Dekoninck, Nikola Jovanović, Ivo Petrov, and Martin Vechev. MathConstruct: Challenging LLM reasoning with constructive proofs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *PMLR*, pp. 2742–2769, 2025.

Felix A. Behrend. On sets of integers which contain no three terms in arithmetical progression. *Proceedings of the National Academy of Sciences*, 32(12):331–332, 1946.

Nils Bengone, Amine Brouk, Max Grinsztajn, Térence Helbert, Bao Lugherini, Arpad Rimmel, and Joanna Tomasik. Shifted S-templates and improved lower bounds for Schur numbers. *arXiv preprint arXiv:2607.15034*, 2026.

Anurag Bishnoi, Jozefien D'haeseleer, Dion Gijswijt, and Aditya Potukuchi. Blocking sets, minimal codes and trifferent codes. *Journal of the London Mathematical Society*, 2024. doi: 10.1112/jlms.12938. URL https://arxiv.org/abs/2301.09457.

Anurag Bishnoi, Bartlomiej Kielak, Benedek Kovacs, Zoltan Lorant Nagy, Gabor Somlai, Mate Vizer, and Zeyu Zheng. The generalized trifference problem. *arXiv preprint arXiv:2505.07706*, 2025. URL https://arxiv.org/abs/2505.07706.

Thomas F. Bloom. Erdős problems. https://www.erdosproblems.com/, 2021. Community database snapshot accessed September 16, 2026.

Boris Bukh and Ting-Wei Chao. Sharp density bounds on the finite field Kakeya problem. *Discrete Analysis*, (26):1–9, 2021. doi: 10.19086/da.30707. URL https://arxiv.org/abs/2108.00074.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

François Chollet, Mike Knoop, Gregory Kamradt, Bryan Landers, and Henry Pinkard. ARC-AGI-2: A new challenge for frontier AI reasoning systems. *arXiv preprint arXiv:2505.11831*, 2025.

Henry Cohn. Tables of spherical codes, 2026. URL https://spherical-codes.org/. Accessed 23 September 2026.

Henry Cohn and Noam Elkies. New upper bounds on sphere packings I. *Annals of Mathematics*, 157:689–714, 2003.

Francesc Comellas. The degree/diameter problem: table of large graphs. https://web.mat.upc.edu/francesc.comellas/delta-d/taula_delta_d.html, 2026.

D. Davis, P. Ivanisvili, Terence Tao, et al. Optimization constants in mathematics. https://github.com/teorth/optimizationproblems, 2026. Database of constants and bounds. Accessed September 16, 2026.

Miroslav Dimitrov. New classes of binary sequences with high merit factor. *arXiv preprint arXiv:2206.12070*, 2022. URL https://arxiv.org/abs/2206.12070.

Yves Edel. Extensions of generalized product caps. *Designs, Codes and Cryptography*, 31:5–14, 2004.

Yves Edel. Tables of caps and codes. https://www.yvesedel.de/, 2026.

Jordan S. Ellenberg and Dion Gijswijt. On large subsets of $\mathbb{F}_q^n$ with no three-term arithmetic progression. *Annals of Mathematics*, 185(1):339–343, 2017.

Christian Elsholtz and Péter Pál Pach. Caps and progression-free sets in $\mathbb{Z}_m^n$. *Designs, Codes and Cryptography*, 88:2133–2170, 2020. arXiv:1903.08266.

Epoch AI. FrontierMath: Tiers 1–4, version 2. https://epoch.ai/benchmarks/frontiermath-tier-4-v2, 2026a. Dataset description. Accessed September 16, 2026.

Epoch AI. FrontierMath: Open problems. https://epoch.ai/frontiermath/open-problems, 2026b. Benchmark description and FAQ. Accessed September 16, 2026.

Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610:47–53, 2022.

Moritz Firsching, Paul Lezeau, Salvatore Mercuri, Miklós Z. Horváth, Yaël Dillies, Calle Sönne, Eric Wieser, Fred Zhang, Thomas Hubert, Blaise Agüera y Arcas, and Pushmeet Kohli. Formal conjectures: An open and evolving benchmark for verified discovery in mathematics, 2026. URL https://arxiv.org/abs/2605.13171.

Harold Fredricksen and Melvin M. Sweet. Symmetric sum-free partitions and lower bounds for Schur numbers. *The Electronic Journal of Combinatorics*, 7(1):R32, 2000. doi: 10.37236/1510.

Mikhail Ganzhinov. Highly symmetric lines. *Linear Algebra and its Applications*, 722:12–37, 2025. doi: 10.1016/j.laa.2025.05.002.

Bogdan Georgiev, Javier Gómez-Serrano, Terence Tao, and Adam Zsolt Wagner. Mathematical exploration and discovery at scale. *arXiv preprint arXiv:2511.02864*, 2025. URL https://arxiv.org/abs/2511.02864.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, et al. FrontierMath: A benchmark for evaluating advanced mathematical reasoning in AI. *arXiv preprint arXiv:2411.04872*, 2024.

Daniel M. Gordon. La Jolla coverings repository. Zenodo. https://doi.org/10.5281/zenodo.19735294, 2026.

Daniel M. Gordon, Greg Kuperberg, and Oren Patashnik. New constructions for covering designs. *Journal of Combinatorial Designs*, 3(4):269–284, 1995. arXiv:math/9502238.

Greg Kamradt. OpenAI's GPT-6 Astra on ARC-AGI-3. https://arcprize.org/blog/astra, 2026. ARC Prize. Published September 3; accessed September 15, 2026.

Iskandar Karapetyan and Karen Karapetyan. Complete caps in $AG(n, 3)$. In *Proceedings of the International Conference on Computer Science and Information Technologies*, pp. 135–136, 2023. doi: 10.51408/csit2023_30. URL https://csit.am/2023/proceedings/DMCA/DMCA_6.pdf.

Clement W. H. Lam, Larry Thiel, and Stan Swiercz. The non-existence of finite projective planes of order 10. *Canadian Journal of Mathematics*, 41(6):1117–1123, 1989.

John Leech and Neil J. A. Sloane. Sphere packings and error-correcting codes. *Canadian Journal of Mathematics*, 23:718–745, 1971.

Nando Leijenhorst and David de Laat. Solving clustered low-rank semidefinite programs arising from polynomial optimization. *Mathematical Programming Computation*, 16(3):503–534, 2024. doi: 10.1007/s12532-024-00264-w.

Laszlo Lovasz. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25 (1):1–7, 1979. doi: 10.1109/TIT.1979.1055985.

Thang Luong, Dawsen Hwang, Hoang H. Nguyen, Golnaz Ghiasi, Yuri Chervonyi, Insuk Seo, Junsu Kim, et al. Towards robust mathematical reasoning. *arXiv preprint arXiv:2511.01846*, 2025.

Jicheng Ma, Guohua Wang, Xinhua Feng, Yiming Liu, Zhichao Hu, and Yuhong Liu. EternalMath: A living benchmark of frontier mathematics that evolves with human discovery, 2026. URL https://arxiv.org/abs/2601.01400.

Alice Miller, R. Julian R. Abel, Ivaylo Valkov, and Douglas Fraser. Implementing the MOLS table for n up to 500. *Symmetry*, 16(12):1678, 2024.

Mirka Miller and Jozef Širáň. Moore graphs and beyond: A survey of the degree/diameter problem. *The Electronic Journal of Combinatorics*, DS14, 2013.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, et al. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Tom Packebusch and Stephan Mertens. Low autocorrelation binary sequences. *Journal of Physics A: Mathematical and Theoretical*, 49(16):165001, 2016.

Viresh Pati, Zhengyu Li, Piyush Jha, Rahul Garg, Yatharth Sejpal, and Vijay Ganesh. MathConstraint: Automated generation of verified combinatorial reasoning instances for LLMs. *arXiv preprint arXiv:2605.08498*, 2026.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Sven C. Polak and Alexander Schrijver. New lower bound on the Shannon capacity of C_7 from circular graphs. *Information Processing Letters*, 143:37–40, 2019. URL https://arxiv.org/abs/1808.07438.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024.

Fred Rowley. An improved lower bound for $S(7)$ and some interesting templates. *arXiv preprint arXiv:2107.03560*, 2021. URL https://arxiv.org/abs/2107.03560.

Imre Z. Ruzsa. Solving a linear equation in a set of integers I. *Acta Arithmetica*, 65(3):259–282, 1993.

Shubhangi Saraf and Madhu Sudan. An improved lower bound on the size of Kakeya sets over finite fields. *Analysis & PDE*, 1(3):375–379, 2008.

Johannes Schmitt, Gergely Bérczi, Jasper Dekoninck, Jeremy Feusi, Tim Gehrunger, et al. IMProofBench: Benchmarking AI on research-level mathematical proof generation. *arXiv preprint arXiv:2509.26076*, 2025. Version 2, July 2026.

T. Stead. The Heilbronn problem: Configuration archive, 2026. URL https://math.tejstead.com/heilbronn/. Accessed 23 September 2026.

Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969.

Ravi Tandon. Strengthening recursive constructions for zero-error Shannon capacity. *arXiv preprint arXiv:2608.30273*, 2026. URL https://arxiv.org/abs/2608.30273.

D. T. Todorov. Four mutually orthogonal Latin squares of order 14. *Journal of Combinatorial Designs*, 20(8):363–367, 2012.

Nina Versluis. On the cap set problem. *Bachelor thesis, Delft University of Technology*, 2017.

Erik Y. Wang, Sumeet R. Motwani, James V. Roggeveen, Eliot Hodges, Dulhan Jayalath, Charles London, Kalyan Ramakrishnan, Jakob Foerster, Cheng Zhang, Flaviu Cipcigan, Philip Torr, and Alessandro Abate. Measuring progress in reasoning toward mathematical discovery with automatic verification. *arXiv preprint arXiv:2603.15617*, 2026.

V. A. Zinoviev and T. Ericson. New lower bounds for contact numbers in small dimensions. *Problems of Information Transmission*, 35:287–294, 1999.

Appendix

Reading guide

A	Task catalogue	Fourteen families, examples and parameters controlling problem size	14
B	Representative constructions	Four examples from object to score	22
C	Detailed results	Model comparisons, size–quality curves and sensitivity	26
D	Verification	Compact formats and algebraic certificates	40
E	References and bounds	Literature sources, construction baselines and the LABS target	42
F	Instance parameters	Main and smaller-instance parameter settings	46
G	Evaluation settings	Model settings and an example prompt	49
H	Code and web access	Tool-assisted results and token usage	51

A Task catalogue

Each family page gives the objective, a checked small example, parameters controlling problem size and one evaluated instance with its reference values. Small diagrams illustrate the mathematical property. In the tables, the baseline is the better of an implemented mathematical construction and the ten-second search result. It can be weaker than a published frontier, which remains the scoring reference. The final column gives this baseline's relative quality, rather than a model score. Gap closed uses the scoring reference as its starting point, as defined in Section 4.2.

family	construction	objective	class	reference; bound or target
cap set	$S \subseteq \mathbb{F}_3^d$; no three collinear points (cap set)	$\lvert S\rvert$	R	published; proven bound
spherical codes	vectors with exact coordinates; pairwise angle $\geq \theta$; 60° or variable	count	R	mixed; proven bound
matrix multiplication	bilinear scheme for $(n \times m)$ by $(m \times p)$ matrices; computes the product exactly	rank r (min)	S	published; proven bound
degree–diameter	graph with max degree d; diameter $\leq k$	vertices	S	published; proven bound
covering	k-subsets (blocks) of $[v]$; every t-subset lies in a block	blocks (min)	S	published; proven bound
Schur	k-colouring of $\{1..N\}$; no monochromatic $x, y, x+y$	N	R	published; proven bound
$\mathbb{F}_q^n$ AP-free	$S \subseteq \mathbb{F}_q^n$, $q \in \{5, 7\}$; no 3-term progression	$\lvert S\rvert$	R	mixed; proven bound
MOLS	Latin squares of order n, n not a prime power; pairwise orthogonal	number of squares	R	published; proven bound
LABS	± 1 sequence of length N; well-formed sequence	merit factor	S	construction; conjectured target
Heilbronn	n points in a square or triangle; distinct points in the domain	min triangle area	S	construction; trivial bound
corners	$S \subseteq [n]^2$; no corner	$\lvert S\rvert$	R	construction; trivial bound
linear-equation-free	$S \subseteq \{1..n\}$; no non-trivial solution of the specified linear equation	$\lvert S\rvert$	R	construction; trivial bound
Shannon	code in $\{0, \ldots, q-1\}^d$; every pair separates on the cycle	$\lvert C\rvert$	R	construction; proven bound
trifference	ternary code of length n; every triple has at least m coordinates with three distinct symbols	$\lvert C\rvert$	U	construction; proven bound

Table 3: The fourteen families. R denotes search-resistant, S families that can benefit from search and U unclassified search sensitivity. These labels concern the reference search procedures studied here. Spherical codes and Heilbronn each contain two variants. Reference types may depend on the parameters.

A.1 Cap sets

A large set must avoid every three-point arithmetic progression. (Edel, 2004; Karapetyan & Karapetyan, 2023)



$d = 2$: four points

Maximise $|S|$ for $S \subseteq \mathbb{F}_3^d$ containing no three distinct points x, y, z with $x + y + z = 0$.

Parameters and scaling. Increasing d enlarges the ambient space to 3^d points. Products lift finite caps to higher dimensions.

Evaluated example: $d = 10$.

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
2240	1140	2432	5619	0.921

Reference: published frontier; proven bound.

A.2 Corner-free sets

A dense grid set must avoid a prescribed three-point pattern. (Behrend, 1946)



$n = 5$: nine points

Maximise $|S|$ for $S \subseteq \{0, \ldots, n-1\}^2$ containing no triple $(x, y), (x + t, y), (x, y + t)$ with $t \neq 0$, including negative t.

Parameters and scaling. Increasing n enlarges the grid. Progression-free difference sets supply constructions at many sizes.

Evaluated example: $n = 138$.

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
3120	2504	3120	19044	1.000

Reference: construction baseline; trivial bound.

A.3 Schur colourings

No solution to $x + y = z$ may lie entirely within one colour class. (Rowley, 2021; Bengone et al., 2026)



Maximise N such that $\{1, \ldots, N\}$ can be coloured with k colours without a monochromatic solution of $x + y = z$. Solutions with $x = y$ are also forbidden.

$k = 2,\ N = 4$

Parameters and scaling. Increasing the number of colours k gives new extremal targets. Recursive templates extend smaller colourings.

Evaluated example: $k = 7$.

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
1093	1093	1696	13699	0.644

Reference: published frontier; proven bound.

A.4 Progression-free sets over finite fields

The ambient field changes which additive patterns a set must avoid. (Edel, 2026; Ellenberg & Gijswijt, 2017)



Maximise $|S|$ for $S \subseteq \mathbb{F}_q^n$ containing no three distinct points satisfying $x + z = 2y$.

$q = 5,\ n = 2$

Parameters and scaling. Varying q or n changes the task. Products and algebraic constructions give candidates across dimensions.

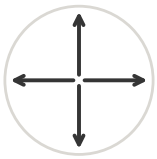
Evaluated example: $q = 5,\ n = 6$.

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
360	360	649	7497	0.555

Reference: published frontier; proven bound.

A.5 Spherical codes

Directions must remain separated while as many vectors as possible are packed. (Zinoviev & Ericson, 1999; Ganzhinov, 2025; Cohn & Elkies, 2003)



$d = 2$: four directions

Fixed angle. Maximise the number of nonzero vectors whose pairwise angles are at least $60°$. After normalisation these give a kissing configuration. Coordinates are integers in $[-1000, 1000]$; at $d = 13$, we also allow $a + b\sqrt{3}$ with integer $|a|, |b| \leq 1000$.

Variable angle. Maximise the number of nonzero vectors in $\mathbb{Z}^d$ satisfying $u \cdot v \leq c\|u\|\|v\|$ for every distinct pair. Coordinates lie in $[-1000, 1000]$, and the prescribed rational value $c = \cos\theta$ is checked exactly.

Parameters and scaling. Dimension d and the angle threshold control the packing problem. Lattices, orbits and signed supports supply constructions.

Evaluated example: $d = 14$.

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
1932	876	1932	3174	1.000

Reference: published frontier; proven bound.

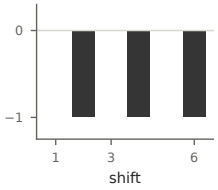
Evaluated example: $d = 12,\ c = 24/100$.

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
28	24	28	1431	1.000

Reference: construction baseline; proven bound.

A.6 Low-autocorrelation binary sequences

A binary sequence should have low autocorrelation at nonzero shifts. (Packebusch & Mertens, 2016)



$N = 7$: autocorrelation

Choose $s \in \{-1, 1\}^N$ to maximise the merit factor $N^2/(2\sum_{k=1}^{N-1} C_k^2)$, where $C_k = \sum_{i=1}^{N-k} s_i s_{i+k}$ is the aperiodic autocorrelation.

Parameters and scaling. Increasing N tests whether a construction retains a high merit factor at longer lengths.

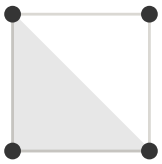
Evaluated example: $N = 280$.

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
5.57769	4.4708	5.57769	12.32	1.000

Reference: construction baseline; conjectured target.

A.7 Heilbronn triangles

Place the points so that the smallest triangle area is as large as possible. (definitions and references in Appendix E)



$n = 4$: minimum area $1/2$

Square. Place n distinct points in $[0, 1]^2$ to maximise the smallest area of a triangle determined by three points. Coordinates are multiples of 10^{-4}.

Triangle. Place n distinct integer-grid points in a specified triangle T to maximise the smallest triangle area, divided by the area of T.

Parameters and scaling. Increasing n introduces more triangle constraints. The square and triangular domains are evaluated separately.

Evaluated example: $n = 46$.

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
0.00047375	0.00047375	0.00047375	0.0227273	1.000

Reference: construction baseline; trivial bound.

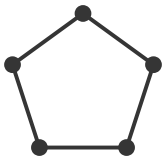
Evaluated example: $n = 32$, triangular domain (Table 21).

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
0.0014885	0.0014885	0.0014885	0.0333333	1.000

Reference: construction baseline; trivial bound.

A.8 Degree–diameter graphs

A sparse graph must keep all vertices close to one another. (Miller & Širáň, 2013; Comellas, 2026)



$d = 2,\ k = 2,\ N = 5$

Maximise the number of vertices of a simple connected undirected graph with maximum degree at most d and diameter at most k.

Parameters and scaling. Varying degree d and diameter k generates new graph problems. Larger diameters test the growth of a construction.

Evaluated example: $d = 4,\ k = 5$.

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
61	61	364	485	0.168

Reference: published frontier; proven bound.

A.9 Linear-equation-free sets

A large integer set must avoid a specified linear relation. (Behrend, 1946; Ruzsa, 1993)



Maximise $|S|$ for $S \subseteq \{1, \ldots, n\}$ such that $ax + by = (a + b)z$ has no solution in S except $x = y = z$.

$a = b = 1,\ n = 9$

Parameters and scaling. The coefficients a, b select the relation; n sets the interval size. Digit constructions can extend to larger intervals.

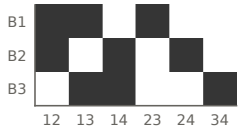
Evaluated example: $a = 3,\ b = 4,\ n = 19044$.

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
630	630	630	19044	1.000

Reference: construction baseline; trivial bound.

A.10 Covering designs

A small collection of blocks must cover every required subset. (Gordon et al., 1995; Gordon, 2026)



Three blocks cover six pairs

Minimise the number of k-element blocks drawn from a v-element set such that every t-element subset is contained in at least one block.

Parameters and scaling. The parameters (v, k, t) control the universe, block size and subsets to cover. Larger settings test covering efficiency.

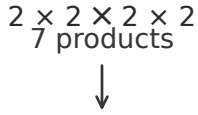
Evaluated example: $v = 18,\ k = 7,\ t = 4$.

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
179	179	126	111	0.704

Reference: published frontier; proven bound.

A.11 Matrix multiplication

A bilinear construction reduces the number of scalar multiplications. (Strassen, 1969; Fawzi et al., 2022)



Strassen: seven products

Minimise the number r of scalar multiplications in a bilinear algorithm for $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{m\times p}$. The identity $AB = \sum_{j=1}^{r}(u_j \cdot A)(v_j \cdot B)W_j$ must hold for all inputs, with each dot product taken over matrix entries. Coefficients are integers in $[-4, 4]$.

Parameters and scaling. Changing the matrix dimensions (n, m, p) defines new multiplication problems. Block composition extends schemes to larger products.

Evaluated example: $n = 4,\ m = 4,\ p = 5$.

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
72	72	61	20	0.847

Reference: published frontier; proven bound.

A.12 Mutually orthogonal Latin squares

When any two squares are superimposed, every ordered pair of symbols must occur exactly once. (Miller et al., 2024; Abel, 2015)



Two squares of order three

Find as many mutually orthogonal Latin squares of order n as possible. Each symbol appears once in every row and column; superimposing any two squares must give all n^2 ordered symbol pairs exactly once.

Parameters and scaling. Changing the order n changes the design problem. Algebraic constructions and combinations of smaller designs supply candidates.

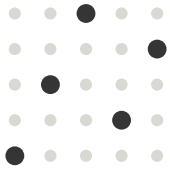
Evaluated example: $n = 14$.

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
4	1	4	12	1.000

Reference: published frontier; proven bound.

A.13 Shannon codes

In the cycle model, adjacent symbols can be confused; every pair of codewords must have a coordinate that distinguishes them. (Lovasz, 1979; Polak & Schrijver, 2019)



C_5^2: five codewords

Maximise the size of a code $C \subseteq \mathbb{Z}_q^d$ such that every two distinct words differ by cyclic distance at least two in some coordinate. Equivalently, C is an independent set in the d-fold strong power of the cycle C_q.

Parameters and scaling. The cycle size q and codeword length d define the task. Finite factors give short certificates for very large products.

Evaluated example: $q = 7,\ d = 24$.

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
1.9411×10^{12}	1.9411×10^{12}	1.9411×10^{12}	3.16219×10^{12}	1.000

Reference: construction baseline; proven bound.

A.14 Trifference codes

Every triple of distinct codewords must display all three symbols in at least m coordinates. (Bishnoi et al., 2024; 2025)



$n = 2,\ m = 2$: three words

Maximise the size of a ternary code $C \subseteq \{0, 1, 2\}^n$ such that, for every three distinct words, at least m coordinates contain all three symbols.

Parameters and scaling. Length n and separation multiplicity m define the task. Linear and concatenated codes admit compact certificates.

Evaluated example: $n = 64,\ m = 1$.

Baseline	10 s search	Reference	Bound or target	Baseline's relative quality
19683	19683	19683	3.72281×10^{11}	1.000

Reference: construction baseline; proven bound.

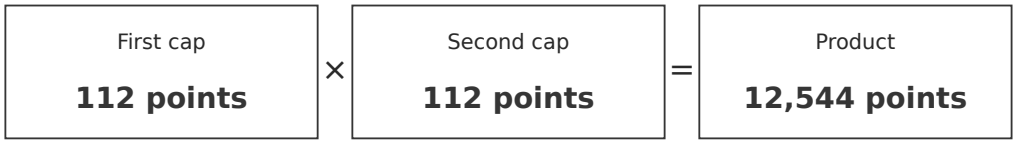
B Representative constructions

B.1 Cap sets: a small ingredient at twice the dimension

Model: GPT-6 Astra, high effort, tool-free. **Instance:** $d = 12$, problem-size experiment.

Task

Construct a subset of $\mathbb{F}_3^{12}$ with no three distinct points summing to zero. The objective is the number of points. The published frontier contains 12,928 points.



6 dimensions + 6 dimensions = 12 dimensions

Construction and verification

The answer takes the Cartesian product of two copies of a 112-point cap $C \subseteq \mathbb{F}_3^6$:

$$S = C \times C, \qquad |S| = 112^2 = 12544.$$

Each finite factor is checked for forbidden triples. In characteristic three, a zero-sum triple in a cap has all three entries equal: three distinct entries are forbidden, and two equal entries force the third to be equal. Applying this fact in each factor shows that any zero-sum triple in $C \times C$ consists of one repeated point.

Both answers at this problem size use this product. The construction makes the dependence on dimension explicit: repeating a factor adds dimensions and multiplies cardinalities.

Measured quality

Model construction	Published frontier	Relative quality
12,544 points	12,928 points	$12544/12928 = \mathbf{0.970}$

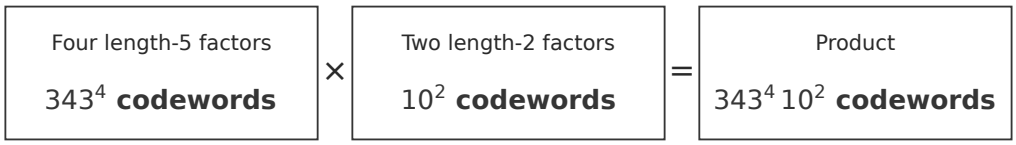
The product describes a large, valid cap set close to the published frontier using only its smaller factors. The size–quality curves in Appendix C.4 compare how this construction and graph constructions behave as their parameters grow.

B.2 Shannon codes: better factors produce a larger code

Model: Qwen3.8-27B, high effort, tool-free. **Instance:** length 24 over seven symbols (C_7^{24}).

Task

Construct codewords of length 24 over $\mathbb{Z}_7$. Every pair must differ by cyclic distance at least two in some coordinate. The objective is the number of codewords.



Codeword length: 4 × 5 + 2 × 2 = 24; each factor is checked separately

Construction and verification

The submitted factors contain 343 words of length five and ten words of length two. Their multiplicities are four and two:

$$4 \cdot 5 + 2 \cdot 2 = 24, \qquad M = 343^4\,10^2 = 1{,}384{,}128{,}720{,}100.$$

The evaluator verifies the finite factors and computes the exact product size. Concatenating words from valid Shannon-code factors preserves the separation condition (Polak & Schrijver, 2019; Tandon, 2026): two distinct product words differ in at least one factor, which supplies the coordinate required for separation.
GPT-6 Astra at medium effort and DeepSeek V4.1 Flash at low effort instead return 10^{12} codewords on this instance. Qwen3.8-27B's choice of factors produces a larger valid code at the same length.

Measured quality

Construction baseline: 1,941,100,559,147 words.
Qwen3.8-27B relative quality: **0.713**.
Size advantage over the two 10^{12}-word answers: **38.4%**.

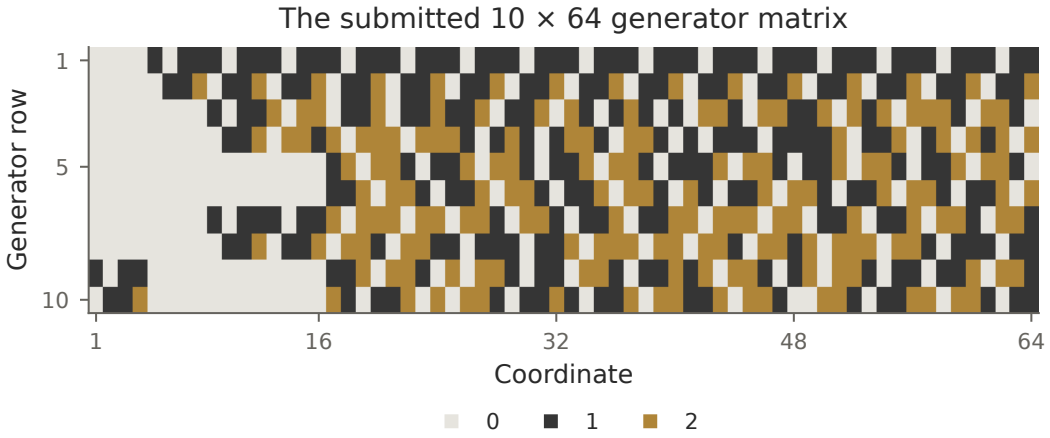
The comparison measures the quality of the finite ingredients and their composition. Even for codes with more than a trillion words, verification gives their exact sizes and makes the improvement measurable.

B.3 Trifference: ten rows describe 59,049 codewords

Model: GPT-6 Astra, high effort, tool-free. **Instance:** $n = 64$, $m = 1$.

Task

Construct a ternary code in which every three distinct words have at least one coordinate containing all three symbols. The construction baseline has 19,683 words.



Construction and verification

The answer is a 10×64 ternary generator matrix G. Its rows span a linear code

$$C = \{uG : u \in \mathbb{F}_3^{10}\}, \qquad \text{rank}(G) = 10, \qquad |C| = 3^{10} = 59049.$$

The coloured matrix above is the submitted generator: grey, black and gold denote 0, 1 and 2. The hyperplane test in Appendix D.4 checks the separation condition without listing every triple of codewords. Both verifiers accepted the construction.

Measured quality

Model construction	Construction baseline	Relative quality
59,049 words	19,683 words	$59049/19683 = \mathbf{3}$

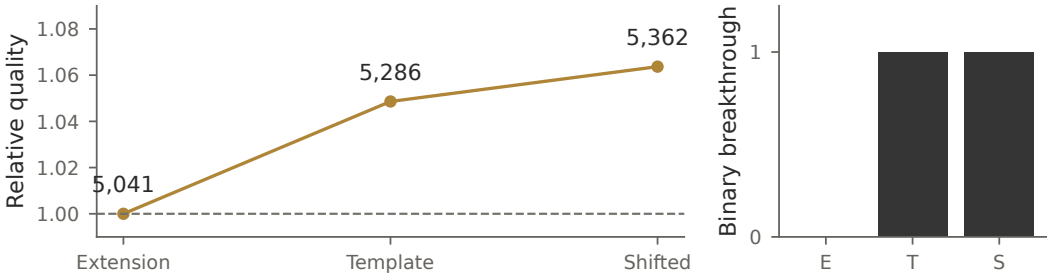
A short certificate describes a code three times as large as the construction baseline, giving relative quality 3. The other three evaluated lengths test whether that quality extends further.

B.4 Schur colourings: successive advances on one scale

Source: published constructions. **Instance:** eight colours.

Task

Colour $\{1, \ldots, N\}$ with eight colours so that no colour class contains a solution of $x + y = z$, including $x = y$. Larger N gives a stronger construction.



Published constructions

Recursive extension of the 1,680-element seven-colour construction reported by Fredricksen & Sweet (2000) gives the historical reference

$$N_0 = 3 \cdot 1680 + 1 = 5041.$$

The template in Rowley (2021) gives $33 \cdot 160 + 6 = 5286$. Shifted templates give $10 \cdot 536 + 2 = 5362$ (Bengone et al., 2026). We compare the published constructions against the same historical reference N_0.

Measured progress

Published construction	N	$N/5041$	$\mathbf{1}\{N > 5041\}$
Recursive extension (Fredricksen & Sweet, 2000)	5,041	1.000	0
Rowley template (Rowley, 2021)	5,286	1.049	1
Shifted template (Bengone et al., 2026)	5,362	1.064	1

Table 4: Selected published Schur constructions with eight colours, scored against the historical reference $N_0 = 5041$. The last column records whether the construction exceeds this reference. Citations identify the accounts used for each construction.

Both advances count as one binary success, while relative quality preserves their different magnitudes. The current benchmark uses 5,362 as the eight-colour reference and extends the challenge by increasing the number of colours under the same construction rule.

C Detailed experimental results

C.1 Quality across the tool-free model comparison

Although every tool-free configuration has a zero breakthrough rate on the 30 published-frontier instances, their mean relative quality spans 0.07–0.76. The two panels place this comparison alongside quality on the 39 construction-baseline instances; Appendix H extends the evaluation to tool-assisted responses.

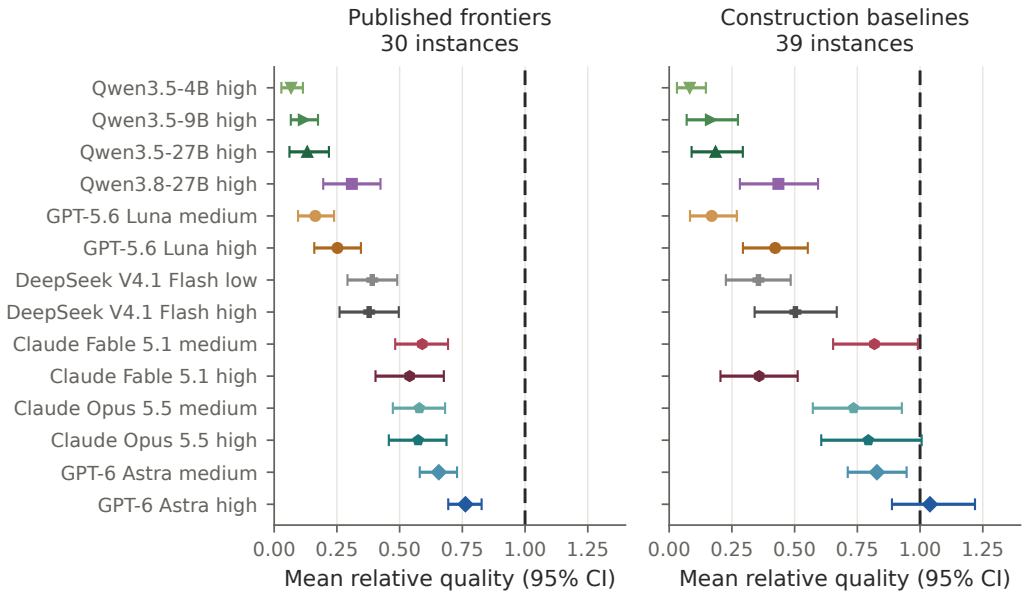


Figure 5: Fourteen-family construction quality for all fourteen tool-free configurations. Whiskers show 95% confidence intervals obtained by resampling instances within each reference group: 30 with published frontiers and 39 with construction baselines.

The overall score in Table 2 combines all 69 instances using the definition in Section 4.

C.2 Performance across parameter ranges

The non-code experiments use three parameter ranges: A1 contains small familiar instances, A2 uses larger settings, and A3 supplies the main evaluation. Family coverage varies across these groups. Appendix C.4 instead varies size within each family, allowing construction quality to be followed as the task grows.

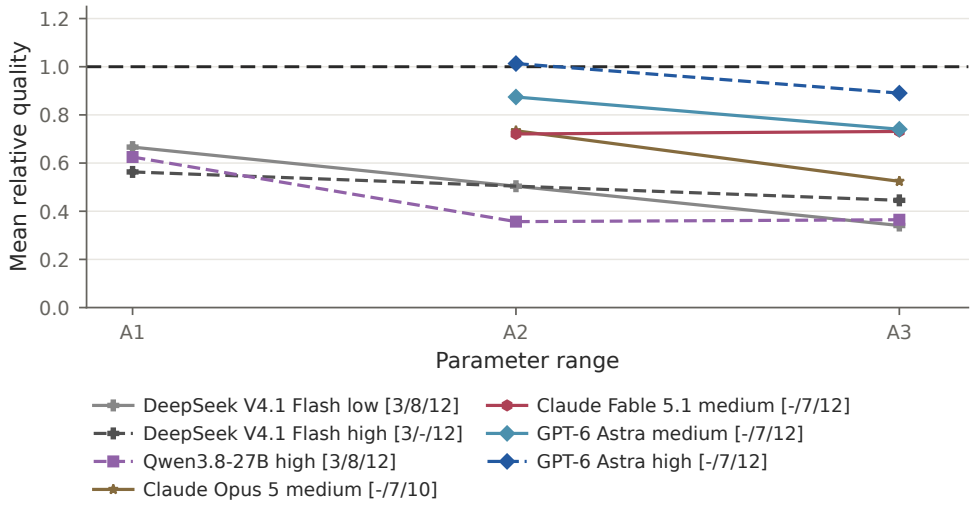


Figure 6: Performance across parameter ranges. Legend brackets give the numbers of evaluated families in A1, A2 and A3, respectively; a dash denotes an unmeasured group. The family sets differ across groups. Claude Opus 5 is an earlier checkpoint included in these auxiliary comparisons.

C.3 Model size and reasoning effort

Qwen3.5's mean quality and validity increase across the three checkpoint sizes. The GPT-6 Astra experiment extends the reasoning-effort comparison to xhigh on shared smaller instances.

On the main 69 instances, Astra medium and high return 68 and 69 valid constructions. To include xhigh effort, we also compare Astra on 25 shared smaller instances. Its mean relative quality is 0.87 at medium, 1.01 at high and 1.06 at xhigh.

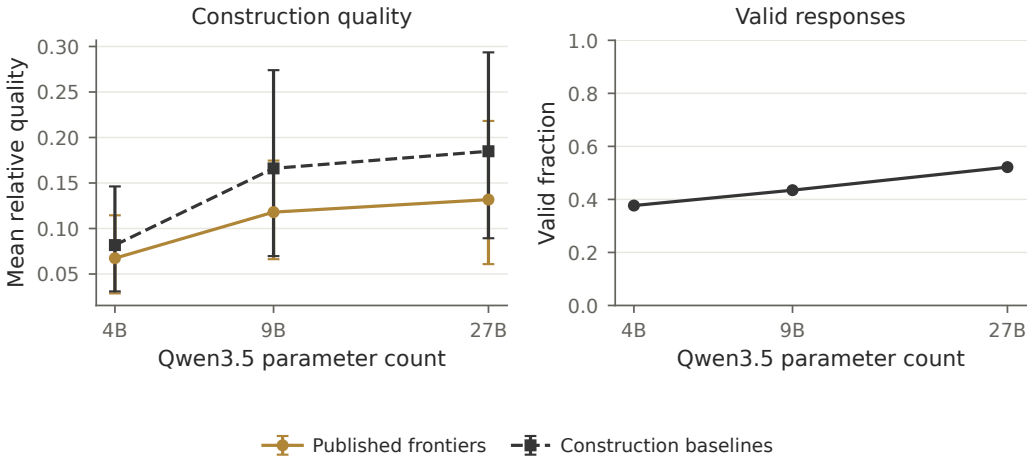


Figure 7: Model-size comparison within Qwen3.5: 4B, 9B and 27B at high effort on the 69 instances. Left: mean relative quality with bootstrap CIs. Right: valid-response fraction. Model size varies while the problems remain fixed.

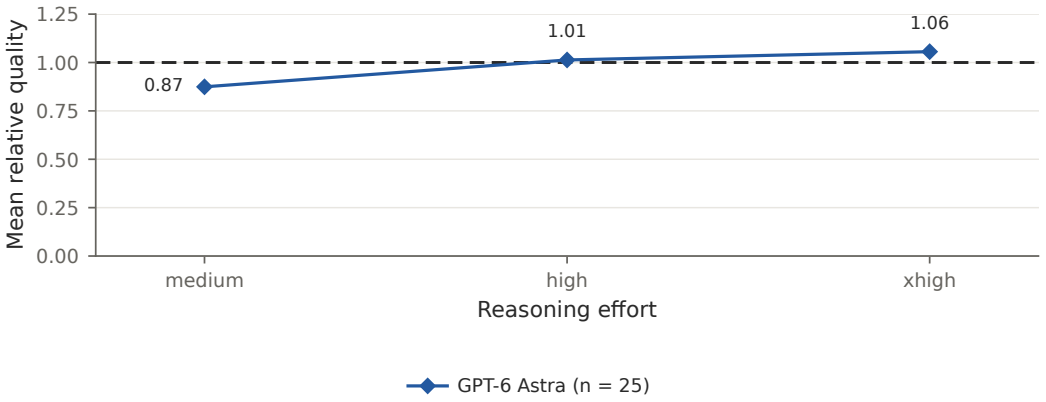


Figure 8: Mean relative quality for GPT-6 Astra at medium, high and xhigh effort on the same 25 smaller instances (A2). Multiple responses are averaged within each instance. All responses are valid at each effort level.

C.4 Size–quality curves: cap sets, spherical codes and graphs

Cap-set products allow GPT-6 Astra to retain quality at larger dimensions. Its graphs with maximum degree three remain valid while falling behind the reference as diameter increases. The lower panels apply the displayed transformations to construction sizes, showing how the model answers and references scale.

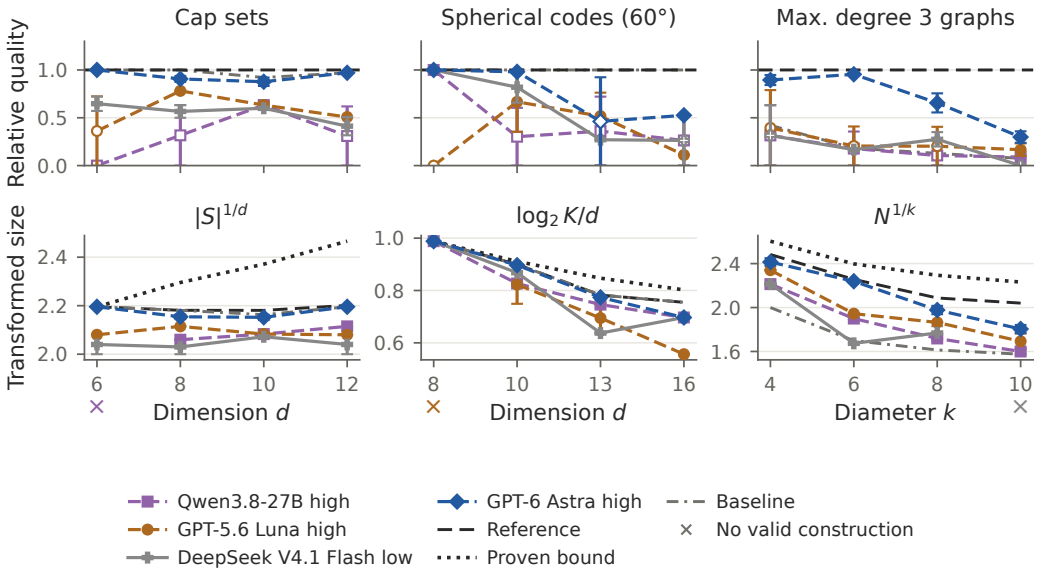


Figure 9: Size–quality curves for cap sets, spherical codes and degree–diameter graphs. Upper panels average two independent responses per size, including zero scores; whiskers show their range and hollow markers indicate at least one invalid response. Lower panels use valid responses only; crosses below the axes mark sizes with no valid construction, using the model's colour. Transformations and line conventions follow Figure 4.

C.5 Size–quality curves: colourings, coverings and integer sets

These curves vary the number of Schur colours, the size of the set covered by a design, and the integer interval from which a progression-free subset is chosen. The integer task supplements the five benchmark families in this experiment. The lower panels show the transformed construction sizes specified above each panel.

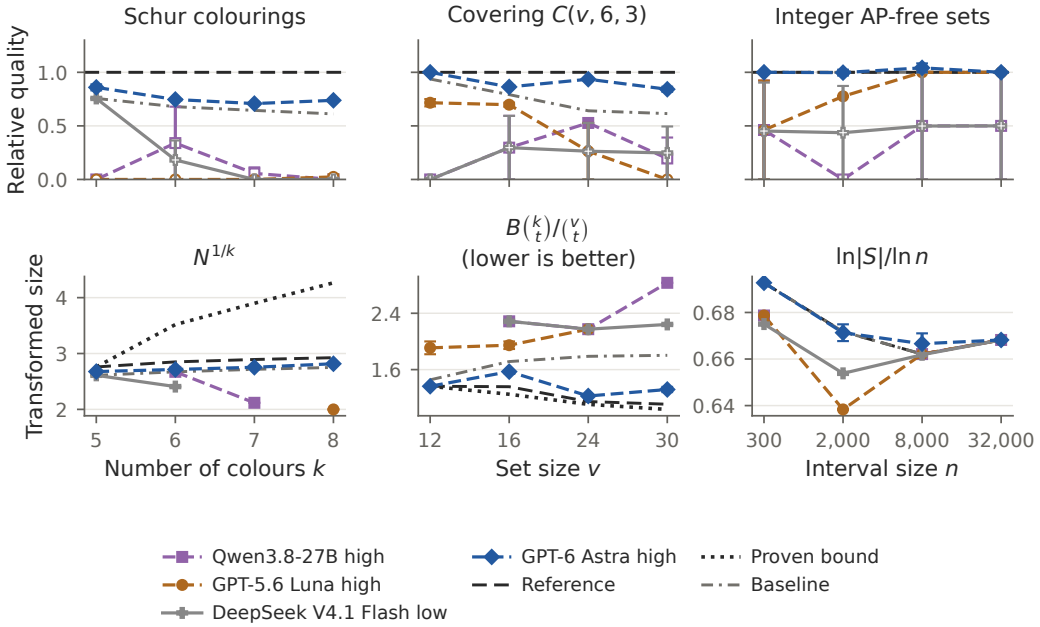


Figure 10: Size–quality curves for Schur colourings, covering designs and integer progression-free sets. Each size has two independent responses per configuration. Upper panels include zero scores; lower panels use valid answers. For Schur, N is the largest coloured integer and k the number of colours. For coverings, B blocks of size k cover every t-subset of a v-element set. For integer sets, $|S|$ elements are selected from $\{1, \ldots, n\}$. Whiskers show the range of the responses included in each panel.

C.6 A changing reference: graphs with maximum degree four

GPT-6 Astra's relative quality is 0.53 at diameter 5, 0.98 at diameter 6 and 0.15 at diameter 7. Between diameters 5 and 7, its graph size changes from 192 to 195 vertices, while the reference grows more quickly.

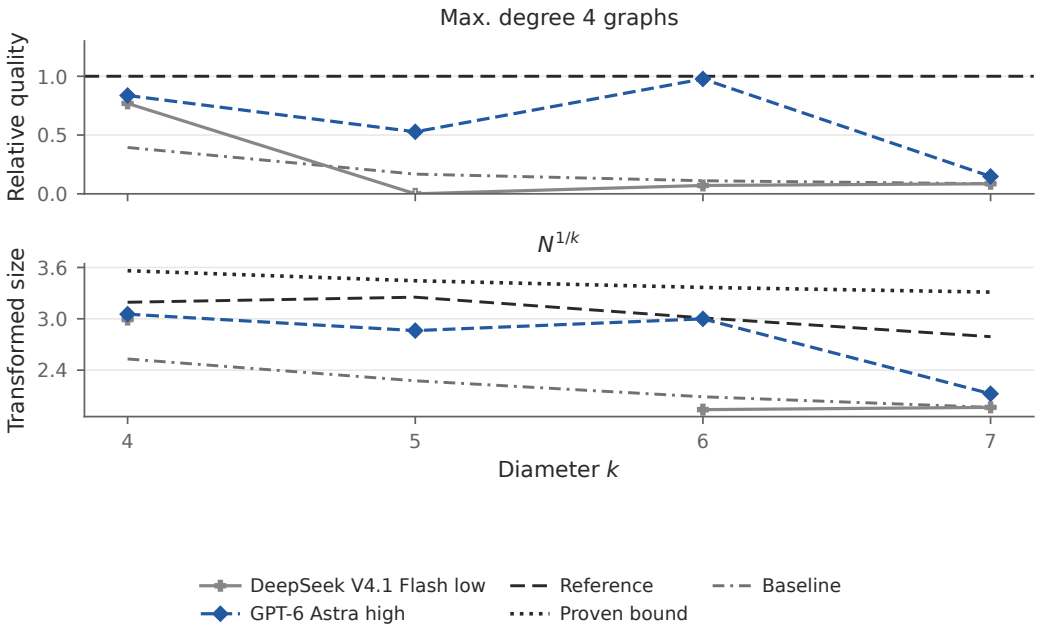


Figure 11: Graphs with maximum degree four at diameters 4–7. The lower panel plots $N^{1/k}$, where N is the number of vertices and k the allowed diameter. Each point represents one response from GPT-6 Astra at high effort or DeepSeek V4.1 Flash at low effort.

C.7 Problem-size experiments: instances and references

Table 5: Problem-size experiments: instances and references. An asterisk marks a construction baseline.

Family	Parameters	Reference	Baseline
cap set	d=6	112	112
cap set	d=8	512	512
cap set	d=10	2432	2240
cap set	d=12	12928	12544
spherical codes (60°)	d=8	240	240
spherical codes (60°)	d=10	510	510
spherical codes (60°)	d=13	1154	1154
spherical codes (60°)	d=16	4320	4320
max. degree 3 graphs	d=3, k=4	38	16
max. degree 3 graphs	d=3, k=6	132	24
max. degree 3 graphs	d=3, k=8	360	46
max. degree 3 graphs	d=3, k=10	1250	94
Schur	k=5	160	121
Schur	k=6	536	364
Schur	k=7	1696	1093
Schur	k=8	5362	3280
covering	v=12, k=6, t=3	15	16
covering	v=16, k=6, t=3	38	48
covering	v=24, k=6, t=3	116	181
covering	v=30, k=6, t=3	225	366
max. degree 4 graphs	d=4, k=4	104	41
max. degree 4 graphs	d=4, k=5	364	61
max. degree 4 graphs	d=4, k=6	745	83
max. degree 4 graphs	d=4, k=7	1320	113
integer AP-free sets	k=3, n=300	52*	52
integer AP-free sets	k=3, n=2000	165*	165
integer AP-free sets	k=3, n=8000	384*	384
integer AP-free sets	k=3, n=32000	1024*	1024

C.8 Problem-size experiments: relative quality

Table 6: Mean relative quality at each problem size. Parentheses give valid responses out of the total; means include zero scores.

Family	Size	Qwen3.8-27B high	GPT-5.6 Luna high	DeepSeek V4.1 Flash low	GPT-6 Astra high
cap set	$d = 6$	0.00 (0/2)	0.36 (1/2)	0.65 (2/2)	1.00 (2/2)
cap set	$d = 8$	0.32 (1/2)	0.78 (2/2)	0.57 (2/2)	0.91 (2/2)
cap set	$d = 10$	0.63 (2/2)	0.63 (2/2)	0.60 (2/2)	0.88 (2/2)
cap set	$d = 12$	0.31 (1/2)	0.51 (2/2)	0.41 (2/2)	0.97 (2/2)
spherical codes (60°)	$d = 8$	1.00 (2/2)	0.00 (0/2)	1.00 (2/2)	1.00 (2/2)
spherical codes (60°)	$d = 10$	0.30 (1/2)	0.67 (2/2)	0.82 (2/2)	0.98 (2/2)
spherical codes (60°)	$d = 13$	0.36 (1/2)	0.52 (2/2)	0.27 (2/2)	0.46 (1/2)
spherical codes (60°)	$d = 16$	0.26 (1/2)	0.11 (2/2)	0.26 (1/2)	0.53 (2/2)
max. degree 3 graphs	$k = 4$	0.32 (1/2)	0.39 (1/2)	0.32 (1/2)	0.89 (2/2)
max. degree 3 graphs	$k = 6$	0.18 (1/2)	0.20 (1/2)	0.17 (2/2)	0.95 (2/2)
max. degree 3 graphs	$k = 8$	0.11 (1/2)	0.20 (1/2)	0.28 (2/2)	0.66 (2/2)
max. degree 3 graphs	$k = 10$	0.09 (2/2)	0.17 (2/2)	0.00 (0/2)	0.30 (2/2)
Schur	$k = 5$	0.00 (0/2)	0.00 (0/2)	0.76 (2/2)	0.86 (2/2)
Schur	$k = 6$	0.34 (1/2)	0.00 (0/2)	0.18 (1/2)	0.75 (2/2)
Schur	$k = 7$	0.06 (1/2)	0.00 (0/2)	0.00 (0/2)	0.71 (2/2)
Schur	$k = 8$	0.00 (0/2)	0.02 (1/2)	0.00 (0/2)	0.74 (2/2)
covering	$v = 12$	0.00 (0/2)	0.72 (2/2)	0.00 (0/2)	1.00 (2/2)
covering	$v = 16$	0.30 (1/2)	0.70 (2/2)	0.30 (1/2)	0.86 (2/2)
covering	$v = 24$	0.53 (2/2)	0.26 (1/2)	0.26 (1/2)	0.94 (2/2)
covering	$v = 30$	0.20 (1/2)	0.00 (0/2)	0.25 (1/2)	0.84 (2/2)
max. degree 4 graphs	$k = 4$	–	–	0.77 (1/1)	0.84 (1/1)
max. degree 4 graphs	$k = 5$	–	–	0.00 (0/1)	0.53 (1/1)
max. degree 4 graphs	$k = 6$	–	–	0.07 (1/1)	0.98 (1/1)
max. degree 4 graphs	$k = 7$	–	–	0.09 (1/1)	0.15 (1/1)
integer AP-free sets	$n = 300$	0.46 (1/2)	0.46 (1/2)	0.45 (1/2)	1.00 (2/2)
integer AP-free sets	$n = 2000$	0.00 (0/2)	0.78 (2/2)	0.44 (1/2)	1.00 (2/2)
integer AP-free sets	$n = 8000$	0.50 (1/2)	1.00 (2/2)	0.50 (1/2)	1.04 (2/2)
integer AP-free sets	$n = 32000$	0.50 (1/2)	1.00 (2/2)	0.50 (1/2)	1.00 (2/2)

C.9 Code quality across increasing lengths

The same finite factors can generate large Shannon products, while trifference certificates represent codes far beyond an explicit list. Relative quality measures their quality at each length. The lower panels express code size M as a per-coordinate growth base $M^{1/d}$ for Shannon and a ternary rate $\log_3 M/n$ for trifference, where d and n are codeword lengths.

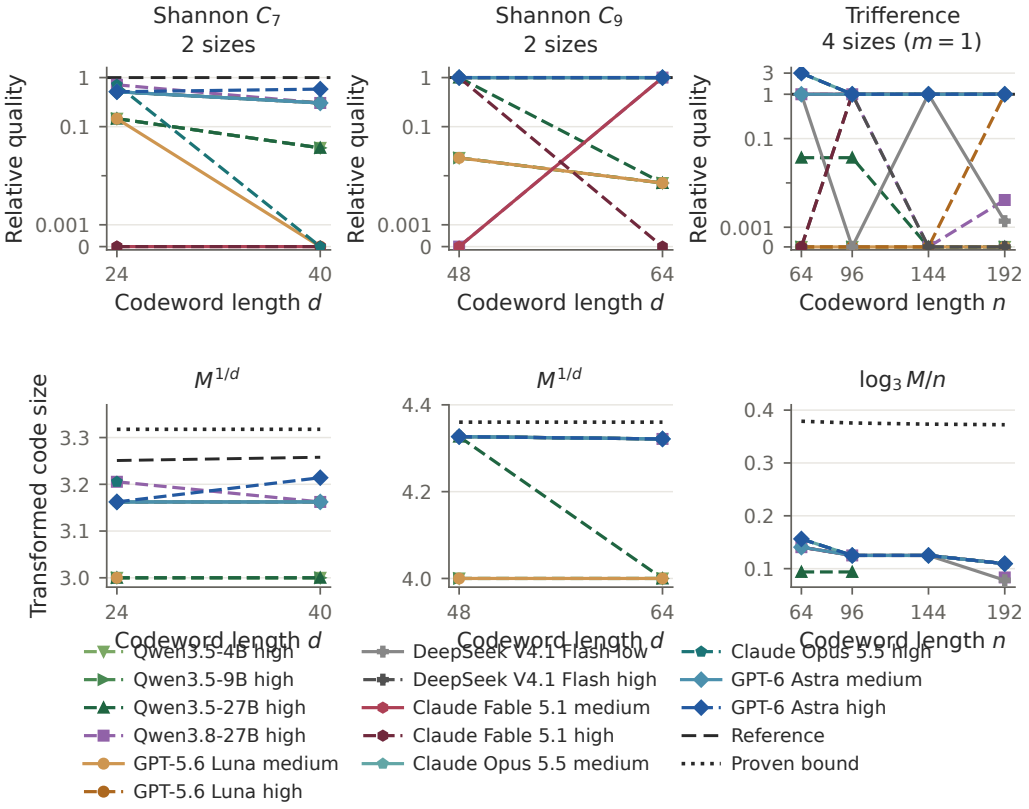


Figure 12: Code-family size–quality curves for all fourteen configurations. Top: relative quality, including zero scores. Bottom: per-coordinate growth for Shannon and ternary rate for trifference, using valid answers only; missing points break the curves. Dark dashed lines are references; dark dotted lines are proven upper bounds. Shannon has two codeword lengths per cycle size; trifference has four lengths at $m = 1$. Each point represents one response.

C.10 Code-family results

Table 7 compares all fourteen tool-free configurations on four Shannon and four trifference instances. Appendix B.2 gives the Shannon product construction and its verification.

configuration	Shannon relative quality	valid	Trifference relative quality	valid	words at $n = 192$
Qwen3.5-4B high	0.053	4/4	0.000	0/4	0
Qwen3.5-9B high	0.053	4/4	0.000	0/4	0
Qwen3.5-27B high	0.297	4/4	0.019	2/4	0
Qwen3.8-27B high	0.504	3/4	0.501	3/4	4.30×10^7
GPT-5.6 Luna medium	0.044	3/4	0.000	0/4	0
GPT-5.6 Luna high	0.705	4/4	0.250	1/4	1.05×10^{10}
DeepSeek V4.1 Flash low	0.705	4/4	0.500	3/4	1.43×10^7
DeepSeek V4.1 Flash high	0.705	4/4	0.250	1/4	0
Claude Fable 5.1 medium	0.250	1/4	1.000	4/4	1.05×10^{10}
Claude Fable 5.1 high	0.250	1/4	0.750	3/4	1.05×10^{10}
Claude Opus 5.5 medium	0.705	4/4	1.500	4/4	1.05×10^{10}
Claude Opus 5.5 high	0.678	3/4	1.500	4/4	1.05×10^{10}
GPT-6 Astra medium	0.705	4/4	1.000	4/4	1.05×10^{10}
GPT-6 Astra high	0.774	4/4	1.500	4/4	1.05×10^{10}

Table 7: Tool-free code-family results on four instances per family, including zero scores. A ratio of 1 matches the construction baseline. The last column gives the trifference code size at length 192, or zero for failure.

Without tools, GPT-6 Astra at high effort and Claude Opus 5.5 at medium and high effort each produce 59,049 trifference codewords at length 64, three times the baseline. All three configurations match the baselines at the other three lengths. Astra medium and Claude Fable 5.1 medium match the baselines at all four lengths, certifying 3^{21} codewords at length 192. GPT-5.6 Luna at medium effort has no valid trifference answer, while Luna high matches the baseline only at length 192. Figure 12 shows how code size scales with length and compares the results with the references and proven bounds.

Tables 8 and 9 give the individual outcomes behind these averages.

Model / effort	$(7, 24)$	$(7, 40)$	$(9, 48)$	$(9, 64)$
Qwen3.5-4B high	0.145	0.037	0.023	0.007
Qwen3.5-9B high	0.145	0.037	0.023	0.007
Qwen3.5-27B high	0.145	0.037	1.000	0.007
Qwen3.8-27B high	0.713	0.304	$0.000^{\dagger}$	1.000
GPT-5.6 Luna medium	0.145	$0.000^{\dagger}$	0.023	0.007
GPT-5.6 Luna high	0.515	0.304	1.000	1.000
DeepSeek V4.1 Flash low	0.515	0.304	1.000	1.000
DeepSeek V4.1 Flash high	0.515	0.304	1.000	1.000
Claude Fable 5.1 medium	$0.000^{\dagger}$	$0.000^{\dagger}$	$0.000^{\dagger}$	1.000
Claude Fable 5.1 high	$0.000^{\dagger}$	$0.000^{\dagger}$	1.000	$0.000^{\dagger}$
Claude Opus 5.5 medium	0.515	0.304	1.000	1.000
Claude Opus 5.5 high	0.713	$0.000^{\dagger}$	1.000	1.000
GPT-6 Astra medium	0.515	0.304	1.000	1.000
GPT-6 Astra high	0.515	0.582	1.000	1.000

Table 8: Shannon relative quality on the four evaluated instances. A dagger marks an invalid or budget-exhausted response, scored zero.

Model / effort	64	96	144	192
Qwen3.5-4B high	$0.000^{\dagger}$	$0.000^{\dagger}$	$0.000^{\dagger}$	$0.000^{\dagger}$
Qwen3.5-9B high	$0.000^{\dagger}$	$0.000^{\dagger}$	$0.000^{\dagger}$	$0.000^{\dagger}$
Qwen3.5-27B high	0.037	0.037	$0.000^{\dagger}$	$0.000^{\dagger}$
Qwen3.8-27B high	1.000	1.000	$0.000^{\dagger}$	0.004
GPT-5.6 Luna medium	$0.000^{\dagger}$	$0.000^{\dagger}$	$0.000^{\dagger}$	$0.000^{\dagger}$
GPT-5.6 Luna high	$0.000^{\dagger}$	$0.000^{\dagger}$	$0.000^{\dagger}$	1.000
DeepSeek V4.1 Flash low	1.000	$0.000^{\dagger}$	1.000	0.001
DeepSeek V4.1 Flash high	$0.000^{\dagger}$	1.000	$0.000^{\dagger}$	$0.000^{\dagger}$
Claude Fable 5.1 medium	1.000	1.000	1.000	1.000
Claude Fable 5.1 high	$0.000^{\dagger}$	1.000	1.000	1.000
Claude Opus 5.5 medium	3.000	1.000	1.000	1.000
Claude Opus 5.5 high	3.000	1.000	1.000	1.000
GPT-6 Astra medium	1.000	1.000	1.000	1.000
GPT-6 Astra high	3.000	1.000	1.000	1.000

Table 9: Trifference relative quality at separation multiplicity $m = 1$. A dagger marks an invalid or budget-exhausted response, scored zero.

C.11 Failures and answer formats

Zero-score responses arise from invalid mathematical constructions, malformed descriptions, or empty and budget-exhausted answers, with the reason for each response recorded in the released data. A format error can obscure a valid construction, as in DeepSeek V4.1 Flash's length-96 trifference response at low effort, which supplies generator rows as integer arrays rather than the required digit strings. Expressing those rows in the required format gives a valid 19,683-word code, although the reported score uses the submitted representation.

C.12 Smaller trifference instances

Four smaller trifference instances test the quality of finite codes that can serve as ingredients for larger constructions. Table 10 evaluates GPT-6 Astra at medium effort on $(n, m) = (30, 1), (36, 2), (42, 3), (56, 1)$, complementing the larger instances in the main evaluation.

Astra matches the construction baselines at lengths 30, 36 and 42 with valid 729-word codes, and triples the reference at length 56 with a 6,561-word code. These finite constructions can be represented explicitly, while algebraic certificates (Appendix D.4) extend evaluation to larger codes whose words cannot all be listed within a response.

Length n	Separation m	Reference	Model	Relative quality	Valid
30	1	729	729	1.000	yes
36	2	729	729	1.000	yes
42	3	729	729	1.000	yes
56	1	2187	6561	3.000	yes

Table 10: GPT-6 Astra at medium effort on four smaller trifference instances. Code sizes are compared with construction baselines.

C.13 Claude Opus 5 on shared instances

We also report an auxiliary comparison with the earlier Claude Opus 5 checkpoint at medium effort on 45 shared instances.

		Mean relative quality		
configuration	Overall score	Published frontiers (15)	Construction baselines (30)	Valid fraction
Claude Opus 5 medium	52.41	0.48	0.55	0.87
Qwen3.5-4B high	7.69	0.08	0.07	0.38
Qwen3.5-9B high	16.96	0.14	0.18	0.47
Qwen3.5-27B high	17.07	0.17	0.17	0.53
Qwen3.8-27B high	40.98	0.37	0.43	0.64
GPT-5.6 Luna medium	20.36	0.18	0.21	0.64
GPT-5.6 Luna high	32.77	0.23	0.37	0.71
DeepSeek V4.1 Flash low	34.30	0.43	0.30	0.80
DeepSeek V4.1 Flash high	45.40	0.34	0.51	0.78
Claude Fable 5.1 medium	75.64	0.54	0.86	0.87
Claude Fable 5.1 high	36.89	0.54	0.29	0.44
Claude Opus 5.5 medium	60.16	0.58	0.61	0.87
Claude Opus 5.5 high	71.38	0.72	0.71	0.87
GPT-6 Astra medium	74.57	0.60	0.82	0.98
GPT-6 Astra high	91.48	0.74	1.00	1.00

Table 11: Comparison with Claude Opus 5 at medium effort on 45 shared instances: 15 with published frontiers and 30 with construction baselines. The overall score is 100 times the mean relative quality over this subset.

C.14 Sensitivity to family selection

We examine the effect of including Kakeya as a scored family in Table 12. This comparison uses the twelve non-code families, with and without Kakeya, on identical model responses. Spherical-code variants share one family label, as do the two Heilbronn domains; each distinct task and parameter setting retains equal weight. Kakeya affects only the mean relative quality on instances with construction baselines and the mean gap closed, leaving the published-frontier results unchanged.

configuration	Relative quality $+K$	Relative quality $-K$	Gap closed $+K$	Gap closed $-K$
Qwen3.5-4B high	0.088	0.096	0.000	0.000
Qwen3.5-9B high	0.191	0.202	0.000	0.000
Qwen3.5-27B high	0.184	0.192	0.000	0.000
Qwen3.8-27B high	0.425	0.418	0.005	0.004
GPT-5.6 Luna medium	0.203	0.207	0.000	0.000
GPT-5.6 Luna high	0.419	0.408	0.002	0.002
DeepSeek V4.1 Flash low	0.283	0.292	0.000	0.000
DeepSeek V4.1 Flash high	0.470	0.510	0.006	0.007
Claude Fable 5.1 medium	0.880	0.868	0.013	0.014
GPT-6 Astra medium	0.838	0.822	0.006	0.007
GPT-6 Astra high	1.015	1.014	0.017	0.017

Table 12: Effect of including Kakeya in the aggregate for the eleven configurations with control results. $+K$ includes Kakeya and $-K$ excludes it. Relative quality is averaged over instances with construction baselines; both settings use the same responses from the twelve non-code families and equal instance weights. Published-frontier scores are identical in the two cases.

C.15 Gap closed and progress toward the LABS target

Gap closed starts at the same reference as relative quality and measures logarithmic progress toward a proven bound. Table 13 separates the two reference groups. The combined value gives each eligible instance equal weight, so the groups contribute in proportion to their 30 and 34 instances. Invalid answers and valid answers that do not exceed the reference contribute zero.

The five LABS instances instead use the conjectured merit-factor target of 12.32. We compute progress from each construction baseline toward this target using the same logarithmic formula, and report it separately. These five values do not enter the proven-bound mean in Table 2.

	Gap closed to proven bounds			LABS target
Configuration	All 64	Published 30	Construction 34	5 instances
Qwen3.5-4B high	0.000	0.000	0.000	0.000
Qwen3.5-9B high	0.000	0.000	0.000	0.000
Qwen3.5-27B high	0.000	0.000	0.000	0.000
Qwen3.8-27B high	0.004	0.000	0.007	0.000
GPT-5.6 Luna medium	0.000	0.000	0.000	0.000
GPT-5.6 Luna high	0.002	0.000	0.004	0.000
DeepSeek V4.1 Flash low	0.000	0.000	0.000	0.000
DeepSeek V4.1 Flash high	0.006	0.000	0.011	0.000
Claude Fable 5.1 medium	0.012	0.000	0.023	0.000
Claude Fable 5.1 high	0.003	0.000	0.006	0.000
Claude Opus 5.5 medium	0.006	0.000	0.011	0.000
Claude Opus 5.5 high	0.009	0.000	0.017	0.000
GPT-6 Astra medium	0.006	0.000	0.011	0.000
GPT-6 Astra high	0.016	0.000	0.030	0.006
GPT-6 Astra high + tools	0.063	0.000	0.118	0.401
GPT-5.6 Luna high + tools	0.047	0.000	0.088	0.247
Claude Opus 5.5 high + tools	0.075	0.000	0.140	0.186

Table 13: Mean progress from the reference. The first three numeric columns use proven mathematical bounds; the last uses the conjectured LABS target. Each mean includes zero scores.

D Verification of constructions

The accepted formats can express constructions that match the references for all 69 evaluated instances. Appendix E.1 documents these reference constructions; the representations below include explicit objects, compact constructions and algebraic certificates.

D.1 Verifier validation

The verifiers use separate code for mathematical checks and objective calculations, and share code for extracting answers and expanding compact descriptions. They agree for all model constructions that pass the primary verifier and all 69 reference constructions.

Tests include forbidden configurations, duplicates, incorrect dimensions and types, false cardinality claims and invalid certificates. Spherical-code tests exercise exact angle boundaries. For small Shannon and trifference codes, exhaustive checks compare verifier decisions with the defining constraints, and certificate tests compare the certified result with an explicitly expanded code. All deliberately invalid test cases are rejected, and the exhaustive and certificate-expansion checks agree with the verifiers. These checks and expected outcomes accompany the release.

D.2 Compact representations

Mathematical ingredients

Compact descriptions specify a construction through its mathematical ingredients:

- *Orbits*: apply coordinate permutations and sign changes to generator vectors, with duplicate vectors counted once.
- *Products*: take the Cartesian product of smaller cap sets.
- *Digit sets*: select integers by their base-b digits, optionally restricting the sum of squared digits as in Behrend-type constructions.
- *Difference sets*: form a corner-free grid $\{(x, y) : x - y \in D\}$ from a specified set D.
- *Cayley graphs*: specify cyclic-group orders and generators, including inverses for undirected edges.

Descriptions are expanded before verification, except for the algebraic code certificates below. Expansion is limited to 60,000 elements, with any smaller family-specific limit taking precedence. Digit enumeration allows at most 480,000 search nodes. Expansion and verification together have a 60-second limit; exceeding it scores zero. Prompts state the accepted representation.

Exact quadratic coordinates. For the thirteen-dimensional kissing instance, a coordinate may be written as $a + b\sqrt{3}$, encoded by an integer pair (a, b) with $|a|, |b| \leq 1000$. Both verifiers check the angle constraints exactly in $\mathbb{Q}(\sqrt{3})$, without floating-point tolerances. This format represents the full 1,154-point published construction (Zinoviev & Ericson, 1999); its JSON answer fits within 128k tokens and passes both verifiers within the 60-second limit. Submissions may contain at most 20,000 vectors.

D.3 Code products and explicit codewords

Shannon codes may be listed explicitly, expanded as products of up to 4,096 words, or specified by factors C_i and integer powers r_i. The factors must satisfy $\sum_i r_i \dim(C_i) = d$. Verifying each factor establishes the full product's separation property, and its exact size is $\prod_i |C_i|^{r_i}$. Certificates may list at most 16,384 factor words in total and describe dimension at most 256. The supported alphabets are $q \in \{5, 7, 9\}$, with C_5 used for calibration. We compute a rational upper bound on $\vartheta(C_q)^d$ (Lovasz, 1979) and round it down to an integer, since code sizes are integers.

The explicit-expansion format for trifference accepts up to 6,561 words of length at most 128, or a ternary generator matrix with at most eight rows. For a verified linear code, translating a triple to include zero reduces verification to pairs of codewords. Nonlinear codes require the direct triple check. Both procedures have the same 60-second limit.

D.4 Linear trifference certificates

The main trifference instances use algebraic certificates to represent codes beyond the explicit-list limit. A linear-code certificate specifies a ternary generator matrix with at most 16 rows, rank $k \leq 12$, length $n \leq 1024$ and $m \in \{1, 2, 3\}$. The verifier computes the rank and certifies the exact size 3^k.

Hyperplane verification

For a full-rank ternary generator matrix G, translate a triple to $(0, uG, vG)$ and put $z = u + v$, $a = u - v$. Coordinate j separates the triple exactly when $(zG)_j = 0$ and $(aG)_j \neq 0$. The case $z = 0$ requires the full code to have minimum distance at least m. For each nonzero projective z, restrict G to the coordinates where zG is zero. The remaining requirement is equivalent to that restriction having rank $k-1$ and minimum distance at least m. This extends the minimal-code characterisation of trifference (Bishnoi et al., 2024; 2025) into a direct check. Verification enumerates $(3^k - 1)/2$ projective directions. For $m \leq 3$, reduced row-echelon form allows exact distance checking using only combinations with at most $m-1$ nonzero pivot coefficients. A second verifier checks the rank and distance conditions independently.

test construction	words	primary (s)	independent (s)
rank 12, $m = 1$	531,441	5.61	8.66
rank 12, $m = 3$	531,441	21.63	37.70
concatenation, $n = 1024$	5.154×10^{47}	< 0.01	< 0.01

Table 14: Measured verification times for reference constructions. Both implementations check the certificate and exact code size. The length-1024 example illustrates verification at a scale beyond the evaluated model answers.

D.5 Reed–Solomon concatenation

Construction and separation guarantee

A Reed–Solomon concatenation certificate supplies a verified inner code of length a and size $q = 3^r$, a monic irreducible polynomial defining $\mathbb{F}_q$, and outer parameters $1 \leq K \leq N \leq q$. Every triple of distinct inner codewords must be separated in at least b coordinates. The outer code evaluates all polynomials of degree below K at N distinct field elements. Replacing its symbols by the q inner words gives exactly q^K words of length aN, with zero padding to n allowed.

The separation guarantee follows from the outer code's distance. Two distinct outer polynomials agree at most $K - 1$ times, so their distance is at least $D = N - K + 1$. If three outer words have t coordinates where all three symbols differ, their three pairwise distances sum to at most $2N + t$ and at least $3D$. Hence $t \geq N - 3K + 3$, and the certificate is valid when $b(N - 3K + 3) \geq m$. This gives an exact construction and count without expanding its words. Both checkers verify the inner code and field irreducibility. For irreducibility, one implementation uses Frobenius/gcd tests and the other uses monic trial division. The representation supports a single concatenation layer.

What the certificate establishes

Length	Number of words	Guaranteed separation
aN	q^K	$b(N - 3K + 3)$

The certificate is accepted when its guaranteed separation $b(N - 3K + 3)$ is at least m. The finite inner code and irreducible polynomial make both the cardinality and the separation claim checkable.

Representation limits. At length 64, the `bench-v1.0` formats admit at most $3^{12} = 531{,}441$ words (relative quality 27), attained by Opus 5.5. Direct and explicit formats obey this cap. For concatenation, $3^r \leq \lfloor 2(3/2)^a \rfloor$, $N \leq \min(3^r, \lfloor 64/a \rfloor)$, $K \leq \lfloor (N + 2)/3 \rfloor$ and $r \leq 12$ give $rK \leq 12$. This is a representation limit, not a known mathematical optimum; scores are not clipped at reference parity.

E References, bounds and targets

E.1 Verified reference constructions

For every one of the 69 evaluated instances, we verified a construction matching its reference, covering all 30 published frontiers and all 39 construction baselines. Each object uses an accepted answer format, fits within the 128k output limit and passes both verifiers within 60 seconds. Together, these constructions establish that a relative quality of 1 is attainable throughout the evaluation under the allowed answer formats and verification time. All 69 objects and their verification results are included with the benchmark.

Reference group	Verified instances	Largest answer (tokens)
Published frontiers	30 / 30	77,528
Construction baselines	39 / 39	19,149
Total	**69 / 69**	**77,528**

Table 15: Reference constructions for all 69 instances. Each object attains relative quality 1 and passes both verifiers within 60 seconds. Token counts use the larger of `cl100k_base` and `o200k_base` for the serialised answer.

E.2 Published constructions

Table 16 gives selected published frontiers and the proven bounds used to measure gap closed. Values follow the listed parameter order.

Family / parameters	Reference and source	Bound and source
Cap set, $d = 10, 11$	2432, 5504; Edel and Karapetyan–Karapetyan constructions (Edel, 2004; Karapetyan & Karapetyan, 2023).	5619, 16857; proven bounds from Versluis (2017) and slicing.
Kissing, $d = 13, 14, 15$	1154, 1932, 2564; constructions (Zinoviev & Ericson, 1999; Ganzhinov, 2025; Leech & Sloane, 1971).	2064, 3174, 4853; proven bounds (Leijenhorst & de Laat, 2024).
Matrix multiplication, $(4, 4, 5)$, $(5, 5, 5)$	61, 93; schemes from the Lille FMM catalogue.	20, 25; proven lower bounds from tensor flattenings.
Degree–diameter, $(4, 5)$, $(7, 3)$, $(3, 7)$, $(4, 4)$	364, 168, 196, 104; graphs from Comellas (2026).	485, 302, 382, 161; proven Moore bounds (Miller & Širáň, 2013).
$\mathbb{F}_5^n$ AP-free, $n = 5, 6$	194, 649; affine charts of Edel's projective caps (Edel, 2026; Elsholtz & Pach, 2020).	1296, 7497; proven bounds from the polynomial-method count (Ellenberg & Gijswijt, 2017).
MOLS, $n = 10, 14, 18$	2, 4, 5; constructions (Miller et al., 2024; Todorov, 2012; Abel, 2015).	8, 12, 17; proven $n - 2$ or $n - 1$ bounds, including Lam et al. (1989).
Schur, $k = 6, 7, 8$	536, 1696, 5362; partitions (Fredricksen & Sweet, 2000; Rowley, 2021; Bengone et al., 2026).	1896 at $k = 6$; general factorial upper bound at $k = 7, 8$; proven.
Covering, (v, k, t): $(18, 7, 4)$, $(24, 6, 3)$, $(30, 6, 3)$	126, 116, 225; block lists from the La Jolla repository (Gordon, 2026).	111, 112, 210; proven lower bounds recorded in the repository (Gordon, 2026).

Table 16: Selected published frontiers and proven bounds. Values follow the listed parameter order; minimisation tasks use lower bounds.

E.3 Construction baselines and controls

Construction baselines are verified outputs of documented construction and search procedures, providing reproducible comparison points across the evaluation. Published LABS sequences (Dimitrov, 2022), spherical-code tables (Cohn, 2026) and Heilbronn configurations (Stead, 2026) provide stronger constructions at some evaluated parameters. References remain unchanged within `bench-v1.0`; exceeding a construction baseline alone does not establish a mathematical record.

Family / parameters	Reference and source	Bound or target and source
Kakeya control, $(q, n) = (13, 3)$, $(7, 4)$, $(11, 4)$	Expert constructions of sizes 703, 553, 2741 (Saraf & Sudan, 2008); construction baselines.	595, 375, 2105; proven lower bounds (Bukh & Chao, 2021).
LABS	Rotated Legendre construction or reference search (Packebusch & Mertens, 2016).	Merit factor 12.32; conjectured target.
Heilbronn (both domains)	Expert construction or reference search; see Appendix A.	$1/(n-2)$; trivial proven bound from triangulation.
Linear-equation-free sets; corners	Digit constructions or reference search (Behrend, 1946; Ruzsa, 1993).	n and n^2; trivial proven bounds.
Spherical codes	Expert construction or reference search.	Cap-packing bound; proven.

Table 17: Construction baselines and controls, with proven bounds or the conjectured LABS target.

E.4 Constructing the code references

Shannon references optimise products of small codes: the Polak–Schrijver 367-word code in C_7^5, its projections, and the 81-word code $\{(i, j, 2i + 4j \bmod 9) : i, j \in \mathbb{Z}_9\}$ in C_9^3. The ten-second search modifies these small codes and recomputes their best products. The resulting references reach roughly 4.8×10^{40} words and have short certificates. Recent recursive improvements (Tandon, 2026) suggest further composition methods. At $(q, d) = (7, 24)$, the `bench-v1.0` baseline uses $367^4 \times 107$; the 108-word factor in C_7^4 due to Vesel and Žerovnik, listed in Polak & Schrijver (2019, Table 1), gives the stronger product $367^4 \times 108$, explaining the corresponding tool-assisted improvement over this baseline.

For the shorter trifference instances, references include 729-word linear codes of lengths 26, 35 and 41 with separation multiplicities 1, 2 and 3, and a 2,187-word code of length 49. The first three augment a published generator array with selected coordinates; the last uses projective columns of weight one or two. Zero padding gives codes at the required lengths.

At lengths 64, 96, 144 and 192 with $m = 1$, the reference library combines 27-word and 81-word inner codes of lengths 9 and 14, finite-geometry constructions, and a 6,561-word length-56 construction generated by GPT-6 Astra. The source constructions are included in the release. References maximise code size over the library's inner codes and outer parameters and were fixed before evaluating these lengths. This procedure does not use the search-time budget, so trifference is excluded from the timed comparison. Astra medium matches all four references through concatenation (Table 18). Better inner codes can improve these results. Table 14 shows that both verifiers also handle much larger constructions within 60 seconds.

instance	length	reference words	verified words	Relative quality	generation (s)
T1	64	19,683	19,683	1.000	52.3
T2	96	531,441	531,441	1.000	53.8
T3	144	387,420,489	387,420,489	1.000	37.0
T4	192	10,460,353,203	10,460,353,203	1.000	63.8

Table 18: GPT-6 Astra at medium effort on four trifference lengths. The evaluator checks each submitted construction and computes its exact size. Ratios use the construction baselines; invalid or budget-exhausted responses score zero.

E.5 Reference searches and algebraic constructions

Expert constructions can leave the baseline unchanged even when search improves. Let r_t be the search objective after t seconds and z_t the better of that result and the expert construction. Table 19 compares both quantities at 10 and 600 seconds using the same expert construction. In dimension ten, for example, cap-set search improves from 1140 to 1202 points, both below the 2240-point product construction. The baseline therefore stays at 2240. For Schur with six, seven and eight colours, both search budgets return 364, 1,093 and 3,280 elements, respectively. The longer covering searches reduce the numbers of blocks from 179, 366 and 181 to 178, 361 and 178, respectively. Shannon search leaves all four code sizes unchanged. The reported gains concern the specific search procedures tested.

Algebraic construction baselines. For corners, translated ternary digit sets give AP-free difference sets D and corner-free grids $\{(x, y) : x - y \in D\}$. For spherical codes, candidate vectors combine selected nonzero coordinates and signs, binary sign vectors and coordinate axes; pairwise angles are checked exactly. These procedures were calibrated on 17 separate parameter settings after model responses were collected, then applied uniformly to all 20 evaluation instances.

Every reference construction passes both verifiers and is shared by all models. Calibration is separate from the timed searches, which use the same algebraic baseline at both budgets.

family	class	runs	instances	mean Δ_r	max Δ_r	mean Δ_z	max Δ_z
cap set	R	3	2	0.108	0.163	0.000	0.000
spherical codes	R	6	5	0.000	0.000	0.000	0.000
corners	R	3	3	0.019	0.024	0.000	0.000
linear-equation-free sets	R	3	3	0.016	0.019	0.016	0.019
Schur	R	3	3	0.000	0.000	0.000	0.000
$\mathbb{F}_q^n$ AP-free	R	3	2	0.041	0.083	0.041	0.083
MOLS	R	3	3	0.000	0.000	0.000	0.000
Shannon	R	4	4	0.000	0.000	0.000	0.000
LABS	S	3	3	0.058	0.086	0.000	0.000
Heilbronn	S	6	6	0.687	0.914	0.687	0.914
matrix multiplication	S	3	1	0.000	0.000	0.000	0.000
degree–diameter	S	3	2	0.000	0.000	0.000	0.000
covering	S	3	3	0.012	0.017	0.012	0.017

Table 19: Search gains for seeds 0–2, with all four evaluated instances included for Shannon. R denotes search-resistant families; S denotes families that may benefit from search, for the procedures studied here. Δ_r compares raw 600- and 10-second search objectives; Δ_z compares the baselines after including the expert construction. Both are log improvement ratios, with direction adjusted for minimisation. Repeats are averaged within each instance before computing the means; maxima are over paired runs.

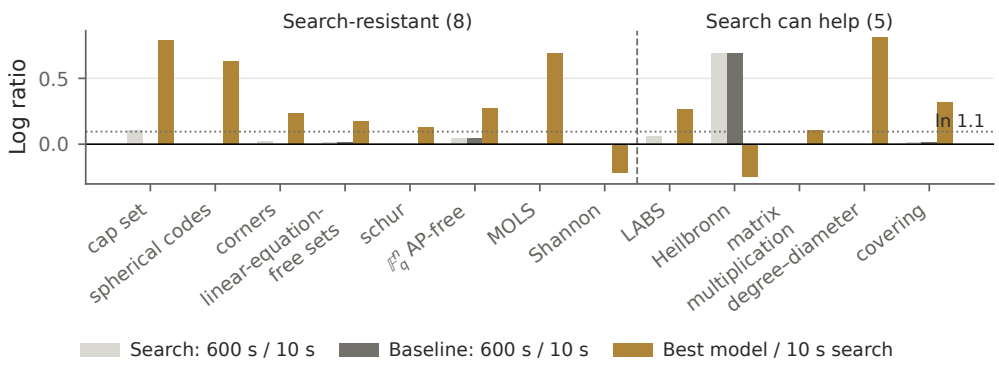


Figure 13: Effect of increasing reference search time. Light grey shows the improvement in the search result from 10 to 600 seconds. Dark grey shows the improvement after taking the better of the search result and the established construction at each budget. Ochre compares the best valid tool-free answer with the 10-second search result. For minimisation, ratios are inverted before taking logarithms; log ratios are then averaged over distinct instances. The dotted line marks a 1.1-fold improvement.

F Instance parameters and resource limits

We use three parameter ranges: A1 contains small familiar instances, while A2 and A3 increase size. Family selection for A3 screened the first sampled setting for a ratio of at least 1.4 between the implemented baseline z and a bound or target b: b/z for maximisation and z/b for minimisation. This screening identifies families with room for improvement; the remaining instances vary the parameters within those families. Answers must also fit the representation limits and be verifiable within 60 seconds. Kakeya is evaluated separately as a control. For example, the Kakeya setting $(q, n) = (13, 3)$ has a proven lower bound of 595 (Bukh & Chao, 2021) and a construction of size 703, leaving a ratio of only 1.18.

The main evaluation contains 69 instances: 61 in the twelve non-code families, four Shannon instances and four trifference instances. Each configuration contributes one response per instance under the protocol in Appendix G. The released manifest specifies the selected instances. Table 20 lists the instances, and Table 21 specifies the triangular domains. Kakeya provides a separate control for the family-selection comparison. Smaller-instance parameters appear in Table 22.

Table 20: The 69 instances in the main evaluation. Each configuration contributes one response per instance.

Family	Parameters
Cap sets	$d = 10$
Cap sets	$d = 11$
Cap sets	$d = 9$
Corners	$n = 138$
Corners	$n = 183$
Corners	$n = 150$
Corners	$n = 162$
Corners	$n = 185$
Schur	$k = 7$
Schur	$k = 6$
Schur	$k = 8$
$\mathbb{F}_q^n$ AP-free	$n = 6,\ q = 5$
$\mathbb{F}_q^n$ AP-free	$n = 5,\ q = 5$
$\mathbb{F}_q^n$ AP-free	$n = 4,\ q = 7$
Spherical (60°)	$d = 14$
Spherical (60°)	$d = 15$
Spherical (variable)	$c = 24/100,\ d = 12$
Spherical (variable)	$c = 39/100,\ d = 12$
Spherical (variable)	$c = 36/100,\ d = 14$
Spherical (variable)	$c = 39/100,\ d = 14$
Spherical (variable)	$c = 36/100,\ d = 12$
Spherical (60°)	$d = 13$
LABS	$N = 280$
LABS	$N = 375$
LABS	$N = 357$
LABS	$N = 281$
LABS	$N = 240$
Heilbronn (square)	$n = 46$
Heilbronn (square)	$n = 26$
Heilbronn (square)	$n = 40$
Heilbronn (square)	$n = 50$
Heilbronn (square)	$n = 48$
Heilbronn (triangle)	$n = 32, T_1$
Heilbronn (triangle)	$n = 34, T_2$
Heilbronn (triangle)	$n = 31, T_3$
Heilbronn (triangle)	$n = 40, T_4$
Heilbronn (triangle)	$n = 26, T_5$
Degree–diameter	$d = 4,\ k = 5$
Degree–diameter	$d = 7,\ k = 3$

Family	Parameters
Degree–diameter	$d = 4,\ k = 4$
Degree–diameter	$d = 3,\ k = 7$
Degree–diameter	$d = 6,\ k = 3$
Degree–diameter	$d = 5,\ k = 4$
Degree–diameter	$d = 3,\ k = 6$
Linear-equation-free	$a = 3,\ b = 4,\ n = 19044$
Linear-equation-free	$a = 2,\ b = 5,\ n = 18883$
Linear-equation-free	$a = 2,\ b = 3,\ n = 17965$
Linear-equation-free	$a = 3,\ b = 4,\ n = 13312$
Linear-equation-free	$a = 3,\ b = 4,\ n = 6248$
Covering designs	$k = 7,\ t = 4,\ v = 18$
Covering designs	$k = 6,\ t = 3,\ v = 30$
Covering designs	$k = 6,\ t = 3,\ v = 24$
Matrix multiplication	$m = 4,\ n = 4,\ p = 5$
Matrix multiplication	$m = 5,\ n = 5,\ p = 5$
Matrix multiplication	$m = 5,\ n = 4,\ p = 5$
MOLS	$n = 14$
MOLS	$n = 10$
MOLS	$n = 18$
MOLS	$n = 15$
MOLS	$n = 12$
MOLS	$n = 20$
Shannon	$d = 24,\ q = 7$
Shannon	$d = 40,\ q = 7$
Shannon	$d = 48,\ q = 9$
Shannon	$d = 64,\ q = 9$
Trifference	$m = 1,\ n = 64$
Trifference	$m = 1,\ n = 96$
Trifference	$m = 1,\ n = 144$
Trifference	$m = 1,\ n = 192$

F.1 Triangular domains

Table 21: Triangular domains used in the main evaluation. Coordinates are integers.

Domain	Vertex 1	Vertex 2	Vertex 3
T_1	$(9026, 8197)$	$(1769, 9178)$	$(7298, 1276)$
T_2	$(578, 9933)$	$(8598, 7337)$	$(2075, 588)$
T_3	$(411, 7999)$	$(9004, 9890)$	$(3031, 713)$
T_4	$(7922, 8793)$	$(9917, 1548)$	$(1510, 3942)$
T_5	$(526, 8526)$	$(8696, 8279)$	$(8084, 1688)$

Each triangle is specified by its three integer-coordinate vertices. The objective is the minimum triangle area divided by the area of the containing domain.

F.2 Smaller-instance parameter draws

Table 22: Three representative smaller-instance draws per tier, in seed order 0, 1, 2. Trifference uses the lengths listed in the main evaluation table.

Family	Tier	Representative parameters
Cap sets	A1	$d = 3$; $d = 3$; $d = 4$
	A2	$d = 8$; $d = 7$; $d = 7$
Corner-free sets	A1	$n = 23$; $n = 15$; $n = 22$
	A2	$n = 66$; $n = 50$; $n = 79$
Schur colourings	A1	$k = 4$; $k = 4$; $k = 3$
	A2	$k = 5$; $k = 5$; $k = 6$
Progression-free sets over finite fields	A1	$q = 5,\ n = 4$; $q = 5,\ n = 4$; $q = 5,\ n = 4$
	A2	$q = 7,\ n = 3$; $q = 7,\ n = 4$; $q = 7,\ n = 4$
Spherical codes (60°)	A1	$d = 7$; $d = 7$; $d = 7$
	A2	$d = 11$; $d = 12$; $d = 9$
Spherical codes (variable angle)	A1	$d = 6,\ c = 37/100$; $d = 6,\ c = 35/100$; $d = 6,\ c = 48/100$
	A2	$d = 10,\ c = 47/100$; $d = 10,\ c = 45/100$; $d = 9,\ c = 20/100$
Low-autocorrelation binary sequences	A1	$N = 19$; $N = 25$; $N = 29$
	A2	$N = 68$; $N = 74$; $N = 91$
Heilbronn triangles (square)	A1	$n = 7$; $n = 6$; $n = 5$
	A2	$n = 18$; $n = 17$; $n = 18$
Heilbronn triangles (triangle)	A1	$n = 9$; $n = 12$; $n = 11$
	A2	$n = 16$; $n = 14$; $n = 12$
Degree–diameter graphs	A1	$d = 4,\ k = 3$; $d = 3,\ k = 3$; $d = 3,\ k = 3$
	A2	$d = 10,\ k = 2$; $d = 5,\ k = 3$; $d = 3,\ k = 5$
Linear-equation-free sets	A1	$a = 3,\ b = 4,\ n = 129$; $a = 1,\ b = 2,\ n = 101$; $a = 2,\ b = 3,\ n = 155$
	A2	$a = 3,\ b = 5,\ n = 1394$; $a = 1,\ b = 3,\ n = 1445$; $a = 3,\ b = 4,\ n = 1439$
Covering designs	A1	$v = 12,\ k = 6,\ t = 3$; $v = 13,\ k = 4,\ t = 2$; $v = 10,\ k = 5,\ t = 3$
	A2	$v = 16,\ k = 6,\ t = 3$; $v = 15,\ k = 7,\ t = 4$; $v = 16,\ k = 6,\ t = 3$
Matrix multiplication	A1	$n = 2,\ m = 2,\ p = 4$; $n = 3,\ m = 3,\ p = 3$; $n = 2,\ m = 2,\ p = 4$
	A2	$n = 3,\ m = 3,\ p = 5$; $n = 3,\ m = 3,\ p = 4$; $n = 4,\ m = 4,\ p = 5$
Mutually orthogonal Latin squares	A1	$n = 12$; $n = 10$; $n = 10$
	A2	$n = 15$; $n = 14$; $n = 14$
Shannon codes	A1	$q = 7,\ d = 3$; $q = 9,\ d = 3$; $q = 7,\ d = 4$
	A2	$q = 9,\ d = 12$; $q = 7,\ d = 8$; $q = 7,\ d = 12$

G Evaluation settings and example prompt

G.1 Inference settings

The fourteen tool-free configurations follow Section 3.5. For each instance, we retain the first completed response or generation-budget stop. Completed responses are verified; invalid answers and generation-budget exhaustion score zero and are not retried. Network and service failures are unscored, and the evaluation may resume or be resubmitted after recovery. Bounded transport reconnection is permitted, so an evaluation can involve more than one generation request. Usage records include all reported tokens, with unreported usage on interrupted requests marked as unknown. Code-family, thirteen-dimensional kissing and Claude Fable 5.1 high and Opus 5.5 evaluations use a two-hour generation deadline, followed by a separate 60-second verification limit. Appendix H describes the tool-assisted protocol.

All configurations receive identical mathematical prompts and format instructions. API models receive a mathematical-construction system message; CLI models retain their native system instructions. No follow-up prompts are used to extend completed answers. Tool-free CLI runs disable tools in their configuration: Claude Code uses `--tools ""` with a strict MCP configuration, while Codex disables execution, web and MCP tools.

Claude Fable 5.1. Claude Code 2.1.269 at medium and high effort, using independent sessions and the model identifier `claude-fable-5-1`. Medium-effort responses were collected on 12–17 September 2026 and high-effort responses on 20 September 2026.

Claude Opus 5.5. Claude Code 2.1.280 at medium and high effort, with model identifier `claude-opus-5-5`, collected on 22–23 September 2026. Each configuration contains 69 fresh responses, with the same prompts, output budget and verification rules as the other tool-free evaluations.

GPT-6 Astra and GPT-5.6 Luna. Codex CLI 0.154.0 at the reasoning-effort levels indicated in the results.

DeepSeek V4.1 Flash. Chat-completions API with reasoning enabled at low or high effort.

Qwen models. vLLM 0.29.0 at high reasoning effort. The 27B checkpoints use FP8 weights; serving settings are supplied with the release.

Reference searches. Each search uses a fixed random seed and one process, with iterated greedy construction or local optimisation as appropriate to the family. We report results at 10 and 600 seconds.

G.2 Example prompt

Prompts specify the problem, parameters and supported answer format. The following prompt was used for the ten-dimensional cap-set instance.

Model prompt

```
Problem (Cap-Set, d=10). Find a subset S of F_3^10
(vectors of length 10 with entries in {0,1,2}) of
maximum possible size containing no three distinct
vectors x, y, z with x + y + z = 0 (mod 3)
coordinate-wise -- a cap set. Output: a JSON list of the
vectors, each written as a string of 10 digits (at most
20000 vectors are accepted).
Output format: a JSON object {"answer": X} where X is a
JSON array of strings of digits 0/1/2, all of the stated
length.
An answer that violates the stated property, is
malformed, contains duplicates, or is out of range
scores zero. Among valid answers, a better objective
scores higher. Give one answer.
Compact form allowed: {"product": [S1, S2, ...]} where
S_i is a list of digit strings in dimension d_i with d_1
+ ... = d; the set is all concatenations.
```

In the tool-free track, each prompt also states that tools are unavailable and requests a single JSON answer.

H Code and web access

Code and web access improve all three models' overall scores. Claude Opus 5.5 achieves the highest overall score, while GPT-6 Astra comes closest to the published frontiers. Family and instance comparisons distinguish these patterns.

H.1 Protocol

We evaluated GPT-6 Astra and GPT-5.6 Luna at high effort through Codex CLI 0.154.0 on 18 September 2026, with code execution and live web search. Each instance has an independent environment with four CPU threads, 16 GiB of memory and a two-hour wall-clock limit, with no limit on total output tokens. An evaluation may contain multiple model and tool interactions. Models can inspect constructions and stop when ready, saving their submission as a JSON file of at most 32 MiB. When the evaluation ends or reaches a resource limit, we score the saved construction if it was written within the time limit and passes verification. Missing, late or invalid submissions receive zero. Both verifiers and their 60-second verification limit are the same as in the tool-free track.

Claude Opus 5.5 was evaluated at high effort through Claude Code 2.1.280 on 22–23 September 2026. It uses native web search and an MCP shell connected to the same CPU environment and resource limits. The native system instructions and tool interfaces therefore differ between Claude Code and Codex. Claude's per-response output limit is 128k; there is no cumulative limit across the trajectory. At an output or resource stop, the saved construction is scored under the same rule.

Mathematical prompts, parameters and formats match the tool-free evaluation; we replace the no-tools instruction with: "Save your final answer as a single JSON object to /workspace/answer.json." Prompts give no reference values, strategies or budget reminders, and environments contain no benchmark library, reference objects or prior answers. Subagents are disabled and bounded reconnection is enabled. Network and service failures are unscored. We retain the first result that can be scored and do not repeat an evaluation to improve its answer.

Web access permits retrieval and adaptation of published constructions. Benchmark materials were available online during the Opus 5.5 evaluation, but not during the Astra and Luna evaluations. An audit of Opus 5.5's recorded interactions found no evidence of direct access to these materials.

H.2 Construction quality

GPT-6 Astra returns 69 valid constructions, improving 62 and tying 7 tool-free outcomes. GPT-5.6 Luna returns 66 valid constructions, with 61 improvements, 5 ties and 3 declines. Luna's three zeroes comprise two invalid submissions and one evaluation that exceeded the memory limit before producing a submission. Both verifiers agree on every scored outcome.

Opus 5.5 returns 69 valid constructions, with 62 improvements and 7 ties against its high-effort tool-free answers. Its final covering evaluation reaches the two-hour limit with a valid 225-block construction saved, matching the reference and receiving full relative quality. Its mean relative quality is 0.985 on published frontiers and 2.428 on construction baselines, matching 23 published references and exceeding 33 construction baselines.

Mean relative quality on instances with published frontiers reaches 0.998 for Astra and 0.961 for Luna, matching 29 and 23 references without exceeding any. On the 39 construction baselines, Astra improves 33 and Luna 25, reaching mean ratios of 1.765 and 1.372. Luna exceeds Astra on 7 Heilbronn instances, with a higher mean across the ten instances in that family. These comparisons include web retrieval, code execution and the different generation budgets of the two tracks.

		Astra high		Luna high		Opus 5.5 high	
Family	n	No tools	Tools	No tools	Tools	No tools	Tools
cap set	3	0.923	0.984	0.464	0.946	0.877	0.955
LABS	5	0.985	1.374	0.000	1.189	0.379	1.166
Heilbronn	10	0.684	2.138	0.196	2.270	0.456	2.936
spherical codes	8	1.313	1.642	0.284	1.420	0.927	1.784
corners	5	1.026	1.109	0.961	1.078	1.009	1.117
matrix multiplication	3	0.895	1.000	0.000	0.886	0.595	1.000
degree–diameter	7	0.752	1.000	0.257	1.000	0.476	0.992
linear-equation-free	5	0.974	1.477	0.564	0.597	0.971	1.387
$\mathbb{F}_q^n$ AP-free	3	1.062	1.168	0.691	1.052	0.902	1.231
MOLS	6	0.550	1.000	0.433	0.958	0.592	0.958
Schur	3	0.731	1.000	0.058	1.000	0.246	1.000
covering	3	0.883	1.000	0.111	0.861	0.312	1.000
Shannon	4	0.774	1.002	0.705	0.998	0.678	1.002
trifference	4	1.500	3.000	0.250	0.750	1.500	7.500

Table 23: GPT-6 Astra, GPT-5.6 Luna and Claude Opus 5.5 at high effort, with and without tools on the 69 instances. Family means average relative quality over instances, including zeroes.

For the length-64 trifference example in Section 3.3, tools increase Astra's code size from 59,049 to 177,147 codewords, raising relative quality from 3 to 9. Opus 5.5 constructs 531,441 codewords, with relative quality 27 against the same construction baseline. This reaches the maximum size supported by the `bench-v1.0` formats at length 64, rather than a known mathematical optimum (Appendix D.5). This instance and the ten Heilbronn instances account for more than its net score advantage over Astra; the other 58 instances slightly favour Astra. Opus remains first when any single family is excluded. Luna matches the construction baselines at lengths 64, 96 and 192, but its length-144 certificate is invalid. All submitted objects are released.

H.3 Token usage

Table 24 sums reported token usage across all model turns in each model's 69 evaluations. Reasoning tokens are included in output totals, and cached tokens are included in input totals. For Claude Code, we also include model usage reported by its native web-search tool. The final deadline-stopped Opus evaluation lacks that aggregate, so its retained streamed usage and the resulting total are lower bounds, marked with a star. Figure 3 describes model generation, including reported search-related model usage; it does not measure external CPU computation or search-service work.

Measure	Astra high	Luna high	Opus 5.5 high
Input tokens	34,575,392	98,019,732	157,011,011*
Cached input tokens (included above)	31,453,056	91,690,496	149,741,729*
Output tokens	445,123	866,731	1,702,589*
Reasoning tokens (included above)	181,416	425,435	668,352*

Table 24: Reported token usage at high effort with code and web access. Stars indicate totals with some unreported usage; actual consumption may be higher.

H.4 Individual instances

Parameters and reference types are listed in Appendix F; the release includes every submitted object and its verification result.

Table 25: Per-instance relative quality for GPT-6 Astra, GPT-5.6 Luna and Claude Opus 5.5 at high effort. P: published frontier; C: construction baseline. A dagger marks an evaluation assigned zero under the protocol in Section 3.5.

			Astra high		Luna high		Opus 5.5 high	
Family	Parameters	Ref.	No tools	Tools	No tools	Tools	No tools	Tools
cap set	$d = 9$	P	0.932	1.000	0.739	1.000	0.932	0.946
cap set	$d = 10$	P	0.921	1.000	0.000†	0.921	0.785	0.921
cap set	$d = 11$	P	0.916	0.951	0.654	0.916	0.916	0.997
LABS	$N = 240$	C	0.931	1.568	0.000†	1.306	0.931	1.067
LABS	$N = 280$	C	0.985	1.507	0.000†	1.253	0.000†	1.123
LABS	$N = 281$	C	0.989	1.517	0.000†	1.328	0.000†	1.517
LABS	$N = 357$	C	0.996	1.076	0.000†	0.877	0.966	1.068
LABS	$N = 375$	C	1.022	1.201	0.000†	1.183	0.000†	1.054
Heilbronn (square)	$n = 26$	C	0.774	2.251	0.320	2.413	0.272	2.887
Heilbronn (square)	$n = 40$	C	0.837	3.052	0.096	1.801	0.463	1.793
Heilbronn (square)	$n = 46$	C	1.193	3.018	0.089	3.232	0.497	2.066
Heilbronn (square)	$n = 48$	C	1.173	1.952	0.130	0.207	0.491	4.470
Heilbronn (square)	$n = 50$	C	0.793	2.930	0.238	4.572	0.685	5.950
Heilbronn (triangle)	$n = 26$, T_5	C	0.224	1.234	0.149	1.159	0.120	1.252
Heilbronn (triangle)	$n = 31$, T_3	C	0.468	1.903	0.115	2.066	0.454	1.630
Heilbronn (triangle)	$n = 32$, T_1	C	0.322	1.623	0.180	2.137	0.456	1.993
Heilbronn (triangle)	$n = 34$, T_2	C	0.425	1.707	0.250	2.246	0.535	2.847
Heilbronn (triangle)	$n = 40$, T_4	C	0.635	1.711	0.398	2.864	0.590	4.475
spherical codes	$d = 13$	P	0.924	1.000	0.270	1.000	0.827	1.000
spherical codes	$d = 14$	P	0.652	1.000	0.188	1.000	0.768	1.000
spherical codes	$d = 15$	P	0.667	1.000	0.164	1.000	0.913	1.000
spherical codes	$c = 24/100,\ d = 12$	C	1.964	2.786	0.857	2.750	0.000†	2.571
spherical codes	$c = 36/100,\ d = 12$	C	1.105	1.276	0.368	1.105	1.105	1.230
spherical codes	$c = 39/100,\ d = 12$	C	1.184	1.533	0.421	1.000	1.105	1.520
spherical codes	$c = 36/100,\ d = 14$	C	1.806	2.113	0.000†	1.806	0.000†	2.766
spherical codes	$c = 39/100,\ d = 14$	C	2.197	2.432	0.000†	1.697	2.697	3.182
corners	$n = 138$	C	1.058	1.078	1.000	1.058	1.000	1.078
corners	$n = 150$	C	1.019	1.083	1.000	1.053	1.019	1.097
corners	$n = 162$	C	1.052	1.117	1.000	1.078	1.019	1.121
corners	$n = 183$	C	1.000	1.138	0.838	1.113	1.008	1.145
corners	$n = 185$	C	1.000	1.127	0.966	1.090	1.000	1.142
matrix multiplication	$m = 4,\ n = 4,\ p = 5$	P	0.938	1.000	0.000†	0.762	0.938	1.000
matrix multiplication	$m = 5,\ n = 4,\ p = 5$	P	0.894	1.000	0.000†	0.894	0.000†	1.000
matrix multiplication	$m = 5,\ n = 5,\ p = 5$	P	0.853	1.000	0.000†	1.000	0.845	1.000
degree–diameter	$d = 3,\ k = 6$	P	0.955	1.000	0.333	1.000	0.485	1.000
degree–diameter	$d = 3,\ k = 7$	P	0.857	1.000	0.622	1.000	0.551	1.000
degree–diameter	$d = 4,\ k = 4$	P	0.769	1.000	0.000†	1.000	0.548	0.952
degree–diameter	$d = 4,\ k = 5$	P	0.558	1.000	0.000†	1.000	0.558	1.000
degree–diameter	$d = 5,\ k = 4$	P	0.802	1.000	0.000†	1.000	0.302	0.991
degree–diameter	$d = 6,\ k = 3$	P	0.730	1.000	0.559	1.000	0.495	1.000
degree–diameter	$d = 7,\ k = 3$	P	0.595	1.000	0.286	1.000	0.393	1.000
linear-equation-free	$a = 3,\ b = 4,\ n = 6248$	C	1.032	1.338	0.135	1.020	0.000†	1.433
linear-equation-free	$a = 3,\ b = 4,\ n = 13312$	C	0.766	1.445	0.822	0.000†	1.199	1.248
linear-equation-free	$a = 2,\ b = 3,\ n = 17965$	C	0.717	1.421	0.478	1.036	0.964	1.423
linear-equation-free	$a = 2,\ b = 5,\ n = 18883$	C	1.576	1.576	0.841	0.931	1.576	1.603

			Astra high		Luna high		Opus 5.5 high	
Family	Parameters	Ref.	No tools	Tools	No tools	Tools	No tools	Tools
linear-equation-free	$a = 3,\ b = 4,\ n = 19044$	C	0.778	1.603	0.544	0.000†	1.117	1.230
$\mathbb{F}_q^n$ AP-free	$n = 4,\ q = 7$	C	1.417	1.505	1.427	1.155	0.971	1.699
$\mathbb{F}_q^n$ AP-free	$n = 5,\ q = 5$	P	0.804	1.000	0.644	1.000	0.773	0.995
$\mathbb{F}_q^n$ AP-free	$n = 6,\ q = 5$	P	0.963	1.000	0.000†	1.000	0.963	1.000
MOLS	$n = 10$	P	1.000	1.000	0.500	1.000	1.000	1.000
MOLS	$n = 12$	P	0.400	1.000	0.400	1.000	0.400	1.000
MOLS	$n = 14$	P	0.250	1.000	0.250	1.000	0.500	1.000
MOLS	$n = 15$	P	0.500	1.000	0.500	1.000	0.500	1.000
MOLS	$n = 18$	P	0.400	1.000	0.200	1.000	0.400	1.000
MOLS	$n = 20$	P	0.750	1.000	0.750	0.750	0.750	0.750
Schur	$k = 6$	P	0.746	1.000	0.000†	1.000	0.000†	1.000
Schur	$k = 7$	P	0.708	1.000	0.174	1.000	0.000†	1.000
Schur	$k = 8$	P	0.739	1.000	0.000†	1.000	0.739	1.000
covering	$k = 6,\ t = 3,\ v = 24$	P	0.991	1.000	0.333	0.800	0.935	1.000
covering	$k = 6,\ t = 3,\ v = 30$	P	0.818	1.000	0.000†	0.784	0.000†	1.000
covering	$k = 7,\ t = 4,\ v = 18$	P	0.840	1.000	0.000†	1.000	0.000†	1.000
Shannon	$d = 24,\ q = 7$	C	0.515	1.009	0.515	0.991	0.713	1.009
Shannon	$d = 40,\ q = 7$	C	0.582	1.000	0.304	1.000	0.000†	1.000
Shannon	$d = 48,\ q = 9$	C	1.000	1.000	1.000	1.000	1.000	1.000
Shannon	$d = 64,\ q = 9$	C	1.000	1.000	1.000	1.000	1.000	1.000
trifference	$m = 1,\ n = 64$	C	3.000	9.000	0.000†	1.000	3.000	27.000
trifference	$m = 1,\ n = 96$	C	1.000	1.000	0.000†	1.000	1.000	1.000
trifference	$m = 1,\ n = 144$	C	1.000	1.000	0.000†	0.000†	1.000	1.000
trifference	$m = 1,\ n = 192$	C	1.000	1.000	1.000	1.000	1.000	1.000