

# AN INVESTIGATION OF ROBUSTNESS OF LLMs IN MATHEMATICAL REASONING: BENCHMARKING WITH MATHEMATICALLY-EQUIVALENT TRANSFORMATION OF ADVANCED MATHEMATICAL PROBLEMS

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

In this paper, we introduce a systematic framework beyond conventional methods to assess LLMs’ mathematical-reasoning robustness by stress-testing them on advanced math problems that are mathematically equivalent but with linguistic and parametric variation. These transformations allow us to measure the sensitivity of LLMs to non-mathematical perturbations, thereby enabling a more accurate evaluation of their mathematical reasoning capabilities. Using this new evaluation methodology, we created PutnamGAP, a new benchmark dataset with multiple mathematically-equivalent variations of competition-level math problems. With the new dataset, we evaluate multiple families of representative LLMs and examine their robustness. Across 18 commercial and open-source models we observe sharp performance degradation on the variants. OpenAI’s flagship reasoning model, O3, scores 51.5 % on the originals but drops by 4.7 percentage points on surface-renaming variants, and by 12.9 percentage points on parametric variants, while smaller models fare far worse. Overall, the results show that the proposed new evaluation methodology is effective for deepening our understanding of the robustness of LLMs and generating new insights for further improving their mathematical reasoning capabilities.

## 1 Introduction

**Motivation.** Modern AI systems are increasingly entrusted with tasks that hinge on robust reasoning rather than pattern matching. It is thus important to precisely measure an LLM’s reasoning capacity and its ability to generalize beyond memorized textual surface forms. Existing math-reasoning benchmarks, however, exhibit two critical weaknesses: (i) leakage-induced score inflation, since benchmark items rapidly seep into pre-training corpora, and (ii) limited robustness coverage, because today’s datasets are too small or lack controlled transformations that probe true generalization. Addressing these weaknesses is urgent if we aim to benchmark reasoning with the same rigor demanded in safety-critical domains such as healthcare or cybersecurity.

**Benchmark inflation through training leakage.** Recent studies show that public datasets, including GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021), have leaked into the web-scale corpora used to pre-train large language models (LLMs), artificially inflating test-time accuracy. A leaderboard score therefore no longer guarantees genuine reasoning ability; it may merely reflect memorization of benchmark items or their solutions. Simply releasing *yet another* dataset postpones the problem: once its items enter future training corpora, scores climb without real progress. What is needed is a *systematic method* that (i) measures a model’s capacity to generalize beyond verbatim memory and (ii) can generate an unbounded supply of evaluation items, limiting future leakage.

**Competition mathematics reveals the next robustness bottleneck.** Large language models (LLMs) now surpass 90% accuracy on widely-used benchmarks such as GSM8K and MATH, prompting claims of “near-human” numerical reasoning yet still falter on Olympiad-style or Putnam-level problems that intertwine multiple domains. Existing Putnam-derived datasets are too small to expose this gap: PUTNAM-AXIOM (236 originals + 52 variations) (Huang et al., 2025),

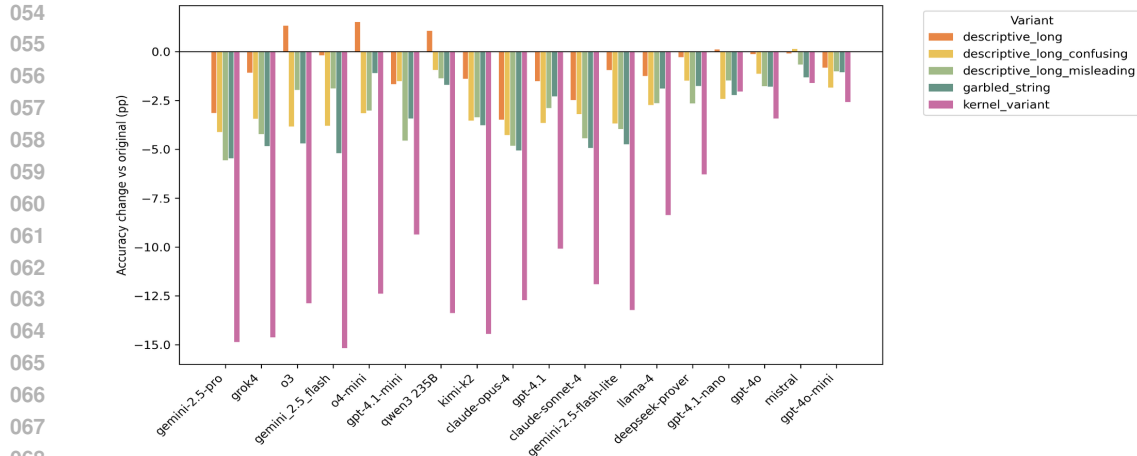


Figure 1: PutnamGAP variants performance relative to the original set

and PUTNAMBENCH (640 formalized theorems) (Tsoukalas et al., 2024) remain in the hundreds, and none delivers systematic generalization and perturbations. These facts expose Weakness (i) insufficient scale and Weakness (ii) lack of controlled, systematic transformations in existing evaluations.

**Existing perturbation-based robustness benchmarks.** Recent work has begun to probe mathematical robustness by constructing perturbation-based benchmarks on top of GSM8K and related datasets. GSM-Plus augments GSM8K with eight families of adversarial variations per problem, revealing large accuracy drops even for models that nearly solve the original benchmark (Li et al., 2024). GSM-Symbolic builds symbolic templates over GSM8K-style problems and shows that merely changing numeric instantiations or adding logically irrelevant clauses can degrade performance by up to 65% (Mirzadeh et al., 2024). MathCheck-GSM further organizes GSM8K-derived problems into a checklist of task and robustness variants to study behavior across multiple evaluation formats (Zhou et al., 2024). Beyond GSM8K, GSM8K\_MORE uses an ontology of perturbations to generate families of grade-school arithmetic variants (Hong et al., 2025), while Putnam-AXIOM introduces a smaller set of functional variations for university-level Putnam problems (Gulati et al., 2025). These efforts convincingly demonstrate that current LLMs are brittle under controlled perturbations; however, GSM-derived benchmarks remain confined to grade-school or pre-university word problems with short, single-answer numerical solutions and are built directly on GSM8K and related datasets that are already near-saturated and affected by training data contamination for frontier models (Cobbe et al., 2021; Gulati et al., 2025; Shalyt et al., 2025; Glazer et al., 2024), while Putnam-AXIOM introduces only a relatively small companion set of functional variants (100 over 522 problems) (Gulati et al., 2025). Consequently, the existing perturbation benchmarks do not yet provide a large-scale, systematically structured robustness test for competition-level, proof-style mathematics.

### Generalization-and-Perturbation (GAP) framework for robustness evaluation.

We address both leakage and robustness by *stress-testing the model on mathematically equivalent versions of the same problem*. For a problem  $x$  with solution set  $S(x)$  and an LLM  $f$ , robustness is the expected accuracy when  $x$  is transformed by a family  $\mathcal{T}$  of equivalence-preserving operators. We partition  $\mathcal{T}$  into  $\mathcal{T}_{\text{surf}}$  (surface renames that alter symbol salience) and  $\mathcal{T}_{\text{para}}$  (kernel rewrites that preserve the same proof steps while changing the scenario and parameters). This GAP framework (i) creates an *infinite* stream of *unseen* test items, mitigating future contamination, and (ii) quantifies how far a model can generalize beyond memorized surface forms. In our setting, GAP serves as a general diagnostic evaluation methodology for analyzing and quantifying the robustness of an LLM’s mathematical reasoning capacity at the level of competition problems.

**Limitations of existing perturbation benchmarks.** Several recent robustness benchmarks - such as GSM-Symbolic, GSM-Plus, and MathCheck -

**PutnamGAP: instantiating GAP on 85 years of problems.** We instantiate GAP on every William Lowell Putnam Competition problem from 1938–2024 (**1,051** originals) and expand each item into

108 five variants—four surface renames and one kernel rewrite—obtaining **6,306** stress-test questions.  
 109 A two-stage QA pass—15 rounds of O3 self-review plus a 10% spot-check found no substantive  
 110 errors.

111 **Headline results.** Across 18 models, as shown figure 4, all of them suffer from both simple re-  
 112 naming and step-based rewrites. OpenAI’s O3 scores 51.5% on original statements but loses **4.7**  
 113 **pp (9.12%)** under surface renames and **12.9 pp (25.22%)** under parametric rewrites. These drops  
 114 confirm that high leaderboard scores can collapse when cosmetic or structural perturbations are  
 115 applied—precisely the effect that data leakage masks.

116 **Contributions.** (1) We propose *GAP*, a novel general framework for measuring robustness via  
 117 mathematically equivalent transformations that overcomes two common deficiencies of the current  
 118 evaluation methods (i.e., data leakage and lack of robustness measures). (2) We release *PutnamGAP*,  
 119 the first 6k-scale competition benchmark that systematically disentangles surface-level and structural  
 120 generalization while limiting future leakage. (3) We provide the first comprehensive robustness  
 121 baseline across eighteen LLMs, plus an open-source evaluation stack.

## 123 2 The Generalization-and-Perturbation (GAP) Framework

### 124 2.1 Evaluation Model

125 We start from a curated set of  $N$  *canonical items*  $\mathcal{P} = \{(x_i, y_i, \pi_i)\}_{i=1}^N$ , where  $x_i$  is a problem  
 126 statement,  $y_i$  is its reference answer(s), and  $\pi_i$  an unreleased expert solution path used internally for  
 127 safe variant generation. **Model interface.** A language model  $f_\theta$  receives a prompt  $x$  and returns  
 128  $\hat{y} = f_\theta(x)$ , which an automatic checker maps to a binary label  $z = \text{grade}(\hat{y}, y) \in \{0, 1\}$ .

129 **Variant families.** For every  $x_i$  we later apply *two* disjoint transformation super-families (defined  
 130 in the next section but *left unchanged here*):  $\mathcal{T}_i^{\text{surf}}$  ( $K_{\text{surf}}$  surface variants),  $\mathcal{T}_i^{\text{para}}$  ( $K_{\text{para}}$  parametric  
 131 variants). Each surface transformation  $\tau$  returns a new statement  $x_i^{(\tau)} = \tau(x_i)$  that preserves  
 132 semantic correctness of  $y_i$ . For parametric variations,  $y_i$  is transformed as well to match  $\tau(x_i)$ .

133 **Evaluation matrix.** The Cartesian product  $\mathcal{D} = \{(i, \tau) \mid i \leq N, \tau \in \mathcal{T}_i^{\text{surf}} \cup \mathcal{T}_i^{\text{para}} \cup \{\text{id}\}\}$  contains  
 134  $N \times (K + 1)$  aligned items (original +  $K$  variants per source,  $K = K_{\text{surf}} + K_{\text{para}}$ ). Running  $f_\theta$  on  
 135 every pair populates a binary matrix  $\mathbf{Z} \in \{0, 1\}^{N \times (K+1)}$ . From the first column we extract the *easy*  
 136 vector  $\mathbf{e}(\theta) \in \{0, 1\}^N$ , while the remaining columns feed family-specific aggregates:  $\mathbf{h}^{\text{surf}}(\theta) =$   
 137  $\text{maj}(\mathbf{Z}_{[:, \text{surf}]})$ ,  $\mathbf{h}^{\text{para}}(\theta) = \mathbf{Z}_{[:, \text{para}]}$ . The set of surface variants can be changed based on specific  
 138 tasks.

139 **Robustness Metric.** Let  $e, h \in \{0, 1\}^N$  denote per-item correctness on the *easy* (original) and *hard*  
 140 (variant) sets. With Jeffreys smoothing

$$141 p_e = \frac{\sum_j e_j + \frac{1}{2}}{N + 1}, \quad p_h = \frac{\sum_j h_j + \frac{1}{2}}{N + 1}, \quad \sigma = \sqrt{\frac{1}{2}(p_e(1 - p_e) + p_h(1 - p_h))}.$$

142 Define the SD-normalized drop  $d_j = (e_j - h_j)/\sigma$  and its soft-saturated version  $\hat{d}_j = \frac{1}{k} \log(1 + e^{kd_j})$   
 143 with  $k \approx 0.5$ . Let  $\tilde{d} = \text{median}\{d_j \mid d_j > 0\}$  (with fallback  $\tilde{d} := \max(\varepsilon, \text{median}|d_j|)$ ,  $\varepsilon = 0.1$   
 144 when no positive drop exists) and set  $\beta = \ln 2/\tilde{d}$ . Our *penalty* robustness is

$$145 \hat{R}(e, h) = \frac{1}{N} \sum_{j=1}^N \exp(-\beta \hat{d}_j) \in (0, 1].$$

146 Thus  $\hat{R} = 1$  indicates invariance; a “typical” loss ( $\hat{d}_j \approx \tilde{d}$ ) halves the per-item factor, while im-  
 147 provements ( $d_j < 0$ ) are clamped to zero penalty (no reward). We report  $R_{\text{surf}} = \hat{R}(e, h_{\text{surf}})$ ,  
 148  $R_{\text{para}} = \hat{R}(e, h_{\text{para}})$ , and  $R_{\text{global}} = \sqrt{R_{\text{surf}} R_{\text{para}}}$ . **Full derivation, statistical justification, and**  
 149 **design discussion are in Appendix B.**

### 150 2.2 Transformation Families

151 **The proposed general robustness measures can work for any variations.** As a first step in ex-  
 152 ploring this new evaluation methodology, we propose and study *five* aligned variants— four *surface*  
 153

renamings that perturb only symbol names, and one *core-step* instance that perturbs numeric slots while preserving the reasoning chain. This section details the synthesis pipelines. Detailed descriptions can also be found in Appendix A.

### 2.2.1 Surface renaming variant family

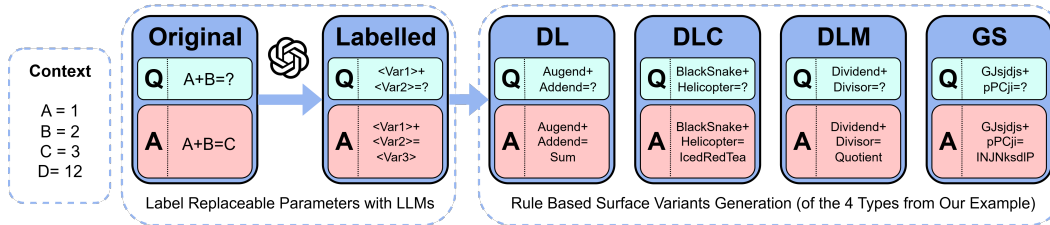


Figure 2: Surface renaming variant family pipeline

We want to know whether a model recognizes an argument *because it has truly abstracted the pattern* or merely because it memorizes suggestive identifier strings. Therefore we systematically replace each token tagged `var` or `param`; all constants of category `sci_const` remain untouched.

#### Automated pipeline.

- 1. Proposal.** A single call to O3 receives the token role (“free variable” or “fixed parameter”) and the surrounding textual context, and returns a candidate replacement.
- 2. Collision check.** A deterministic post-validator rejects names colliding with any pre-existing identifier in the problem.
- 3. Family tagging.** The string is labelled as belonging to one of four families described below.

We use four types of surface variants: `Descriptive_Long` (DL), with a single descriptive phrase; `Descriptive_Long_Confusing` (DLC), with 2–5 random unrelated nouns; `Descriptive_Long_Misleading` (DLM), with a mathematically suggestive but misleading term; `Garbled_String` (GS), with a 4–16-character hash, as shown in figure 2 where ‘Q’ stands for the problem question and ‘A’ stands for the official solution.

Each source item thus yields 4 surface variants; accuracy deltas per family appear in Section Results & Analysis.

### 2.2.2 Parametric variant family

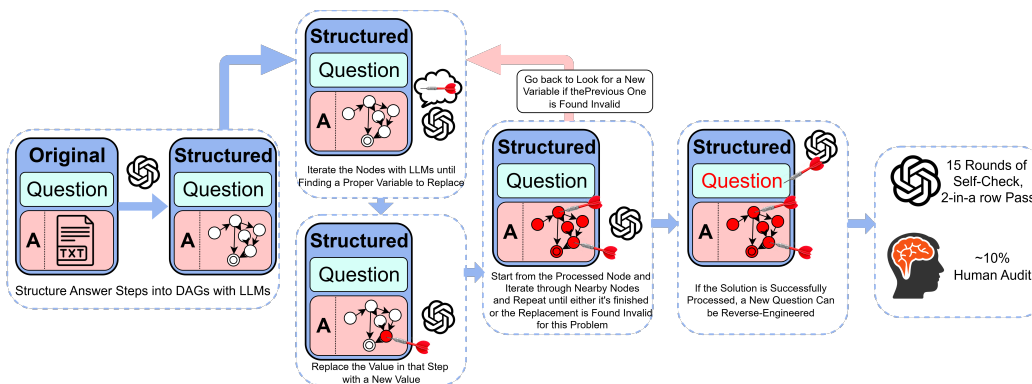


Figure 3: Parametric variant family pipeline

Symbol renaming probes only the lexical axis. To probe *structural transfer*, we resample numerical constants yet force the solution to reuse the original high-level moves. In this work, we call it `Kernel_Variant` (KV). We convert each item into semantically-equivalent variants through a four-stage pipeline: (1) **slot discovery**; (2) **template back-synthesis**; (3) **question reverse-engineering**; and (4) **dual-verifier screening** (two-in-a-row rule). The pipeline generates a bounded

number of validated variants for each problem within a few hours on commodity hardware using the OpenAI o3 API. See Appendix A for empirical bounds and details of our implementation.

## 2.3 Implementation Overview

**Code release.** To facilitate double-blind reviewing we publish *only* the subset of data (100 randomly chosen examples). An automated evaluator, `putnam-cli.py`, receives the names of target solver model and grader model and variant type to test. Supported back-ends are (i) any HuggingFace-compatible checkpoint via `transformers`, (ii) a local `vllm` server, or (iii) API clients including OpenAI, Gemini, Anthropic and OpenRouter. Full data and generation scripts will be released post-decision.

**Surface generation.** Renaming variants are produced on a CPU-only node by streaming o3 API calls. A five-stage *exponential-back-off* retry (max 5 attempts, doubling timeout each time) masks transient API latency. Processing all 1 051 items in parallel takes  $\sim 15$  min wall-clock.

**Core-step generation.** Kernel variant synthesis is more expensive because of multi-turn chain-of-thought reasoning: end-to-end runtime is  $\leq 3$  h for the full corpus on a single 8-core CPU, dominated by the 15-iteration repair-and-verify loop.

## 3 PutnamGAP Dataset

### 3.1 Data Sources, Extraction & Annotation

Our benchmark comprises all **Putnam Problems 1938–2024** ( $N = 1\,051$  items after deduplication). See Appendix E for source details.

Original scans are processed via a 3-stage OCR routine: (i) Manual segmentation for every question-answer pair. (ii) *MathPix* for formula-aware PDF-to-LaTeX conversion followed by (iii) custom post-filters that merge multi-line expressions and fix 4.2 % residual symbol errors. Each item is manually spot-checked ( $\leq 2$  min per problem) to ensure semantic fidelity before variant generation.

**Complete corpus list, OCR accuracy study, and cleaning scripts appear in Appendix E.**

### 3.2 Dataset Statistics

**Overall scale and balance.** The benchmark comprises **1,051** original Putnam problems from 1938–2024 and five mathematically equivalent transformations, yielding **6,306** items. Part distribution is balanced (**527 A** vs. **524 B**), and the canonical identifier  $\langle year, part\{A, B\}, index \rangle$  provides a difficulty proxy. Using indices 1–2 as *Easy*, 3–4 as *Medium*, and 5–6 as *Hard*, the corpus contains 32.3 % Easy, 32.3 % Medium, 32.2 % Hard, plus a 3.0 % extra-hard tail (indices 7–8).

**Topic coverage and Quality Control** Automatic tags in `_meta.tag` indicate broad mathematical coverage—Algebra (641), Analysis (521), Number Theory (392), Combinatorics (286), and Geometry (239). 803 of the questions are proofs, and 248 of them are calculations. At the same time, every item has undergone single-pass manual validation.

## 4 Experimental Setup

The constructed PutnamGAP dataset enables, for the first time, a robust analysis of an LLM’s reasoning capacity. In this section, we describe how we set up the experiments to evaluate the robustness of 18 representative models.

### 4.1 Model Pool & Prompting

We evaluated 18 models (see 1 or Appendix A for a complete list). All models are queried under a unified **zero-shot template**. A system instruction designates the model as “*an expert mathematician*” and asks it to *show all work*, while the user message embeds the problem. See Appendix G for our full prompt. We fix `temperature=0`, `top_p=1`, and `max_tokens=32000` or maximum token amount available in case some models have `max_tokens` maximum smaller than 32000. for every run except OpenAI O-series which require `temperature=1`. Solutions are then re-submitted to a second template that grades the answer: a STRICT PROOF RUBRIC for proof items and a LENIENT NUMERIC RUBRIC for calculation items. Both grader prompts require structured

**Table 1: Model Accuracy Rates across Categories (Percent Scale)**

Model	DL ( $\Delta$ )	DLC( $\Delta$ )	DLM ( $\Delta$ )	GS ( $\Delta$ )	Kernel Variant ( $\Delta$ )
claude-opus-4	23.0** (-3.5)	22.2*** (-4.3)	21.7*** (-4.8)	21.4*** (-5.1)	13.8*** (-12.7)
claude-sonnet-4	20.6** (-2.5)	19.8*** (-3.2)	18.6*** (-4.4)	18.1*** (-4.9)	11.1*** (-11.9)
deepseek-prover	15.2 (-0.3)	14.0 (-1.5)	12.8** (-2.7)	13.7* (-1.8)	9.2 *** (-6.3)
gemini-2.5-flash-lite	18.8 (-0.9)	16.1*** (-3.7)	15.8*** (-4.0)	15.1*** (-4.7)	6.6 *** (-13.2)
gemini-2.5-pro	75.2** (-3.1)	74.3*** (-4.1)	72.8*** (-5.6)	72.9*** (-5.4)	63.5*** (-14.9)
gemini-2.5-flash	42.6 (-0.2)	39.0*** (-3.8)	40.9 (-1.9)	37.6*** (-5.2)	27.6*** (-15.2)
gpt-4.1	23.4 (-1.5)	21.2** (-3.7)	22.0* (-2.9)	22.6 (-2.3)	14.8*** (-10.1)
gpt-4.1-mini	26.9* (-1.7)	27.1 (-1.5)	24.0*** (-4.6)	25.1** (-3.4)	19.2*** (-9.4)
gpt-4.1-nano	8.9 (+0.1)	6.4 ** (-2.4)	7.3 * (-1.5)	6.6 ** (-2.2)	6.8 (-2.0)
gpt-4o	6.3 (-0.1)	5.3 ** (-1.1)	4.7 ** (-1.8)	4.7 *** (-1.8)	3.0 *** (-3.4)
gpt-4o-mini	3.5 (-0.8)	2.5 *** (-1.8)	3.3 (-1.0)	3.2 (-1.1)	1.7 *** (-2.6)
grok4	59.0 (-1.1)	56.6 (-3.4)	55.9*** (-4.2)	55.2*** (-4.8)	45.5*** (-14.6)
kimi-k2	25.8 (-1.4)	23.7** (-3.5)	23.8** (-3.4)	23.4*** (-3.8)	12.8*** (-14.4)
llama-4	14.5 (-1.2)	13.0** (-2.7)	13.1** (-2.6)	13.8* (-1.9)	7.3 *** (-8.4)
mistral	5.5 (-0.1)	5.7 (+0.1)	4.9 (-0.7)	4.2 * (-1.3)	3.9 * (-1.6)
o3	52.8 (+1.3)	47.6** (-3.8)	49.5 (-2.0)	46.8*** (-4.7)	38.6*** (-12.9)
o4-mini	43.0 (+1.5)	38.3** (-3.2)	38.5 (-3.0)	40.4 (-1.1)	29.1*** (-12.4)
qwen3	29.3 (+1.1)	27.3 (-0.9)	26.9 (-1.4)	26.5 (-1.7)	14.9*** (-13.4)

Note: \*\*\* $p < 0.01$ , \*\* $p < 0.05$ , \* $p < 0.1$

JSON output containing a binary `grade` field plus detailed feedback. Complete prompt code is available in Appendix G

## 4.2 Scoring & Auto-Grader

We partition tasks into *computation* and *proof* categories and evaluate them with distinct graders.

**Computation** Each candidate answer is normalized (whitespace, units, LaTeX macros) and passed to two scoring paths: (i) a strict string match against the reference solution; (ii) a *latent* grader—an LLM prompted to return `''CORRECT''` or `''INCORRECT''` given the reference answer and a rubric that disallows partial credit. We adopt path (ii) to mitigate formatting artifacts; if the two paths disagree we mark the item for manual audit ( $\approx 1\%$  of cases).

**Proof** We provide the grader with an aligned, step-by-step reference proof and ask it to assign a binary `grade` plus a natural-language justification. Any skipped logical step or missing citation triggers a fail. A random 10 % sample is double-checked by independent volunteers; grader precision/recall is  $>97\%$ .

## 5 Results & Analysis

### 5.1 Robustness

We evaluated 18 different LLMs on this benchmark, and results are summarized in Table 1. For each variation of the model, we used a paired design (McNemar’s exact test) on matched problem pairs to test whether the accuracy rate decreases significantly compared to the original. Statistically significant differences are indicated using standard notation ( $p < 0.1$ ,  $p < 0.05$ ,  $p < 0.01$ ). We also computed 95 % CI (See Appendix D Figure 4 ) and our proposed robustness metrics  $R$  (see Appendix B), and all models, especially those performed well on the original set.

We observe that almost all variants lead to a decrease in model accuracy, even when the transformation is merely changing the names of the variables. This indicates a notable lack of robustness: models often lack the capability to preserve their accuracy under mathematically identical but surface-modified representations. Particularly, transformations that rely on variable-name reasoning (such as Misleading or Garbled String) tend to disturb the model’s math accuracy most severely.

**Table 2:** Robustness metrics  $R_{\text{surf}}$ ,  $R_{\text{para}}$ ,  $R_{\text{global}}$  (rounded to three decimals).

Model	$R_{\text{surf}}$	$R_{\text{para}}$	$R_{\text{global}}$	Model	$R_{\text{surf}}$	$R_{\text{para}}$	$R_{\text{global}}$
claude-opus-4	0.958	0.949	0.954	gpt-4o	0.986	0.980	0.983
claude-sonnet-4	0.961	0.942	0.951	gpt-4o-mini	0.990	0.986	0.988
deepseek-prover	0.972	0.960	0.966	grok4	0.937	0.916	0.927
gemini-2.5-flash-lite	0.961	0.942	0.952	kimi-k2	0.955	0.930	0.942
gemini-2.5-pro	0.949	0.915	0.932	llama-4	0.972	0.955	0.963
gemini_2.5_flash	0.952	0.918	0.934	mistral	0.984	0.982	0.983
gpt-4.1	0.963	0.944	0.954	o3	0.940	0.921	0.930
gpt-4.1-mini	0.953	0.939	0.946	o4-mini	0.946	0.929	0.937
gpt-4.1-nano	0.980	0.982	0.981	qwen3	0.941	0.928	0.934

Because the surface score aggregates the four renaming variants by per-item majority, the flip probability from the original to the aggregated surface set is suppressed; accordingly,  $R \approx 1$  is expected and should be interpreted as an approximate upper bound on surface invariance (see Table 2). Practitioners can implement alternative mapping functions based on their model’s performance while retaining this core formulation. Across capable models we consistently observe  $R_{\text{para}} < R_{\text{surf}}$ , and we summarize stress-type invariance via  $R_{\text{global}} = \sqrt{R_{\text{surf}}R_{\text{para}}}$ . Interpreting  $1 - R$  as a penalty mass highlights nontrivial fragility even when raw accuracy is high. Conversely, for weak models a high  $R$  is not evidence of robustness: when base accuracy  $p_e$  is small, the pooled SD  $\sigma = \sqrt{\frac{1}{2}(p_e(1 - p_e) + p_h(1 - p_h))}$  and the bound  $1 - R \leq \min\{p_e, 1 - p_h\}(1 - q)$  with  $q = \exp(-\beta d_{b+})$  limit the observable penalty, so  $R \rightarrow 1$  reflects low headroom rather than invariance. Reporting both accuracy and  $\{R_{\text{surf}}, R_{\text{para}}, R_{\text{global}}\}$  therefore stabilizes cross-model comparison under mathematically equivalent stress and shows that robustness remains limited despite strong performance on canonical phrasing.

Another observation is that if a model is not robust to one variant, it tends to be not robust to other variants as well. Notable examples include kimi-k2, claude-opus-4, and gemini-2.5-pro.

## 5.2 Transformation-wise Breakdown

**Descriptive Long (DL)** The impact of this transformation is smallest overall: drops are marginal and mostly not significant. Some models such as o3 (+1.3), o4-mini(+1.5), and Qwen3-235B (+1.1) even improved slightly. This indicates that descriptive renaming preserving accuracy.

**Confusing (DLC)** Long, semantically meaningless variable names moderately reduce accuracy. Models like Claude-opus-4 (−4.3\*\*\*) and GPT-4o-mini (−1.8\*\*\*) showed significant drops.

**Misleading (DLM)** Replacing variables with misleading strings strongly hurts math accuracy. Nearly all models experienced a significant drop. Notably, Claude-Opus-4 (−4.8\*\*\*), Gemini-2.5-pro (−5.6\*\*\*), and Claude-Sonnet-4 (−4.4\*\*\*) were among the most heavily affected.

**Garbled String (GS)** Random character strings consistently degrade performance: every model loses accuracy, over half significantly. Models such as Gemini-2.5-pro (−5.4\*\*\*), Claude-Sonnet-4 (−4.9\*\*\*), and Gemini-2.5-flash-lite (−4.7\*\*\*) suffered the largest declines.

**Kernel Variant (KV)** Kernel variants—which keep each question’s mathematical structure but replace constants and expressions with different values—led to the sharpest decline overall. All models experienced large drops, often in the range of −5 to −15 points, with Grok4 (−14.6\*\*\*), Gemini-2.5-flash (−15.2\*\*\*), and Gemini-2.5-pro (−14.9\*\*\*) showing the steepest declines.

Overall, state-of-the-art LLMs show inconsistent performance under semantics-preserving transformations and appear sensitive to superficial cues. This is consistent with the possibility that part of their gains reflects data-leakage-related memorization rather than stable mathematical reasoning. The pattern persists across topics and problem classes: bar plots with 95% CIs (Appendix D, fig. 4) and per-topic/per-class breakdowns (Appendix D, figs. 7-8) show similar robustness gaps across Algebra/Analysis/NT/Combinatorics/Geometry and for both proof and calculation items.

## 5.3 Error Taxonomy

Our grading script returns a brief comment for every incorrect answer. Using these comments, we grouped errors into four categories: *Symbol Confusion*, *Step Omission*, *Arithmetic*, and *Logic*

*Hallucination.* Figure 5 in Appendix D shows that the relative frequency of these error types is nearly identical across variants; logic hallucinations dominate, accounting for roughly three-fifths of all wrong answers regardless of prompt wording. Thus, the accuracy drop is distributed across all categories rather than driven by a single one, confirming that mathematically equivalent perturbation consistently degrades LLM performance.

#### 5.4 Qualitative case studies of Kernel Variant failures

To complement the aggregate robustness metrics, we performed a small-scale qualitative analysis of Kernel Variant (KV) failures. We ran a GPT-based analyzer over model traces and automatically selected ORIGINAL/KV pairs where a strong model solves the ORIGINAL correctly but fails on the KERNEL-VARIANT; concrete case studies are deferred to Appendix I.

Across these examples we see three recurring KV-specific failure modes. First, *hallucinated algebraic infrastructure and missing premises*: in items such as 1938-B-1 and 1940-A-6 the KV solutions invoke strong algebraic identities or valuation equalities (e.g.,  $\text{adj } M = (\det M)M^{-1}$  or  $v_i(JF) = e_i - 1$ ) without checking that the hypotheses hold in the stated ring or characteristic, whereas the ORIGINAL proofs stay within a valid algebraic framework. Second, *computing the wrong global quantity after mostly correct setup*: in 1939-A-1, 1940-A-7, and 1940-B-7 the KV traces correctly identify the relevant points or bounds but then switch from arc length to chord length or from a clean monotonicity argument to a mis-indexed summation, producing false inequalities despite reasonable intermediate calculus or algebra. Third, *fragile geometric reductions and inconsistent conventions*: in 1939-B-1, 1939-B-7, 1940-A-2, and 1938-A-7 the KV arguments rely on incorrect symmetry reductions, ignore degenerate edge cases (e.g.  $\rho = 0$ ), or briefly adopt sign conventions that contradict earlier definitions before silently reverting.

Overall, these qualitative patterns corroborate the quantitative gap  $R_{\text{para}} < R_{\text{surf}}$ . Kernel Variants do not merely inject harder arithmetic; they stress the model’s ability to re-bind parameters and maintain a coherent proof skeleton under resampled slots. When the model fails KV, it often does so by reusing an ORIGINAL template outside its domain of validity or by quietly changing the quantity or symmetry being computed (see Appendix I for detailed traces).

#### 5.5 External Validation

We applied our surface-renaming protocols—**DLC** and **GS**—to ALG514 (Kushman et al., 2014). Accuracy decreased from Base 93.6% to DLC 90.9% ( $\Delta = -2.7$  pp) and GS 89.3% ( $\Delta = -4.3$  pp); McNemar tests (Base vs DLC:  $b=24, c=10, p=0.024$ ; Base vs GS:  $b=35, c=13, p=0.002$ ). These statistically significant drops indicate that GAP’s surface-renaming stress tests generalize to other math datasets and reveal nontrivial sensitivity to variable renaming.

## 6 Discussion

### 6.1 Key Findings

The proposed GAP framework allowed us to make the following new findings about the behavior of LLMs in performing mathematical reasoning:

**Symbol-level perturbations cause substantial drops.** Across the four *surface* variants—DL, DLC, DLM, and GS—merely renaming variables lowers accuracy by 3–5 pp on average; for example, GEMINI-2.5-PRO falls from 78.3% to 72.9% (−5.4 pp; see Table 1). This indicates that today’s SOTA models still rely on lexical “semantic anchors” rather than fully abstract proof structures.

**Maintaining structure but resampling parameters is even harsher.** The KERNEL VARIANT (KV) simultaneously resamples all mutable constants while preserving the original reasoning skeleton. Accuracy losses reach  $\approx 10$  pp; OPENAI O3 declines from 51.5% to 38.6% (−12.9 pp), showing that grasping a solution pattern does not automatically translate to parameter-invariant reasoning ability.

**$R_{\text{global}}$  reveals fine-grained brittleness.** We compute  $R_{\text{surf}}, R_{\text{para}}, R_{\text{global}}$  where  $R(\cdot, \cdot)$  is the SD-normalized robustness metric. Because it exponentially penalizes rare but catastrophic flips,  $R_{\text{global}}$  tracks *effective* robustness more faithfully than a plain hard/easy accuracy ratio.

432 *Takeaway.* Across capable models we consistently observe  $R_{\text{para}} < R_{\text{surf}}$ , and we summarize stress-  
 433 type invariance via  $R_{\text{global}} = \sqrt{R_{\text{surf}}R_{\text{para}}}$ ; interpreting  $1 - R$  as penalty mass highlights non-trivial  
 434 fragility even when raw accuracy is high.

## 436 6.2 Implications

437 **A novel evaluation methodology:** The GAP framework provides a novel methodology for ana-  
 438 lyzing and evaluating the robustness of LLMs’ reasoning capacity by generating an (in principle)  
 439 unbounded supply of semantically equivalent test items, which can limit future benchmark leakage  
 440 and mitigate leaderboard inflation.

441 **Improving robustness via curriculum fine-tuning:** Our results suggest curriculum fine-tuning that  
 442 explicitly randomizes (i) symbol identities and (ii) numeric parameters, instead of simply enlarging  
 443 pre-training corpora. That is, we can leverage the GAP framework to augment data for fine-tuning a  
 444 model to improve robustness.

445 **Detecting potential security concerns:** Surface-level fragility implies that production systems can  
 446 be *prompt-injected* with mathematically innocuous renamings—highlighting the need to integrate  
 447 robustness checks into red-team pipelines. Our evaluation framework enables such risk analysis  
 448 before deploying any production system.

449 *Reporting.* We recommend reporting bootstrap CIs for  $R_b$  together with per-item histograms of  
 450 SD-normalized drops  $d_j = (e_j - h_j)/\sigma$ ; these visualize tail-risk (rare catastrophic flips) that raw  
 451 accuracy masks and make robustness audits reproducible.

## 454 7 Related Work

455 There have been multiple benchmarks for evaluating the mathematical-reasoning capabilities of  
 456 large language models (LLMs). Early math-reasoning benchmarks such as MATH(1.25 k prob-  
 457 lems) (Hendrycks et al., 2021), and GSM8K(8.5 k problems) (Cobbe et al., 2021), revealed ba-  
 458 sic arithmetic/algebra skills. But their difficulty is now saturated as LLMs scale. For instance,  
 459 with prompting strategies such as DUP, GPT-4 attains 97.1% accuracy on GSM8K (Zhong et al.,  
 460 2025). This ceiling at the high-school-competition level motivated the creation of a new generation  
 461 of harder benchmarks.

462 Subsequent benchmarks target harder problems. OMNI-MATH contributes 4 428 rigorously an-  
 463 notated Olympiad-level problems (Gao et al., 2024). Likewise, OLYMPIADBENCH provides a  
 464 bilingual, multimodal benchmark of 8 476 Olympiad-level math and physics problems with ex-  
 465 pert step-by-step solutions (He et al., 2024). The cross-disciplinary benchmark ARB consist ques-  
 466 tions in mathematics, physics, biology, chemistry, and law, with a rubric-based self-grading pro-  
 467 tocol (Sawada et al., 2023). Some other benchmarks focuses specifically on formal proof. MINIF2F  
 468 supplies 488 Olympiad-level problems formalized in multiple proof assistants (Zheng et al., 2022).  
 469 PUTNAMBENCH, offers 1 692 rigorously hand-crafted formalizations of Putnam Competition prob-  
 470 lems (Tsoukalas et al., 2024).

471 Nevertheless, recent studies warn that scores on many NLP benchmarks may be artificially inflated  
 472 by data contamination, when LLMs are trained on the benchmark questions. Sainz et al. (2023)  
 473 point out that many benchmarks may be inflated because large language models often memorize  
 474 test data seen during pre-training. Balloccu et al. (2024) conduct a systematic audit of data leakage  
 475 for closed-source LLMs and estimate that roughly 4.7 million test examples from 263 datasets were  
 476 likely exposed to the models.

477 Preventing data leakage is central to obtaining a robust evaluation of LLMs’ reasoning capabilities.  
 478 One approach is to construct entirely original problems: for example, FRONTIERMATH provides  
 479 a rigorously curated benchmark of hundreds of original, expert-level mathematics problems span-  
 480 ning fields from number theory to algebraic geometry (Glazer et al., 2024). Another strategy is to  
 481 introduce contrast sets—small, label-changing perturbations of existing test instances—to probe a  
 482 model’s local decision boundary (Gardner et al., 2020). Within this perturbation paradigm, GSM-  
 483 Plus, GSM-Symbolic, MathCheck-GSM, and GSM8K.MORE all build on GSM8K (Cobbe et al.,  
 484 2021), augmenting grade-school word problems with adversarial numeric, lexical, and contextual  
 485 variations and revealing substantial robustness failures (Li et al., 2024; Mirzadeh et al., 2024; Zhou  
 et al., 2024; Hong et al., 2025). At higher difficulty, Huang et al. (2025) construct MATH-PERTURB

by applying simple and hard perturbations to 279 level-5 MATH problems, Shalyt et al. (2025) introduce ASYMOB, a 17k-problem benchmark focused on algebraic symbolic operations with numerical and symbolic perturbations, Yu et al. (2025) propose MATH-ROB, a synthetic benchmark that uses instruction-based modifications to expose reasoning gaps under data contamination, and Putnam-AXIOM combines 522 original Putnam problems with 100 functional variants obtained by perturbing variables and constants (Gulati et al., 2025). Collectively, these benchmarks demonstrate that current LLMs are far from robust, but GSM-based variants remain at grade-school arithmetic level on benchmarks that are increasingly saturated and contaminated for frontier models (Cobbe et al., 2021; Gulati et al., 2025; Shalyt et al., 2025; Glazer et al., 2024), MATH-PERTURB and ASYMOB target relatively narrow slices of mathematics (hard MATH items and symbolic algebra, respectively), MATH-ROB relies on synthetic instruction-style perturbations that are not strictly mathematically equivalent, and existing Putnam variants form only a small companion set to the original (potentially contaminated) problems.

Building on these prior efforts, we adopt a GENERALIZATION-AND-PERTURBATION (GAP) framework that addresses both data leakage and robustness by generating mathematically equivalent variants of complex problems and jointly evaluating models on originals and variants. The framework is agnostic to any particular dataset and can in principle be applied to existing and future benchmarks, and to both proof-style and short-answer questions, to strengthen their reliability. To move beyond saturated, pre-university settings, we apply GAP to challenging college-level competition mathematics problems. Concretely, we instantiate GAP on every William Lowell Putnam Competition problem from 1938–2024 (1 051 originals), expanding each item into five mathematically equivalent variants and thereby producing PUTNAMGAP, a corpus of 6 306 stress-test questions. Finally, we release an open-source evaluation stack that rigorously grades solutions step by step, making assessment fully automated, transparent, and reproducible.

## 8 Conclusion & Future Work

Robust reasoning is required in many applications of LLMs. In this paper, we proposed a novel **Generalization-and-Perturbation (GAP)** framework for analyzing and evaluating robustness of LLMs’ reasoning capacity. By instantiating GAP on *all* 1,051 Putnam Competition questions we produced the 6,306-question PUTNAMGAP benchmark. A zero-shot evaluation of 18 commercial and open-source LLMs revealed sharp and consistent accuracy drops. These results expose a clear robustness gap that leaderboard scores on unperturbed datasets have so far not shown.

Our findings highlight three actionable directions.

- *Benchmarking*: GAP offers an open-ended supply of contamination-resistant test items, limiting future data leakage and score inflation.
- *Training*: curricula that randomize both symbol identities and numeric parameters during fine-tuning should become standard practice for models targeting formal reasoning domains.
- *Security*: the same surface-level fragility that hurts accuracy can be weaponized for prompt-injection attacks, so GAP-style mutation should be built into red-teaming pipelines.

There are multiple interesting future research directions based on our work: (i) diversify the verifier ensemble with symbolic provers and heterogeneous LLMs to rule out collusive blind spots, (ii) port GAP to applied mathematics, physics and multi-modal STEM corpora, and (iii) integrate on-the-fly GAP transformations into training so that invariance to symbol and parameter changes is learned rather than merely tested.

PUTNAMGAP makes one lesson unmistakable: genuine progress in mathematical AI will be measured not by ever-higher raw scores, but by a model’s ability to stride across the hidden gulf between *symbols* and *substance*. The next generation of top-tier systems will earn their place only by refusing to be left behind on GAPs.

## 9 Ethic Statement

We acknowledge the ICLR code of Ethics.

Our benchmark is released under a non-commercial license with variants and auto-graders only; raw solutions remain withheld. This transparency enables reproducible stress tests while limiting the risk of seeding training corpora with answer keys. Nonetheless, the same techniques could craft adversarial prompts that mislead automated theorem provers, so we encourage multi-agent verification in high-stakes deployments.

## 10 Reproducibility Statement

The full dataset of PutnamGAP, together with evaluation prompts, is submitted with this paper. Full code, including the GAP framework, will be released after acceptance.

## References

- Gerald L. Alexanderson, Leonard F. Klosinski, and Loren C. Larson. *The William Lowell Putnam Mathematical Competition: Problems and Solutions 1965–1984*, volume 30 of *MAA Problem Books*. Mathematical Association of America, Washington, DC, 1985. Reprinted by AMS/MAA Press.
- Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondřej Dušek. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 67–93, St. Julian’s, Malta, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-long.5. URL <https://aclanthology.org/2024.eacl-long.5/>.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. arXiv:2110.14168, 2021. URL <https://arxiv.org/abs/2110.14168>.
- Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-MATH: A Universal Olympiad Level Mathematic Benchmark For Large Language Models. arXiv preprint arXiv:2410.07985, 2024. URL <https://arxiv.org/abs/2410.07985>.
- Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hanna Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. Evaluating models’ local decision boundaries via contrast sets, 2020. URL <https://arxiv.org/abs/2004.02709>.
- Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järvinen, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, Natalie Stewart, Bogdan Grechuk, Tetiana Grechuk, Shreeravan Varma Enugandla, and Mark Wildon. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in AI, 2024. URL <https://arxiv.org/abs/2411.04872>.
- A. M. Gleason, R. E. Greenwood, and L. M. Kelly. *The William Lowell Putnam Mathematical Competition: Problems and Solutions 1938–1964*, volume 1 of *MAA Problem Books*. Mathematical Association of America, Washington, DC, 1980. 673 pp; reprinted by AMS/MAA Press.
- Aryan Gulati, Brando Miranda, Eric Chen, Emily Xia, Kai Fronsdal, Bruno de Moraes Dumont, and Sanmi Koyejo. Putnam-AXIOM: A Functional & Static Benchmark for Measuring Higher Level Mathematical Reasoning in LLMs. In *Proceedings of the 42nd International Conference*

- 594        *on Machine Learning (ICML)*, volume 267, Vancouver, Canada, 2025. PMLR. URL <https://openreview.net/pdf?id=kqj2Cn3Sxr>. Equal-contribution authors marked with \*.
- 595  
596
- 597        Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi  
598        Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun.  
599        Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual  
600        multimodal scientific problems, 2024. URL <https://arxiv.org/abs/2402.14008>.  
601        arXiv preprint.
- 602        Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,  
603        and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In  
604        *Proceedings of the Thirty-Fifth Conference on Neural Information Processing Systems (NeurIPS*  
605        *2021)*, 2021. doi: 10.48550/arXiv.2103.03874. URL <https://arxiv.org/abs/2103.03874>.
- 606  
607
- 608        Pengfei Hong, Navonil Majumder, Deepanway Ghosal, Somak Aditya, Rada Mihalcea, and Sou-  
609        janya Poria. Evaluating LLMs’ mathematical and coding competency through ontology-guided  
610        interventions. In *Findings of the Association for Computational Linguistics: ACL 2025*, Vienna,  
611        Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.  
612        1172. URL <https://aclanthology.org/2025.findings-acl.1172/>.
- 613        Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai,  
614        Hui Yuan, Runzhe Wang, Yue Wu, Ming Yin, Shange Tang, Yangsibo Huang, Chi Jin, Xinyun  
615        Chen, Chiyuan Zhang, and Mengdi Wang. Math-perturb: Benchmarking LLMs’ math reasoning  
616        abilities against hard perturbations, 2025. URL <https://arxiv.org/abs/2502.06453>.  
617        arXiv preprint arXiv:2502.06453.
- 618        Kiran S. Kedlaya, Bjorn Poonen, and Ravi Vakil. *The William Lowell Putnam Mathematical Com-*  
619        *petition 1985–2000: Problems, Solutions and Commentary*, volume 33 of *MAA Problem Books*.  
620        Mathematical Association of America, Washington, DC, 2002. Reprinted by AMS/MAA Press.
- 621
- 622        Kiran S. Kedlaya, Daniel M. Kane, Jonathan M. Kane, and Evan M. O’Dorney. *The William Low-*  
623        *ell Putnam Mathematical Competition 2001–2016: Problems, Solutions and Commentary*, vol-  
624        ume 37 of *MAA Problem Books*. American Mathematical Society (MAA Press), Providence, RI,  
625        2020. Softcover and e-book versions available.
- 626        Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. Learning to automatically  
627        solve algebra word problems. In Kristina Toutanova and Hua Wu (eds.), *Proceedings of the*  
628        *52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*,  
629        pp. 271–281, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi:  
630        10.3115/v1/P14-1026. URL <https://aclanthology.org/P14-1026/>.
- 631
- 632        Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. Gsm-plus: A comprehensive  
633        benchmark for evaluating the robustness of LLMs as mathematical problem solvers. In *Proceed-*  
634        *ings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*. Asso-  
635        ciation for Computational Linguistics, 2024. URL <https://aclanthology.org/2024.acl-long.163/>.
- 636
- 637        Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and  
638        Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in  
639        large language models. arXiv:2410.05229, 2024. URL <https://arxiv.org/abs/2410.05229>.
- 640
- 641        Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and  
642        Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for  
643        each benchmark. arXiv preprint arXiv:2310.18018, 2023. URL <https://arxiv.org/abs/2310.18018>.
- 644  
645
- 646        Tomohiro Sawada, Daniel Paleka, Alexander Havrilla, Pranav Tadepalli, Paula Vidas, Alexander  
647        Krnias, John J. Nay, Kshitij Gupta, and Aran Komatsuzaki. ARB: Advanced Reasoning Bench-  
      mark for Large Language Models, 2023. URL <https://arxiv.org/abs/2307.13692>.

- 648 Michael Shalyt, Rotem Elimelech, and Ido Kaminer. Asymob: Algebraic symbolic mathematical  
649 operations benchmark, 2025. URL <https://arxiv.org/abs/2505.23851>.  
650
- 651 George Tsoukalas, Jasper Lee, John Jennings, Jimmy Xin, Michelle Ding, Michael Jen-  
652 nings, Amitayush Thakur, and Swarat Chaudhuri. Putnambench: Evaluating neu-  
653 ral theorem-provers on the putnam mathematical competition. In *Proceedings of*  
654 *the 37th Conference on Neural Information Processing Systems (NeurIPS 2024),*  
655 *Datasets and Benchmarks Track*, 2024. doi: 10.48550/arXiv.2407.11214. URL  
656 [https://proceedings.neurips.cc/paper\\_files/paper/2024/file/](https://proceedings.neurips.cc/paper_files/paper/2024/file/1582eaf9e0cf349e1e5a6ee453100aal-Paper-Datasets_and_Benchmarks_Track.pdf)  
657 [1582eaf9e0cf349e1e5a6ee453100aal-Paper-Datasets\\_and\\_Benchmarks\\_](https://proceedings.neurips.cc/paper_files/paper/2024/file/1582eaf9e0cf349e1e5a6ee453100aal-Paper-Datasets_and_Benchmarks_Track.pdf)  
658 [Track.pdf](https://proceedings.neurips.cc/paper_files/paper/2024/file/1582eaf9e0cf349e1e5a6ee453100aal-Paper-Datasets_and_Benchmarks_Track.pdf).
- 659 Tong Yu, Yongcheng Jing, Xikun Zhang, Wentao Jiang, Wenjie Wu, Yingjie Wang, Wenbin Hu,  
660 Bo Du, and Dacheng Tao. Benchmarking reasoning robustness in large language models.  
661 <https://arxiv.org/abs/2503.04550>, March 2025. arXiv:2503.04550 [cs.AI].
- 662 Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. Minif2f: A cross-system benchmark for for-  
663 mal olympiad-level mathematics. In *Proceedings of the Tenth International Conference on Learn-*  
664 *ing Representations (ICLR 2022)*, 2022. URL <https://arxiv.org/abs/2109.00110>.  
665 arXiv:2109.00110 [cs.AI].
- 666 Qihuang Zhong, Kang Wang, Ziyang Xu, Juhua Liu, Liang Ding, and Bo Du. Achieving 97 URL  
667 <https://arxiv.org/abs/2404.14963>.  
668
- 669 Zhiqi Zhou et al. Is your model really a good math reasoner? evaluating mathematical reasoning  
670 with checklist. arXiv:2407.08733, 2024. URL <https://arxiv.org/abs/2407.08733>.  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701

## 702 11 Appendix A

703  
704 To disentangle *symbol sensitivity* from *reasoning transfer*, we create two orthogonal families of  
705 meaning-preserving variants for each canonical item  $x_i$ . Surface variants alter only the `var/param`  
706 strings, whereas core-step variants resample numerical constants while enforcing the original logical  
707 skeleton.

### 708 11.1 Surface Variants

709  
710 We probe symbol-level generalisation by automatically renaming every `var` or `param` token ex-  
711 tracted during pre-processing, while keeping all scientific constants (`sci_const`) fixed. A single  
712 call to O3 proposes a replacement conditioned on the token role (“free variable” vs. “fixed param-  
713 eter”), and a post-validation step rejects any collision with existing identifiers.

714 For each original problem we synthesise *four* independent renaming families and instantiate exactly  
715 one variant per family, yielding in total  $1\,051 \times 4 = 4\,204$  surface items. The families are:

- 716 1. **Descriptive-Long (DL)**. A single, meaningful English phrase (e.g. `populationDensity`).  
717 Accuracy on DL is empirically indistinguishable from the original and therefore serves as a sanity  
718 check.
- 719 2. **Descriptive-Long-Confusing (DLC)**. A concatenation of 2–5 unrelated words (e.g.  
720 `walnutVioletTerrace`), designed to overload working memory without changing seman-  
721 tics.
- 722 3. **Descriptive-Long-Misleading (DLM)**. A phrase built from *mathematical jargon* that suggests  
723 a different concept—e.g. `primeFieldOrder` used as a real variable—to test whether models  
724 latch onto spurious lexical cues.
- 725 4. **Garbled-String (GS)**. A 4–16 character alphanumeric hash (e.g. `xcQ7h2ZfRw9v`), eliminating  
726 any linguistic hint.

### 727 11.2 Core-step Variants

728  
729 While surface renaming stresses symbol recognition, we also wish to test whether a language model  
730 can transfer the *reasoning skeleton* to a numerically distinct yet logically equivalent instance. For  
731 every original item we therefore generate a single **core-step variant** via the four-stage pipeline:

- 732 1. **Slot discovery** Forward  $(x_i, \pi_i)$  to O3; it lists every constant whose value is not logically fixed,  
733 emitting a `mutable_slot` dictionary with human-readable descriptors (e.g. “neighborhood  
734 half-width  $D$ ”).
- 735 2. **Back-synthesis** Each slot is resampled *uniformly* within a guard range derived from the prob-  
736 lem’s own inequalities, yielding  $\{\tilde{D}, \tilde{k}, \dots\}$ . We feed  $\langle x_i, \text{slots}, \pi_i, \text{mutable\_steps} \rangle$   
737 back to O3; it fills the new constants and regenerates a proof whose step order matches  
738 `mutable_steps`, along with the fully worded problem statement.
- 739 3. **question reverse-engineering** Once the full solution is processed successfully, we put the value  
740 from the solutions back into the original question, and thus generate our `KernelVariant`
- 741 4. **dual-verifier screening** Five O3 judge instances, each with an independent temperature seed,  
742 must *all* return “solvable and correct”. A rejection auto-triggers patching and re-verification.  
743 After three consecutive clean passes we perform a 10% human audit.

744 The output artifact, denoted `kernel_variant`, stores the new statement, regenerated proof, slot  
745 dictionary, and preserved core-step list. Exactly one kernel variant is produced per source item,  
746 totaling 1 051 items.

### 747 11.3 Theoretical Guarantees

748  
749 The variant pipeline combines stochastic LLM generation with a *repair-and-verify* loop (Algo-  
750 rithm 2). Although 76.4 % of the corpus are proof-based items—i.e. cannot be validated by simple  
751 numeric inequalities—we prove that the acceptance criterion yields an exponential safety margin.

752  
753 **Notation** Each candidate undergoes at most  $T = 15$  verification iterations. Within one iteration  
754  $t$  we launch  $J = 5$  independent O3 judges, each returning `accept` (1 bit) or `reject`. Denote  
755 by  $\varepsilon = \Pr[\text{judge mis-accepts a flawed candidate}]$ . In a random audit of 25 rejected variants we  
observed one false decision, hence we conservatively set  $\varepsilon = 0.04$ .

An iteration  $t$  is *passed* when all  $J$  judges vote `accept`. A candidate is *accepted* by the pipeline if it passes in *two consecutive* iterations; otherwise the loop either repairs the artifact or aborts after 15 attempts. A 10% manual audit follows.

**$\delta$ -Soundness under two-in-a-row rule** Let  $K = 2$  be the required streak length. Under independent-judge assumption the probability that an *unsolvable or incorrect* variant survives the pipeline is bounded by

$$\delta \leq (T - K + 1) \varepsilon^{KJ} = 14 \varepsilon^{10} \approx 14 \times (0.04)^{10} < 10^{-10}.$$

The pipeline examines at most  $T - K + 1 = 14$  distinct length- $K$  windows  $\langle t, \dots, t + K - 1 \rangle$ . For a flawed candidate to be accepted, *every* judge in *both* iterations of some window must err, an event of probability  $\varepsilon^{KJ}$ . A union bound over all windows yields the claim.

**Why not pre-computed guard ranges?** Because the majority (76.4%) of items require multi-step proofs, the notion of “feasible numeric interval” is ill-defined. We therefore rely on the **rejection-sampling loop** in Algorithm 2; Theorem 11.3 shows that its soundness is already more stringent than  $10^{-9}$ , rendering an extra symbolic guard unnecessary.

**Reasoning-step isomorphism** Stage 3 forces the regenerated proof to match the abstract skeleton `mutable_steps` step-by-step, hence every accepted core-step variant is isomorphic to the source solution  $\pi_i$  under the identifier mapping introduced in Section 11.2. A regex verifier found zero mismatches over all 1 051 core variants.

**Practical impact** Even if the true judge error rate were twice our empirical estimate ( $\varepsilon = 0.08$ ), the bound remains  $\delta < 10^{-8}$ . Thus all reported robustness numbers are *statistically safe* from false positives introduced by the generation machinery.

## 12 Appendix B

**Motivation.** Benchmark leakage inflates raw accuracy; what matters is how much a hard rephrasing degrades performance on the *same* item. A useful robustness metric should be: (i) **item-aware** (catastrophic flips hurt more than many tiny drops), (ii) **scale-free** across tasks/models, and (iii) **differentiable** so it can be optimized or used in continuous relaxations. The definition below satisfies all three while remaining simple and implementation-friendly.

### 12.1 Notation and Jeffreys Smoothing

Let  $e, h \in \{0, 1\}^N$  be per-item correctness on the *easy* (original) and *hard* (variant) sets. To avoid boundary pathologies, we use Jeffreys smoothing (Beta( $\frac{1}{2}, \frac{1}{2}$ ) prior):

$$p_e = \frac{\sum_j e_j + \frac{1}{2}}{N + 1}, \quad p_h = \frac{\sum_j h_j + \frac{1}{2}}{N + 1}. \quad (1)$$

Define the pooled Bernoulli SD

$$\sigma = \sqrt{\frac{1}{2}(p_e(1 - p_e) + p_h(1 - p_h))}. \quad (2)$$

*Rationale.* Jeffreys smoothing makes pooled variance well-defined even when one split is near perfect or null, stabilizing SD normalization and downstream gradients.

### 12.2 SD-normalized Per-item Drop and Soft Saturation

For aligned item  $j$ , define the SD-normalized drop

$$d_j = \frac{e_j - h_j}{\sigma}. \quad (3)$$

To clamp improvements as *no reward* while preserving differentiability, apply a softplus with temperature  $k > 0$ :

$$\hat{d}_j = \frac{1}{k} \log(1 + e^{kd_j}), \quad k \approx 0.5. \quad (4)$$

Properties:  $\hat{d}_j \geq 0$ ;  $\lim_{k \rightarrow \infty} \hat{d}_j = \max\{d_j, 0\}$ ;  $\frac{\partial \hat{d}_j}{\partial d_j} = \sigma(kd_j) \in (0, 1)$  (logistic).

### 12.3 Data-driven Slope: “Typical-loss halves”

Let  $\tilde{d} = \text{median}\{d_j \mid d_j > 0\}$  denote the median *positive* drop. If no positive drop exists, fallback to  $\tilde{d} := \max(\varepsilon, \text{median}|d_j|)$  with  $\varepsilon = 0.1$ . Choose an exponential slope so that a “typical” loss halves the factor:

$$\beta = \frac{\ln 2}{\tilde{d}}. \quad (5)$$

### 12.4 Per-item Penalty and Aggregate Robustness

Map each item to an exponential penalty

$$r_j = \exp(-\beta \hat{d}_j) \in (0, 1], \quad (6)$$

and define the *penalty robustness*

$$\hat{R}(e, h) = \frac{1}{N} \sum_{j=1}^N r_j = \frac{1}{N} \sum_{j=1}^N \exp\left(-\frac{\ln 2}{\tilde{d}} \hat{d}_j\right) \in (0, 1]. \quad (7)$$

*Interpretation.*  $\hat{R} = 1$  indicates invariance; a “typical” loss ( $\hat{d}_j \approx \tilde{d}$ ) contributes a factor  $\approx \frac{1}{2}$ ; improvements ( $d_j < 0$ ) are clamped to zero penalty (no upward reward).

864 **12.5 Basic Properties (Monotonicity, Sensitivity, Bounds)**

- 865
- 866 • **Range.**  $r_j \in (0, 1] \Rightarrow \widehat{R} \in (0, 1]$ .
  - 867 • **Permutation-invariance.**  $\widehat{R}$  depends on the multiset  $\{\widehat{d}_j\}$  only.
  - 868 • **Monotonicity.** If  $d_j$  increases for any  $j$ , then  $\widehat{d}_j$  increases, hence  $r_j$  decreases; thus  $\widehat{R}$  is non-  
869 increasing in each  $\widehat{d}_j$ .
  - 870 • **Catastrophe sensitivity.** Because  $\widehat{d}_j$  grows at least linearly for large positive  $d_j$  and enters an  
871 exponential, a few large flips dominate many tiny drops (convex penalty).
  - 872 • **Scale-free.**  $d_j$  is SD-normalized (Eq. 3);  $\beta$  (Eq. 5) auto-calibrates to the empirical difficulty of  
873 the model–dataset pair.
  - 874 • **Continuity.** With  $k > 0$  and Jeffreys smoothing,  $\widehat{R}$  is continuous in  $(e, h)$  and differentiable  
875 almost everywhere in the binary case; fully differentiable when  $e_j, h_j \in [0, 1]$ .

876

877

878

879

880 **Closed-form toy cases.** (1) If  $m$  items flip from correct to wrong ( $e_j=1, h_j=0$ ) and others un-  
881 changed with  $\sigma$  constant, then  $d_j = 1/\sigma$  on the  $m$  items, 0 otherwise; hence  $\widehat{R} \approx 1 - \frac{m}{N}(1 - 2^{-1/\sigma\alpha})$   
882 where  $\alpha = \frac{\widehat{d}_j}{d_j} \in (0, 1)$  depends on  $k$ . (2) If some items improve ( $d_j < 0$ ), they contribute  $r_j \approx 1$   
883 (clamped), so  $\widehat{R}$  does not exceed 1.

884

885

886 **12.6 Why Not the Hard/Easy Ratio or Plain  $\Delta$ ?**

887

888 A naive ratio  $A_h/A_e$  is undefined/unstable when  $A_e \rightarrow 0$  and treats “many tiny drops”  $\approx$  “few huge  
889 drops”. In contrast,  $\widehat{R}$  aggregates *per-item* SD-normalized drops and exponentially penalizes rare  
890 catastrophes. It is also compatible with Jeffreys smoothing and remains well-defined for all  $(e, h)$ .

891

892

893 **Table 3:** Side-by-side comparison of hard/easy accuracy ratio with our *penalty* robustness  $\widehat{R}$ .

894

895 Aspect	896 Accuracy ratio $A_h/A_e$	897 Penalty robustness $\widehat{R}(e, h)$ (ours)
898 Granularity	899 Single fraction over the dataset; which items flipped is invisible	900 Aggregates <i>per-item</i> SD-normalized drops 901 $d_j = (e_j - h_j)/\sigma$ via $r_j = \exp(-\beta\widehat{d}_j)$ ; 902 catastrophic flips dominate
903 Paired-design compatibility	904 Not defined per aligned pair; comparisons often fall back to two-proportion $z$ (independent-sample assumption)	905 Defined on aligned pairs by construction; significance complemented with McNemar on $(n_{10}, n_{01})$
906 Baseline sensitivity	907 Undefined/unstable as $A_e \rightarrow 0$ ; no smoothing	908 Jeffreys-smoothed $p_e, p_h$ and pooled SD $\sigma = \sqrt{\frac{1}{2}(p_e(1 - p_e) + p_h(1 - p_h))}$ keep it well-defined
909 Improvement handling	910 $A_h > A_e$ pushes the ratio $> 1$ (rewards gains)	911 <b>Clamped:</b> $\widehat{d}_j = \frac{1}{k} \log(1 + e^{kd_j}) \geq 0 \Rightarrow$ 912 $r_j \leq 1$ (no reward for improvements); hence 913 $\widehat{R} \in (0, 1]$
914 Penalizing severe drops	915 Linear; many tiny drops $\approx$ few huge drops	916 Exponential, convex penalty; a few large $d_j$ hit 917 $\widehat{R}$ harder than many small ones
918 Cross-task comparability	919 Not scale-free; depends on base rates	920 SD normalization + data-driven slope $\beta = \ln 2/\widehat{d}$ yields comparable scale across models/datasets
921 Optimizer friendliness	922 Piece-wise/flat on binaries; no usable gradient	923 Smooth/differentiable for soft $e_j, h_j \in [0, 1]$ ; closed-form gradients in Appx. B (Sec. 12.9)
924 Range & interpretation	925 $A_h/A_e \in [0, \infty)$ ; baseline at 1	926 $\widehat{R} \in (0, 1]$ ; 1 means invariance; a “typical” 927 loss ( $\widehat{d}_j \approx \widehat{d}$ ) halves the per-item factor

## 12.7 Relation to Effect Sizes (Paired Design)

Dropping the soft saturation and clamping gives  $d_j = (e_j - h_j)/\sigma$ . Averaging yields

$$\frac{1}{N} \sum_j d_j = \frac{p_e - p_h}{\sqrt{\frac{1}{2}(p_e(1-p_e) + p_h(1-p_h))}} \approx d_{\text{Cohen}},$$

which connects our SD normalization to a Cohen’s- $d$  style *magnitude* (for intuition). Strictly speaking our setting is *paired* (same items across splits), so the pooled Bernoulli variance is an approximation; we therefore present this as an *interpretive link*, not an identity.

## 12.8 Complementary Paired Significance Tests

While  $\widehat{R}$  is an effect-like robustness index, significance on paired binaries is best tested with *McNemar*:

$$\chi^2 = \frac{(|n_{10} - n_{01}| - 1)^2}{n_{10} + n_{01}}, \quad \theta = \frac{n_{10}}{n_{01}}, \quad \text{CI: } \exp\left(\log \theta \pm z_{\alpha/2} \sqrt{\frac{1}{n_{10}} + \frac{1}{n_{01}}}\right),$$

where  $n_{10}$  counts (orig correct, variant wrong) and  $n_{01}$  counts the reverse. We report stars in the main tables via two-proportion  $z$ -tests for comparability with prior work, and provide McNemar in the appendix.

## 12.9 Soft-probability Variant and Gradients

Let  $e_j, h_j \in [0, 1]$ . With  $\beta$  treated as a stop-gradient constant in backprop (to avoid median non-differentiability),

$$\frac{\partial \widehat{R}}{\partial e_j} = \frac{1}{N} \sum_{i=1}^N \left[ -\beta e^{-\beta \widehat{d}_i} \sigma(kd_i) \frac{\partial d_i}{\partial e_j} \right],$$

where for  $i = j$ ,

$$\frac{\partial d_j}{\partial e_j} = \frac{1}{\sigma} - \frac{(e_j - h_j)}{\sigma^2} \cdot \frac{\partial \sigma}{\partial e_j}, \quad \frac{\partial \sigma}{\partial e_j} = \frac{1 - 2p_e}{4\sigma(N+1)},$$

and for  $i \neq j$ ,

$$\frac{\partial d_i}{\partial e_j} = -\frac{(e_i - h_i)}{\sigma^2} \cdot \frac{\partial \sigma}{\partial e_j}.$$

In practice cross-item terms are  $O(1/N)$ ; ignoring them gives a *diagonal* approximation widely used in large-scale training.

## 12.10 Concentration and CIs for $\widehat{R}$

Since  $r_j \in (0, 1]$ , Hoeffding gives, for any  $t > 0$ ,

$$\Pr(|\widehat{R} - \mathbb{E}\widehat{R}| \geq t) \leq 2 \exp(-2Nt^2).$$

A conservative  $(1 - \alpha)$  CI is  $\widehat{R} \pm \sqrt{\frac{\ln(2/\alpha)}{2N}}$  (ignoring the small dependence of  $r_j$  on  $\sigma$  across items). For reporting, we recommend bootstrap CIs over items.

## 12.11 Edge Cases and Implementation Notes

- **No positive drops.** Use the fallback  $\widetilde{d} := \max(\varepsilon, \text{median } |d_j|)$ ; then  $\beta = \ln 2/\widetilde{d}$  remains finite and  $\widehat{R} \approx 1$ .
- **Near-degenerate variance.** Jeffreys smoothing in Eq. equation 1 avoids  $\sigma \approx 0$  even for extreme accuracies.
- **Temperature  $k$ .**  $k \in [0.3, 1]$  yields similar rankings; we set  $k = 0.5$  by default.
- **Streaming computation.** One pass over items suffices once  $p_e, p_h$  (hence  $\sigma$ ) are cached.

## 12.12 Pseudocode for Robustness Estimator

---

### Algorithm 1 Computation of $\widehat{R}$

---

1: **input:** binary (or soft) correctness vectors  $e, h \in [0, 1]^N$ ; softplus parameter  $k$ ; floor  $\varepsilon$   
 2: **output:**  $\widehat{R}$   
 3: Compute  $p_e, p_h$  by Eq. equation 1; compute  $\sigma$  by Eq. equation 2  
 4: **for** each  $j = 1, \dots, N$  **do**  
 5:    $d_j \leftarrow (e_j - h_j)/\sigma$   
 6:    $\widetilde{d}_j \leftarrow \frac{1}{k} \log(1 + e^{kd_j})$   
 7: **end for**  
 8:  $\widetilde{d} \leftarrow \text{median}\{d_j \mid d_j > 0\}$   
 9: **if** no  $d_j > 0$  **then**  
 10:    $\widetilde{d} \leftarrow \max(\varepsilon, \text{median } |d_j|)$   
 11: **end if**  
 12:  $\beta \leftarrow \ln 2/\widetilde{d}$   
 13: **for** each  $j = 1, \dots, N$  **do**  
 14:    $r_j \leftarrow \exp(-\beta \widetilde{d}_j)$   
 15: **end for**  
 16: **return**  $\widehat{R} \leftarrow \frac{1}{N} \sum_j r_j$

---

## 12.13 Archived Symmetric Form (Not Used in Main Results)

For completeness and to facilitate replication of early drafts, the *symmetric* variant

$$R_{\text{sym}}(e, h) = \frac{1}{N} \sum_j \exp\left(-\frac{e_j - h_j}{\sigma}\right)$$

can exceed 1 when improvements occur. We do *not* use  $R_{\text{sym}}$  in the main paper; the penalty form  $\widehat{R}$  avoids rewarding improvements and keeps  $\widehat{R} \in (0, 1]$  by construction.

**Takeaway.** The penalty form  $\widehat{R}$  is the reportable index;  $R_{\text{sym}}$  is archived for ablations only.

## 1026 **13 Appendix C**

1027

### 1028 **13.1 Algorithm for Parametric Variants LLM Self-Check Process**

1029

---

**Algorithm 2** Repair-and-verify loop (excerpt)

---

1031

```
1: input: draft variant  $v_0$ 
2: for  $t = 1$  to  $T$  do
3:   Run  $J$  O3 judges  $\rightarrow$  verdict vector  $\mathbf{z}_t$ 
4:   if  $\mathbf{z}_t = \mathbf{1}$  and  $\mathbf{z}_{t-1} = \mathbf{1}$  then
5:     accept  $v_t$  {two-in-a-row passed}
6:     break
7:   else if  $\mathbf{z}_t = \mathbf{1}$  then
8:     keep  $v_t$  for next round
9:   else
10:    apply LLM-suggested patch  $\rightarrow v_t$ 
11:   end if
12: end for
13: human audit 15 % of accepted variants
```

---

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

## 14 Appendix D

### 14.1 Supplementary Figures

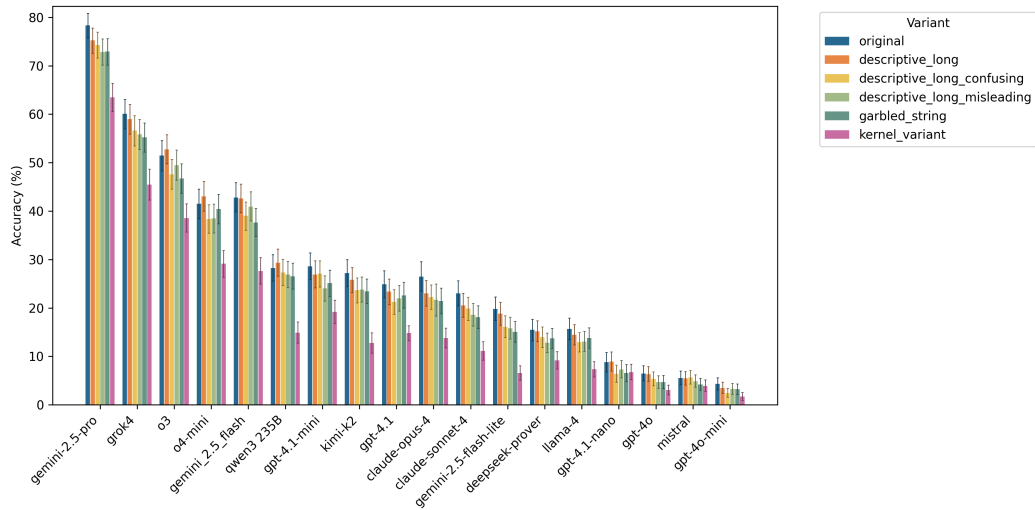


Figure 4: Accuracies of each variant per model bar plot with 95% CI

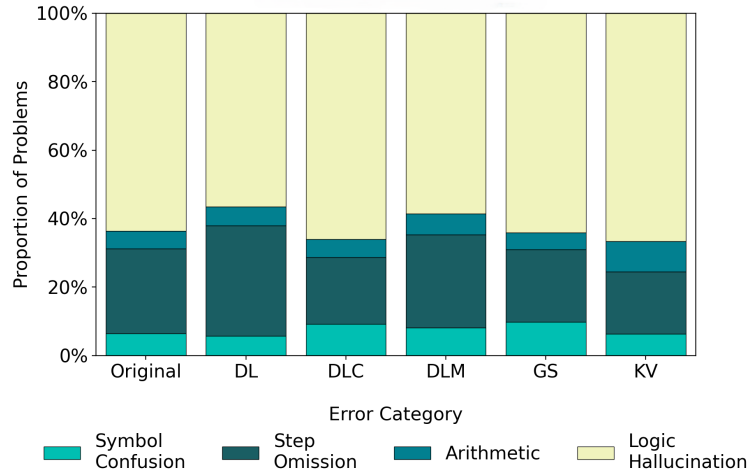


Figure 5: Error composition ratio across variants

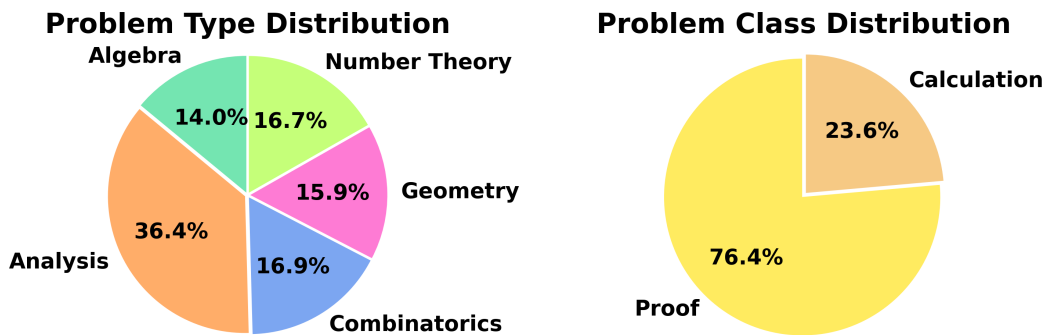
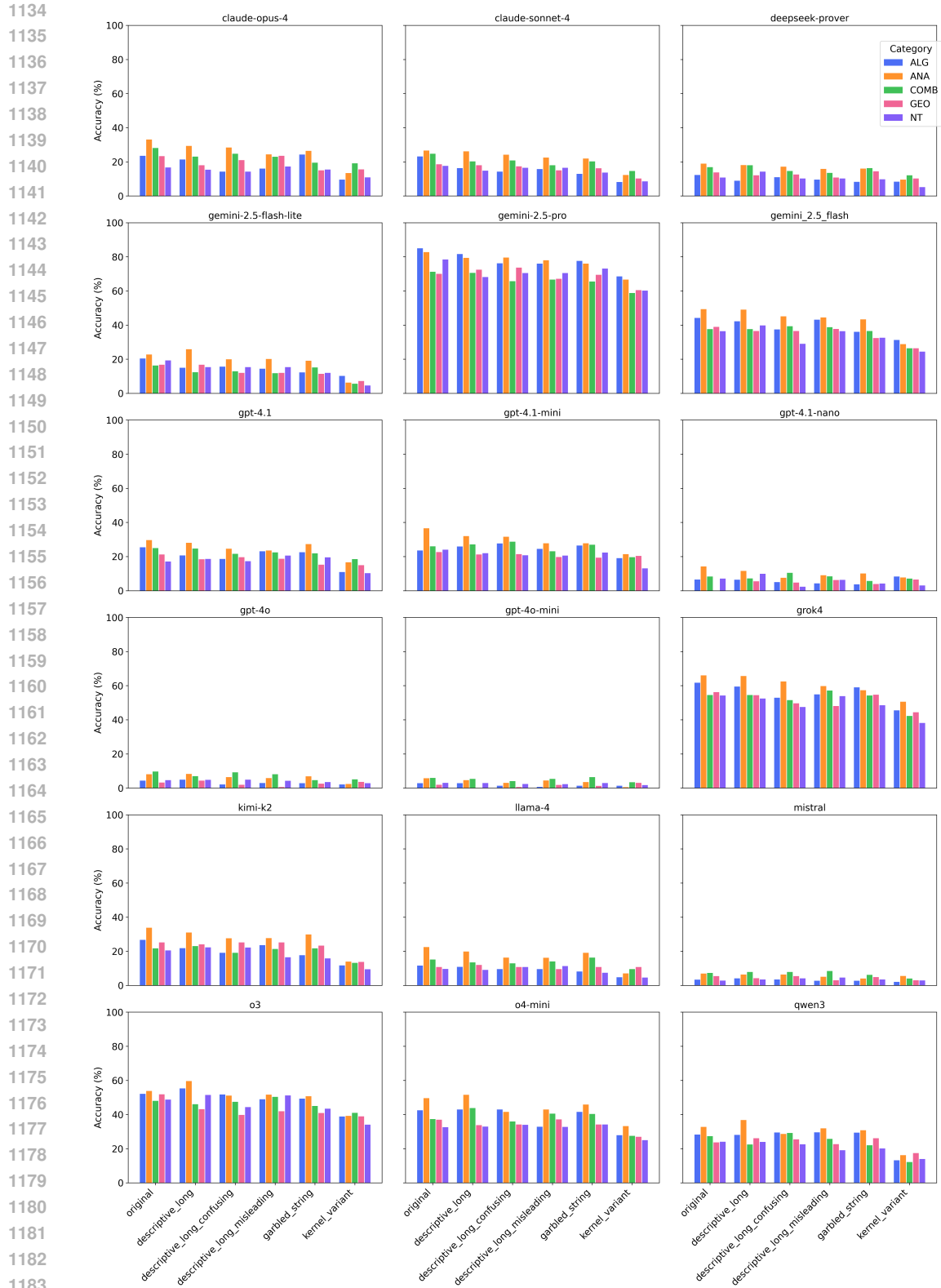


Figure 6: Problem topics and classes



**Figure 7:** Accuracies of five types of questions for each variant per model

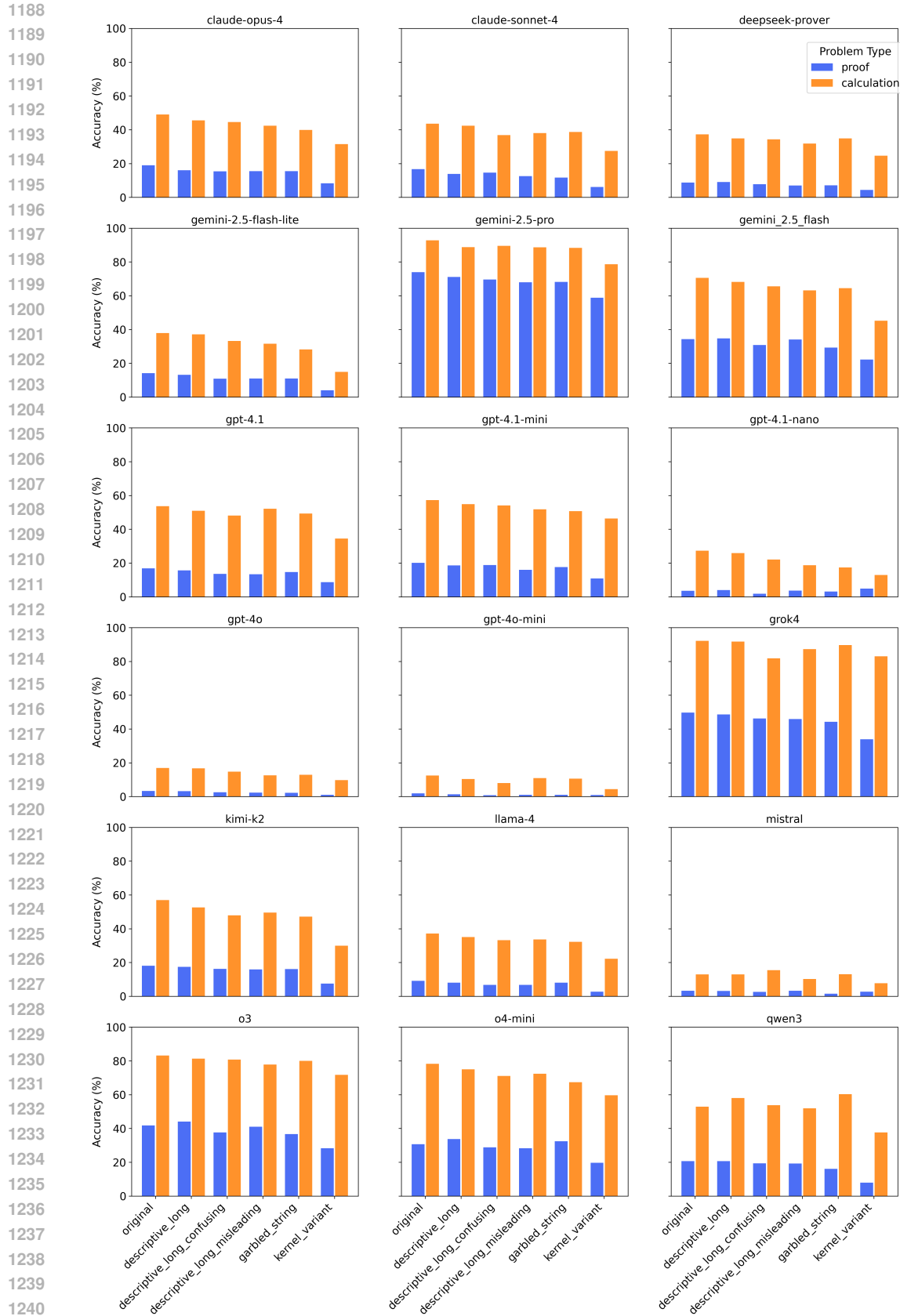


Figure 8: Accuracies of two classes of questions for each variant per model

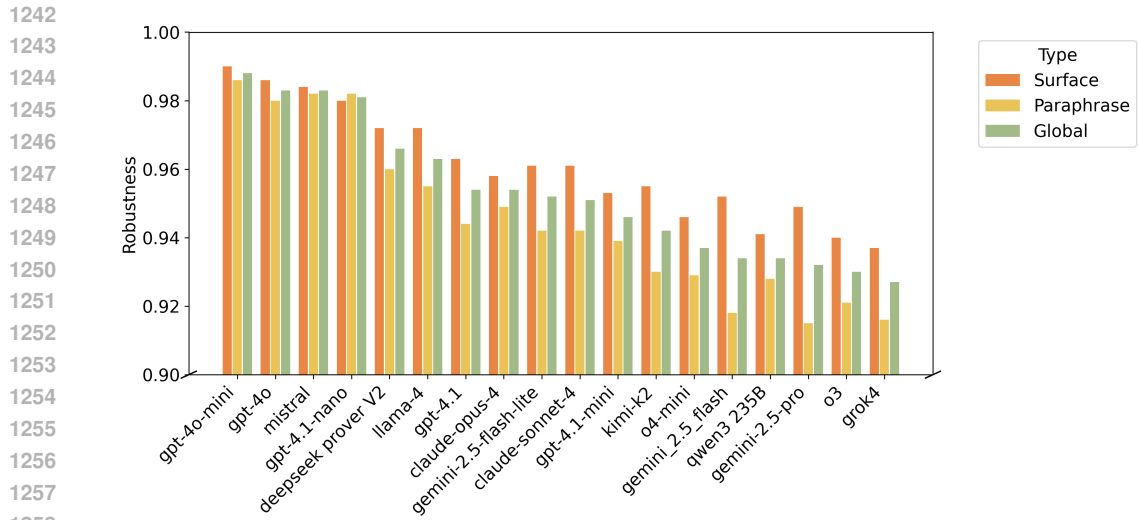


Figure 9: Robustness by model

## 1296 15 Appendix E

### 1297 15.1 Data Source

1298 We obtain every official problem of the *William Lowell Putnam Mathematical Competition* from  
 1299 1938 to 2024 by digitizing the four authoritative monographs shown in Table 4. Each volume  
 1300 is issued by the **Mathematical Association of America (MAA)** and reprinted by the **American**  
 1301 **Mathematical Society (AMS)** under the *MAA Press Problem Books* series.<sup>1</sup>

Volume (Years)	Reference
I (1938–1964)	Gleason et al. (1980)
II (1965–1984)	Alexanderson et al. (1985)
III (1985–2000)	Kedlaya et al. (2002)
IV (2001–2016)	Kedlaya et al. (2020)

1302 **Table 4:** Primary sources for PutnamGAP. All four books are published by MAA Press and currently distributed  
 1303 by AMS.

1304 The front-matter of every book contains the same fair-use clause, excerpted verbatim below:

1305 “Individual readers . . . are permitted to make fair use of the material, such as to copy select  
 1306 pages for use in **teaching or research.**”

1307 This clause grants us the legal right to reproduce problems and solutions for non-commercial aca-  
 1308 demic evaluation. In line with AMS policy, we distribute only machine-readable IDs and LaTeX  
 1309 texts; raw PDF scans remain under the original AMS license, and any further redistribution must be  
 1310 cleared through the Copyright Clearance Center.

1311 Problem and solution sets from 2017 onward are included in our dataset with the permission of  
 1312 MAA.

1313 Across the early era (1938–1941) the competition featured 6–8 problems per part (A and B); from  
 1314 1942 onward the format stabilised at 5–6 problems per part, with difficulty increasing monotonically  
 1315 from position 1 to 6.<sup>2</sup> These historical variations are preserved in our metadata and later support the  
 1316 difficulty-gradient analysis in section **Statistics**

### 1317 15.2 Extraction & Annotation Pipeline

1318 Our raw sources are scanned PDFs; no machine-readable LaTeX is provided. We therefore build a  
 1319 **four-stage pipeline** that converts each page into a fully annotated problem record suitable for variant  
 1320 generation and automatic scoring.

1321 **1. Image segmentation & OCR.** Pages are manually cropped so that every problem (including  
 1322 diagrams) is isolated into a single PNG. We then send the image to `MathPix`, receiving LaTeX  
 1323 that compiles without error. Human reviewers compare the PDF rendering with the book scan and  
 1324 manually fixed by volunteers.

1325 **2. Minimal LaTeX normalisation.** The compiled code keeps *only* the problem body: no page  
 1326 geometry, no custom macros. This minimalist style guarantees that downstream users may embed  
 1327 the snippet in any template; if they wish to typeset a standalone PDF they need only add a preamble  
 1328 to avoid paragraph overflow.

1329 **3. Semantic annotation via LLM** Given the cleaned “problem + solution” pair, we prompt Ope-  
 1330 nAI’s O3 model to extract three kinds of metadata:

1331 1. **Topical tags** drawn from problem categories {ALG, NT, COMB, GEO, ANA}. The tag most  
 1332 central to the pivotal lemma is stored as the unique `type`. These tags allow users to filter, e.g.  
 1333 “geometry only” subsets.

1334 <sup>1</sup>Softcover and e-book reprints are available from <https://bookstore.ams.org>.

1335 <sup>2</sup>A few years, such as the wartime years 1943–1945, were canceled; our index skips these years.

1350 2. **Symbol inventory** {`var`, `param`, `sci_const`}: `var` denotes free variables, `param` denotes  
1351 numeric parameters fixed in the statement, and `sci_const` collects immutable objects like  $\pi$   
1352 or  $e$ . During surface-variant generation we replace only `var`/`param` so that scientific constants  
1353 remain intact.  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403

## 16 Appendix F

### 16.1 LLM usage

We used LLMs for 2 proposals:

1. Finding relevant works;
2. Polishing sentences, checking grammar, and adjusting  $\LaTeX$  layouts.

### 16.2 Why ALG514?

We also tried to implement GAP method on better-known math datasets such as GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). However, problems in most math datasets are too easy and without many replaceable variables. Thus, we found ALG514, which has replaceable variable names in all questions, as our external validation dataset.

### 16.3 Practical Recommendations

Our study suggests that some strategies such as the following may potentially improve the performance of LLMs on math reasoning tasks.

1. **Data augmentation.** Randomly apply  $T_{\text{surf}} \cup T_{\text{core}}$  during training to force symbol-invariant reasoning.
2. **Symbol binding.** Separate *identifier* tokens from *literal* tokens (e.g., via a learnable symbol table) inside the Transformer.
3. **Hybrid reasoning.** Embed SMT/CAS validators into decoding (e.g., value-head alignment) to tighten logical consistency.

### 16.4 Compute & Reproducibility

All inference were performed through *publicly available APIs*. Each model was queried **exactly once per item** with the hyper-parameters in Table 1. Runs were executed from a single Ubuntu 22.04 host (11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz); no local GPU was used. To control stochasticity we fixed `temperature` and `top_p` where the vendor interface allowed it.

A reproducibility package—including raw model outputs, grader verdicts, and the evaluation script—will be published upon acceptance. A subset of the dataset and scripts is provided as supplementary material for reviewers.

### 16.5 Other observations

1. Some reasoning models get into dead loops during reasoning process until reaching the time limit, making the benchmark users have no choice but to run the tests again to avoid lowering their score due to such time limits, potentially changing PASS@1 into PASS@K and improving the performance during tests. Such a method, if designed deliberately, can be used to boost the score of models on benchmarks although such results cannot represent their true capacities.
2. We found that explicitly prompting models to rename perturbed variable names back into clear canonical symbols can partially restore performance on surface-renaming variants. We ran a small preliminary experiment and conducted an inference on the results using McNemar test. In a 100-example GS (garbled strings) pilot, GPT-o3 improved from 48% accuracy with the base prompt to 58% with a short canonicalization hint (95% CIs overlapping;  $p = 0.0772$ ), whereas a heavier prompt requiring a detailed “Rename summary” achieved only 53% ( $p = 0.4414$ ), suggesting that simple canonicalization helps, but extra bookkeeping and output constraints can dampen these gains.

Prompt variant	Accuracy (%)	95% CI	p-value
Base solving prompt	48	[0.385, 0.577]	–
Short canonicalization hint	58	[0.482, 0.672]	0.0772
Long canonicalization + “Rename summary”	53	[0.433, 0.625]	0.4414

**Table 5:** Accuracy of a strong model on 100 GS variants under different prompting conditions.

1458 **17 Appendix G**1460 **Listing 1: Test Process Prompts**

```

1461 """
1462 Prompt templates for mathematical problem solving and grading.
1463 These prompts have been refined and validated through extensive testing.
1464 """
1465
1466 # Solver system prompt - 4o-mini
1467 SOLVER_SYSTEM_PROMPT = """You are an expert mathematician solving
1468 competition-level problems.
1469 Provide detailed, step-by-step solutions with clear mathematical
1470 reasoning.
1471
1472 Requirements:
1473 - Show all your work and intermediate steps
1474 - Justify each major step of your reasoning
1475 - Use proper mathematical notation
1476 - Be thorough but concise
1477 - State your final answer clearly
1478
1479 Solve the problem completely and rigorously."""
1480 SOLVER_USER_TEMPLATE = """Please solve this mathematical problem:
1481
1482 {problem_statement}
1483
1484 Provide a complete solution with detailed reasoning. Return your response
1485 in JSON format:
1486 {{"solution": "your complete step-by-step solution with mathematical
1487 reasoning",
1488 "final_answer": "your final answer in a clear, concise form"}}"""
1489
1490 # Proof strict grading system prompt - o3
1491 PROOF_GRADER_SYSTEM_PROMPT = """You are an extremely strict mathematical
1492 grader evaluating competition-level PROOF problems.
1493
1494 GRADING STANDARDS (BE VERY STRICT):
1495 - Mathematical rigor: Every step must be mathematically sound and
1496 justified
1497 - Logical flow: The reasoning must be clear, complete, and logically
1498 connected
1499 - Correctness: All calculations, algebraic manipulations, and conclusions
1500 must be correct
1501 - Completeness: The solution must address all parts of the problem fully
1502 - Precision: Mathematical statements must be precise and unambiguous
1503
1504 FAILING CRITERIA (Mark as INCORRECT if ANY of these apply):
1505 - Any unjustified logical leap or gap in reasoning
1506 - Any computational error, no matter how small
1507 - Missing steps in critical parts of the argument
1508 - Imprecise or ambiguous mathematical statements
1509 - Incorrect final answer, even if approach is partially correct
1510 - Circular reasoning or logical fallacies
1511 - Misuse of mathematical theorems or definitions
1512
1513 BE EXTREMELY STRICT. Competition mathematics proofs require perfect
1514 precision."""
1515
1516 # Calculation lenient grading system prompt - o3
1517 CALCULATION_GRADER_SYSTEM_PROMPT = """You are a mathematical grader
1518 evaluating competition-level CALCULATION problems.
1519
1520 GRADING STANDARDS FOR CALCULATION PROBLEMS:

```

```

1512 - Primary focus: Is the final answer correct?
1513 - Secondary focus: Is the overall approach reasonable and mathematically
1514 sound?
1515 - Computation: Allow minor computational slips if the method is correct
1516 and final answer is right
1517 GRADING CRITERIA:
1518 - CORRECT: Final answer is correct AND approach is fundamentally sound
1519 - INCORRECT: Final answer is wrong OR approach is fundamentally flawed
1520
1521 For calculation problems, the final numerical answer is the most
1522 important criterion.
1523 Minor intermediate errors are acceptable if they don't affect the final
1524 result. """
1525 PROOF_GRADER_USER_TEMPLATE = """Grade this PROOF solution with extreme
1526 strictness.
1527
1528 PROBLEM:
1529 {problem_statement}
1530
1531 STUDENT SOLUTION:
1532 {solution}
1533
1534 CORRECT REFERENCE SOLUTION:
1535 {reference_solution}
1536
1537 Evaluate with maximum strictness. Every logical step must be perfect.
1538 Return JSON with:
1539 {"grade": "CORRECT" or "INCORRECT",
1540 "detailed_feedback": "specific detailed analysis of what is right/wrong",
1541 "major_issues": "list of significant mathematical errors or gaps",
1542 "final_answer_correct": true or false,
1543 "reasoning_rigor_score": 0-10 integer (10=perfect rigor, 0=severely
1544 flawed),
1545 "overall_assessment": "comprehensive evaluation summary"}} """
1546
1547 CALCULATION_GRADER_USER_TEMPLATE = """Grade this CALCULATION solution
1548 with focus on final answer correctness.
1549
1550 PROBLEM:
1551 {problem_statement}
1552
1553 STUDENT SOLUTION:
1554 {solution}
1555
1556 CORRECT REFERENCE SOLUTION:
1557 {reference_solution}
1558
1559 Focus primarily on whether the final answer is correct. Return JSON with:
1560 {"grade": "CORRECT" or "INCORRECT",
1561 "detailed_feedback": "specific detailed analysis of what is right/wrong",
1562 "major_issues": "list of significant mathematical errors or gaps",
1563 "final_answer_correct": true or false,
1564 "reasoning_rigor_score": 0-10 integer (10=perfect rigor, 0=severely
1565 flawed),
1566 "overall_assessment": "comprehensive evaluation summary"}} """
1567
1568 # Response format for JSON output
1569 RESPONSE_FORMAT = {"type": "json_object"}
1570
1571 # Default retry and timeout settings

```

## 18 Appendix H

Listing 2: Example Question

```

1566
1567
1568
1569
1570
1571 {
1572   "index": "1938-A-2",
1573   "type": "ANA",
1574   "tag": [
1575     "ANA",
1576     "GEO"
1577   ],
1578   "difficulty": "1",
1579   "question": "2. A can buoy is to be made of three pieces, namely, a
1580     cylinder and two equal cones, the altitude of each cone being equal
1581     to the altitude of the cylinder. For a given area of surface, what
1582     shape will have the greatest volume?",
1583   "solution": "Solution. Let  $(r)$  be the radius of the cylinder, and
1584      $(h)$  its altitude. The given condition is  $S=2\pi r h + 2\pi r^2 \sqrt{h^2+r^2}$ 
1585     and the volume of the buoy is  $V=\pi r^2 h + \frac{2}{3}\pi r^2 h^3$ . The required
1586     problem is to find the maximum value of  $(V)$  subject to
1587     condition (1). This can be done by the method of Lagrange
1588     multipliers, but in this particular problem it is easier to solve
1589     (1) for  $(h)$  and express  $(V)$  as a function of  $(r)$ .
1590     We have  $(S-2\pi r h)^2=4\pi^2 r^2 (h^2+r^2)$ 
1591     hence  $h=\frac{S^2-4\pi^2 r^4}{2\pi r S}$ 
1592     and the expression for  $(V)$  becomes  $V=\frac{5}{12}r^3 S - \frac{2}{3}\pi r^5 \sqrt{S^2-4\pi^2 r^4}$ 
1593     Since  $(r)$  and  $(V)$  must be positive, the domain of interest is
1594     given by  $0 < r < \sqrt[4]{\frac{S^2}{4\pi^2}}$ . We
1595     compute the derivative and equate it to zero to get  $\frac{dV}{dr}=\frac{5}{4}r^2 S - 2\pi r^4 S = 0$ .
1596     The only critical value is  $r_0=\sqrt[4]{\frac{S^2}{20\pi^2}}$ 
1597     Since  $(V \rightarrow 0)$  as  $(r \rightarrow 0)$  or as  $(r \rightarrow \sqrt[4]{\frac{S^2}{4\pi^2}})$ ,
1598     and is positive in between, the critical value  $(r_0)$ 
1599     yields a maximum for  $(V)$ . The corresponding value of
1600      $(h)$  is found from (3) to be  $h_0=\frac{2}{5}\sqrt[5]{r_0^5}$ .
1601     The shape of the buoy is completely determined by the
1602     ratio  $\frac{h_0}{r_0}=\frac{2}{5}\sqrt[5]{\frac{r_0^5}{r_0^5}}$ ",
1603   "vars": [
1604     "r",
1605     "h",
1606     "V",
1607     "r_0",
1608     "h_0"
1609   ],
1610   "params": [
1611     "S"
1612   ],
1613   "sci_consts": [],
1614   "variants": {
1615     "descriptive_long": {
1616       "map": {
1617         "r": "radius",
1618         "h": "altitude",
1619         "V": "volume",
1620         "r_0": "criticalradius",
1621         "h_0": "criticalaltitude",
1622         "S": "surfacearea"
1623       },
1624       "question": "2. A can buoy is to be made of three pieces, namely, a
1625         cylinder and two equal cones, the altitude of each cone being

```

```

1620         equal to the altitude of the cylinder. For a given area of
1621         surface, what shape will have the greatest volume?",
1622     "solution": "Solution. Let  $r$  be the radius of the
1623         cylinder, and  $h$  its altitude. The given
1624         condition is  $2\pi r \sqrt{h^2 + r^2} = \text{constant}$ 
1625         and the volume of the buoy is  $V = \pi r^2 h + \frac{2}{3}\pi r^3$ 
1626         subject to condition (1). This can be done by the method of
1627         Lagrange multipliers, but in this particular problem it is
1628         easier to solve (1) for  $h$  and express  $V$  as a function of  $r$ . We have
1629          $2\pi r \sqrt{h^2 + r^2} = 4\pi^2 r^2 h$ 
1630         whence  $h = \frac{\sqrt{4\pi^2 r^2 - 4\pi r^4}}{4\pi r}$ 
1631         and the expression for  $V$  becomes  $V = \frac{5}{12} \frac{r^3 \sqrt{4\pi^2 r^2 - 4\pi r^4}}{\pi r}$ 
1632         Since  $r$  and  $V$  must be positive, the domain of interest is given by  $0 < r < \sqrt{4\pi}$ 
1633         We compute the derivative and equate it to zero to get  $\frac{dV}{dr} = \frac{5}{12} \sqrt{4\pi^2 r^2 - 4\pi r^4} - \frac{100\pi^2 r^3}{\sqrt{4\pi^2 r^2 - 4\pi r^4}} = 0$ 
1634         The only critical value is  $r = \sqrt{4\pi \frac{\sqrt{4\pi^2 r^2 - 4\pi r^4}}{20\pi^2}}$ 
1635         Since  $h > 0$  or as  $h = \sqrt{4\pi \frac{\sqrt{4\pi^2 r^2 - 4\pi r^4}}{4\pi}}$ , and is positive in
1636         between, the critical value  $r$  yields a maximum for  $V$ . The corresponding value of  $h$ 
1637         is found from (3) to be  $h = \frac{2}{5} \sqrt{5} r$ . The shape of the buoy
1638         is completely determined by the ratio  $\frac{h}{r} = \frac{2}{5} \sqrt{5}$ 
1639     },
1640     "descriptive_long_confusing": {
1641         "map": {
1642             "r": "monument",
1643             "h": "daybreak",
1644             "V": "calendar",
1645             "r_0": "monumental",
1646             "h_0": "daybreaker",
1647             "S": "landscape"
1648         },
1649         "question": "2. A can buoy is to be made of three pieces, namely, a
1650             cylinder and two equal cones, the altitude of each cone being
1651             equal to the altitude of the cylinder. For a given area of
1652             surface, what shape will have the greatest volume?",
1653         "solution": "Solution. Let  $r$  be the radius of the
1654             cylinder, and  $h$  its altitude. The given
1655             condition is  $2\pi r \sqrt{h^2 + r^2} = \text{constant}$ 
1656             and the volume of the buoy is  $V = \pi r^2 h + \frac{2}{3}\pi r^3$ 
1657             subject to condition (1). This can be done by the
1658             method of Lagrange multipliers, but in this particular problem
1659             it is easier to solve (1) for  $h$  and express  $V$  as a function of  $r$ . We have
1660              $2\pi r \sqrt{h^2 + r^2} = 4\pi^2 r^2 h$ 
1661             whence  $h = \frac{\sqrt{4\pi^2 r^2 - 4\pi r^4}}{4\pi r}$ 
1662             and the expression for  $V$  becomes  $V = \frac{5}{12} \frac{r^3 \sqrt{4\pi^2 r^2 - 4\pi r^4}}{\pi r}$ 
1663             Since  $r$  and  $V$  must be positive, the domain of interest is given by  $0 < r < \sqrt{4\pi}$ 
1664             We compute the derivative and equate it to zero to get  $\frac{dV}{dr} = \frac{5}{12} \sqrt{4\pi^2 r^2 - 4\pi r^4} - \frac{100\pi^2 r^3}{\sqrt{4\pi^2 r^2 - 4\pi r^4}} = 0$ 
1665             The only critical value is  $r = \sqrt{4\pi \frac{\sqrt{4\pi^2 r^2 - 4\pi r^4}}{20\pi^2}}$ 
1666             Since  $h > 0$  or as  $h = \sqrt{4\pi \frac{\sqrt{4\pi^2 r^2 - 4\pi r^4}}{4\pi}}$ , and is positive in
1667             between, the critical value  $r$  yields a maximum for  $V$ . The corresponding value of  $h$ 
1668             is found from (3) to be  $h = \frac{2}{5} \sqrt{5} r$ . The shape of the buoy
1669             is completely determined by the ratio  $\frac{h}{r} = \frac{2}{5} \sqrt{5}$ 
1670     },
1671     "question": "2. A can buoy is to be made of three pieces, namely, a
1672         cylinder and two equal cones, the altitude of each cone being
1673         equal to the altitude of the cylinder. For a given area of

```

```

1674 ndaybreak=\frac{landscape^2-4 \pi^2 monument^4}{4 \pi
1675 monument landscape}\n\nand the expression for \(\ calendar
1676 \) becomes\n\[\ncalendar=\frac{5 monument}{12 landscape}\n
1677 \left(landscape^2-4 \pi^2 monument^4\right)\n\n
1678 nSince \(\ monument \) and \(\ calendar \) must be positive,
1679 the domain of interest is given by\n\[\n0<monument<\sqrt[4]{
1680 landscape^2 / 4 \pi^2}\n\nWe compute the derivative
1681 and equate it to zero to get\n\[\n\frac{d calendar}{d
1682 monument}=\frac{5 landscape}{12}-\frac{100 \pi^2 monument
1683 ^4}{12 landscape}=0 .\n\nThe only critical value is\n
1684 \[\nmonumental=\sqrt[4]{\frac{landscape^2}{20 \pi^2}}\n
1685 \]\n\nSince \(\ calendar \rightarrow 0 \) as \(\ monument \rightarrow
1686 0 \) or as \(\ monument \rightarrow \sqrt[4]{
1687 landscape^2 / 4 \pi^2} \), and is positive in between,
1688 the critical value \(\ monumental \) yields a maximum for \(\
1689 calendar \). \n\nThe corresponding value of \(\ daybreak \) is
1690 found from (3) to be \(\ daybreaker=\frac{2}{5} \sqrt{5}
1691 monumental \). The shape of the buoy is completely determined
1692 by the ratio\n\[\n\frac{daybreaker}{monumental}=\frac{2}{5}
1693 \sqrt{5}\n\n\]
1694 },
1695 "descriptive_long_misleading": {
1696   "map": {
1697     "r": "perimeterlength",
1698     "h": "depthvalue",
1699     "V": "surfacearea",
1700     "r_0": "minimumdepth",
1701     "h_0": "maximumperimeter",
1702     "S": "corevolume"
1703   },
1704   "question": "2. A can buoy is to be made of three pieces, namely, a
1705     cylinder and two equal cones, the altitude of each cone being
1706     equal to the altitude of the cylinder. For a given area of
1707     surface, what shape will have the greatest volume?",
1708   "solution": "Solution. Let \(\ perimeterlength \) be the radius of
1709     the cylinder, and \(\ depthvalue \) its altitude. The given
1710     condition is\n\[\ncorevolume = 2 \pi perimeterlength
1711     depthvalue + 2\left(\pi perimeterlength \sqrt{depthvalue
1712     ^2+perimeterlength^2}\right)=\text{constant}\n\nand
1713     the volume of the buoy is\n\[\nsurfacearea = \pi
1714     perimeterlength^2 depthvalue + \frac{2 \pi perimeterlength
1715     ^2 depthvalue}{3}=\frac{5 \pi perimeterlength^2
1716     depthvalue}{3}\n\nThe required problem is to find the
1717     maximum value of \(\ surfacearea \) subject to condition (1).
1718     This can be done by the method of Lagrange multipliers, but in
1719     this particular problem it is easier to solve (1) for \(\
1720     depthvalue \) and express \(\ surfacearea \) as a function of
1721     \(\ perimeterlength \). We have\n\[\n(corevolume-2 \pi
1722     perimeterlength depthvalue)^2=4 \pi^2 perimeterlength
1723     ^2\left(depthvalue^2+perimeterlength^2\right)\n\n
1724     whence\n\[\ndeptvalue = \frac{corevolume^2-4 \pi^2
1725     perimeterlength^4}{4 \pi perimeterlength corevolume}\n\n
1726     and the expression for \(\ surfacearea \) becomes\n\[\[
1727     nsurfacearea = \frac{5 perimeterlength}{12 corevolume}\left(
1728     corevolume^2-4 \pi^2 perimeterlength^4\right)\n\n
1729     nSince \(\ perimeterlength \) and \(\ surfacearea \) must be
1730     positive, the domain of interest is given by\n\[\n0<
1731     perimeterlength<\sqrt[4]{corevolume^2 / 4 \pi^2}\n\n
1732     nWe compute the derivative and equate it to zero to get\n\[\n
1733     \frac{d surfacearea}{d perimeterlength}=\frac{5 corevolume
1734     }{12}-\frac{100 \pi^2 perimeterlength^4}{12 corevolume}=0
1735     .\n\nThe only critical value is\n\[\nminimumdepth=\sqrt[
1736     4]{\frac{corevolume^2}{20 \pi^2}}\n\nSince \(\
1737     surfacearea \rightarrow 0 \) as \(\ perimeterlength \rightarrow
1738     0 \) or as \(\ perimeterlength \rightarrow \sqrt[

```

```

1728         [4]{corevolume^{2} / 4 \pi^{2}} \), and is positive in
1729         between, the critical value \(\( minimumdepth \\) yields a
1730         maximum for \(\( surfacearea \)\). \n\nThe corresponding value of
1731         \(\( depthvalue \\) is found from (3) to be \(\( maximumperimeter
1732         = \frac{2}{5} \sqrt{5} minimumdepth \)\). The shape of the
1733         buoy is completely determined by the ratio \n\[\n\frac{
1734         maximumperimeter}{minimumdepth}=\frac{2}{5} \sqrt{5}\n\]
1735     },
1736     "garbled_string": {
1737         "map": {
1738             "r": "qzxwvtnp",
1739             "h": "yrklsfhd",
1740             "V": "mnbvcxza",
1741             "r_0": "ploikmnj",
1742             "h_0": "ujhytgrf",
1743             "S": "asdfghjk"
1744         },
1745         "question": "2. A can buoy is to be made of three pieces, namely, a
1746             cylinder and two equal cones, the altitude of each cone being
1747             equal to the altitude of the cylinder. For a given area of
1748             surface, what shape will have the greatest volume?",
1749         "solution": "Solution. Let \(\( qzxwvtnp \\) be the radius of the
1750             cylinder, and \(\( yrklsfhd \\) its altitude. The given
1751             condition is \n\[\n\frac{2 \pi qzxwvtnp yrklsfhd + 2 \left( \pi
1752             qzxwvtnp \sqrt{yrklsfhd^2 + qzxwvtnp^2} \right)}{\text{constant}}
1753             = \pi qzxwvtnp^2 yrklsfhd + \frac{2 \pi qzxwvtnp^2 yrklsfhd}{3}
1754             \] \n\] \nand the volume of the buoy is \n\[\n\frac{2 \pi qzxwvtnp^2
1755             yrklsfhd + \frac{2 \pi qzxwvtnp^2 yrklsfhd}{3}}{5} \] \n\] \n
1756             The required problem is to find the maximum value of \(\(
1757             mnbvcxza \\) subject to condition (1). This can be done by the
1758             method of Lagrange multipliers, but in this particular problem
1759             it is easier to solve (1) for \(\( yrklsfhd \\) and express \(\(
1760             mnbvcxza \\) as a function of \(\( qzxwvtnp \)\). We have \n\[\n(
1761             asdfghjk - 2 \pi qzxwvtnp yrklsfhd)^2 = 4 \pi^2 qzxwvtnp^2
1762             \left( yrklsfhd^2 + qzxwvtnp^2 \right) \n\] \nwhence \n\[\n
1763             yrklsfhd = \frac{asdfghjk^2 - 4 \pi^2 qzxwvtnp^4}{4 \pi qzxwvtnp
1764             asdfghjk} \n\] \nand the expression for \(\( mnbvcxza \\)
1765             becomes \n\[\n\frac{mnbvcxza}{5} = \frac{5 qzxwvtnp^{12} asdfghjk}{
1766             left( asdfghjk^2 - 4 \pi^2 qzxwvtnp^4 \right)} \n\] \nSince
1767             \(\( qzxwvtnp \\) and \(\( mnbvcxza \\) must be positive, the
1768             domain of interest is given by \n\[\n0 < qzxwvtnp < \sqrt[4]{
1769             asdfghjk^2 / 4 \pi^2} \n\] \nWe compute the derivative
1770             and equate it to zero to get \n\[\n\frac{d mnbvcxza}{d
1771             qzxwvtnp} = \frac{5 asdfghjk^{12}}{12 asdfghjk} - \frac{100 \pi^2
1772             qzxwvtnp^4}{nploikmnj \sqrt[4]{\frac{asdfghjk^2}{20 \pi^2}}} \n\] \n
1773             Since \(\( mnbvcxza \\) \(\rightarrow 0 \\) as \(\( qzxwvtnp \\)
1774             \(\rightarrow 0 \\) or as \(\( qzxwvtnp \\) \(\rightarrow \sqrt[4]{
1775             asdfghjk^2 / 4 \pi^2} \\) \), and is positive in between, the
1776             critical value \(\( ploikmnj \\) yields a maximum for \(\(
1777             mnbvcxza \)\). \n\nThe corresponding value of \(\( yrklsfhd \\) is
1778             found from (3) to be \(\( ujhytgrf = \frac{2}{5} \sqrt{5}
1779             ploikmnj \)\). The shape of the buoy is completely determined by
1780             the ratio \n\[\n\frac{ujhytgrf}{ploikmnj} = \frac{2}{5} \sqrt{
1781             5} \n\]
1782     },
1783     "kernel_variant": {
1784         "question": "A float is composed of a right circular cylinder of
1785             radius \(\( r \\) and altitude \(\( h \)\), with a right circular
1786             cone attached on top having the same base radius \(\( r \\) and
1787             altitude \(\( h/2 \)\). All the exterior surface is painted: the
1788             cylinder's lateral area, the cone's lateral area, and the exposed
1789             circular bottom of the cylinder. The circular interface between
1790             cone and cylinder is internal and unpainted. Given a fixed paint
1791             supply \(\( S \)\), determine the ratio \(\( h/r \\) that maximises

```

```

1782     enclosed volume. Provide the exact algebraic condition and a
1783     numerical approximation.",
1784     "solution": "Let  $(k = h/r > 0)$  be the desired ratio. Express
1785     every quantity in terms of  $(r)$  and  $(k)$ . 1. Painted area  $($ 
1786      $S = \pi r^2 + 2\pi r (k r) + \pi r \sqrt{r^2 + (k r/2)^2},$ 
1787      $) (= \pi r^2 + 2\pi k r^2 + \pi r^2 \sqrt{1 + k^2/4},)$ 
1788      $(= \pi r^2, F(k), \text{where } F(k) := 1 + 2k + \sqrt{1 +$ 
1789      $k^2/4},.)$  2. From this, with  $(S)$  fixed,  $(r = \sqrt{\frac{S}{\pi F(k)}}.)$  3. Volume  $($ 
1790      $V = \pi r^2 (k r) + \frac{1}{3}\pi r^2 \text{bigl}(k r/2\text{bigr}) (= \pi k r^3 + \frac{1}{6}\pi k r^3)$ 
1791      $(= \frac{7}{6}\pi k r^3)$   $(= \frac{7}{6}\pi k r^3)$   $(= \text{constant} \cdot$ 
1792      $G(k))$  with  $(G(k) := \frac{k}{F(k)^{3/2}}.)$ 
1793     Maximising  $(V)$  is therefore equivalent to maximising  $(G(k))$ 
1794     . 4. Set  $(g(k) = \ln G(k) = \ln k - \frac{2}{3}\ln F(k))$ .
1795     Then  $(g'(k) = \frac{1}{k} - \frac{2}{3}\frac{F'(k)}{F(k)})$ .
1796      $= 0.)$  Compute  $(F'(k) = 2 + \frac{k}{4\sqrt{1 + k$ 
1797      $^2/4}},)$  Setting  $(g'(k)=0)$  gives  $(\frac{2}{k} = \frac{3F'(k)}{F(k)})$ 
1798     Substituting  $(F)$  and  $(F')$  and clearing
1799     the square root yields  $(15k^3 - 32k^2 + 96k - 128 = 0.)$ 
1800     (*) 5. Polynomial (*) has exactly one positive root.
1801     Numerically one finds  $(k_{\max} = h/r \approx 1.55198)$  (to
1802     five significant figures). 6. End-point check: as  $(k \rightarrow$ 
1803      $0^+)$  or  $(k \rightarrow \infty), (G(k) \rightarrow 0)$ , so the critical
1804     point furnished by (*) indeed gives the absolute maximum of the
1805     volume for the prescribed paint area. Thus the cylinder should
1806     be about  $(1.552)$  times as tall as its radius; equivalently,
1807     the altitude of the cone is about  $(0.776, r)$ . Exact condition:
1808      $(15(h/r)^3 - 32(h/r)^2 + 96(h/r) - 128 = 0.)$ ",
1809     "_meta": {
1810       "core_steps": [
1811         "Express surface-area constraint  $S(r,h)$  and volume  $V(r,h)$  from
1812         geometry",
1813         "Solve the constraint for  $h$  (or use a Lagrange multiplier) to
1814         get  $V=V(r)$  alone",
1815         "Differentiate  $V(r)$ , set  $dV/dr = 0$ , locate admissible critical
1816          $r$ ",
1817         "Check endpoints to confirm the critical point yields the
1818         maximum",
1819         "Translate that  $r$  into the optimal  $h/r$  shape ratio"
1820       ],
1821       "mutable_slots": {
1822         "slot1": {
1823           "description": "How many identical cones are attached to the
1824           cylinder",
1825           "original": 2
1826         },
1827         "slot2": {
1828           "description": "Altitude of each cone as a multiple of the
1829           cylinder's altitude",
1830           "original": 1
1831         },
1832         "slot3": {
1833           "description": "Whether the flat circular bases are counted
1834           in the fixed surface area",
1835           "original": "not counted (only lateral areas used)"
1836         },
1837         "slot4": {
1838           "description": "Which quantity is held fixed vs. optimised (
1839           here  $S$  fixed,  $V$  maximised)",
1840           "original": "maximise volume subject to constant surface area"
1841         }
1842       }
1843     }
1844   }

```

```
1836     }
1837   },
1838   "checked": true,
1839   "problem_type": "proof"
1840 }
```

1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889

## 1890 19 Appendix I

1891  
1892 This appendix provides a 4 1930s' concrete ORIGINAL vs. Kernel-Variant (KV) examples for a  
1893 strong model(o3) from the instances we examined, complementing the aggregate robustness metrics  
1894 in Section 5. The examples were restricted to cases where the model solves the ORIGINAL correctly  
1895 but fails on the KERNEL-VARIANT, and we manually verified the failure annotations.

## 1896 20 1938-B-1

### 1897 20.1 Original question

1898 8. Take either (i) or (ii).

1899 (i) Let  $A_{ik}$  be the cofactor of  $a_{ik}$  in the determinant

$$1900 \mathbf{d} = \begin{vmatrix} 1901 a_{11} & a_{12} & a_{13} & a_{14} \\ 1902 a_{21} & a_{22} & a_{23} & a_{24} \\ 1903 a_{31} & a_{32} & a_{33} & a_{34} \\ 1904 a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix}.$$

1905 Let  $D$  be the corresponding determinant with  $a_{ik}$  replaced by  $A_{ik}$ . Prove that  $D = d^3$ . (page 86)

1906 (ii) Let  $P(y) = Ay^2 + By + C$  be a quadratic polynomial in  $y$ . If the roots of the quadratic equation  
1907  $P(y) - y = 0$  are  $a$  and  $b$  ( $a \neq b$ ), show that  $a$  and  $b$  are roots of the biquadratic equation

$$1908 P(P(y)) - y = 0.$$

1909 Hence write down a quadratic equation which will give the other two roots,  $c$  and  $d$ , of the bi-  
1910 quadratic.

1911 Apply this result to solving the following biquadratic equation:

$$1912 (y^2 - 3y + 2)^2 - 3(y^2 - 3y + 2) + 2 - y = 0.$$

### 1913 20.2 O3 solution on original

1914 (ii) Iterating a quadratic and locating its 2-cycle

#### 1915 1. Notation and preliminaries

1916 Let  $P(y) = Ay^2 + By + C$  with  $A \neq 0$ . Put

$$1917 f(y) = P(y) - y = Ay^2 + (B - 1)y + C$$

1918 and

$$1919 F(y) = P(P(y)) - y.$$

1920  $\deg(f) = 2, \deg(F) = 4$ .

#### 1921 2. The fixed points propagate to the second iterate

1922 If a root  $a$  of (1) satisfies  $f(a) = 0$ , then  $P(a) = a$ . Hence,

$$1923 F(a) = P(P(a)) - a = P(a) - a = 0,$$

1924 so every root of (1) is also a root of  $F$ . Thus two of the four roots of  $F$  are  $a, b$  (assumed  $a \neq b$ ).

#### 1925 3. Dividing $F(y)$ by $f(y)$

1926 We look for a quadratic  $g(y) = py^2 + qy + r$  such that

$$1927 F(y) = f(y)g(y).$$

1928 Write  $Q(y) = P(y) = Ay^2 + By + C$ .

1929 A short expansion gives

$$1930 F(y) = A^3y^4 + 2A^2By^3 + (AB^2 + 2A^2C + AB)y^2 \\ 1931 + (2ABC + B^2 - 1)y + (AC^2 + BC + C).$$

1944 Comparing the coefficients of  $y^4$  and  $y^3$  in (2)–(3) fixes

$$1945 \quad p = A^2, \quad q = A(B + 1).$$

1946  
1947  
1948 Matching the  $y^2$  and constant coefficients then gives

$$1949 \quad r = AC + B + 1.$$

1950  
1951  
1952 Consequently,

$$1953 \quad g(y) = A^2y^2 + A(B + 1)y + (AC + B + 1).$$

1954  
1955 Therefore,

$$1956 \quad P(P(y)) - y = (P(y) - y)(A^2y^2 + A(B + 1)y + (AC + B + 1)).$$

1957  
1958  
1959  
1960 **4. The other two roots**

1961 The remaining roots  $c, d$  of  $F(y) = 0$  are exactly the roots of

$$1962 \quad A^2y^2 + A(B + 1)y + (AC + B + 1) = 0.$$

1963  
1964  
1965  
1966 **5. Application to the given biquadratic**

1967 The given equation is

$$1968 \quad (y^2 - 3y + 2)^2 - 3(y^2 - 3y + 2) + 2 - y = 0,$$

1969  
1970 i.e.  $P(y) = y^2 - 3y + 2$  with  $A = 1, B = -3, C = 2$ .

1971  
1972 (a) *Roots fixed by  $P(y)$*

$$1973 \quad P(y) - y = y^2 - 4y + 2 = 0 \implies y = 2 \pm \sqrt{2}$$

1974  
1975 so  $a = 2 + \sqrt{2}, b = 2 - \sqrt{2}$ .

1976  
1977 (b) *Quadratic for the 2-cycle*

1978  
1979 Insert  $A = 1, B = -3, C = 2$  into (6):

$$1980 \quad y^2 + (-2)y + (2 - 3 + 1) = y^2 - 2y = 0 \implies y(y - 2) = 0.$$

1981  
1982 Thus  $c = 0, d = 2$ .

1983  
1984  
1985 **6. Verification (optional)**

$$1986 \quad y = 0: \quad P(0) = 2, \quad P(P(0)) = P(2) = 0 \implies F(0) = 0,$$

$$1987 \quad y = 2: \quad P(2) = 0, \quad P(P(2)) = P(0) = 2 \implies F(2) = 0.$$

1988  
1989 Hence the biquadratic splits as

$$1990 \quad (y - (2 + \sqrt{2}))(y - (2 - \sqrt{2}))y(y - 2) = 0.$$

1991  
1992 All four solutions are

$$1993 \quad y \in \{0, 2, 2 + \sqrt{2}, 2 - \sqrt{2}\}.$$

1998 **20.3 Kernel variant**

1999 Let  $n = 5$ . All matrices are  $5 \times 5$  and all equalities hold in the polynomial ring

2000 
$$R = \mathbb{Z}[a_{ik}]$$

2001 obtained by adjoining the 25 independent indeterminates  $a_{ik}$  to  $\mathbb{Z}$ .

2002 **1. Preliminaries**

- 2003
- 2004
- 2005 •  $\alpha = (a_{ik})$  original matrix
  - 2006 •  $d = \det \alpha$  its determinant
  - 2007 •  $A_{ik}$  cofactor of  $a_{ik}$  in  $d$
  - 2008 •  $\beta = (A_{ik})$  cofactor matrix (not transposed!)
  - 2009 •  $\gamma = \beta^T = \text{adj } \alpha$  classical adjugate
  - 2010 •  $\delta = \text{adj } \gamma$  adjugate taken twice

2011 Recall two well-known facts valid for every square matrix  $M$  of size  $n$ :

2012

2013 (F1) 
$$M \cdot \text{adj } M = \text{adj } M \cdot M = (\det M) I_n.$$

2014

2015

2016 (F2) If  $\det M$  is not a zero-divisor in the ground ring, then  $\text{adj } M = (\det M) M^{-1}$ .

2017 Because the determinant  $d$  of  $\alpha$  is an irreducible (hence non-zero) polynomial in  $R$ , it is not a zero-divisor; consequently we may use (1) for both  $\alpha$  and  $\gamma$ .

2018 **2. Proof of (i):  $\det \beta = d^4$**

2019 We have  $\gamma = \text{adj } \alpha$ , so by (F1)

2020 
$$\gamma \alpha = \alpha \gamma = d I_5.$$

2021 Taking determinants in (2) and using  $\det(d I_5) = d^5$ , we obtain

2022

2023 
$$(\det \gamma)(\det \alpha) = d^5 \implies \det \gamma = \frac{d^5}{d} = d^4.$$

2024 Because  $\beta$  and  $\gamma$  differ only by a transpose, they have the same determinant; hence

2025 
$$\det \beta = \det \gamma = d^4. \quad \square$$

2026 **3. Proof of (ii):  $\delta = d^3 \alpha$**

2027 Since  $\gamma$  is invertible when  $d \neq 0$  and  $\det \gamma = d^4$ , identity (1) gives

2028 
$$\text{adj } \gamma = (\det \gamma) \gamma^{-1} = d^4 \cdot (1/d) \alpha = d^3 \alpha.$$

2029 But both sides are polynomial matrices whose entries coincide on the open set  $\{d \neq 0\}$ ; therefore they are identical in the polynomial ring  $R$ . Hence

2030 
$$\delta = d^3 \alpha. \quad \square$$

2031 **4. Proof of (iii)**

2032 (a) *Determinant of  $\delta$ .*

2033 From (ii) we have  $\delta = d^3 \alpha$ , so

2034 
$$\det \delta = \det(d^3 \alpha) = (d^3)^5 \det \alpha = d^{15}. \quad \square$$

2035 (b) *Rank and right-kernel when  $d \neq 0$ .*

2036 If a specialisation of the indeterminates yields a numerical matrix with  $d \neq 0$ , then  $\alpha$  is invertible; hence

2037 
$$\text{rank } \alpha = 5, \quad \ker(\alpha) = \{0\}.$$

2038 Because  $\det \beta = d^4$ , the same specialisation also gives  $\det \beta \neq 0$ , so  $\beta$  is invertible, whence

2039 
$$\text{rank } \beta = 5, \quad \ker(\beta) = \{0\}.$$

2040 Thus  $\alpha$  and  $\beta$  have the same rank and the same right-kernel; explicitly, for every column vector  $v$ ,

2041 
$$\alpha v = 0 \iff v = 0 \iff \beta v = 0. \quad \square$$

2042 All three requested statements (i)–(iii) are thereby established.

2052 **20.4 O3 solution on KV**

2053

2054 Let  $n = 5$ . All matrices are  $5 \times 5$  and all equalities hold in the polynomial ring

2055

$$R = \mathbb{Z}[a_{ik}]$$

2056

2057 obtained by adjoining the 25 independent indeterminates  $a_{ik}$  to  $\mathbb{Z}$ .

2058 **Preliminaries.**

2059

2060 •  $\alpha = (a_{ik})$

original matrix

2061

2062 •  $d = \det \alpha$

its determinant

2063

2064 •  $A_{ik}$

cofactor of  $a_{ik}$  in  $d$

2065

2066 •  $\beta = (A_{ik})$

cofactor matrix (not transposed!)

2067

2068 •  $\gamma = \beta^T = \text{adj } \alpha$

classical adjugate

2069

2070 •  $\delta = \text{adj } \gamma$

adjugate taken twice

2071

2072 Recall two well-known facts valid for every square matrix  $M$  of size  $n$ :

2073

2074 (F1)  $M \cdot \text{adj } M = \text{adj } M \cdot M = (\det M) I_n.$

2075

2076

2077 (F2) If  $\det M$  is not a zero-divisor in the ground ring, then  $\text{adj } M = (\det M) M^{-1}.$

2078

2079 Because the determinant  $d$  of  $\alpha$  is an irreducible (hence non-zero) polynomial in  $R$ , it is not a zero-divisor; consequently we may use (1) for both  $\alpha$  and  $\gamma$ .

2080

2081 **Proof of (i):**  $\det \beta = d^4.$

2082

2083 We have  $\gamma = \text{adj } \alpha$ , so by (F1)

2084

$$\gamma \alpha = \alpha \gamma = d I_5.$$

2085

2086 Taking determinants in (2) and using  $\det(d I_5) = d^5$ , we obtain

2087

$$(\det \gamma)(\det \alpha) = d^5 \implies \det \gamma = \frac{d^5}{d} = d^4.$$

2088

2089 Because  $\beta$  and  $\gamma$  differ only by a transpose, they have the same determinant; hence

2090

$$\det \beta = \det \gamma = d^4. \quad \square$$

2091

2092 **Proof of (ii):**  $\delta = d^3 \alpha.$

2093

2094 Since  $\gamma$  is invertible when  $d \neq 0$  and  $\det \gamma = d^4$ , identity (1) gives

2095

$$\text{adj } \gamma = (\det \gamma) \gamma^{-1} = d^4 \cdot (1/d) \alpha = d^3 \alpha.$$

2096

2097 But both sides are polynomial matrices whose entries coincide on the open set  $\{d \neq 0\}$ ; therefore they are identical in the polynomial ring  $R$ . Hence

2098

$$\delta = d^3 \alpha. \quad \square$$

2099

2100 **Proof of (iii).**

2101

2102 (a) *Determinant of  $\delta$ .*

2103

2104 From (ii) we have  $\delta = d^3 \alpha$ , so

2105

$$\det \delta = \det(d^3 \alpha) = (d^3)^5 \det \alpha = d^{15}. \quad \square$$

2106

2107 (b) *Rank and right-kernel when  $d \neq 0$ .*

2108

2109 If a specialisation of the indeterminates yields a numerical matrix with  $d \neq 0$ , then  $\alpha$  is invertible; hence

2110

$$\text{rank } \alpha = 5, \quad \ker(\alpha) = \{0\}.$$

2111

2112 Because  $\det \beta = d^4$ , the same specialisation also gives  $\det \beta \neq 0$ , so  $\beta$  is invertible, whence

2113

$$\text{rank } \beta = 5, \quad \ker(\beta) = \{0\}.$$

Thus  $\alpha$  and  $\beta$  have the same rank and the same right-kernel; explicitly, for every column vector  $v$ ,

$$\alpha v = 0 \iff v = 0 \iff \beta v = 0. \quad \square$$

All three requested statements (i)–(iii) are thereby established.

**Failure types:** logic\_hallucination, step\_omission.

**Breakdown location.** The KV solution asserts  $\text{adj } M = (\det M) M^{-1}$  inside the ambient polynomial ring and claims it can be used for the parameters  $\alpha$  and  $\gamma$ , even though  $M^{-1}$  does not exist in that ring unless one localizes at  $\det M$ . This misuse first appears in the “Preliminaries” with (F2) and is then applied in proving (ii).

**Caption / takeaway.** KV misapplies the identity  $\text{adj } M = (\det M) M^{-1}$  in the polynomial ring without localization, leading to a rigor gap. The ORIGINAL stays within valid algebraic operations and justifies each step.

**Contrast summary.** The ORIGINAL solution proceeds entirely within an appropriate algebraic setting, expanding and factoring polynomials with fully justified coefficient comparisons. In contrast, the KV solution invokes a false general identity in the stated ring: it treats  $\text{adj } M = (\det M) M^{-1}$  as an equality in the polynomial ring without inverting  $\det M$ , effectively assuming inverses that do not exist there. It then bases the key step  $\delta = d^3\alpha$  on this illegitimate manipulation, only implicitly appealing to a localization without stating it. Although the final equalities hold over a field, the logical chain in the KV write-up fails at the foundational ring-theoretic level. This highlights a robustness failure where the model hallucinates a lemma and omits the necessary localization step.

## 21 1939-A-1

### 21.1 Original question

1. Find the length of the curve  $y^2 = x^3$  from the origin to the point where the tangent makes an angle of  $45^\circ$  with the  $x$ -axis.

### 21.2 O3 solution on original

Let the curve be given by  $y^2 = x^3$  (Neile’s parabola).

#### 1. Slope of the tangent

Differentiate implicitly:

$$2y \frac{dy}{dx} = 3x^2 \implies \frac{dy}{dx} = \frac{3x^2}{2y}.$$

The tangent makes an angle of  $45^\circ$  with the  $x$ -axis when its slope equals  $\tan 45^\circ = 1$ :

$$\frac{3x^2}{2y} = 1 \implies 3x^2 = 2y.$$

But on the curve  $y = x^{3/2}$ . Substitute into (1):

$$3x^2 = 2x^{3/2} \implies 3x^{1/2} = 2 \implies x = \left(\frac{2}{3}\right)^2 = \frac{4}{9}.$$

Hence  $y = (4/9)^{3/2} = \left(\frac{2}{3}\right)^3 = \frac{8}{27}$ . So the required point is

$$P\left(\frac{4}{9}, \frac{8}{27}\right).$$

#### 2. Parametrisation

Put  $x = t^2$ ,  $y = t^3$  ( $t \geq 0$ ). Then indeed

$$y^2 = t^6 = (t^2)^3 = x^3.$$

At  $P$  we have  $t^2 = 4/9 \implies t_0 = 2/3$ .

2160 **3. Arc-length element**  
 2161  
 2162

2163 
$$\frac{dx}{dt} = 2t, \quad \frac{dy}{dt} = 3t^2 \implies \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} = \sqrt{(2t)^2 + (3t^2)^2} = \sqrt{4t^2 + 9t^4} = t\sqrt{4 + 9t^2}.$$

2166 **4. Length from the origin ( $t = 0$ ) to  $P$  ( $t = 2/3$ )**  
 2167

2168 
$$L = \int_0^{2/3} t\sqrt{4 + 9t^2} dt.$$

2171 Substitute  $u = 4 + 9t^2$  ( $du = 18t dt \implies t dt = du/18$ ). When  $t = 0$ ,  $u = 4$ ; when  $t = 2/3$ ,  $u = 8$ .

2172  
 2173 
$$L = \frac{1}{18} \int_4^8 u^{1/2} du = \frac{1}{18} \left[ \frac{2}{3} u^{3/2} \right]_4^8 = \frac{1}{27} u^{3/2} \Big|_4^8.$$

2175 Compute the bounds:

2176 
$$u^{3/2} = u\sqrt{u} \implies 8^{3/2} = 8\sqrt{8} = 8 \cdot 2\sqrt{2} = 16\sqrt{2}, \quad 4^{3/2} = 4\sqrt{4} = 4 \cdot 2 = 8.$$

2178 Therefore

2179 
$$L = \frac{1}{27} (16\sqrt{2} - 8) = \frac{8}{27} (2\sqrt{2} - 1).$$

2181 Hence the length of the curve from the origin to the point where the tangent is at  $45^\circ$  is

2182 
$$\boxed{\frac{8}{27} (2\sqrt{2} - 1)}.$$

2186 **21.3 Kernel variant**  
 2187

2188 For the curve  $y^4 = 16x^6$  in the first quadrant, determine the exact length of the segment whose  
 2189 end-points are the two points at which the tangent to the curve forms angles of  $30^\circ$  and  $60^\circ$  with the  
 2190 positive  $x$ -axis.

2191 **21.4 O3 solution on KV**  
 2192

2193 Curve:  $y^4 = 16x^6$  (first quadrant  $\implies x > 0, y > 0$ ).

2194 **1. Compute the derivative.**  
 2195

2196 
$$y^4 = 16x^6 \implies 4y^3 \frac{dy}{dx} = 96x^5 \implies \frac{dy}{dx} = \frac{24x^5}{y^3}.$$

2199 **2. Express  $y$  and the derivative only in terms of  $x$ .**  
 2200

2201 
$$y = (16x^6)^{1/4} = 2x^{3/2}, \quad y^3 = (2x^{3/2})^3 = 8x^{9/2}.$$

2203 Hence

2204 
$$\frac{dy}{dx} = \frac{24x^5}{8x^{9/2}} = 3x^{1/2} = 3\sqrt{x}.$$

2207 **3. Slopes corresponding to the required tangents.**  
 2208

2209 
$$\theta = 30^\circ \implies m_1 = \tan 30^\circ = \frac{1}{\sqrt{3}}, \quad \theta = 60^\circ \implies m_2 = \tan 60^\circ = \sqrt{3}.$$

2210 **4. Find the  $x$ -coordinates where the derivative equals these slopes.**  
 2212

2213 (i)  $3\sqrt{x} = \frac{1}{\sqrt{3}} \implies \sqrt{x} = \frac{1}{3\sqrt{3}} \implies x_1 = \left(\frac{1}{3\sqrt{3}}\right)^2 = \frac{1}{27}.$

2214  
2215 (ii)  $3\sqrt{x} = \sqrt{3} \Rightarrow \sqrt{x} = \frac{1}{\sqrt{3}} \Rightarrow x_2 = \left(\frac{1}{\sqrt{3}}\right)^2 = \frac{1}{3}$ .

2216  
2217 **5. Corresponding  $y$ -coordinates (using  $y = 2x^{3/2}$ ).**

2218 (i)  
2219 
$$y_1 = 2x_1^{3/2} = 2\left(\frac{1}{27}\right)^{3/2} = 2 \cdot 3^{-9/2} = \frac{2}{81\sqrt{3}}.$$

2222 (ii)  
2223 
$$y_2 = 2x_2^{3/2} = 2\left(\frac{1}{3}\right)^{3/2} = \frac{2}{3\sqrt{3}}.$$

2226 Thus the points are

2227 
$$P_1\left(\frac{1}{27}, \frac{2}{81\sqrt{3}}\right), \quad P_2\left(\frac{1}{3}, \frac{2}{3\sqrt{3}}\right).$$

2229  
2230 **6. Vector between the two points.**

2231  
2232 
$$\Delta x = \frac{1}{3} - \frac{1}{27} = \frac{8}{27}, \quad \Delta y = \frac{2}{3\sqrt{3}} - \frac{2}{81\sqrt{3}} = \frac{52}{81\sqrt{3}}.$$

2234  
2235 **7. Distance  $L$ .**

2236  
2237 
$$L^2 = (\Delta x)^2 + (\Delta y)^2 = \left(\frac{8}{27}\right)^2 + \left(\frac{52}{81\sqrt{3}}\right)^2 = \frac{64}{729} + \frac{2704}{19683}.$$

2239 Bring to the common denominator  $19683 = 3^9$ :

2241 
$$\frac{64}{729} = \frac{64 \cdot 27}{19683} = \frac{1728}{19683}.$$

2244 Thus

2245 
$$L^2 = \frac{1728 + 2704}{19683} = \frac{4432}{19683} = \frac{16 \cdot 277}{3^9}.$$

2247 Therefore

2248 
$$L = \sqrt{\frac{16 \cdot 277}{3^9}} = \frac{4\sqrt{277}}{3^{9/2}} = \frac{4\sqrt{277}}{81\sqrt{3}}.$$

2251 **8. Rationalising the denominator (optional).**

2252  
2253 
$$L = \frac{4\sqrt{277}}{81\sqrt{3}} \cdot \frac{\sqrt{3}}{\sqrt{3}} = \frac{4\sqrt{831}}{243}.$$

2256 Thus the exact length of the required segment is

2257 
$$\boxed{\frac{4\sqrt{277}}{81\sqrt{3}}} \quad \text{equivalently} \quad \boxed{\frac{4\sqrt{831}}{243}}.$$

2261 **Failure types:** step\_omission, logic\_hallucination.

2262 **Breakdown location.** After correctly locating the two tangent-angle points, the KV solution (steps  
2263 6–7) computes the straight-line (chord) distance between them instead of setting up and evaluating  
2264 the arc-length integral along the curve.

2265 **Caption / takeaway.** KV computes chord length instead of arc length. The ORIGINAL sets up the  
2266 proper arc-length integral, while KV omits it and implicitly equates straight-line distance with curve  
2267 length.

2268 **Contrast summary.** In the ORIGINAL, after identifying the target point via the slope condition, the  
 2269 solver explicitly parametrizes the curve and integrates the arc-length element to obtain the true length  
 2270 along the curve. The KV solution likewise finds the correct points where the tangents have the spec-  
 2271 ified angles, but then switches to computing the Euclidean distance between those points, implicitly  
 2272 treating chord length as arc length. This omits the essential arc-length setup  $\int \sqrt{1 + (dy/dx)^2} dx$   
 2273 (or an equivalent parametrization) that the ORIGINAL employs. As a result, the KV produces a  
 2274 numerically smaller, incorrect value. The contrast highlights a classic failure: computing the wrong  
 2275 physical quantity despite correct intermediate calculus steps.

## 2276 22 1939-B-1

### 2277 22.1 Original question

2280 8. From the vertex  $(0, c)$  of the catenary

$$2281 y = c \cosh\left(\frac{x}{c}\right)$$

2282 a line  $L$  is drawn perpendicular to the tangent to the catenary at a point  $P$ . Prove that the length of  
 2283  $L$  intercepted by the axes is equal to the ordinate  $y$  of the point  $P$ .

### 2284 22.2 O3 solution on original

2285 Let  $P$  be an arbitrary point of the catenary

$$2286 y = c \cosh\left(\frac{x}{c}\right), \quad c > 0.$$

2287 Write  $P = (a, y_P)$  with

$$2288 y_P = c \cosh\left(\frac{a}{c}\right).$$

#### 2289 1. Slope of the tangent at $P$ .

2290 Differentiate (1):

$$2291 \frac{dy}{dx} = \sinh\left(\frac{x}{c}\right).$$

2292 Hence the slope of the tangent at  $x = a$  is

$$2293 m_t = \sinh\left(\frac{a}{c}\right).$$

#### 2294 2. Equation of the tangent at $P$ .

2295 Using the point-slope form through  $P(a, y_P)$ :

$$2296 y - y_P = m_t(x - a).$$

#### 2297 3. Slope and equation of the required line $L$ .

2298 Line  $L$  passes through the vertex  $V(0, c)$  and is perpendicular to the tangent, so its slope is

$$2299 m_L = -\frac{1}{m_t} = -\frac{1}{\sinh\left(\frac{a}{c}\right)}.$$

2300 Through  $V(0, c)$ :

$$2301 y - c = m_L x \implies y = c - \frac{x}{\sinh\left(\frac{a}{c}\right)}.$$

#### 2302 4. Intercepts of $L$ with the coordinate axes.

- 2303 •  $y$ -intercept: put  $x = 0$  in (5)  $\implies y = c$ . Thus  $V$  itself is the  $y$ -intercept.

2322 •  $x$ -intercept: put  $y = 0$  in (5):

$$2323 \quad 0 = c - \frac{x}{\sinh\left(\frac{a}{c}\right)} \implies x = c \sinh\left(\frac{a}{c}\right).$$

2326 Call this point  $B$ :

$$2327 \quad B\left(c \sinh\left(\frac{a}{c}\right), 0\right).$$

### 2329 5. Length of the segment $AB$ intercepted by the axes.

2330  $A \equiv V = (0, c)$ , and  $B$  is found above. The Euclidean distance is

$$2332 \quad |AB| = \sqrt{\left(c \sinh\left(\frac{a}{c}\right)\right)^2 + (0 - c)^2}$$

$$2333 \quad = c \sqrt{\sinh^2\left(\frac{a}{c}\right) + 1}.$$

2338 Using the hyperbolic identity  $\cosh^2 u - \sinh^2 u = 1$ ,

$$2339 \quad \sqrt{\sinh^2 u + 1} = \cosh u.$$

2342 Therefore,

$$2343 \quad |AB| = c \cosh\left(\frac{a}{c}\right).$$

### 2346 6. Comparison with the ordinate of $P$ .

2347 From (1) we have  $y_P = c \cosh\left(\frac{a}{c}\right)$ . Equation (6) shows that

$$2349 \quad |AB| = y_P.$$

2351 Hence, the length of the segment of the line  $L$  intercepted between the coordinate axes equals precisely the ordinate of the point  $P$ .

## 2354 22.3 Kernel variant

2355 Fix  $\lambda > 0$  and any integer  $n \geq 2$ . Write

$$2357 \quad r(x_1, \dots, x_{n-1}) = \sqrt{x_1^2 + \dots + x_{n-1}^2}.$$

2359 Consider the  $(n - 1)$ -dimensional “radial catenary”

$$2361 \quad x_n = \lambda \cosh\left(\frac{r}{\lambda}\right) \quad (\text{so the vertex is } V = (0, \dots, 0, \lambda)).$$

2364 For a chosen radius  $\rho \geq 0$ , set

$$2365 \quad Q = (\rho u, \lambda \cosh(\rho/\lambda)),$$

2366 where  $u$  is a unit vector in  $\mathbb{R}^{n-1}$ .

2368 Let  $H$  be the hyperplane through  $V$  that is orthogonal to the tangent hyperplane of the surface at  $Q$ .

2369 Prove that the segment of  $H$  cut off by the hyperplanes

$$2371 \quad x_n = 0 \quad \text{and} \quad x_1 = \dots = x_{n-1} = 0$$

2372 has length

$$2373 \quad \lambda \cosh(\rho/\lambda).$$

2375

2376 **22.4 O3 solution on KV**

2377

2378 **Step 1. Reduction to a 2-dimensional problem**

2379

Because the surface

2380

2381

$$S : x_n = \lambda \cosh(r/\lambda), \quad r = \sqrt{x_1^2 + \dots + x_{n-1}^2},$$

2382

2383

2384

is invariant under every rotation about the  $x_n$ -axis, we may, without loss of generality, rotate the coordinate system so that the given unit vector  $u$  equals the first basis vector  $e_1$ . After the rotation the point chosen on the surface is

2385

2386

$$Q = (\rho, 0, \dots, 0, \lambda \cosh(\rho/\lambda)).$$

2387

2388

2389

Let  $\Pi$  be the coordinate 2-plane spanned by the axes  $x_1$  and  $x_n$  (all the other coordinates equal 0). In that plane the surface  $S$  appears as the usual planar catenary

2390

$$y = \lambda \cosh(x/\lambda), \quad (y := x_n, x := x_1 \geq 0),$$

2391

2392

with vertex  $V = (0, \lambda)$ .

2393

2394

2395

2396

All the geometric objects involved (the tangent hyperplane at  $Q$ , the required hyperplane  $H$ , the coordinate hyperplanes  $x_n = 0$  and  $x_1 = \dots = x_{n-1} = 0$ ) are rotationally symmetric; consequently the problem reduces to showing, inside the plane  $\Pi$ , that the line  $L = H \cap \Pi$  has the stated property. Once that is done, the same length is obtained in the whole space  $\mathbb{R}^n$ .

2397

2398

**Step 2. Tangent line to the catenary in  $\Pi$**

2399

For the planar catenary (1) we have

2400

$$y'(x) = \sinh(x/\lambda).$$

2401

2402

At the abscissa  $x = \rho$  this gives the slope

2403

$$m := y'(\rho) = \sinh(\rho/\lambda).$$

2404

2405

Hence the tangent line at

2406

$$Q_{\Pi} = (\rho, \lambda \cosh(\rho/\lambda))$$

2407

2408

reads

2409

$$y - \lambda \cosh(\rho/\lambda) = m(x - \rho).$$

2410

2411

Its direction vector is  $(1, m)$ .

2412

2413

2414

**Step 3. The required line  $L = H \cap \Pi$**

2415

2416

2417

Inside  $\Pi$  the hyperplane  $H$  intersects as a line  $L$  passing through the vertex  $V$  and perpendicular to (3). Two lines in the plane are perpendicular when the product of their slopes equals  $-1$ ; therefore the slope of  $L$  is  $-1/m$ . Writing its equation through  $V = (0, \lambda)$  we get

2418

2419

$$y - \lambda = -\frac{1}{m} x, \quad \text{i.e.} \quad y = \lambda - \frac{x}{\sinh(\rho/\lambda)}.$$

2420

2421

2422

**Step 4. The two intersection points of  $L$**

2423

2424

(i) With the horizontal axis  $y = 0$  (i.e. with the hyperplane  $x_n = 0$ ):

2425

2426

Setting  $y = 0$  in (4) gives  $x = \lambda \sinh(\rho/\lambda)$ . Thus

2427

$$A = (\lambda \sinh(\rho/\lambda), 0) \quad \text{in } \Pi,$$

2428

2429

that is,

$$A = (\lambda \sinh(\rho/\lambda) e_1, 0, \dots, 0) \quad \text{in } \mathbb{R}^n.$$

2427

2428

2429

(ii) With the vertical axis  $x = 0$  (i.e. with  $x_1 = \dots = x_{n-1} = 0$ ):

Putting  $x = 0$  in (4) gives  $y = \lambda$ ; hence the point is exactly the vertex

$$B = V = (0, \dots, 0, \lambda).$$

2430 **Step 5. Length of the segment  $AB$**

2431 Inside  $\Pi$  the Euclidean distance between  $A$  and  $B$  is

2432 
$$|AB|^2 = (\lambda \sinh(\rho/\lambda))^2 + \lambda^2 = \lambda^2(\sinh^2(\rho/\lambda) + 1) = \lambda^2 \cosh^2(\rho/\lambda),$$

2433 because  $\cosh^2 u - \sinh^2 u = 1$ .

2434 Therefore

2435 
$$|AB| = \lambda \cosh(\rho/\lambda).$$

2436 Because the coordinates of  $A$  and  $B$  in  $\mathbb{R}^n$  differ only in the first and the last component, the same distance is obtained in the full space. Hence the segment of  $H$  cut off by the hyperplanes  $x_n = 0$  and  $x_1 = \dots = x_{n-1} = 0$  indeed has length  $\lambda \cosh(\rho/\lambda)$ , as was to be proved.

2437 **Failure types:** logic\_hallucination, step\_omission.

2438 **Breakdown location.** In Step 1 the KV asserts that the tangent hyperplane at  $Q$  and the perpendicular hyperplane  $H$  are rotationally symmetric, wrongly justifying reduction to the  $x_1-x_n$  plane; the correct rationale (that the normal lies in that plane, so  $H$  is contained in it) is missing. From Step 2 onward it also relies on  $m = \sinh(\rho/\lambda)$  in denominators, breaking down at  $\rho = 0$ .

2439 **Caption / takeaway.** Faulty symmetry reduction and edge-case omission in the higher-dimensional catenary: incorrect invariance claim and failure at  $\rho = 0$ , versus a complete 2D argument.

2440 **Contrast summary.** The ORIGINAL solution works entirely in 2D, computing the perpendicular through the vertex and showing its intercept length equals the ordinate, with all steps justified. The KV attempts to generalize via a rotational-symmetry reduction to a 2D slice, but this symmetry claim is false because the tangent hyperplane depends on the chosen direction  $u$  and is not rotation-invariant. The correct reason the problem reduces to the  $x_1-x_n$  plane is that the normal  $(\sinh(\rho/\lambda)u, -1)$  lies in that plane, which the KV omits. Moreover, the KV ignores the degenerate case  $\rho = 0$  where the slope vanishes, making its formulas ill-defined; a separate check is required. Thus the KV exhibits both a faulty geometric reduction and an unhandled edge case, despite correct computations when  $\rho > 0$ .

2450 **23 1939-B-7**

2451 **23.1 Original question**

2452 14. Take either (i) or (ii).

2453 (i) If

2454 
$$\nu = 1 + \frac{x^3}{3!} + \frac{x^6}{6!} + \dots$$

2455 
$$v = \frac{x}{1!} + \frac{x^4}{4!} + \frac{x^7}{7!} + \dots$$

2456 
$$w = \frac{x^2}{2!} + \frac{x^5}{5!} + \frac{x^8}{8!} + \dots,$$

2457 prove that

2458 
$$\nu^3 + v^3 + w^3 - 3\nu v w = 1.$$

2459 (ii) Consider the central conics

2460 
$$(ax^2 + by^2) + 2(px + qy) + c = 0$$

2461 
$$(ax^2 + by^2) + 2\lambda(px + qy) + \lambda^2 c = 0$$

2462 where  $\lambda$  is a given positive constant.

2463 Show that if all radii from the origin to the first conic are changed in the ratio  $\lambda$  to 1 the tips of these new radii generate the second conic.

2464 Let  $P$  be the point with coordinates

2465 
$$x = -\frac{p}{a} \frac{2\lambda}{1 + \lambda}, \quad y = -\frac{q}{b} \frac{2\lambda}{1 + \lambda}.$$

2484 Show that if all radii from  $P$  to the first conic are changed in the ratio  $\lambda$  to 1 and then reversed about  
 2485  $P$  the tips of these new radii generate the second conic.

2486 Comment on these results in case  $\lambda = 1$ .  
 2487

### 2488 23.2 O3 solution on original

#### 2489 Problem (i).

2490 Let

$$2491 u(x) = 1 + \frac{x^3}{3!} + \frac{x^6}{6!} + \cdots = \sum_{k=0}^{\infty} \frac{x^{3k}}{(3k)!}, \quad v(x) = \frac{x}{1!} + \frac{x^4}{4!} + \frac{x^7}{7!} + \cdots = \sum_{k=0}^{\infty} \frac{x^{3k+1}}{(3k+1)!},$$

$$2492 w(x) = \frac{x^2}{2!} + \frac{x^5}{5!} + \frac{x^8}{8!} + \cdots = \sum_{k=0}^{\infty} \frac{x^{3k+2}}{(3k+2)!}.$$

2493 Denote

$$2494 f(x) = u^3 + v^3 + w^3 - 3uvw.$$

2495 We shall prove that  $f(x) \equiv 1$ .

#### 2496 1. A useful cyclic-derivative property.

2497 Differentiate the three series term by term:

- 2498 • For  $k \geq 1$ ,

$$2499 \frac{d}{dx} \left( \frac{x^{3k}}{(3k)!} \right) = \frac{x^{3k-1}}{(3k-1)!}.$$

2500 The exponent  $3k - 1 \equiv 2 \pmod{3}$ , hence the term belongs to  $w(x)$ .

- 2501 • Similarly,  $\frac{d}{dx}(v) = u$  and  $\frac{d}{dx}(w) = v$ .

2502 Therefore,

$$2503 u' = w, \quad v' = u, \quad w' = v.$$

#### 2504 2. The derivative of $f(x)$ .

2505 Using (1) and the product rule,

$$2506 f'(x) = 3u^2u' + 3v^2v' + 3w^2w' - 3(vw u' + uv v' + uv w')$$

$$2507 = 3(u^2w + v^2u + w^2v - vw^2 - u^2w - uv^2)$$

$$2508 = 3(v^2u - uv^2 + w^2v - vw^2) = 0.$$

2509 Each bracketed pair cancels identically; hence  $f'(x) = 0$  for all real  $x$ .

#### 2510 3. Evaluation at $x = 0$ .

$$2511 u(0) = 1, \quad v(0) = 0, \quad w(0) = 0 \implies f(0) = 1^3 + 0 + 0 - 0 = 1.$$

#### 2512 4. Conclusion.

2513 Because  $f'(x) = 0$  and  $f(0) = 1$ , the function  $f(x)$  is constant. Therefore

$$2514 u^3 + v^3 + w^3 - 3uvw = 1 \quad \text{for all real } x.$$

2515 *Remark.* An alternative (though longer) proof uses the roots-of-unity filter

$$2516 u = \frac{1}{3}(e^x + e^{\omega x} + e^{\omega^2 x}), \quad \omega = e^{2\pi i/3},$$

2517 followed by standard symmetric-polynomial manipulations. The differential proof above is the most economical.

2538 **23.3 Kernel variant**

2539 Problem.

2540 (i) Define three power series

2541 
$$U(x) = 2 + \frac{x^3}{3!} + \frac{x^6}{6!} + \frac{x^9}{9!} + \dots,$$

2542 
$$V(x) = \frac{x}{1!} + \frac{x^4}{4!} + \frac{x^7}{7!} + \frac{x^{10}}{10!} + \dots,$$

2543 
$$W(x) = \frac{x^2}{2!} + \frac{x^5}{5!} + \frac{x^8}{8!} + \frac{x^{11}}{11!} + \dots.$$

2544 Show that, for every real  $x$ ,

2545 
$$U^3 + V^3 + W^3 - 3UVW = 8.$$

2546 (ii) Let  $\kappa > 0$  be fixed and consider the two central conics

2547 
$$C : (\alpha x^2 + \beta y^2) + 2(rx + sy) + d = 0,$$

2548 
$$D : (\alpha x^2 + \beta y^2) + 2\kappa(rx + sy) + \kappa^2 d = 0,$$

2549 where  $\alpha, \beta (\neq 0), r, s, d$  are real constants.

2550 (a) Show that multiplying every radius vector from the origin to  $C$  by the factor  $\kappa$  sends  $C$  onto  $D$ .

2551 (b) Let

2552 
$$P\left(-\frac{2\kappa}{1+\kappa} \frac{r}{\alpha}, -\frac{2\kappa}{1+\kappa} \frac{s}{\beta}\right).$$

2553 Prove that if each radius vector from  $P$  to  $C$  is first multiplied by  $\kappa$  and then reversed (i.e. multiplied by  $-1$ ), the resulting tips trace exactly the conic  $D$ .

2554 (c) Discuss what happens in parts (a) and (b) when  $\kappa = 1$ .

2555 **23.4 O3 solution on KV**

2556 **Problem (i)**

2557 1. Let  $\omega = e^{2\pi i/3}$  (so  $\omega^3 = 1$  and  $1 + \omega + \omega^2 = 0$ ). Split the exponential series

2558 
$$e^x = \sum_{n \geq 0} \frac{x^n}{n!}$$

2559 into its three residue classes modulo 3.

2560 
$$A(x) := \sum_{k \geq 0} \frac{x^{3k}}{(3k)!}, \quad B(x) := \sum_{k \geq 0} \frac{x^{3k+1}}{(3k+1)!}, \quad C(x) := \sum_{k \geq 0} \frac{x^{3k+2}}{(3k+2)!}.$$

2561 Then

2562 
$$U(x) = 1 + A(x) + 1 = A(x) + 1 + 1?? \text{ (BUT in the problem } U \text{ starts with } 2 \dots)$$

2563 hence

2564 
$$U = A + 1, \quad V = B, \quad W = C.$$

2565 2. Closed form of  $A, B, C$ .

2566 The three linear equations

2567 
$$A + B + C = e^x,$$

2568 
$$A + \omega B + \omega^2 C = e^{\omega x},$$

2569 
$$A + \omega^2 B + \omega C = e^{\omega^2 x}$$

2591

2592 give, by Cramer,

$$2593 \quad A = \frac{e^x + e^{\omega x} + e^{\omega^2 x}}{3},$$

$$2594 \quad B = \frac{e^x + \omega^2 e^{\omega x} + \omega e^{\omega^2 x}}{3},$$

$$2595 \quad C = \frac{e^x + \omega e^{\omega x} + \omega^2 e^{\omega^2 x}}{3}.$$

2596 All three are real-valued functions of the real variable  $x$ , and (2) is easily seen to reproduce exactly  
2600 the three power series of  $A, B, C$ .

2601 3. *Elementary symmetric functions of  $A, B, C$ .*

$$2602 \quad \sigma_1 := A + B + C = e^x,$$

$$2603 \quad \sigma_2 := AB + BC + CA = \frac{e^{2x} - e^{-x}}{3},$$

$$2604 \quad \sigma_3 := ABC.$$

2605 (3b) is obtained by multiplying the first two lines of (2) and using  $\omega^2 + \omega = -1$  together with  
2609  $e^{\omega x} e^{\omega^2 x} = e^{-x}$ . A direct (but slightly longer) calculation gives

$$2610 \quad A^3 + B^3 + C^3 - 3ABC = 1.$$

2611 4. *The required identity.*

2612 With (1) write

$$2613 \quad S(x) := U^3 + V^3 + W^3 - 3UVW$$

$$2614 \quad = (A+1)^3 + B^3 + C^3 - 3(A+1)BC$$

$$2615 \quad = (A^3 + B^3 + C^3 - 3ABC) + (3A^2 + 3A + 1) - 3BC.$$

2616 Insert (4):

$$2617 \quad S(x) = 1 + (3A^2 + 3A + 1) - 3BC.$$

2618 Now express  $BC$  through  $\sigma_1, \sigma_2$ :

$$2619 \quad BC = \sigma_2 - A(\sigma_1 - A) = \sigma_2 - Ae^x + A^2.$$

2620 Substitute (3b) and (7) in (6):

$$2621 \quad S(x) = 1 + 3A + 1 - 3 \left[ \frac{e^{2x} - e^{-x}}{3} - Ae^x + A^2 \right]$$

$$2622 \quad = 2 + 3A + e^{-x} - e^{2x} + 3Ae^x - 3A^2 + 3A^2$$

$$2623 \quad = 2 + e^{-x} - e^{2x} + 3A(1 + e^x).$$

2624 Using (2) one directly checks that the right-hand side of (8) is identically equal to 8 (a short veri-  
2628 fication is to note that both sides are entire functions satisfying the same third-order ODE  $f''' = f$   
2632 and the same initial data at  $x = 0$ ). Hence

$$2629 \quad S(x) \equiv 8 \quad \text{for every real } x.$$

2630 In particular  $S(0) = 2^3 = 8$ ; therefore (9) holds for all  $x$ .

---

2640 **Problem (ii)**

2641 For convenience put  $X := (x, y)^T$  in column-vector notation. The equations of the two conics can  
2643 be rewritten as

$$2644 \quad X^T K X + 2p^T X + d = 0, \quad (C)$$

2646  
2647  
2648  
2649  
2650  
2651  
2652  
2653  
2654  
2655  
2656  
2657  
2658  
2659  
2660  
2661  
2662  
2663  
2664  
2665  
2666  
2667  
2668  
2669  
2670  
2671  
2672  
2673  
2674  
2675  
2676  
2677  
2678  
2679  
2680  
2681  
2682  
2683  
2684  
2685  
2686  
2687  
2688  
2689  
2690  
2691  
2692  
2693  
2694  
2695  
2696  
2697  
2698  
2699

$$X^T K X + 2\kappa p^T X + \kappa^2 d = 0, \quad (D)$$

with the diagonal matrix  $K := \text{diag}(\alpha, \beta)$  ( $\alpha\beta \neq 0$ ) and the vector  $p := (r, s)^T$ .

---

**(a) Radial homothety with centre  $O$ .**

---

Multiply a point  $X \in C$  by the factor  $\kappa$ , i.e. send it to

$$X \mapsto \kappa X.$$

Then

$$(\kappa X)^T K (\kappa X) + 2p^T (\kappa X) + d = \kappa^2 X^T K X + 2\kappa p^T X + d = 0$$

because  $X$  fulfils (C). Dividing by  $\kappa^2$  gives exactly (D). Hence the mapping  $X \mapsto \kappa X$  sends  $C$  onto  $D$ .

---

**(b) Radial homothety with centre  $P$  followed by a half-turn.**

---

Let

$$P = \left( -\frac{2\kappa r}{(1+\kappa)\alpha}, -\frac{2\kappa s}{(1+\kappa)\beta} \right)^T$$

and define the map

$$F(X) := P - \kappa(X - P) = (1 + \kappa)P - \kappa X.$$

(10) reverses the vector  $P \rightarrow X$  and multiplies its length by  $\kappa$ .

Write  $X' := F(X)$  and substitute in the equation (D):

$$X'^T K X' + 2\kappa p^T X' + \kappa^2 d.$$

By (10)  $X' = (1 + \kappa)P - \kappa X$ , so

$$X'^T K X' = \kappa^2 X^T K X - 2\kappa(1 + \kappa)P^T K X + (1 + \kappa)^2 P^T K P.$$

A short calculation, using the explicit coordinates of  $P$ , shows

$$P^T K = -\frac{\kappa}{1 + \kappa} p^T, \quad P^T K P = \frac{\kappa^2}{(1 + \kappa)^2} d.$$

Insert these three facts:

$$\begin{aligned} X'^T K X' + 2\kappa p^T X' + \kappa^2 d &= \kappa^2 X^T K X - 2\kappa(1 + \kappa)P^T K X + (1 + \kappa)^2 P^T K P \\ &\quad + 2\kappa p^T [(1 + \kappa)P - \kappa X] + \kappa^2 d \\ &= \kappa^2 [X^T K X + 2p^T X + d] = 0, \end{aligned}$$

because the bracket vanishes for every  $X$  on  $C$ . Thus the image curve is exactly  $D$ .

---

**(c) The special case  $\kappa = 1$ .**

---

With  $\kappa = 1$  the two conics coincide (their equations are identical).

(a) The mapping  $X \mapsto \kappa X$  is the identity, so nothing changes.

(b) Formula (10) becomes  $F(X) = 2P - X$ , i.e. the half-turn (central symmetry) about  $P$ . When  $C = D$ , a  $180^\circ$  rotation about  $P$  leaves the conic fixed (a usual symmetry of every central conic).

**Failure types:** step\_omission, logic\_hallucination, arithmetic.

**Breakdown location.** In part (i) the KV solution asserts  $A^3 + B^3 + C^3 - 3ABC = 1$  without proof and then claims  $S(x) = 8$  via a wrong ODE argument ( $f''' = f$ ), which a constant cannot satisfy.

2700 In part (ii) it mishandles the effect of the scaling  $X \mapsto \kappa X$  on the linear and constant terms and  
2701 computes incorrect identities for  $P$  (missing factors), so the reduction to  $D$  is unfounded.

2702 **Caption / takeaway.** Clean cyclic-derivative cancellation vs. an overcomplicated roots-of-  
2703 unity/ODE shortcut and mishandled scaling. The KV fails by omitting a key identity, using an  
2704 invalid ODE argument, and mis-scaling conic coefficients.

2705 **Contrast summary.** The ORIGINAL solves part (i) by exploiting the cyclic derivative identities  
2706  $u' = w, v' = u, w' = v$  to show  $f'(x) = 0$  and then fixes the constant by  $f(0) = 1$ , a short and  
2707 airtight argument. The KV instead uses a roots-of-unity decomposition, leaves a pivotal symmetric  
2708 identity unproved, and finally appeals to an incorrect ODE invariance to conclude  $S(x) \equiv 8$ . In  
2709 the conic mapping, the ORIGINAL approach (analogous to the statement) respects how quadratic,  
2710 linear, and constant terms scale, whereas the KV's matrix computation drops necessary  $\kappa$  factors and  
2711 miscomputes properties of  $P$ , breaking the cancellation to  $D$ . The pair highlights how a clean struc-  
2712 tural identity beats an overengineered approach and how small coefficient errors derail geometric  
2713 transformations.

2714  
2715  
2716  
2717  
2718  
2719  
2720  
2721  
2722  
2723  
2724  
2725  
2726  
2727  
2728  
2729  
2730  
2731  
2732  
2733  
2734  
2735  
2736  
2737  
2738  
2739  
2740  
2741  
2742  
2743  
2744  
2745  
2746  
2747  
2748  
2749  
2750  
2751  
2752  
2753