Can LLMs Reliably Grade Olympiad Proofs? A Study of Mathematical Verification with LLMs

Azim Ospanov[1,2*] **Zijin Feng**[1†] **Ding Ding**[1,3] **Chengwu Liu**[1] **Haoli Bai**[1]
Jiacheng Sun[1] **Lifeng Shang**[1] **Farzan Farnia**[2,†]
[1]Huawei Foundation Model Department
[2]The Chinese University of Hong Kong
[3]The Hong Kong University of Science and Technology

Abstract

Large Language Models (LLMs) have made remarkable progress on solving competition-level mathematics, yet their ability to verify natural-language mathematical proofs remains relatively underexplored. This gap is especially pronounced for Olympiad-level proofs, which require meticulous assessment of each claim in a long reasoning chain. Current proof verification thus primarily relies on expert inspection, which is costly to scale. In this paper, we present a systematic study of LLM-based verification in this setting. Specifically, we introduce a human-curated dataset of USAMO 2025 candidate solutions with expert grading, and conduct a broad evaluation of existing LLM verifiers across frontier models. We also propose *TROJAN*, an iterative self-critique pipeline for generating high-fidelity adversarial proofs at scale. Our comprehensive evaluation covers 20 models spanning seven metrics, two prompt styles, three inference methods, and three prompt templates, showing that state-of-the-art models still leave considerable room for improvement. To narrow this gap, we propose *MABGrader*, a Multi-Armed Bandit framework that reframes grading as arm selection over the discrete score space. *MABGrader* outperforms the SOTA ProofGrader by 26.4% in Quadratic Weighted Kappa and reduces grading error by 14%. Code and dataset are available at `https://github.com/aziksh-ospanov/TROJAN`.

1 Introduction

Mathematical problem-solving is a core capability of large language models (LLMs). Recent frontier models have demonstrated remarkable proficiency in competition-level mathematics, including the International Mathematical Olympiad (IMO) and the United States of America Mathematical Olympiad (USAMO) [58, 24]. Much of this progress, however, is measured by final-answer correctness, which is a limited proxy for complex mathematical reasoning since a solution may reach the correct answer while containing logical gaps, unjustified claims, or subtle errors [47, 33]. Moreover, many proving problems do not admit a final answer, requiring costly expert verification.

These motivate scalable methods for evaluating natural-language mathematical solutions. In this work, we study the verification of natural-language mathematical solutions. Specifically, an LLM-based verifier receives a mathematical problem, a candidate solution, and optionally a reference solution and a marking scheme; it must then identify reasoning errors and assign a fair grade [33, 4]. Recent work has approached this task by constructing proof-verification benchmarks and designing LLM-based graders: MathArena [4] evaluates LLMs on competition proof problems using expert graders to assess model-generated solutions, ProofAutoGrader [32] constructs IMO-GradingBench and studies automatic grading with reference solutions and guidelines, and ProofGrader [33] analyzes how model choice, grading schemes, reference solutions, and ensembling affect agreement with expert.

*First author: azimospanov@outlook.com
†Corresponding authors: fengzijinn@gmail.com, farnia@cse.cuhk.edu.hk

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

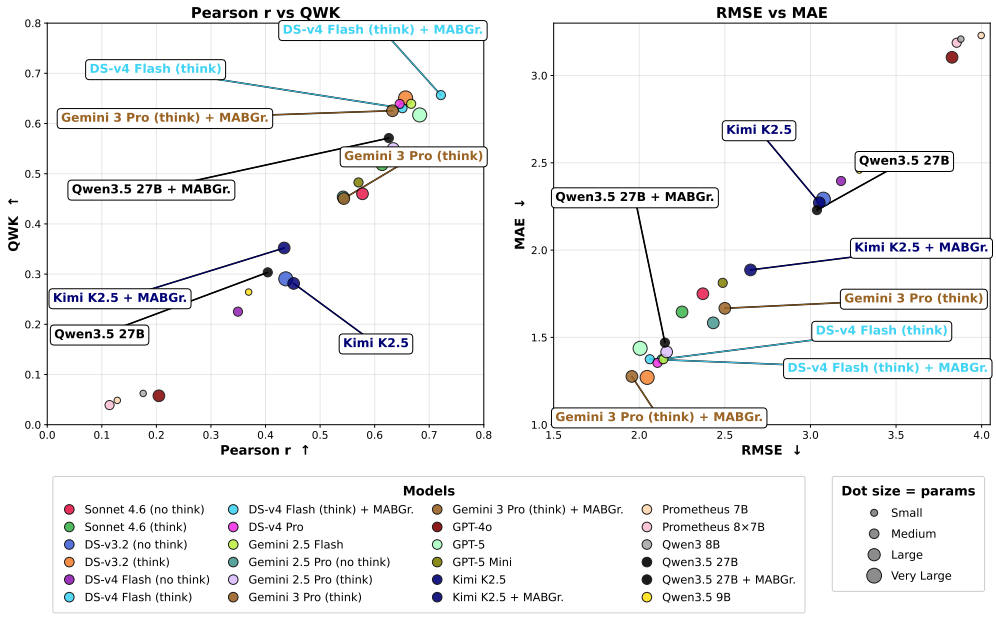

Figure 1: Performance of models on *Trojan-Bench(H)* , comparing Majority@16 with our `MABGr.` inference method across 4 metrics. Color and marker size denotes the model and parameter scale, resp. The spread indicates considerable room for improvement, while `MABGr.` show consistent gains.

Despite this progress, two challenges remain for Olympiad-level mathematical verification. First, existing datasets often collect LLM-generated candidate solutions and rely on human experts to grade them, which is costly, difficult to scale, and provides limited control over the error distribution [4, 33]. This makes it difficult to diagnose which types of mistakes LLM can or cannot detect. Second, verification itself remains difficult. Competition proofs are long, multi-step arguments in which a single mistake or omission can invalidate later conclusions or alter the appropriate partial credit. A verifier must thus go beyond surface plausibility by localizing the failure point and assessing its effect on the final grade. Figure 1 illustrates this gap. Existing approaches, including prompt engineering, PRM/ORM-based selection [29, 57], and LLM-judge ensembles [8, 22], can improve verification performance, but often require careful finetuning and stacking of different models. These limitations motivate diagnostic benchmarks for targeted error analysis and inference methods that improve grading without modifying the underlying base model.

In this work, we present a systematic study of LLM-based verification for Olympiad-level mathematics. We introduce a human-curated benchmark dataset derived from the USAMO 2025, where rubric-aligned adversarial errors, such as imprecise notation and unjustified lemma applications, are deliberately injected into candidate solutions. This design enables targeted evaluation of whether LLM judges can detect specific reasoning flaws and assign appropriate partial credit, while providing greater control over error type and severity than spontaneous LLM-generated mistakes. To scale data generation, we propose an iterative self-critique pipeline called *TROJAN* (**T**argeted adve**R**sarial err**O**r in**J**ection for m**A**th evaluatio**N**) that produces adversarial solutions with human-like flaws. We show that fine-tuning on this synthetic data improves agreement with expert grading. We further propose *MABGrader*, an agentic Multi-Armed Bandit framework for mathematical verification. Rather than producing a verification judgment in one pass, *MABGrader* formulates verification as an optimal selection problem over candidate judgments, where each arm corresponds to a possible grade with supporting error analysis. Through iterative sampling and reward optimization, it allocates more computation to uncertain cases and improves grade selection without modifying the underlying base model. Our contributions are summarized as follows:

- We introduce *Trojan-Bench(H)*, a human-curated dataset of error-injected USAMO 2025 solutions for evaluating how LLM verifiers detect subtle reasoning flaws and assign grades.
- We propose *TROJAN*, a self-critique pipeline for generating high-fidelity adversarial samples, and show that training on the resulting *Trojan-Bench(S)* improves alignment with experts.
- We conduct a systematic evaluation of 20 models across multiple prompts, inference methods, and metrics, measuring their verification abilities on Olympiad-level mathematics.

- We introduce *MABGrader*, a Multi-Armed Bandit algorithm that mitigates LLM verification instability and provides a robust alternative to single- and multi-shot grading methods.

2 Related Works

Benchmarks for Mathematical Reasoning. The mathematical evaluation includes foundational datasets such as GSM8K [9] and MATH [29], alongside advanced benchmarks like AIME [10], HARDMath2 [46], and CollegeMath [49]. Recent research has focused on competition-level mathematics through specialized benchmarks including Omni-MATH [18], Olympiad-Bench [23], and HARP [63]. Modern frameworks such as MathArena [4] provide model rankings based on challenging problems sampled from fresh Olympiads. Beyond benchmarking, recent breakthroughs have led to systems that participate directly in the International Mathematical Olympiad (IMO) [24, 21, 53]. These advancements demonstrate that models are moving toward achieving performance comparable to top human competitors.

LLM-as-a-Judge in mathematics. LLM-as-a-Judge frameworks utilize large language models to evaluate responses based on metrics such as correctness and fairness [22, 65, 28, 52]. Recent studies on LLM reliability show that while these models achieve high correlation with human experts on standard tasks [48, 17], they remain susceptible to issues such as sycophancy [43]. However, when grading complex mathematical theorems, many researchers continue to rely on human experts, a practice that limits the scalability of such evaluation pipelines [33, 32, 59, 42].

Verification in formal mathematics. Complementing informal evaluation, formal mathematics utilizes proof assistants such as Lean [34], Coq [5], and Isabelle [35] to achieve automated verification. Benchmarks in this domain rely on the underlying language kernel to confirm proof validity, offering an objective alternative to natural language assessment [64, 39, 54, 62]. Recent research also employs compiler feedback to guide LLM generation [38, 30] and integrates informal reasoning steps into formal proofs to enhance verifiability [40, 25, 6].

Industry standards on fair grading. According to [13], in math competitions with binary grading, the agreement is approximately 90.4%. As we expand this to a 0–7 scale, the agreement and correlation will naturally be lower. Although those figures are not directly translatable to our setting, the consensus is that prestigious math Olympiads and new mathematical proofs should have a high rate of agreement among judges and reviewers. The intended gold standard involves multiple graders, a rubric, and a reconciliation process, which is stricter than single-rater "good enough" labeling. For general educational purposes, [60] reports an absolute $QWK_{human,LLM} \geq 0.70$ as an acceptable metric for automated scoring. This is not a universal rule of thumb, as different domains have different requirements. We use these guidelines as a conservative lower bound for model evaluation. In this work, we systematically examine the current state of LLM grading to clarify the performance of automated judging against human labels.

3 Problem Formulation

In this work, we study the verification of natural-language mathematical solutions. Given a mathematical problem p, a candidate solution $\tilde{s}$, and optionally a reference solution `Ref` and a marking scheme `MS`, the goal is for an LLM-based verifier to identify reasoning errors and assign a grade that fairly reflects the correctness and completeness of the solution. Following common practice in [4, 32, 33], grades range from 0 to 7.

We consider two verification settings, illustrated in Figure 2. In the Full setting, corresponding to the `Ref+MS` configuration of [33], the verifier is provided with both the reference solution and the marking scheme specifying point allocations and deductions. In the Blind setting, corresponding to the `None` configuration of [33], the verifier receives only the problem and candidate solution, without access to either a reference solution or a marking scheme.

4 Dataset Construction

To comprehensively evaluate whether LLMs can verify Olympiad-level proofs and assign appropriate grades, in this section we describe the construction of our human-curated *Trojan-Bench(H)* benchmark and the proposed data generation pipeline *TROJAN*, which we use to obtain dataset *Trojan-Bench(S)* .

Problem: Let k and d be positive integers. Prove that there exists a positive integer N such that for every odd integer $n > N$, the digits in the base-$2n$ representation of n^k are all greater than d.

Full Setting	Blind Setting
Reference solution: We claim one such N is $\boxed{N = 2^{k-1}(d+1)}$ ...	NO REFERENCE SOLUTION
Marking Scheme: *Initial closed form expression for each digit [1 pt]*: Uses the base-2n representation...	NO MARKING SCHEME
Candidate Solution: Begin by letting a_i be the digit at the ith position from the...	**Candidate Solution:** Begin by letting a_i be the digit at the ith position from the...

Figure 2: Example verification inputs for USAMO 2025 Problem 1 under the Full and Blind settings. In the Full setting, the verifier receives the problem, reference solution, marking scheme, and candidate solution. In the Blind setting, the verifier receives only the problem and candidate solution.

4.1 Human curated dataset

We construct *Trojan-Bench(H)* to support controlled evaluation of LLM-based proof verification. Unlike existing datasets that collect LLM-generated solutions and annotate them with human grades, we use a backward construction procedure: starting from golden reference solutions, we map proof components to marking-scheme criteria and systematically inject errors that correspond to targeted point deductions. We build our dataset *Trojan-Bench(H)* from the USAMO 2025 competition set, which contains six proof-based problems across multiple areas of mathematics. Grades range from 0 to 7 according to the marking schemes. We use the USAMO 2025 problems, reference solutions, and marking schemes provided by MathArena [4].

Concretely, we construct candidate solutions by injecting subtle errors into the golden solution, disrupting the proof structure and leading to specific point deductions. For each problem, we randomly sample a target grade and introduce errors designed to match the corresponding marking-scheme deductions. Furthermore, when constructing faulty solutions, we applied heavier penalties to fundamental parts of the proof, such as operating in base-$2n$ in Problem 1 or establishing the correct definition and structure of cupcake segments in Problem 6. A logical gap in a fundamental step typically causes a cascade of point losses across other criteria. The 2026 version of the USAMO in [4] follows a similar logic by specifying overarching deductions.

To account for the inherent variance in individual grading, we utilized two independent annotators for the final dataset. We followed the data creation pipeline from [45] to increase the validity and reliability of the final set. However, we omitted the final step of non-expert validation due to the specialized nature of mathematical proofs. The result is a set of 48 human-verified USAMO 2025 candidate solutions with a corresponding human grade and a justification. We provide a detailed description of the creation pipeline and aggregated metrics in the following sections, and a full datasheet can be found in the Appendix Section A.

Table 1: Inter-annotator agreement on our human-curated benchmark *Trojan-Bench(H)*

Metric	QWK	Pearson r	RMSE	MAE	$\text{WTA}_{\leq 1}$	$\text{WTA}_{\leq 2}$	Exact Match	Bias
Value	0.76	0.89	1.09	0.62	0.85	0.94	0.58	± 0.08

Expert Inter-Rater Agreement. Table 1 provides the agreement metrics for the human-curated dataset following the procedure in [45]. The results show a Quadratic Weighted Kappa (QWK) of 0.76 and a Pearson correlation of 0.89, which exceed the performance of any of the evaluated LLMs and indicate strong expert consensus. Scoring consistency is further reflected in the Within Tolerance Agreement, where 85% of scores fall within 1 point ($WTA_{\leq 1}$) and 94% fall within

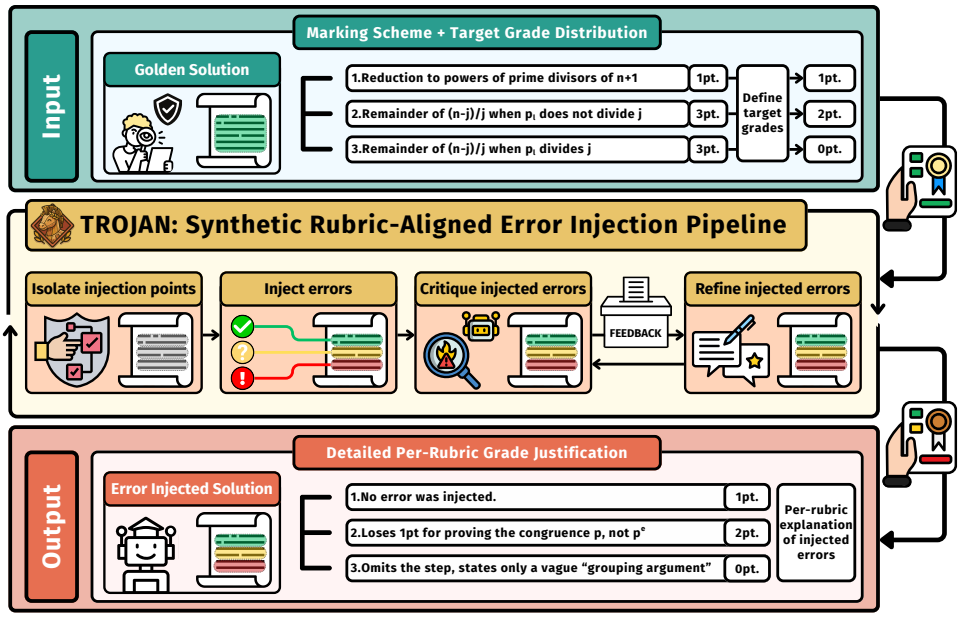

Figure 3: Overview of our synthetic data generation pipeline *TROJAN*. Given a problem, reference solution, marking scheme, and target score profile, the pipeline injects rubric-aligned errors, performs iterative self-critique and refinement, and outputs an adversarial proof with a grade justification.

2 points ($WTA_{\leq 2}$). Combined with a low systematic bias of ± 0.08, these metrics establish the human-curated set as a reliable gold standard for assessing automated grading systems.

4.2 Synthetic dataset generation

To scale benchmark construction beyond manual error injection, we introduce *TROJAN*, an LLM-based pipeline for generating adversarial mathematical proofs. Given a mathematical problem, a reference golden solution, a marking scheme, and a target score profile, *TROJAN* outputs a proof whose errors are calibrated to be consistent with the specified score profile, together with a structured grade justification. Figure 3 provides an overview of the pipeline.

The score profile specifies the target point allocation across marking-scheme criteria. The generator LLM is given one human-authored error-injection example for one-shot prompting and then analyzes the marking scheme. For criteria assigned full marks, it is allowed to closely paraphrase the corresponding part of the reference solution; for each criterion with points losses, it is tasked with injecting errors calibrated to the target score. The same LLM then performs self-critique using a separate prompt. The goal is to isolate the parts that contain mistakes and produce an explicit self-judgment. Its output is an alignment judgment together with revision guidance. In the refinement stage, the model applies the suggested revisions to the corresponding proof blocks. This critique-refinement loop terminates once the self-critique model accepts the current candidate $n_{\text{alignment}}$ consecutive rounds; if alignment is broken before then, the streak is discarded and refinement continues with the latest critique. The maximum number of critique-refinement rounds is bounded by n_{iter}.

The pipeline is general and can be applied to any problem set with valid reference solutions and marking schemes. We construct a large synthetic dataset *Trojan-Bench(S)* from four competitions: USAMO 2025, USAMO 2026, IMO 2025, and IMC 2025, with problem statements, reference solutions, with data sourced from the MathArena repository [4]. When evaluating the synthetic data, we treat this grade and justification as ground-truth labels. Experiments on the subset of *Trojan-Bench(S)* show that it closely correlates with *Trojan-Bench(H)*, suggesting good alignment.

Table 2: Correlation and agreement of the synthetically-generated benchmark *Trojan-Bench(S)* with expert graders.

Metric	QWK	Pearson r	RMSE	MAE	$\text{WTA}_{\leq 1}$	$\text{WTA}_{\leq 2}$	Exact Match	Bias
Value	0.78	0.87	1.20	0.77	0.83	0.90	0.67	0.11

Table 3: Inter-annotator agreement on the synthetically-generated benchmark *Trojan-Bench(S)*.

Metric	QWK	Pearson r	RMSE	MAE	$\text{WTA}_{\leq 1}$	$\text{WTA}_{\leq 2}$	Exact Match	Bias
Value	0.78	0.94	0.70	0.49	1.00	1.00	0.51	± 0.00

Audit of *Trojan-Bench(S)*. To further validate the quality of the proposed *TROJAN*, we performed a manual audit of the *Trojan-Bench(S)* subset. We sampled 50 problems and annotated using similar procedures as in the construction and audit of *Trojan-Bench(H)*. Each problem was verified by two experts and we present the correlation, error and bias metrics between LLM and human assigned grades in Table 2. We observe that generally the correlation is high and $\sim 83\%$ of problems fall within one grading points. This generally indicates that the proposed rubric-based error injection helps LLMs inject relevant and appropriate synthetic errors into the correct solutions. We also report the inter-annotation agreement in Table 3 between the experts in this audit and show that the consensus was high when grading, with all graded problems falling into $WTA_{\leq 1}$. Combined, this shows robustness and overall high quality of synthetically injected errors via *TROJAN*.

5 Our Approach

To address the difficulty of assigning fair grades to Olympiad-level problems, we implemented a Multi-Armed Bandit (MAB) algorithm for grade selection. Even with access to a mark scheme and a reference solution, producing a grade outright is often hard. Prior work also suggests that evaluating a pre-assigned grade together with its justification is easier than scoring from scratch, since the candidate grade provides a concrete anchor for assessment. [9, 29]. Independently, evidence from the LLM-as-a-judge and educational assessment literatures shows that aggregating multiple independent evaluators of the same candidate reduces individual-judge bias and improves agreement with expert ratings [55, 65, 52, 16].

Motivated by this, we first considered an A/B-style voting scheme that, given a grade range, argues for one grade against another. This approach is token-hungry: its token cost grows linearly with the number of comparisons, and it wastes pulls on clearly sub-optimal arms whose exploration cost is high. We therefore adopt UCB [27, 3], a well-studied bandit algorithm which, to our knowledge, has not previously been applied to mathematical grading. We compare them in the Appendix section N.

At the core of our algorithm is a judgment-defense setup. For each candidate grade $g \in \{0, \ldots, 7\}$, we prompt an LLM to justify the grade and record its justification along with a self-reported confidence. In the next phase, the same model, now prompted as a judging oracle, selects arms and assigns rewards. This recovers the classic stochastic bandit setting. After each round, *MABGrader* aggregates the oracle's rewards and updates the arm statistics, balancing exploration and exploitation to converge on the grade closest to the underlying true grade.

At round t, let $N_g(t)$ denote the number of times arm g has been pulled prior to round t, let $r_s \in [0, 1]$ denote the reward observed at pull s, and let

$$\hat{\mu}_g(t) \;=\; \frac{1}{N_g(t)} \sum_{s=1}^{t-1} r_s \, \mathbb{1}[g_s = g] \,, \qquad N_g(t) \;=\; \sum_{s=1}^{t-1} \mathbb{1}[g_s = g]$$

denote the empirical mean and pull count of arm g. UCB first plays each arm once, then at every subsequent round selects

$$g_t \;=\; \arg\max_{g \in \{0,\ldots,7\}} \Big[\underbrace{\hat{\mu}_g(t)}_{\text{exploitation}} \;+\; \underbrace{c\sqrt{\tfrac{2 \ln t}{N_g(t)}}}_{\text{exploration}} \Big],$$

breaking ties uniformly at random. The default $c = 1$ recovers the classical UCB1 bound of [3], which yields $\mathcal{O}(\sqrt{KT \ln T})$ regret for $K = 8$ arms with rewards in $[0, 1]$; the oracle's 1-5 fairness Likert is rescaled to $[0, 1]$ via $r = (\tilde{r} - 1)/4$. The parameter c is user-controlled and can be adapted to different problems. The design is compatible with any single-arm sampling bandit, including Exp3, ϵ-greedy, and Thompson Sampling. For this study we chose a single, well-established algorithm and leave a comparison with alternatives to future work. We provide more details in the Appendix Section I.

6 Experiments

6.1 Experimental Setup

Inference Settings. In all experiments, we fix the sampling parameters at $top_p = 0.95$ and temperature $t = 0.1$ to minimize LLM-induced variance during the grading process. Unless otherwise noted, we employ the "thinking" versions of the evaluated models. For models that support adjustable reasoning effort, the setting was fixed to "high". Maximum token limits vary by model and are detailed in the Appendix. The default testing environment follows the MathArena template [4]. Every run was initialized with a seed of 42. We explicitly indicate where an alternative template was used for comparison.

All multi-sample experiments are performed at @16, where we sample 16 independent attempts from scratch. To evaluate Best-of-N methods, we utilize two Outcome Reward Models (ORMs): Skywork-Reward-Llama-3.1-8Bv0.2 [31] and ArmoRM-Llama3-8Bv0.1 [56]. We also incorporate two Process Reward Models (PRMs): math-shepherd-mistral-7b-prm [57] and RLHFlow/Llama3.1-8B-PRMDeepseek-Data [14]. For our main *MABGrader* results we used 16 number of iterations and $c = 0.25$ guided by our ablation result.

Evaluation Metrics. We use seven metrics to measure alignment between verifier-generated grades and ground truth grades.

$$\text{QWK} = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}}, \qquad \text{WTA}_\tau = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{|y_i - \hat{y}_i| \le \tau\}, \qquad \text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \qquad \text{Pearson}_r = \frac{\sum_i (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_i (y_i - \bar{y})^2}\sqrt{\sum_i (\hat{y}_i - \bar{\hat{y}})^2}}, \qquad \text{EM} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{y_i = \hat{y}_i\}.$$

Here, n is the number of grading samples; y_i and $\hat{y}_i$ the ground-truth and verifier-generated grades for the i-th sample, respectively; and $\bar{y}$ and $\bar{\hat{y}}$ their corresponding means. QWK measures ordinal agreement while penalizing larger grade discrepancies, with O and E denoting the observed and expected agreement matrices, respectively, and weights $w_{i,j} = (i-j)^2/(N-1)^2$ over the $N = 8$ possible grades. RMSE measures overall grading error, WTA_τ measures the fraction of predictions within tolerance τ, Pearson r measures linear correlation, MAE measures average absolute grading error, and EM measures exact grade agreement. We also report Bias, defined as $\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)$, with positive and negative values indicating over- and under-grading, respectively.

Data Generation. For synthetic data generation, we utilized Claude Opus 4.6 (Think) [1]. Because existing literature indicates a high self-validation bias [7], we exclude this model from our evaluation suite. We use a temperature of $t = 0.7$ for data generation and refinement, while the critic module employs a more conservative $t = 0.3$ to reduce variance in the feedback. We also finetuned two models, Qwen3.5-9B and 27B SFT, on *Trojan-Bench(S)*. The model cards are specified in Appendix.

6.2 Analysis of *Trojan-Bench(S)*

LLM performance on the human-annotated benchmark. We present the performance of 14 models on the human-annotated benchmark using the QWK metric in Figure 4. In this section, we focus primarily on the correlation metric and provide a detailed breakdown of all other metrics in the Appendix Sections D, E and F. Our evaluation spans two scenarios, Blind and Full, and incorporates three sampling methods: CoT@1 (a single inference), MeanCoT@16 (the averaged grade rounded to the nearest integer), and MajorityVote@16 (selecting the grade with the highest frequency).

Consistent with the findings in [33], the Full setting is, on average, a less challenging task than the Blind setting. We observe that models achieve a 0.06 higher absolute correlation and a 24.2% relative improvement in performance when using MajorityVote@16. Interestingly, smaller models such as GPT-5-mini or Qwen3-8B achieve higher correlation when not provided with additional information. Upon further inspection, we found that these models are often unable to follow the gold solution and instead produce hallucinated responses. A deeper case analysis is provided in the Appendix Section J.1. Overall, the models remain below the 0.70 threshold even when supplied with full grading information and instructions, whereas expert annotators achieve a 0.76 agreement rate. Furthermore, LLM performance collapses to a maximum of 0.53 in the Blind setting. This scenario is particularly difficult as the models must both derive a correct solution and grade the attempt based

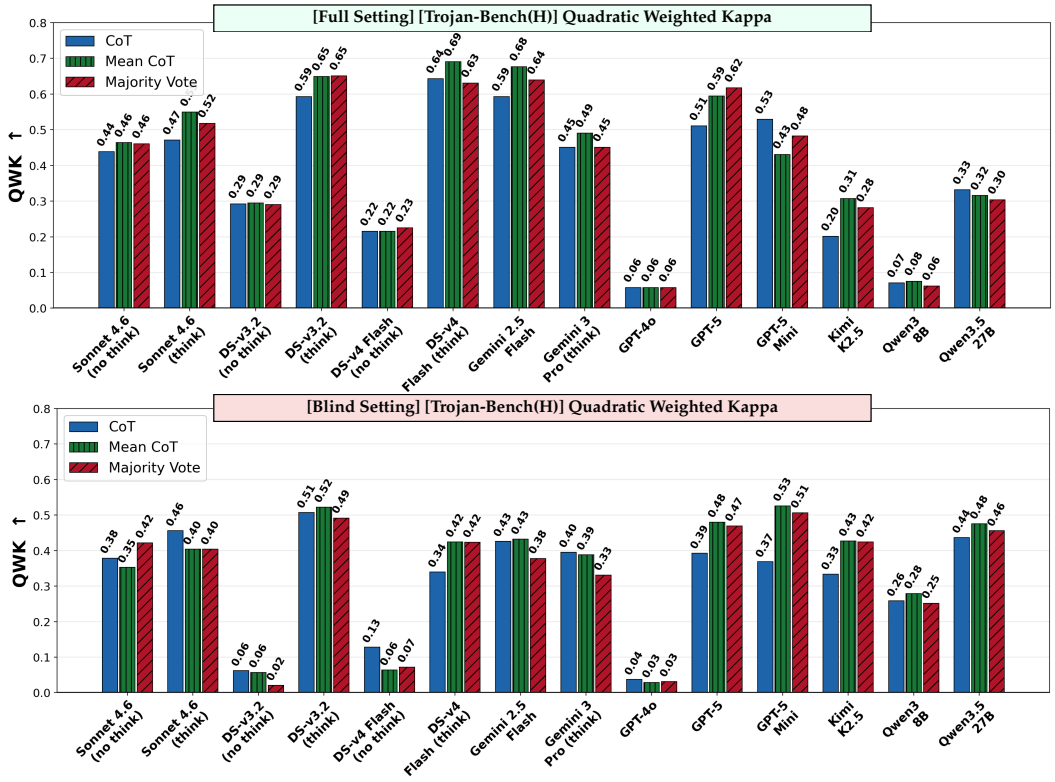

Figure 4: Model correlation with human labels using QWK across Full and Blind settings. The bar plots report CoT@1, MeanCoT@16, and MajorityVote@16 for each tested model.

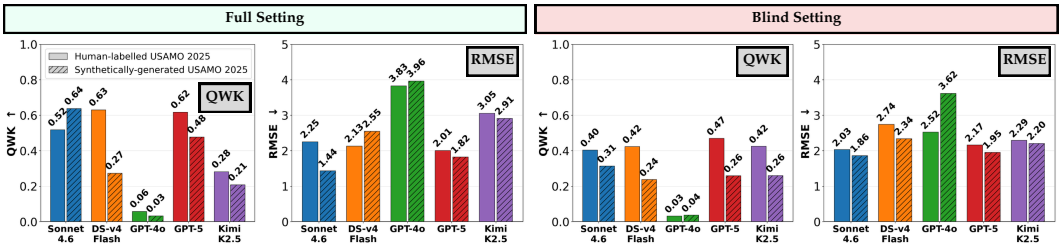

Figure 5: Grading performance on human and synthetic USAMO 2025 subsets ($N = 48$). Reported values represent the outcome of MajorityVote@16 across QWK, RMSE, and Bias metrics.

on that derivation. We observe similar trends in RMSE, Pearson correlation, and Bias, with the full suite of results detailed in the Appendix Sections D, E, and F.

Additionally, we conducted a comparative analysis of the different sampling methods. Because we utilized a lower temperature, the performance gap between single and multi-sampling is relatively small. However, consistent trends remain present in our evaluation. On average, MajorityVote outperforms other sampling techniques, though the primary benefit serves to stabilize the grading output. Specifically, under Full inference, MajorityVote scores 5.46% higher than CoT and 2.36% higher than MeanCoT.

Quality of the Synthetic Dataset. Furthermore, to evaluate the quality of the synthetic dataset, we provide two points of analysis. First, we measured the QWK, RMSE, and Bias on the USAMO 2025 subset of the synthetic data. We kept all sampling parameters consistent across both sets, with each containing 48 samples. We present these results in Figure 5. Our evaluation suggests that on the human-curated benchmark, models achieve a 0.07 lower absolute QWK, a 0.12 higher absolute RMSE, and a 0.04 higher absolute Bias compared to their synthetic counterparts. These results

Table 4: Comparative performance of base and fine-tuned models on *TROJAN* synthetic samples.

	Full Setting			**Blind Setting**		
	QWK (↑)	RMSE (↓)	Bias (≈ 0)	QWK (↑)	RMSE (↓)	Bias (≈ 0)
Qwen3.5-9B (Base)	0.02	4.00	3.23	0.06	2.52	0.63
Qwen3.5-9B (SFT)	**0.35**	**2.08**	**0.27**	**0.07**	**2.40**	**0.39**
Qwen3.5-27B (Base)	0.36	2.98	1.54	-0.11	4.12	-3.40
Qwen3.5-27B (SFT)	**0.47**	**2.10**	**0.77**	**0.06**	**2.50**	**0.21**

Table 5: Sensitivity of QWK, RMSE, and Bias to n_{iter} and n_{align}.

		QWK ↑			RMSE ↓			Bias		
	$n_{\text{align}} \rightarrow$	1	2	3	1	2	3	1	2	3
	1	0.583	—	—	1.414	—	—	0.400	—	—
	3	0.579	0.833	0.776	1.265	0.894	1.049	0.400	0.000	0.100
n_{iter}	5	0.757	**0.884**	0.818	0.949	**0.707**	0.894	0.300	0.100	0.000
	7	0.694	0.683	0.717	1.225	1.140	1.342	0.700	0.500	0.600
	9	0.571	0.735	0.622	1.225	1.140	1.304	-0.300	0.100	0.700

indicate that the human-crafted dataset presents a higher level of difficulty. While most models exhibit similar trends, we observe that DeepSeek-V4-Flash-Thinking shows a significantly lower correlation with synthetic data compared to the human-annotated version. When measuring the correlation of model responses between the expert-labeled and synthetic sets, we observe a Pearson r of 0.89.

Fine-Tuning on *Trojan-Bench(S)*. Additionally, we performed SFT on the synthetic data and present our findings in Table 4. We supervised fine-tuned Qwen3.5-9B and 27B models [44] using LLaMA-Factory [66] on the full set of our synthetic data derived from four competition problems (see Section 4.2). The training mixture includes data from both the full and blind verification settings. The data were split into training and validation sets using a 95/5 ratio. The maximum context length was set to 16K tokens, which is sufficient for our task. The 9B model was fully fine-tuned, while the 27B model was fine-tuned with LoRA using rank-8 adapters. Both models were trained for 2 epochs with an effective global batch size of 16, bfloat16 precision, and a cosine learning-rate schedule with 10% warmup. The peak learning rate was 1e-5 for the full fine-tuning run and 1e-4 for the LoRA run. All participating models were run in "non-thinking" mode, which excludes the reasoning phase typically enclosed in `<think>` tokens.

We observe that after fine-tuning, smaller models demonstrate improved performance across the three primary metrics. Specifically, when trained on synthetic data, models achieve a higher correlation with human grading, a result that is consistent across both small and large model scales. Taken together, these findings suggest that synthetic data provides a high degree of correlation with human-generated labels and that training on this data effectively improves agreement with expert judgment.

***TROJAN* hyperparameter selection.** To choose parameters for our synthetic pipeline generation, we ablated two main parameters. For each pair of parameters we asked human experts to grade 10 outputs and aggregated LLM-assigned score (according to a score profile) and human assigned scores. We report results in Table 5. We selected $n_{iter} = 5$ and $n_{alig.} = 2$ for the *TROJAN* pipeline as this configuration represents the global optimum across all primary evaluation metrics. This setting achieves the highest correlation with human labels (QWK = 0.884) and the lowest error (RMSE = 0.707). While certain alternative configurations provide a zero bias, they do so at the expense of significantly lower correlation and higher error rates.

6.3 Performance of *MABGrader*

Comparison with baselines. Table 6 presents the performance of four models from different providers across the QWK, RMSE, and Bias metrics. To ensure a comprehensive evaluation, we compare *MABGrader* against three primary baselines: zero-shot CoT@1, MajorityVote@16 using three distinct prompt templates, and Best-of-N selection utilizing four reward models applied to the MathArena template outputs. Our results indicate that *MABGrader* consistently achieves the highest

Table 6: Performance on the manually annotated USAMO 2025 dataset in the Full setting, where models are given a reference solution and marking scheme. We report QWK, RMSE, and Bias for four base models: DeepSeek-V4-Flash, Kimi-K2.5, Gemini-3-Pro-Preview, and Qwen3.5-27B.

	QWK (↑)				**RMSE (↓)**				**Bias (≈ 0)**			
	DS V4F	Kimi K2.5	Gem. 3P	Q3.5 27B	DS-V4F	Kimi K2.5	Gem. 3P	Q3.5 27B	DS-V4F	Kimi K2.5	Gem. 3P	Q3.5 27B
Chain-of-Thought (CoT@1)												
MathArena [4]	0.59	0.20	0.45	0.33	2.28	3.29	2.50	3.37	-0.69	2.24	1.35	1.73
ProofGr. [33]	0.64	0.24	0.46	0.36	2.15	3.06	2.51	3.10	-0.81	2.14	1.59	2.02
ProofAGr. [32]	0.52	0.19	0.42	0.32	2.47	3.47	2.84	3.60	-1.15	2.86	1.78	1.97
Chain-of-Thought with Majority Vote Selection (MajorityVote@16)												
MathArena [4]	0.64	0.28	0.45	0.30	2.13	3.05	2.50	3.04	-0.62	2.15	1.38	1.81
ProofGr. [33]	**0.66**	0.27	0.46	0.41	2.14	2.84	2.41	2.99	-0.71	2.21	1.33	1.89
ProofAGr. [32]	0.55	0.26	0.44	0.38	2.40	3.21	2.70	3.38	-1.06	2.34	1.49	1.93
Chain-of-Thought with Best-of-N Selection (BoN@16)												
Skywork [31]	0.61	0.20	0.41	0.29	2.16	3.33	2.64	3.11	-0.22	2.31	1.44	1.98
ArmoRM [56]	0.59	0.15	0.41	0.29	2.26	3.46	2.66	3.16	**-0.19**	2.35	1.58	2.04
RLHFlow [14]	0.60	0.15	0.50	0.27	**2.02**	3.19	2.41	3.18	-0.42	1.63	1.42	1.88
Shepherd [57]	0.61	0.12	0.45	0.31	2.14	3.40	2.49	2.95	-0.10	2.13	1.52	2.04
Proposed Method (MABGrader@16)												
MABGrader	**0.66**	**0.35**	**0.63**	**0.57**	2.06	**2.65**	**1.96**	**2.15**	-0.60	**0.20**	**0.34**	**-0.98**

correlation with human experts, as evidenced by higher QWK scores across all tested models. While *MABGrader* significantly improves RMSE and Bias for Kimi-K2.5, Gemini-3-Pro, and Qwen3.5-27B, we observe that specific reward models such as RLHFlow and ArmoRM yield slightly better results for DeepSeek-V4-Flash in terms of error magnitude and systematic bias. Overall, the proposed method provides a more stable alignment with human grading across diverse model architectures.

7 Limitations and Future Work

Our evaluation pipeline depends on the availability of both a markscheme and a golden reference solution. This assumption is reasonable for Olympiad-level grading, where they are produced alongside the competition, but it does not hold for most real-world applications such as peer review of new theorems or open-ended scientific verification. Writing a high-quality markscheme without access to a reference solution is itself a difficult task and remains an open problem that our work does not address.

The same dependency limits *MABGrader* directly. Both the per-arm justifications and the oracle reward are framed against the reference proof, so the algorithm requires a markscheme that maps cleanly onto a golden solution. *MABGrader* therefore does not apply to the Blind setting, where there is no point of comparison for either the candidate proof or the per-arm justifications. Extending the algorithm to reference-free regimes, for instance by having the grader first synthesize an outline of the expected proof that the oracle can then evaluate against, is a natural direction for future work.

8 Conclusion

We conducted a systematic study of LLM-based verification for Olympiad-level mathematics. We introduced a human-curated benchmark derived from USAMO with markscheme-aligned injected errors and expert grades, and used it to evaluate 20 frontier models across prompting and inference settings. Our results show that current LLM verifiers still fall short of expert-level agreement. We proposed *TROJAN* for scalable adversarial proof generation to support evaluation and SFT and to improve grading alignment. Finally, we introduced *MABGrader*, a Multi-Armed Bandit method for verification that achieves the best overall performance, outperforming ProofGrader with 26.4% higher QWK and 14% less error. Our work suggests paths toward more reliable automated verification.

Acknowledgments

The work of Farzan Farnia is supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China, Project 14210725, and is also supported by CUHK Direct Research Grant with CUHK Project No. 4055164. Moreover, the work is partially supported by a grant under 1+1+1 CUHK-CUHK(SZ)-GDSTC Joint Collaboration Fund. Finally, the authors sincerely thank the anonymous reviewers and metareviewer for their helpful feedback and constructive suggestions.

References

[1] Anthropic. Claude 4.6 opus. `https://www.anthropic.com/news/claude-opus-4-6`, 2026.

[2] Anthropic. Introducing Claude Sonnet 4.6, February 2026.

[3] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multi-armed bandit problem. *Machine Learning*, 2002.

[4] Mislav Balunovic, Jasper Dekoninck, Ivo Petrov, Nikola Jovanovic, and Martin T. Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *CoRR*, abs/2505.23281, 2025.

[5] Yves Bertot and Pierre Castéran. *Interactive theorem proving and program development: Coq'Art: the calculus of inductive constructions*. Springer Science & Business Media, 2013.

[6] Chenrui Cao, Liangcheng Song, Zenan Li, Xinyi Le, Xian Zhang, Hui Xue, and Fan Yang. Reviving dsp for advanced theorem proving in the era of reasoning models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[7] Wei-Lin Chen, Zhepei Wei, Xinyu Zhu, Shi Feng, and Yu Meng. Do llm evaluators prefer themselves for a reason?, 2025.

[8] Zhijun Chen, Xiaodong Lu, Jingzheng Li, Pengpeng Chen, Zhuoran Li, Kai Sun, Yuankai Luo, Qianren Mao, Ming Li, Likang Xiao, Dingqi Yang, Xiao Huang, Yikun Ban, Hailong Sun, and Philip S. Yu. Harnessing multiple large language models: A survey on llm ensemble, 2026.

[9] 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, 2021.

[10] American Mathematics Competitions and Art of Problem Solving. Aime problems and solutions — 2025. `https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions`, 2025.

[11] DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, April 2026.

[12] DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Erhang Li, Fangqi Zhou, Fangyun Lin, Fucong Dai, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoran Wei, Haowei Zhang, Haowen Luo, Haozhe Ji, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, Jialiang Huang, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jingchang Chen, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jinhua Zhu, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexin Huang, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Liang Zhao, Liangsheng Yin, Lihua Guo, Lingxiao Luo, Linwang Ma, Litong Wang, Liyue Zhang, M. S. Di, M. Y Xu, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Panpan Huang, Peixin Cong, Peiyi Wang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, S. H. Liu,

Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Songyang Zhou, Tao Ni, Tao Yun, Tian Pei, Tian Ye, Tianyuan Yue, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjun Gao, Wentao Zhang, Xi Gao, Xiangwen Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyuan Li, Xu Chen, Xuecheng Su, Xuehai Pan, Xuheng Lin, Xuwei Fu, Y. Q. Wang, Yang Zhang, Yanhong Xu, Yanru Ma, Yao Li, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yiliang Xiong, Ying He, Ying Zhou, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuduan Wang, Yue Gong, Yuhan Wu, Yuheng Zou, Yukun Li, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehua Zhao, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhiyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Zizheng Pan, Zongqing Yao, Bei Feng, Hui Li, J. L. Cai, Jiaqi Ni, Lei Xu, Meng Li, Ning Tian, R. J. Chen, R. L. Jin, S. S. Li, Shuang Zhou, Tianyu Sun, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xinnan Song, Xinyi Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, Dongjie Ji, Jian Liang, Jianzhong Guo, Jin Chen, Leyi Xia, Miaojun Wang, Mingming Li, Peng Zhang, Ruyi Chen, Shangmian Sun, Shaoqing Wu, Shengfeng Ye, T. Wang, W. L. Xiao, Wei An, Xianzu Wang, Xiaowen Sun, Xiaoxiang Wang, Ying Tang, Yukun Zha, Zekai Zhang, Zhe Ju, Zhen Zhang, and Zihua Qu. Deepseek-v3.2: Pushing the frontier of open large language models, 2025.

[13] Jasper Dekoninck, Ivo Petrov, Kristian Minchev, Miroslav Marinov, Maria Drencheva, Lyuba Konova, Milen Milenov Shumanov, Kaloyan Tsvetkov, Nikolay Drenchev, Lazar D. Todorov, Kalina Nikolova, Nikolay Georgiev, Vanesa Kalinkova, Margulan Ismoldayev, Mislav Balunovic, and Martin Vechev. The open proof corpus: A large-scale study of LLM-generated mathematical proofs. In *The Fourteenth International Conference on Learning Representations*, 2026.

[14] Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.

[15] Gheorghe Comanici et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025.

[16] Nancy Falchikov and Judy Goldfinch. Student peer assessment in higher education: A meta-analysis comparing peer and teacher marks. *Review of Educational Research*, 70(3):287–322, 2000.

[17] Evan Frick, Tianle Li, Connor Chen, Wei-Lin Chiang, Anastasios Nikolas Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. How to evaluate reward models for RLHF. In *The Thirteenth International Conference on Learning Representations*, 2025.

[18] 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, October 2024. arXiv:2410.07985.

[19] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

[20] Google. A new era of intelligence with gemini 3, November 2025.

[21] Google DeepMind. AI achieves silver-medal standard solving international mathematical olympiad problems. `https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/`, 2024. Accessed: 2025-05-08.

[22] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A survey on llm-as-a-judge, 2025.

[23] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems, 2024.

[24] Yichen Huang and Lin F. Yang. Winning gold at IMO 2025 with a model-agnostic verification-and-refinement pipeline. In *The 5th Workshop on Mathematical Reasoning and AI at NeurIPS 2025*, 2025.

[25] Albert Qiaochu Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Timoth'ee Lacroix, Yuhuai Wu, and Guillaume Lample. Draft, Sketch, and Prove: Guiding formal theorem provers with informal proofs. In *International Conference on Learning Representations*, 2023.

[26] Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models, 2024.

[27] T.L Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[28] Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. In *Forty-second International Conference on Machine Learning*, 2025.

[29] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024.

[30] Yong Lin, Shange Tang, Bohan Lyu, Ziran Yang, Jui-Hui Chung, Haoyu Zhao, Lai Jiang, Yihan Geng, Jiawei Ge, Jingruo Sun, Jiayun Wu, Jiri Gesi, Ximing Lu, David Acuna, Kaiyu Yang, Hongzhou Lin, Yejin Choi, Danqi Chen, Sanjeev Arora, and Chi Jin. Goedel-prover-v2: Scaling formal theorem proving with scaffolded data synthesis and self-correction. *CoRR*, abs/2508.03613, 2025.

[31] Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. *CoRR*, abs/2410.18451, 2024.

[32] Thang Luong, Dawsen Hwang, Hoang H. Nguyen, Golnaz Ghiasi, Yuri Chervonyi, Insuk Seo, Junsu Kim, Garrett Bingham, Jonathan Lee, Swaroop Mishra, Alex Zhai, Clara Huiyi Hu, Henryk Michalewski, Jimin Kim, Jeonghyun Ahn, Junhwi Bae, Xingyou Song, Trieu H. Trinh, Quoc V. Le, and Junehyuk Jung. Towards robust mathematical reasoning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 35418–35442. Association for Computational Linguistics, 2025.

[33] Wenjie Ma, Andrei Cojocaru, Neel Kolhe, Bradley Louie, Robin Said Sharif, Haihan Zhang, Vincent Zhuang, Matei Zaharia, and Sewon Min. Reliable fine-grained evaluation of natural language math proofs. *CoRR*, abs/2510.13888, 2025.

[34] Leonardo de Moura and Sebastian Ullrich. The lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28: 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings*, page 625–635, Berlin, Heidelberg, 2021. Springer-Verlag.

[35] Tobias Nipkow, Lawrence C Paulson, and Markus Wenzel. *Isabelle/HOL: a proof assistant for higher-order logic*, volume 2283. Springer Science & Business Media, 2002.

[36] OpenAI. Gpt-4o system card, 2024.

[37] OpenAI. Openai gpt-5 system card, 2026.

[38] Azim Ospanov, Farzan Farnia, and Roozbeh Yousefzadeh. Apollo: Automated llm and lean collaboration for advanced formal reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[39] Azim Ospanov, Farzan Farnia, and Roozbeh Yousefzadeh. minif2f-lean revisited: Reviewing limitations and charting a path forward. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[40] Azim Ospanov, Zijin Feng, Jiacheng Sun, Haoli Bai, Xin Shen, and Farzan Farnia. Hermes: Towards efficient and verifiable mathematical reasoning in llms, 2025.

[41] Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. OpenWebMath: An open dataset of high-quality mathematical web text. In *The Twelfth International Conference on Learning Representations*, 2023.

[42] Ivo Petrov, Jasper Dekoninck, Lyuben Baltadzhiev, Maria Drencheva, Kristian Minchev, Mislav Balunovic, Nikola Jovanović, and Martin Vechev. Proof or bluff? evaluating LLMs on 2025 USA math olympiad. In *2nd AI for Math Workshop @ ICML 2025*, 2025.

[43] Ivo Petrov, Jasper Dekoninck, and Martin Vechev. Brokenmath: A benchmark for sycophancy in theorem proving with llms, 2025.

[44] Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026.

[45] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[46] James V. Roggeveen, Erik Y. Wang, Will Flintoft, Peter Donets, Lucy S. Nathwani, Nickholas Gutierrez, David Ettel, Anton Marius Graf, Siddharth Dandavate, Arjun Nageswaran, Raglan Ward, Ava Williamson, Anne Mykland, Kacper K. Migacz, Yijun Wang, Egemen Bostan, Duy Thuc Nguyen, Zhe He, Marc L. Descoteaux, Felix Yeung, Shida Liu, Jorge García Ponce, Luke Zhu, Yuyang Chen, Ekaterina S. Ivshina, Miguel Fernandez, Minjae Kim, Kennan Gumbs, Matthew Scott Tan, Russell Yang, Mai Hoang, David Brown, Isabella A. Silveira, Lavon Sykes, Ahmed Roman, William Fredenberg, Yiming Chen, Lucas Martin, Yixing Tang, Kelly Werker Smith, Hongyu Liao, Logan G. Wilson, Alexander Dazhen Cai, Andrea Elizabeth Biju, and Michael P. Brenner. Hardmath2: A benchmark for applied mathematics built by students as part of a graduate class. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[47] Zhihong Shao, Yuxiang Luo, Chengda Lu, Z. Z. Ren, Jiewen Hu, Tian Ye, Zhibin Gou, Shirong Ma, and Xiaokang Zhang. Deepseekmath-v2: Towards self-verifiable mathematical reasoning. *CoRR*, abs/2511.22570, 2025.

[48] Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Yuan Tang, Alejandro Cuadron, Chenguang Wang, Raluca Popa, and Ion Stoica. Judgebench: A benchmark for evaluating LLM-based judges. In *The Thirteenth International Conference on Learning Representations*, 2025.

[49] Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. Mathscale: Scaling instruction tuning for mathematical reasoning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[50] Kimi Team. Kimi k2.5: Visual agentic intelligence, 2026.

[51] Qwen Team. Qwen3.5: Accelerating productivity with native multimodal agents, February 2026.

[52] Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. Judging the judges: Evaluating alignment and vulnerabilities in llms-as-judges, 2025.

[53] Trieu Trinh, Yuhuai Wu, Quoc Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 2024.

[54] George Tsoukalas, Jasper Lee, John Jennings, Jimmy Xin, Michelle Ding, Michael Jennings, Amitayush Thakur, and Swarat Chaudhuri. Putnambench: Evaluating neural theorem-provers on the putnam mathematical competition, 2024.

[55] Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. Replacing judges with juries: Evaluating llm generations with a panel of diverse models, 2024.

[56] Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 10582–10592. Association for Computational Linguistics, 2024.

[57] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9426–9439. Association for Computational Linguistics, 2024.

[58] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[59] Sean Welleck, Jiacheng Liu, Ximing Lu, Hannaneh Hajishirzi, and Yejin Choi. Naturalprover: Grounded mathematical proof generation with language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[60] Scott Wood, Erin Yao, Lisa Haisfield, and Susan Lottridge. Establishing standards of best practice in automated scoring. Technical report, ACT, Inc., July 2021.

[61] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025.

[62] Roozbeh Yousefzadeh, Xuenan Cao, and Azim Ospanov. A lean dataset for international math olympiad: Small steps towards writing math proofs for hard problems, 2025.

[63] Albert S. Yue, Lovish Madaan, Ted Moskovitz, DJ Strouse, and Aaditya K. Singh. Harp: A challenging human-annotated math reasoning benchmark, 2024.

[64] Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. miniF2F: a cross-system benchmark for formal Olympiad-level mathematics, 2022.

[65] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[66] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics.

Contents

A Datasheets **18**

B Distribution of injected errors in *Trojan-Bench(H)* **24**

C Evaluated Model List **24**

D Extended evaluation results **24**

D.1 *Trojan-Bench(H)* 27

D.2 *Trojan-Bench(S)* 31

D.3 Additional evaluation of BoN methods 36

E Extended Bar plots **37**

F Evaluation of LLM correlation with human and other LLM annotators **48**

G Annotation procedure **65**

H Additional details on SFT training **65**

I Hyperparameter c ablation in *MABGrader* **67**

J Case studies **68**

J.1 Behavior of smaller models on Full and Blind settings 68

J.2 Effect of *MABGrader* on grading procedures 72

K Inclusion of USAMO 2025 samples in the training set **74**

L Effect of changing the generator model in *TROJAN* pipeline **74**

M Benchmarking *TROJAN* SFT pipeline against ProofBench **74**

N Effect of Multi-Armed Bandit algorithm on grading **76**

O Prompt templates **77**

A Datasheets

Following the framework of [19] and [41], Table 7 provides the additional information about our expert-annotated dataset and Table 8 provides additional information about our synthetically generated dataset.

Table 7: Datasheet for our expert annotated dataset

Questions	Answers
Motivation	
For what purpose was the dataset created?	The dataset was developed to benchmark the performance of current Large Language Models in grading and evaluating Olympiad-level mathematical proofs.
Who created the dataset and on behalf of which entity?	The authors of this paper.
Who funded the creation of the dataset?	The company where the authors work.
Any other comment?	None.
Composition	
What do the instances that comprise the dataset represent?	USAMO 2025 problems that consist of problem statements, golden solutions, corresponding markschemes, candidate proofs with minor errors and corresponding expert assigned grades.
How many instances are there in total?	48 instances
Does the dataset contain all possible instances or is it a sample of instances from a larger set?	Yes, we present all instances.
What data does each instance consist of?	A problem statement, golden solution, corresponding markscheme, candidate solution with logical errors and corresponding expert grades.
Is there a label or target associated with each instance?	Only human assigned grades.
Is any information missing from individual instances?	No.
Are relationships between individual instances made explicit?	Not applicable.
Are there recommended data splits?	No recommended split.
Are there any errors, sources of noise, or redundancies in the dataset?	No, there are no errors in the proposed dataset to the best of our knowledge. Verification was done by two independent annotators.
Is the dataset self-contained, or does it link to or otherwise rely on external resources?	The dataset is self-contained.
Does the dataset contain data that might be considered confidential?	No.
Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?	No.
Collection Process	
How was the data associated with each instance acquired?	All 48 instances originate from our team and further verified using two-step collection described in [45].

Continued on next page

What mechanisms or procedures were used to collect the data?	The seed problems were taken from USAMO 2025 subset of the MathArena repository [4]. Based on corresponding markschemes, we injected mistakes that would result in a particular loss of points.
If the dataset is a sample from a larger set, what was the sampling strategy?	No, there is no larger version.
Who was involved in the data collection process and how were they compensated?	Authors of this work are responsible for data collection.
Over what time frame was the data collected?	The dataset was collected over a four-month period, from January 2026 through April 2026.
Were any ethical review processes conducted?	No.
Preprocessing/cleaning/labeling	
Was any preprocessing/cleaning/labeling of the data done?	Yes, we manually assigned grades to each sample and preprocessed the data to follow MathArena [4] format.
Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data?	No, all raw data was converted to preprocessed samples.
Is the software that was used to preprocess/clean/label the data available?	We used open-source text editors to perform annotation.
Any other comments?	No.
Uses	
Has the dataset been used for any tasks already?	We evaluated a large number of LLMs and measured their correlation with human labels. Moreover, we tested our proposed *MABGrader* agent.
Is there a repository that links to any or all papers or systems that use the dataset?	Public GitHub and HuggingFace links will be provided at a later date.
What (other) tasks could the dataset be used for?	The dataset may be used for benchmarking alignment with human grades in LLM-as-a-Judge scenarios.
Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?	We do not anticipate future issues emerging from the proposed benchmark dataset.
Are there tasks for which the dataset should not be used?	The data is primarily for benchmarking, so we discourage others from training their models on this dataset.
Any other comments?	No.
Distribution	
Will the dataset be distributed to third parties outside of the entity on behalf of which the dataset was created?	Yes, we will release the dataset on public platforms such as GitHub and HuggingFace at a later date.
How will the dataset will be distributed?	GitHub, HuggingFace.
When will the dataset be distributed?	The dataset will be distributed along with the camera-ready version of the submission.
Will the dataset be distributed under a copyright or other intellectual property license, and/or under applicable terms of use?	Yes, the dataset will be released under the MIT license.

Continued on next page

Have any third parties imposed IP-based or other restrictions on the data associated with the instances?	No.
Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?	No.
Any other comments?	No.
Maintenance	
Who will be supporting/hosting/maintaining the dataset?	The first author of the submission.
How can the owner/curator/manager of the dataset be contacted?	The curator of the dataset may be reached through an email or any other public means, such as GitHub profile.
Is there an erratum?	At this stage there is no associated erratum.
Will the dataset be updated?	Yes, we plan to periodically update the dataset if mistakes are found.
If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances?	Not applicable.
Will older versions of the dataset continue to be supported/hosted/maintained?	Yes.
If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?	Yes, since we release all accompanied information, others may use this data for other uses.
Any other comments?	No.

Table 8: Datasheet for our synthetically generated dataset using *TROJAN*.

Questions	Answers
Motivation	
For what purpose was the dataset created?	The dataset was developed to benchmark the performance of current Large Language Models in grading and evaluating Olympiad-level mathematical proofs.
Who created the dataset and on behalf of which entity?	The authors of this paper.
Who funded the creation of the dataset?	The company where the authors work.
Any other comment?	None.
Composition	
What do the instances that comprise the dataset represent?	USAMO 2025/USAMO 2026/IMO 2025/IMC 2025 problems that consist of problem statements, golden solutions, corresponding markschemes, candidate proofs with minor errors and corresponding expert assigned grades.
How many instances are there in total?	200 instances
Does the dataset contain all possible instances or is it a sample of instances from a larger set?	Yes, we present all instances.
What data does each instance consist of?	A problem statement, golden solution, corresponding markscheme, candidate solution with logical errors and corresponding expert grades.
Is there a label or target associated with each instance?	Only LLM assigned target grades as described in the score profile.
Is any information missing from individual instances?	No.
Are relationships between individual instances made explicit?	Not applicable.
Are there recommended data splits?	No recommended split.
Are there any errors, sources of noise, or redundancies in the dataset?	No, there are no errors in the proposed dataset to the best of our knowledge. Verification was done by a critic LLM, namely Claude-Opus-4.6-Think.
Is the dataset self-contained, or does it link to or otherwise rely on external resources?	The dataset is self-contained.
Does the dataset contain data that might be considered confidential?	No.
Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?	No.
Collection Process	
How was the data associated with each instance acquired?	All 200 instances originate from our team and further verified using critique-alignment process of our *TROJAN* pipeline.

Continued on next page

What mechanisms or procedures were used to collect the data?	The seed problems were taken from USAMO 2025/USAMO 2026/IMO 2025/IMC 2025 subsets of the MathArena repository [4]. Based on corresponding markschemes, LLM injected mistakes that would result in a particular loss of points as described in a given score profile.
If the dataset is a sample from a larger set, what was the sampling strategy?	No, there is no larger version.
Who was involved in the data collection process and how were they compensated?	Authors of this work are responsible for data collection.
Over what time frame was the data collected?	The dataset was collected over a four-month period, from January 2026 through April 2026.
Were any ethical review processes conducted?	No.
Preprocessing/cleaning/labeling	
Was any preprocessing/cleaning/labeling of the data done?	Yes, we manually assigned grades to each sample and preprocessed the data to follow MathArena [4] format.
Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data?	No, all raw data was converted to preprocessed samples.
Is the software that was used to preprocess/clean/label the data available?	We used open-source text editors to perform annotation.
Any other comments?	No.
Uses	
Has the dataset been used for any tasks already?	We evaluated a large number of LLMs and measured their correlation with LLM-assigned labels. Moreover, we tested our proposed *MAB-Grader* agent.
Is there a repository that links to any or all papers or systems that use the dataset?	Public GitHub and HuggingFace links will be provided at a later date.
What (other) tasks could the dataset be used for?	The dataset may be used for benchmarking alignment with human grades in LLM-as-a-Judge scenarios.
Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?	We do not anticipate future issues emerging from the proposed benchmark dataset.
Are there tasks for which the dataset should not be used?	The data is for benchmarking and may be used for training. At this stage we do not see scenarios for which we discourage others to use this data.
Any other comments?	No.
Distribution	
Will the dataset be distributed to third parties outside of the entity on behalf of which the dataset was created?	Yes, we will release the dataset on public platforms such as GitHub and HuggingFace at a later date.
How will the dataset will be distributed?	GitHub, HuggingFace.
When will the dataset be distributed?	The dataset will be distributed along with the camera-ready version of the submission.

Continued on next page

Will the dataset be distributed under a copyright or other intellectual property license, and/or under applicable terms of use?	Yes, the dataset will be released under the MIT license.
Have any third parties imposed IP-based or other restrictions on the data associated with the instances?	No.
Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?	No.
Any other comments?	No.
Maintenance	
Who will be supporting/hosting/maintaining the dataset?	The first author of the submission.
How can the owner/curator/manager of the dataset be contacted?	The curator of the dataset may be reached through an email or any other public means, such as GitHub profile.
Is there an erratum?	At this stage there is no associated erratum.
Will the dataset be updated?	Yes, we plan to periodically update the dataset if mistakes are found.
If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances?	Not applicable.
Will older versions of the dataset continue to be supported/hosted/maintained?	Yes.
If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?	Yes, since we release all accompanied information, others may use this data for other uses.
Any other comments?	No.

B Distribution of injected errors in *Trojan-Bench(H)*

Table 9 shows the distribution of injected errors in the *Trojan-Bench(H)* dataset. During construction, we apply logical gap errors the most, since they are easier to control and spot right away. Moreover, a logical gap is the widest error category that is difficult to define with one term, so many unjustified claims go under this broad umbrella category. "Misapplied theorem" refers to errors with incorrect preliminaries or even a completely wrong theorem used to prove the statement. Conceptual errors refer to fundamental issues with the proof, i.e., wrong constructions or even interpretations of the problem. Computational slips are very minor mistakes that do not affect the grand scheme of things, but still lead to less rigorous proofs. "Incomplete justification" is a mistake in which the proof is only partially complete and technically correct, but fails to explain all conditions/steps.

Table 9: Distribution of injected error types in *Trojan-Bench(H)*. We report them as absolute counts and percentages.

Error Type	Count	Percentage
Logical gap	27	35.06%
Misapplied theorem	18	23.38%
Conceptual error	12	15.58%
Computational slip	8	10.39%
Incomplete justification	6	7.79%
None	6	7.79%
Total	77	100.0%

C Evaluated Model List

Table 10 summarises every grader that participates in our evaluation suite. The list spans 20 distinct models from major providers, covering both open-weight and closed-weight families together with their thinking-enabled and non-thinking variants where available. Each row reports which (dataset, prompting style) combinations a given model was run on. The two datasets are USAMO 2025 (the human-annotated benchmark, *Trojan-Bench(H)*) and the synthetic set (*Trojan-Bench(S)*), and the two styles are the rubric-equipped Full template and the rubric-free Blind template. The final column lists the maximum completion-token budget observed across attempts, which clarifies the inference-cost regime each model operated in. We observe that models rarely hit the maximum token limit.

In total, the table tracks 52 model$\times$(dataset, style) configurations and serves as an index for the per-model results that follow later in this appendix. Coverage is not uniform across providers. Smaller open-weight reward-style models such as the Prometheus family were evaluated on *Trojan-Bench(H)* Full only, since they did not produce usable scores in the rubric-free regime. Frontier thinking-enabled models were evaluated more broadly because their scoring behaviour is the primary subject of our analysis.

D Extended evaluation results

This section reports the full set of correlation and error metrics for every grader in Table 10, broken down by inference style (Full and Blind), dataset (*Trojan-Bench(H)* and *Trojan-Bench(S)*), and aggregation method (single CoT, Mean-CoT@16, and MajorityVote@16). The first two tables of each subsection report the mean and standard deviation of every metric across the 16 sampling attempts at the per-attempt level, which characterises the within-grader variance. The remaining six tables report the aggregated grade under the three aggregation methods, which is the quantity each grader exposes to a downstream user.

Table 10: The list of all 20 evaluated models. A ✓ marks the presence of a sampling_grades.csv-producing run for the indicated (dataset, style) pair. In total 52 model×(dataset, style) runs are reported. The last column reports the maximum `completion_tokens`.

	Trojan-Bench(H)		*Trojan-Bench(S)*		
Model	Full	Blind	Full	Blind	Max tokens
Claude-Sonnet-4.6 [2] (T)	✓	✓	–	–	65,536
Claude-Sonnet-4.6 [2] (T)	✓	✓	✓	✓	23,625
DeepSeek-V3.2 [12] (T)	✓	✓	✓	–	8,192
DeepSeek-V3.2 [12] (T)	✓	✓	✓	–	65,536
DeepSeek-V4-Flash [11] (T)	✓	✓	–	–	65,536
DeepSeek-V4-Flash [11] (T)	✓	✓	✓	✓	65,536
DeepSeek-V4-Pro [11] (T)	✓	–	–	–	65,536
Gemini-2.5-Flash [15]	✓	✓	–	–	32,768
Gemini-2.5-Pro [15] (T)	✓	–	–	–	65,536
Gemini-2.5-Pro [15] (T)	✓	–	–	–	65,536
Gemini-3-Pro-Prev. [20] (T)	✓	✓	–	–	65,536
GPT-4o [36]	✓	✓	✓	✓	4,096
GPT-5 [37]	✓	✓	✓	✓	65,536
GPT-5-Mini [37]	✓	✓	✓	✓	32,768
Kimi-K2.5 [50]	✓	✓	✓	✓	32,768
Prometheus-V2-7B [26]	✓	–	–	–	1024
Prometheus-V2-8×7B [26]	✓	–	–	–	1024
Qwen3-8B [61]	✓	✓	–	–	16,384
Qwen3.5-27B [51]	✓	✓	✓	✓	32,768
Qwen3.5-9B [51]	✓	–	✓	✓	16,384

Table 11: Mean ± standard deviation of Pearson r, Kendall τ, QWK, and MAE across individual CoT sampling attempts per problem. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the human-labelled *Trojan-Bench(H)* dataset against a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE
Claude-Sonnet-4.6 (T)	0.554 ± 0.025	0.451 ± 0.026	0.446 ± 0.024	1.785 ± 0.070
Claude-Sonnet-4.6 (T)	0.604 ± 0.046	0.506 ± 0.043	0.513 ± 0.043	1.623 ± 0.087
DeepSeek-V3.2 (T)	0.435 ± 0.011	0.342 ± 0.009	0.289 ± 0.008	2.292 ± 0.025
DeepSeek-V3.2 (T)	0.622 ± 0.065	0.526 ± 0.058	0.613 ± 0.066	1.357 ± 0.136
deepseek-v4-flash-nothinking	0.347 ± 0.014	0.286 ± 0.020	0.228 ± 0.010	2.363 ± 0.030
DeepSeek-V4-Flash (T)	0.656 ± 0.012	0.544 ± 0.015	0.631 ± 0.021	1.375 ± 0.029
deepseek-v4-pro	0.594 ± 0.078	0.510 ± 0.069	0.586 ± 0.078	1.469 ± 0.133
Gemini-2.5-Flash	0.586 ± 0.068	0.478 ± 0.057	0.572 ± 0.067	1.512 ± 0.153
Gemini-2.5-Pro (T)	0.555 ± 0.075	0.449 ± 0.074	0.462 ± 0.083	1.606 ± 0.146
Gemini-2.5-Pro (T)	0.599 ± 0.062	0.484 ± 0.057	0.515 ± 0.063	1.500 ± 0.133
Gemini-3-Pro-Prev. (T)	0.565 ± 0.032	0.463 ± 0.035	0.470 ± 0.032	1.615 ± 0.077
GPT-4o	0.216 ± 0.026	0.178 ± 0.020	0.062 ± 0.012	3.101 ± 0.032
GPT-5	0.620 ± 0.051	0.522 ± 0.050	0.557 ± 0.056	1.507 ± 0.102
GPT-5-Mini	0.519 ± 0.049	0.436 ± 0.043	0.421 ± 0.046	1.918 ± 0.117
Kimi-K2.5	0.413 ± 0.091	0.328 ± 0.079	0.282 ± 0.067	2.234 ± 0.171
Prometheus-V2-7B	0.020 ± 0.097	0.004 ± 0.083	0.010 ± 0.068	2.857 ± 0.107
Prometheus-V2-8×7B	0.063 ± 0.109	0.060 ± 0.076	0.031 ± 0.053	2.895 ± 0.145
Qwen3-8B	0.180 ± 0.053	0.186 ± 0.039	0.076 ± 0.024	3.070 ± 0.080
Qwen3.5-27B	0.396 ± 0.036	0.319 ± 0.042	0.287 ± 0.043	2.278 ± 0.112
Qwen3.5-9B	0.427 ± 0.069	0.373 ± 0.056	0.306 ± 0.064	2.370 ± 0.162

Table 12: Mean ± standard deviation of RMSE, bias, and exact-match accuracy across individual CoT sampling attempts per problem. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the human-labelled *Trojan-Bench(H)* dataset against a standard markscheme and golden solution.

Model	RMSE	Bias	Exact
Claude-Sonnet-4.6 (T)	2.408 ± 0.050	1.349 ± 0.085	0.228 ± 0.045
Claude-Sonnet-4.6 (T)	2.227 ± 0.096	1.177 ± 0.110	0.277 ± 0.038
DeepSeek-V3.2 (T)	3.068 ± 0.022	2.073 ± 0.035	0.238 ± 0.011
DeepSeek-V3.2 (T)	2.117 ± 0.201	0.260 ± 0.207	0.414 ± 0.039
deepseek-v4-flash-nothinking	3.158 ± 0.029	2.058 ± 0.036	0.232 ± 0.022
DeepSeek-V4-Flash (T)	2.111 ± 0.061	-0.615 ± 0.165	0.380 ± 0.020
deepseek-v4-pro	2.255 ± 0.225	-0.021 ± 0.195	0.401 ± 0.020
Gemini-2.5-Flash	2.217 ± 0.183	0.461 ± 0.209	0.325 ± 0.057
Gemini-2.5-Pro (T)	2.358 ± 0.199	1.245 ± 0.227	0.339 ± 0.036
Gemini-2.5-Pro (T)	2.186 ± 0.145	1.098 ± 0.111	0.330 ± 0.058
Gemini-3-Pro-Prev. (T)	2.454 ± 0.109	1.365 ± 0.088	0.359 ± 0.020
GPT-4o	3.826 ± 0.025	3.064 ± 0.037	0.166 ± 0.013
GPT-5	2.112 ± 0.123	0.960 ± 0.084	0.285 ± 0.034
GPT-5-Mini	2.588 ± 0.126	1.456 ± 0.112	0.227 ± 0.036
Kimi-K2.5	2.998 ± 0.178	1.944 ± 0.189	0.251 ± 0.028
Prometheus-V2-7B	3.564 ± 0.103	1.895 ± 0.272	0.138 ± 0.036
Prometheus-V2-8×7B	3.589 ± 0.131	2.464 ± 0.120	0.150 ± 0.042
Qwen3-8B	3.824 ± 0.055	2.908 ± 0.115	0.157 ± 0.030
Qwen3.5-27B	3.071 ± 0.110	1.903 ± 0.159	0.253 ± 0.017
Qwen3.5-9B	3.145 ± 0.166	2.012 ± 0.186	0.221 ± 0.029

Table 13: Mean ± standard deviation of Pearson r, Kendall τ, QWK, and MAE across individual CoT sampling attempts per problem. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the human-labelled *Trojan-Bench(H)* dataset without access to a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE
Claude-Sonnet-4.6 (T)	0.379 ± 0.043	0.293 ± 0.061	0.349 ± 0.049	1.797 ± 0.057
Claude-Sonnet-4.6 (T)	0.584 ± 0.100	0.481 ± 0.131	0.497 ± 0.100	1.646 ± 0.188
DeepSeek-V3.2 (T)	0.151 ± 0.115	0.089 ± 0.079	0.053 ± 0.045	3.012 ± 0.097
DeepSeek-V3.2 (T)	0.514 ± 0.072	0.399 ± 0.064	0.456 ± 0.069	1.699 ± 0.111
deepseek-v4-flash-nothinking	0.164 ± 0.125	0.112 ± 0.124	0.071 ± 0.058	2.776 ± 0.036
DeepSeek-V4-Flash (T)	0.468 ± 0.072	0.361 ± 0.054	0.396 ± 0.058	2.146 ± 0.143
Gemini-2.5-Flash	0.339 ± 0.147	0.265 ± 0.119	0.336 ± 0.149	1.913 ± 0.196
Gemini-3-Pro-Prev. (T)	0.464 ± 0.033	0.376 ± 0.033	0.369 ± 0.049	2.057 ± 0.095
GPT-4o	0.063 ± 0.069	0.031 ± 0.057	0.040 ± 0.043	2.138 ± 0.055
GPT-5	0.475 ± 0.051	0.351 ± 0.067	0.435 ± 0.054	1.663 ± 0.110
GPT-5-Mini	0.458 ± 0.065	0.337 ± 0.063	0.455 ± 0.066	1.770 ± 0.114
Kimi-K2.5	0.434 ± 0.070	0.319 ± 0.066	0.372 ± 0.079	1.879 ± 0.134
Qwen3-8B	0.196 ± 0.119	0.154 ± 0.093	0.175 ± 0.107	2.442 ± 0.173
Qwen3.5-27B	0.454 ± 0.022	0.352 ± 0.030	0.448 ± 0.023	1.703 ± 0.107

Table 14: Mean $\pm$ standard deviation of RMSE, bias, and exact-match accuracy across individual CoT sampling attempts per problem. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the human-labelled *Trojan-Bench(H)* dataset without access to a standard markscheme and golden solution.

Model	RMSE	Bias	Exact
Claude-Sonnet-4.6 (T)	2.292 $\pm$ 0.052	-0.422 $\pm$ 0.086	0.214 $\pm$ 0.046
Claude-Sonnet-4.6 (T)	2.006 $\pm$ 0.238	-0.609 $\pm$ 0.437	0.146 $\pm$ 0.055
DeepSeek-V3.2 (T)	3.683 $\pm$ 0.107	-2.832 $\pm$ 0.103	0.133 $\pm$ 0.020
DeepSeek-V3.2 (T)	2.243 $\pm$ 0.128	-0.871 $\pm$ 0.139	0.227 $\pm$ 0.048
deepseek-v4-flash-nothinking	3.476 $\pm$ 0.067	-2.505 $\pm$ 0.075	0.156 $\pm$ 0.036
DeepSeek-V4-Flash (T)	2.798 $\pm$ 0.151	-1.406 $\pm$ 0.027	0.229 $\pm$ 0.017
Gemini-2.5-Flash	2.501 $\pm$ 0.254	-0.028 $\pm$ 0.206	0.204 $\pm$ 0.019
Gemini-3-Pro-Prev. (T)	2.772 $\pm$ 0.106	1.568 $\pm$ 0.180	0.240 $\pm$ 0.027
GPT-4o	2.544 $\pm$ 0.053	0.802 $\pm$ 0.065	0.098 $\pm$ 0.017
GPT-5	2.199 $\pm$ 0.109	-0.598 $\pm$ 0.161	0.211 $\pm$ 0.054
GPT-5-Mini	2.327 $\pm$ 0.134	-0.136 $\pm$ 0.136	0.204 $\pm$ 0.042
Kimi-K2.5	2.415 $\pm$ 0.107	-1.001 $\pm$ 0.177	0.182 $\pm$ 0.054
Qwen3-8B	3.143 $\pm$ 0.167	1.044 $\pm$ 0.291	0.196 $\pm$ 0.046
Qwen3.5-27B	2.284 $\pm$ 0.081	-0.151 $\pm$ 0.188	0.266 $\pm$ 0.046

D.1 *Trojan-Bench(H)*

Correlation and error patterns on *Trojan-Bench(H)*. Across both Full and Blind templates, the strongest graders on *Trojan-Bench(H)* are the thinking-enabled variants of frontier models. Under the Full setting, DeepSeek-V4-Flash (Think) leads on most aggregate metrics with mean Pearson r of 0.66, mean QWK of 0.63, and mean MAE of 1.38 across attempts, closely followed by GPT-5 (0.62 Pearson, 0.56 QWK) and DeepSeek-V3.2 (Think) (0.62 Pearson, 0.61 QWK). The non-thinking variants of the same model families lose between 0.05 and 0.20 QWK and pick up systematic over-grading bias, often exceeding $+1.0$ points. Smaller open-weight graders such as Qwen3-8B, Prometheus-V2-7B, and Prometheus-V2-8$\times$7B, together with GPT-4o, collapse to near-zero or negative correlation with the human reference, with MAE close to or exceeding 3.0, which is the level of a constant "award full marks" baseline.

Effect of aggregation. Mean-CoT@16 and MajorityVote@16 produce small but consistent gains over a single CoT call. Gemini-2.5-Flash improves from QWK 0.59 at CoT@1 to 0.68 at Mean-CoT@16 in the Full setting, and DeepSeek-V4-Flash (Think) improves from 0.64 to 0.69. MajorityVote@16 produces similar gains except on graders whose all samples agree, in which case voting cannot denoise a deterministic bias (a pattern we revisit in the case-study section). The Blind setting shows the same ordering of graders as Full but with a uniform downward shift in correlation, and a sign flip in bias for several frontier models. DeepSeek-V3.2 (Think), for example, moves from $+0.27$ Full to -0.88 Blind, which reflects the absence of the rubric anchor.

Smaller graders behave differently on *Trojan-Bench(H)*. A non-trivial ranking shift appears for the small graders. Qwen3-8B, GPT-5 Mini, and Kimi-K2.5 reach better QWK and lower MAE under the Blind template than under the Full template. We discuss this counter-intuitive effect in detail in Section J.1. The short version is that the rubric in the Full template acts as a surface-form anchor that biases small graders toward awarding most of the available points on each rubric line.

Table 15: Performance of language models across multiple correlation and error metrics. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the human-labelled *Trojan-Bench(H)* dataset against a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE	RMSE	Bias	WTA$_{\leq 1}$
Claude-Sonnet-4.6 (T)	0.54	0.44	0.44	1.79	2.43	1.33	0.56
Claude-Sonnet-4.6 (T)	0.55	0.45	0.47	1.73	2.40	1.24	0.58
DeepSeek-V3.2 (T)	0.43	0.35	0.29	2.27	3.04	2.02	0.48
DeepSeek-V3.2 (T)	0.60	0.53	0.59	1.35	2.26	0.27	0.71
DeepSeek-V4-Flash (T)	0.66	0.54	0.64	1.33	2.08	-0.50	0.69
Gemini-2.5-Flash	0.62	0.50	0.59	1.54	2.23	0.63	0.58
Gemini-2.5-Pro (T)	0.60	0.49	0.52	1.52	2.20	1.06	0.65
Gemini-2.5-Pro (T)	0.57	0.45	0.49	1.54	2.20	1.04	0.69
Gemini-3-Pro-Prev. (T)	0.54	0.44	0.45	1.65	2.50	1.35	0.65
GPT-4o	0.20	0.17	0.06	3.10	3.83	3.06	0.31
GPT-5	0.57	0.47	0.51	1.56	2.22	0.98	0.65
GPT-5-Mini	0.63	0.52	0.53	1.67	2.25	1.25	0.54
Kimi-K2.5	0.33	0.22	0.20	2.37	3.29	2.24	0.50
Prometheus-V2-7B	-0.09	-0.12	-0.07	2.92	3.66	1.71	0.35
Prometheus-V2-8×7B	0.12	0.03	0.06	3.00	3.72	2.63	0.38
Qwen3-8B	0.15	0.16	0.07	3.04	3.83	2.83	0.34
Qwen3.5-27B	0.43	0.36	0.33	2.19	2.99	1.73	0.50
Qwen3.5-9B	0.30	0.26	0.20	2.58	3.37	2.16	0.39

Table 16: Performance of language models across multiple correlation and error metrics. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the human-labelled *Trojan-Bench(H)* dataset against a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE	RMSE	Bias	WTA$_{\leq 1}$
Claude-Sonnet-4.6 (T)	0.57	0.47	0.46	1.69	2.35	1.31	0.60
Claude-Sonnet-4.6 (T)	0.65	0.53	0.55	1.54	2.13	1.17	0.60
DeepSeek-V3.2 (T)	0.44	0.34	0.29	2.29	3.06	2.08	0.48
DeepSeek-V3.2 (T)	0.65	0.53	0.65	1.35	1.93	0.23	0.67
DeepSeek-V4-Flash (T)	0.72	0.59	0.69	1.25	1.87	-0.67	0.69
Gemini-2.5-Flash	0.69	0.57	0.68	1.23	1.76	0.44	0.73
Gemini-2.5-Pro (T)	0.59	0.48	0.48	1.52	2.24	1.23	0.69
Gemini-2.5-Pro (T)	0.64	0.53	0.54	1.42	2.07	1.08	0.67
Gemini-3-Pro-Prev. (T)	0.59	0.48	0.49	1.58	2.37	1.33	0.65
GPT-4o	0.20	0.17	0.06	3.10	3.83	3.06	0.31
GPT-5	0.67	0.57	0.59	1.44	2.01	0.98	0.69
GPT-5-Mini	0.55	0.46	0.43	1.88	2.52	1.50	0.52
Kimi-K2.5	0.48	0.36	0.31	2.13	2.83	1.92	0.46
Prometheus-V2-7B	0.09	0.06	0.04	2.48	3.07	1.94	0.42
Prometheus-V2-8×7B	0.13	0.17	0.05	2.73	3.42	2.48	0.33
Qwen3-8B	0.20	0.21	0.08	3.08	3.78	2.92	0.29
Qwen3.5-27B	0.44	0.37	0.32	2.17	2.93	1.83	0.48
Qwen3.5-9B	0.45	0.36	0.31	2.27	3.05	2.02	0.50

Table 17: Performance of language models across multiple correlation and error metrics. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the human-labelled *Trojan-Bench(H)* dataset against a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE	RMSE	Bias	WTA$_{\leq 1}$
Claude-Sonnet-4.6 (T)	0.58	0.47	0.46	1.75	2.37	1.38	0.58
Claude-Sonnet-4.6 (T)	0.61	0.53	0.52	1.65	2.25	1.23	0.56
DeepSeek-V3.2 (T)	0.44	0.34	0.29	2.29	3.08	2.08	0.48
DeepSeek-V3.2 (T)	0.66	0.57	0.65	1.27	2.05	0.10	0.67
DeepSeek-V4-Flash (T)	0.65	0.54	0.63	1.38	2.13	-0.54	0.65
Gemini-2.5-Flash	0.67	0.57	0.64	1.38	2.14	0.63	0.69
Gemini-2.5-Pro (T)	0.54	0.44	0.45	1.58	2.43	1.29	0.63
Gemini-2.5-Pro (T)	0.63	0.51	0.55	1.42	2.16	1.13	0.71
Gemini-3-Pro-Prev. (T)	0.54	0.44	0.45	1.67	2.50	1.38	0.65
GPT-4o	0.20	0.17	0.06	3.10	3.83	3.06	0.31
GPT-5	0.68	0.59	0.62	1.44	2.01	0.98	0.65
GPT-5-Mini	0.57	0.49	0.48	1.81	2.49	1.35	0.56
Kimi-K2.5	0.45	0.35	0.28	2.27	3.05	2.15	0.46
Prometheus-V2-7B	0.13	0.07	0.05	3.23	4.00	3.10	0.31
Prometheus-V2-8×7B	0.11	0.07	0.04	3.19	3.85	2.98	0.25
Qwen3-8B	0.18	0.23	0.06	3.21	3.88	3.04	0.27
Qwen3.5-27B	0.40	0.33	0.30	2.23	3.04	1.81	0.50
Qwen3.5-9B	0.37	0.30	0.26	2.46	3.29	2.04	0.44

Table 18: Performance of language models across multiple correlation and error metrics. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the human-labelled *Trojan-Bench(H)* dataset without access to a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE	RMSE	Bias	WTA$_{\leq 1}$
Claude-Sonnet-4.6 (T)	0.40	0.32	0.38	1.77	2.24	-0.31	0.44
Claude-Sonnet-4.6 (T)	0.49	0.34	0.46	1.60	1.97	0.16	0.56
DeepSeek-V3.2 (T)	0.15	0.04	0.06	2.96	3.64	-2.71	0.29
DeepSeek-V3.2 (T)	0.57	0.50	0.51	1.54	2.12	-0.88	0.56
DeepSeek-V4-Flash (T)	0.40	0.31	0.34	2.25	2.94	-1.38	0.46
Gemini-2.5-Flash	0.42	0.32	0.43	1.86	2.39	-0.01	0.46
Gemini-3-Pro-Prev. (T)	0.48	0.39	0.40	2.00	2.73	1.46	0.52
GPT-4o	0.06	0.00	0.04	2.17	2.55	0.79	0.40
GPT-5	0.45	0.32	0.39	1.71	2.32	-0.87	0.56
GPT-5-Mini	0.37	0.23	0.37	1.92	2.49	-0.29	0.44
Kimi-K2.5	0.41	0.31	0.33	1.92	2.35	-0.92	0.48
Qwen3-8B	0.28	0.21	0.26	2.35	3.13	0.91	0.44
Qwen3.5-27B	0.44	0.34	0.44	1.69	2.31	-0.31	0.52

Table 19: Performance of language models across multiple correlation and error metrics. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the human-labelled *Trojan-Bench(H)* dataset without access to a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE	RMSE	Bias	$\text{WTA}_{\leq 1}$
Claude-Sonnet-4.6 (T)	0.39	0.31	0.35	1.77	2.22	-0.40	0.44
Claude-Sonnet-4.6 (T)	0.48	0.36	0.40	1.63	2.03	-0.21	0.50
DeepSeek-V3.2 (T)	0.20	0.12	0.06	3.00	3.63	-2.83	0.31
DeepSeek-V3.2 (T)	0.61	0.48	0.52	1.52	2.03	-0.85	0.60
DeepSeek-V4-Flash (T)	0.51	0.39	0.42	2.04	2.67	-1.46	0.50
Gemini-2.5-Flash	0.44	0.33	0.43	1.75	2.29	-0.25	0.54
Gemini-3-Pro-Prev. (T)	0.50	0.42	0.39	2.06	2.63	1.56	0.46
GPT-4o	0.05	0.01	0.03	2.10	2.54	0.81	0.44
GPT-5	0.53	0.40	0.48	1.58	2.08	-0.63	0.56
GPT-5-Mini	0.53	0.39	0.53	1.63	2.07	-0.08	0.54
Kimi-K2.5	0.54	0.44	0.43	1.71	2.14	-0.92	0.48
Qwen3-8B	0.33	0.25	0.28	2.08	2.58	1.04	0.46
Qwen3.5-27B	0.49	0.38	0.48	1.56	2.12	-0.10	0.56

Table 20: Performance of language models across multiple correlation and error metrics. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the human-labelled *Trojan-Bench(H)* dataset without access to a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE	RMSE	Bias	$\text{WTA}_{\leq 1}$
Claude-Sonnet-4.6 (T)	0.46	0.38	0.42	1.67	2.15	-0.46	0.48
Claude-Sonnet-4.6 (T)	0.47	0.35	0.40	1.63	2.03	-0.13	0.54
DeepSeek-V3.2 (T)	0.09	0.06	0.02	3.15	3.83	-3.02	0.29
DeepSeek-V3.2 (T)	0.59	0.48	0.49	1.58	2.18	-1.08	0.58
DeepSeek-V4-Flash (T)	0.49	0.39	0.42	2.10	2.74	-1.35	0.48
Gemini-2.5-Flash	0.38	0.29	0.38	1.85	2.56	-0.06	0.52
Gemini-3-Pro-Prev. (T)	0.42	0.35	0.33	2.17	2.86	1.58	0.46
GPT-4o	0.05	0.01	0.03	2.08	2.52	0.79	0.46
GPT-5	0.50	0.38	0.47	1.60	2.17	-0.56	0.54
GPT-5-Mini	0.51	0.38	0.51	1.63	2.24	-0.25	0.56
Kimi-K2.5	0.50	0.41	0.42	1.79	2.29	-1.00	0.50
Qwen3-8B	0.29	0.25	0.25	2.40	3.16	1.27	0.44
Qwen3.5-27B	0.46	0.36	0.46	1.65	2.26	-0.31	0.56

D.2 *Trojan-Bench(S)*

Correlation and error patterns on *Trojan-Bench(S)*. The synthetic benchmark is constructed so that every candidate solution has a target rubric-aligned grade, which makes the Full setting easier than on *Trojan-Bench(H)*. Under the Full template, Claude-Sonnet-4.6 (Think) achieves Pearson r of 0.81 and QWK of 0.75 at the aggregated level; DeepSeek-V3.2 (Think) reaches 0.81 Pearson and 0.77 QWK with MeanCoT@16; and GPT-5 reaches 0.77 Pearson and 0.69 QWK. Both Mean-CoT and MajorityVote lift aggregate quality by a few points relative to a single CoT call. MAE values cluster around or below 1.0 for the strongest graders, which means the synthetic distribution is concentrated where rubric-following alone is enough to recover the target grade.

Failure modes carry over from *Trojan-Bench(H)*. Two of the failure modes we observe on *Trojan-Bench(H)* also appear on the synthetic data. First, GPT-4o has near-zero QWK (0.12) despite a non-trivial Pearson correlation (0.53) and a bias of $+3.37$, which is the signature of a grader that ranks proofs in roughly the right order but inflates every grade to the maximum. Second, the lower-tier graders (Kimi-K2.5, Qwen3.5-27B, GPT-5 Mini) carry a positive bias of about $+1.6$ to $+2.2$, again driven by over-crediting partially correct rubric items. Under the Blind template, the systematic over-grading shrinks because the rubric anchor is removed, and most graders move to a small positive or near-zero bias under MajorityVote@16.

Cross-set consistency. The relative ordering of graders is largely preserved between *Trojan-Bench(H)* and *Trojan-Bench(S)*. The top three graders on the Full template are the same on both datasets, and the bottom three are also the same. The absolute values are higher on the synthetic set because the candidate distribution has lower per-rubric-item ambiguity by construction, but the directional conclusions match. This cross-set consistency is the basis for our claim in the main text that fine-tuning on the synthetic set transfers usefully to the human set.

Table 21: Mean $\pm$ standard deviation of Pearson r, Kendall τ, QWK, and MAE across individual CoT sampling attempts per problem. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the synthetically-generated dataset, *Trojan-Bench(S)*, against a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE
Claude-Sonnet-4.6 (T)	0.772 ± 0.067	0.677 ± 0.079	0.713 ± 0.075	0.843 ± 0.157
DeepSeek-V3.2 (T)	0.729 ± 0.007	0.630 ± 0.007	0.423 ± 0.007	2.008 ± 0.018
DeepSeek-V3.2 (T)	0.768 ± 0.019	0.652 ± 0.021	0.724 ± 0.019	1.020 ± 0.039
DeepSeek-V4-Flash (T)	0.587 ± 0.042	0.488 ± 0.037	0.533 ± 0.036	1.380 ± 0.051
GPT-4o	0.527 ± 0.007	0.454 ± 0.004	0.120 ± 0.003	3.371 ± 0.009
GPT-5	0.745 ± 0.018	0.626 ± 0.017	0.658 ± 0.012	1.121 ± 0.039
GPT-5-Mini	0.657 ± 0.008	0.519 ± 0.007	0.445 ± 0.004	1.779 ± 0.018
Kimi-K2.5	0.692 ± 0.026	0.572 ± 0.024	0.363 ± 0.018	2.246 ± 0.033
Qwen3.5-27B	0.680 ± 0.011	0.572 ± 0.016	0.395 ± 0.006	2.183 ± 0.025
Qwen3.5-9B	0.732 ± 0.017	0.604 ± 0.019	0.528 ± 0.014	1.731 ± 0.055

Table 22: Mean $\pm$ standard deviation of RMSE, bias, and exact-match accuracy across individual CoT sampling attempts per problem. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the synthetically-generated dataset, *Trojan-Bench(S)*, against a standard markscheme and golden solution.

Model	RMSE	Bias	Exact
Claude-Sonnet-4.6 (T)	1.256 ± 0.189	0.609 ± 0.093	0.436 ± 0.080
DeepSeek-V3.2 (T)	2.337 ± 0.020	1.978 ± 0.018	0.080 ± 0.005
DeepSeek-V3.2 (T)	1.398 ± 0.052	0.093 ± 0.037	0.312 ± 0.016
DeepSeek-V4-Flash (T)	1.848 ± 0.068	-0.530 ± 0.041	0.237 ± 0.006
GPT-4o	3.628 ± 0.010	3.371 ± 0.009	0.005 ± 0.000
GPT-5	1.549 ± 0.022	0.769 ± 0.038	0.312 ± 0.019
GPT-5-Mini	2.278 ± 0.022	1.661 ± 0.042	0.195 ± 0.003
Kimi-K2.5	2.592 ± 0.048	2.223 ± 0.036	0.074 ± 0.005
Qwen3.5-27B	2.611 ± 0.013	2.100 ± 0.014	0.126 ± 0.013
Qwen3.5-9B	2.242 ± 0.018	1.498 ± 0.030	0.210 ± 0.024

Table 23: Mean $\pm$ standard deviation of Pearson r, Kendall τ, QWK, and MAE across individual CoT sampling attempts per problem. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the synthetically-generated dataset, *Trojan-Bench(S)*, without access to a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE
Claude-Sonnet-4.6 (T)	0.507 ± 0.067	0.411 ± 0.058	0.493 ± 0.058	1.188 ± 0.066
DeepSeek-V4-Flash (T)	0.464 ± 0.044	0.367 ± 0.044	0.432 ± 0.041	1.561 ± 0.054
GPT-4o	0.427 ± 0.005	0.368 ± 0.015	0.120 ± 0.003	3.034 ± 0.010
GPT-5	0.490 ± 0.024	0.397 ± 0.031	0.459 ± 0.025	1.341 ± 0.079
GPT-5-Mini	0.503 ± 0.013	0.394 ± 0.014	0.423 ± 0.012	1.538 ± 0.029
Kimi-K2.5	0.452 ± 0.035	0.352 ± 0.018	0.391 ± 0.031	1.590 ± 0.052
Qwen3.5-27B	0.379 ± 0.021	0.245 ± 0.021	0.346 ± 0.020	1.787 ± 0.037
Qwen3.5-9B	0.409 ± 0.090	0.336 ± 0.071	0.314 ± 0.070	1.900 ± 0.133

Table 24: Mean $\pm$ standard deviation of RMSE, bias, and exact-match accuracy across individual CoT sampling attempts per problem. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the synthetically-generated dataset, *Trojan-Bench(S)*, without access to a standard markscheme and golden solution.

Model	RMSE	Bias	Exact
Claude-Sonnet-4.6 (T)	1.563 ± 0.065	0.267 ± 0.235	0.264 ± 0.029
DeepSeek-V4-Flash (T)	1.994 ± 0.082	-0.374 ± 0.058	0.203 ± 0.037
GPT-4o	3.332 ± 0.011	2.986 ± 0.013	0.031 ± 0.003
GPT-5	1.709 ± 0.067	0.576 ± 0.043	0.212 ± 0.033
GPT-5-Mini	1.943 ± 0.031	0.990 ± 0.045	0.193 ± 0.030
Kimi-K2.5	1.978 ± 0.076	0.892 ± 0.016	0.175 ± 0.018
Qwen3.5-27B	2.288 ± 0.047	-0.040 ± 0.024	0.196 ± 0.007
Qwen3.5-9B	2.336 ± 0.148	1.290 ± 0.083	0.139 ± 0.025

Table 25: Performance of language models across multiple correlation and error metrics. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the synthetically-generated dataset, *Trojan-Bench(S)*, against a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE	RMSE	Bias	$\text{WTA}_{\leq 1}$
Claude-Sonnet-4.6 (T)	0.81	0.73	0.75	0.78	1.12	0.61	0.80
DeepSeek-V3.2 (T)	0.72	0.62	0.42	2.01	2.35	1.98	0.36
DeepSeek-V3.2 (T)	0.77	0.65	0.73	1.00	1.37	0.05	0.79
DeepSeek-V4-Flash (T)	0.57	0.47	0.52	1.43	1.89	-0.48	0.63
GPT-4o	0.53	0.45	0.12	3.37	3.63	3.37	0.10
GPT-5	0.73	0.61	0.64	1.18	1.58	0.78	0.67
GPT-5-Mini	0.65	0.51	0.44	1.79	2.30	1.67	0.52
Kimi-K2.5	0.67	0.55	0.35	2.24	2.59	2.20	0.31
Qwen3.5-27B	0.69	0.59	0.40	2.19	2.60	2.10	0.35
Qwen3.5-9B	0.75	0.63	0.54	1.75	2.24	1.51	0.52

Table 26: Performance of language models across multiple correlation and error metrics. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the synthetically-generated dataset, *Trojan-Bench(S)*, against a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE	RMSE	Bias	$\text{WTA}_{\leq 1}$
Claude-Sonnet-4.6 (T)	0.74	0.65	0.66	0.86	1.36	0.70	0.78
DeepSeek-V3.2 (T)	0.73	0.63	0.42	2.01	2.32	1.98	0.35
DeepSeek-V3.2 (T)	0.81	0.69	0.77	0.92	1.23	0.05	0.83
DeepSeek-V4-Flash (T)	0.63	0.53	0.59	1.23	1.65	-0.51	0.69
GPT-4o	0.53	0.46	0.12	3.37	3.63	3.37	0.10
GPT-5	0.77	0.66	0.69	1.02	1.45	0.73	0.75
GPT-5-Mini	0.67	0.53	0.46	1.72	2.20	1.64	0.52
Kimi-K2.5	0.70	0.58	0.36	2.25	2.58	2.23	0.31
Qwen3.5-27B	0.69	0.58	0.40	2.17	2.58	2.10	0.37
Qwen3.5-9B	0.73	0.60	0.52	1.70	2.18	1.49	0.52

Table 27: Performance of language models across multiple correlation and error metrics. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the synthetically-generated dataset, *Trojan-Bench(S)*, against a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE	RMSE	Bias	$\text{WTA}_{\leq 1}$
Claude-Sonnet-4.6 (T)	0.72	0.62	0.64	0.90	1.44	0.74	0.76
DeepSeek-V3.2 (T)	0.73	0.63	0.42	2.01	2.33	1.98	0.35
DeepSeek-V3.2 (T)	0.77	0.65	0.72	1.03	1.41	0.08	0.77
DeepSeek-V4-Flash (T)	0.59	0.49	0.54	1.36	1.83	-0.50	0.66
GPT-4o	0.52	0.45	0.12	3.38	3.64	3.38	0.10
GPT-5	0.77	0.65	0.67	1.09	1.52	0.79	0.73
GPT-5-Mini	0.66	0.52	0.45	1.78	2.29	1.67	0.52
Kimi-K2.5	0.70	0.57	0.36	2.26	2.61	2.25	0.30
Qwen3.5-27B	0.69	0.58	0.40	2.14	2.56	2.07	0.37
Qwen3.5-9B	0.73	0.60	0.52	1.76	2.25	1.53	0.52

Table 28: Performance of language models across multiple correlation and error metrics. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the synthetically-generated dataset, *Trojan-Bench(S)*, without access to a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE	RMSE	Bias	$\text{WTA}_{\leq 1}$
Claude-Sonnet-4.6 (T)	0.49	0.37	0.46	1.16	1.55	0.48	0.71
DeepSeek-V4-Flash (T)	0.50	0.41	0.47	1.50	1.90	-0.36	0.60
GPT-4o	0.43	0.36	0.12	3.03	3.32	2.98	0.14
GPT-5	0.50	0.40	0.46	1.36	1.72	0.62	0.63
GPT-5-Mini	0.51	0.40	0.44	1.52	1.90	0.93	0.56
Kimi-K2.5	0.41	0.33	0.36	1.64	2.05	0.91	0.52
Qwen3.5-27B	0.35	0.22	0.32	1.84	2.35	-0.06	0.51
Qwen3.5-9B	0.29	0.24	0.22	2.06	2.50	1.26	0.40

Table 29: Performance of language models across multiple correlation and error metrics. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the synthetically-generated dataset, *Trojan-Bench(S)*, without access to a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE	RMSE	Bias	$\text{WTA}_{\leq 1}$
Claude-Sonnet-4.6 (T)	0.50	0.41	0.50	1.18	1.55	0.09	0.70
DeepSeek-V4-Flash (T)	0.48	0.38	0.46	1.47	1.86	-0.35	0.61
GPT-4o	0.43	0.38	0.12	3.04	3.33	2.99	0.14
GPT-5	0.49	0.41	0.46	1.30	1.68	0.53	0.66
GPT-5-Mini	0.53	0.43	0.45	1.45	1.85	0.99	0.59
Kimi-K2.5	0.52	0.41	0.44	1.45	1.78	0.91	0.58
Qwen3.5-27B	0.39	0.26	0.36	1.67	2.19	-0.10	0.55
Qwen3.5-9B	0.46	0.38	0.36	1.76	2.15	1.25	0.46

Table 30: Performance of language models across multiple correlation and error metrics. Thinking-enabled and non-thinking model variants are denoted by T and T, respectively. Results are evaluated on the synthetically-generated dataset, *Trojan-Bench(S)*, without access to a standard markscheme and golden solution.

Model	Pearson r	Kendall τ	QWK	MAE	RMSE	Bias	$\text{WTA}_{\leq 1}$
Claude-Sonnet-4.6 (T)	0.47	0.37	0.47	1.25	1.63	0.07	0.67
DeepSeek-V4-Flash (T)	0.51	0.41	0.47	1.50	1.91	-0.34	0.58
GPT-4o	0.42	0.36	0.12	3.06	3.35	3.01	0.14
GPT-5	0.49	0.40	0.46	1.35	1.72	0.57	0.64
GPT-5-Mini	0.53	0.41	0.45	1.51	1.89	0.96	0.55
Kimi-K2.5	0.49	0.39	0.44	1.54	1.89	0.80	0.55
Qwen3.5-27B	0.36	0.22	0.33	1.83	2.33	-0.16	0.51
Qwen3.5-9B	0.40	0.33	0.31	1.92	2.36	1.25	0.44

D.3 Additional evaluation of BoN methods

The table below extends the main-text *MABGrader* comparison to the synthetic benchmark. Table 31 reports the same baselines (CoT@1, MajorityVote@16 with three templates, BoN@16 with four reward models) on *Trojan-Bench(S)* Full, and confirms that *MABGrader* continues to lead on QWK and RMSE for the three graders run in this setting. The same pattern we observe on *Trojan-Bench(H)* holds here: token-level sampling baselines (MajorityVote, BoN with off-the-shelf reward models) yield essentially flat improvements over a single CoT call, while the prompt-side hypothesis enumeration in *MABGrader* is what produces the consistent gain across graders.

Table 31: Comparative analysis of model inference performance on synthetically generated data across QWK, RMSE, and Bias metrics for DeepSeek-V4-Flash, Kimi-K2.5, Gemini-3-Pro-Preview, and Qwen3.5-27B. Evaluation is performed *with* a reference solution and a corresponding marking scheme.

	QWK (↑)				**RMSE** (↓)				**Bias** (≈ 0)			
	DS V4F	Kimi K2.5	Gem. 3P	Q3.5 27B	DS-V4F	Kimi K2.5	Gem. 3P	Q3.5 27B	DS-V4F	Kimi K2.5	Gem. 3P	Q3.5 27B
Chain-of-Thought (CoT@1)												
MathArena [4]	0.52	0.35	-	0.40	1.89	2.59	-	2.60	-0.48	2.20	-	2.10
Chain-of-Thought with Majority Vote Selection (MajorityVote@16)												
MathArena [4]	0.54	0.36	-	0.41	1.83	2.56	-	2.55	-0.50	2.25	-	2.20
Chain-of-Thought with Best-of-N Selection (BoN@16)												
Skywork [31]	0.51	0.34	-	0.41	1.81	2.61	-	2.67	-0.52	2.35	-	2.15
ArmoRM [56]	0.55	0.35	-	0.41	1.79	2.67	-	2.64	-0.43	2.32	-	2.11
RLHFlow [14]	0.53	0.35	-	0.39	1.85	2.62	-	2.61	-0.41	2.28	-	2.18
Shepherd [57]	0.52	0.36	-	0.37	1.89	2.66	-	2.70	-0.40	2.30	-	2.20
Proposed Method (MABGrader@16)												
MABGrader	0.58	0.49	-	0.53	1.67	2.24	-	1.90	-0.50	0.71	-	-0.96

E Extended Bar plots

The bar plots in Figures 6 - 17 are a visual companion to the numerical tables of the previous section. Each plot reports a single metric family (correlation, error, or agreement) for one (dataset, style) pair, with three bars per model corresponding to CoT@1, Mean-CoT@16, and MajorityVote@16. We include them so that a reader can compare every grader and every aggregation method at a glance for any single metric, without having to cross-reference the numerical tables.

The plots make the patterns from the previous section visually obvious. Frontier thinking-enabled models cluster in the upper band of the correlation plots and in the lower band of the error plots, whereas smaller graders separate cleanly into a lower-correlation, higher-MAE cluster. The agreement plots (Exact Match and WTA ≤ 1) show that close-but-not-exact agreement with the human reference is achievable for most frontier models, while exact agreement remains low even for the strongest graders. The synthetic-set plots show the same ordering as the human-set plots but with a uniform upward shift in correlation and a downward shift in error, consistent with the cross-set discussion above.

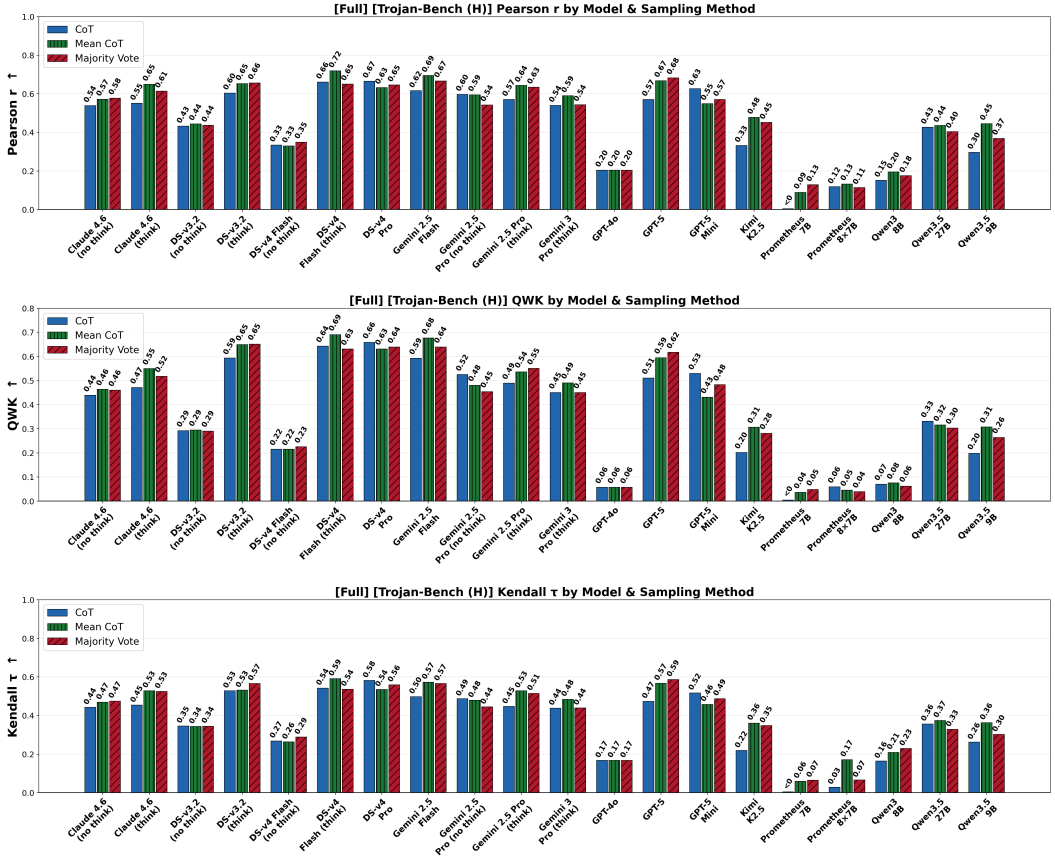

Figure 6: ***Trojan-Bench(H)*** (Full): Correlation metrics (Pearson r, QWK, Kendall τ) by model and sampling method.

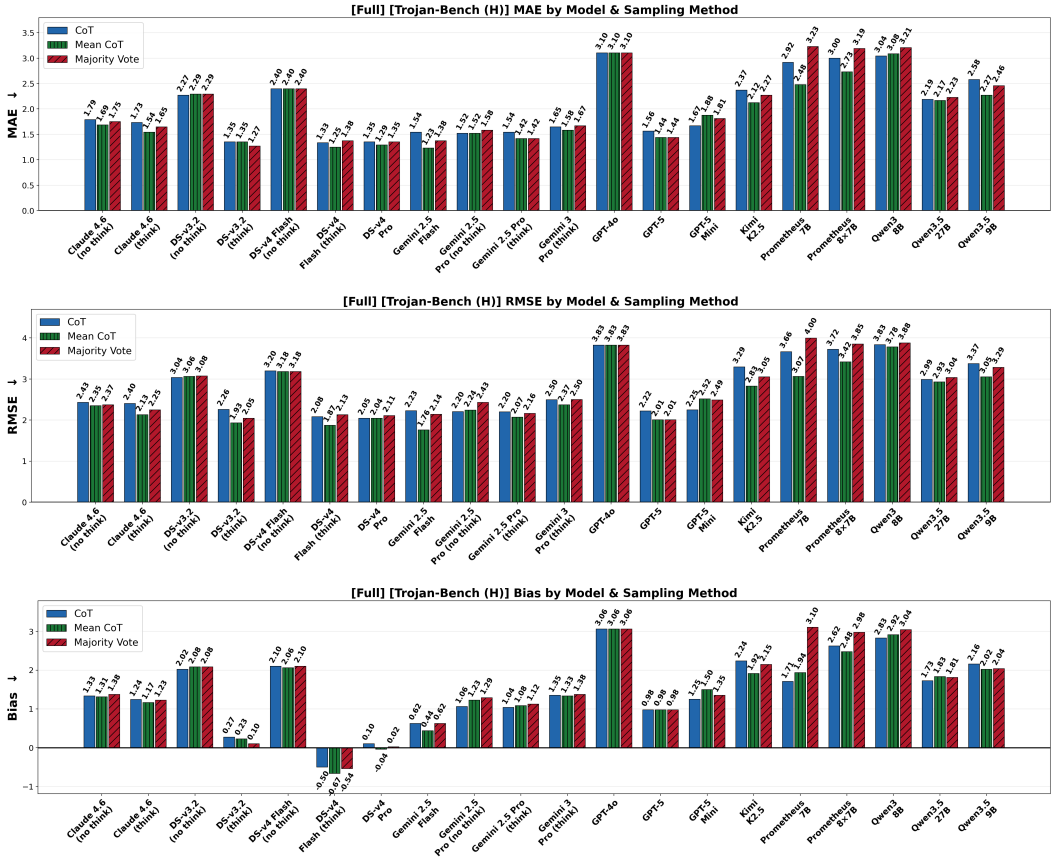

Figure 7: ***Trojan-Bench(H)*** (Full): Error metrics (MAE, RMSE, Bias) by model and sampling method.

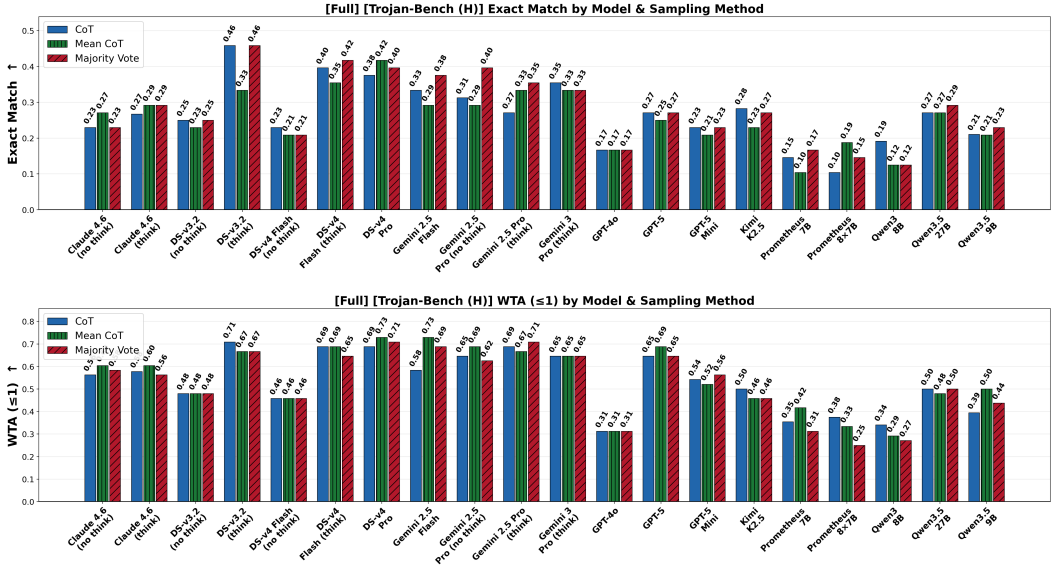

Figure 8: ***Trojan-Bench(H)*** (Full): Agreement metrics (Exact Match, WTA ≤ 1) by model and sampling method.

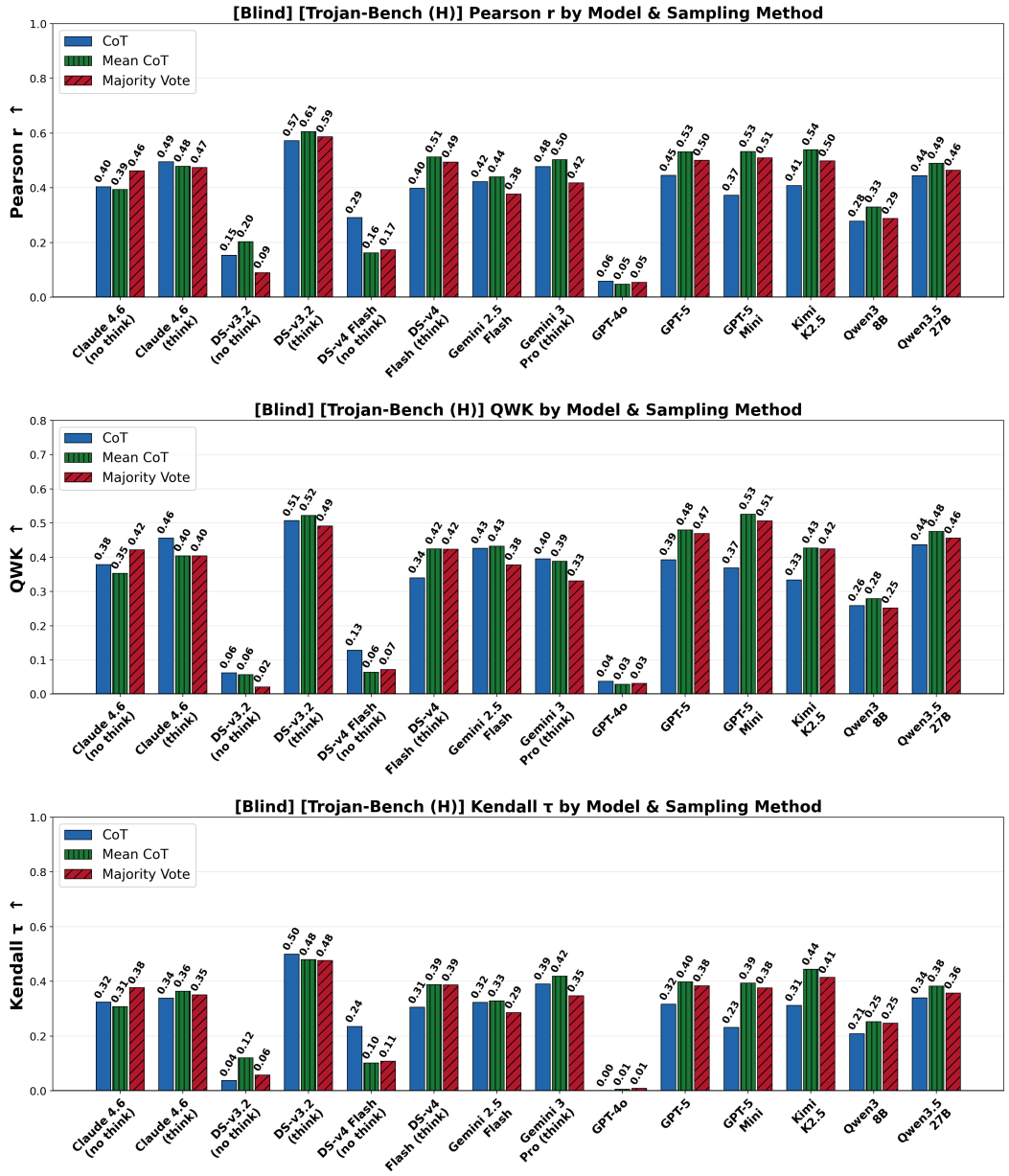

Figure 9: ***Trojan-Bench(H)*** (Blind): Correlation metrics (Pearson r, QWK, Kendall τ) by model and sampling method.

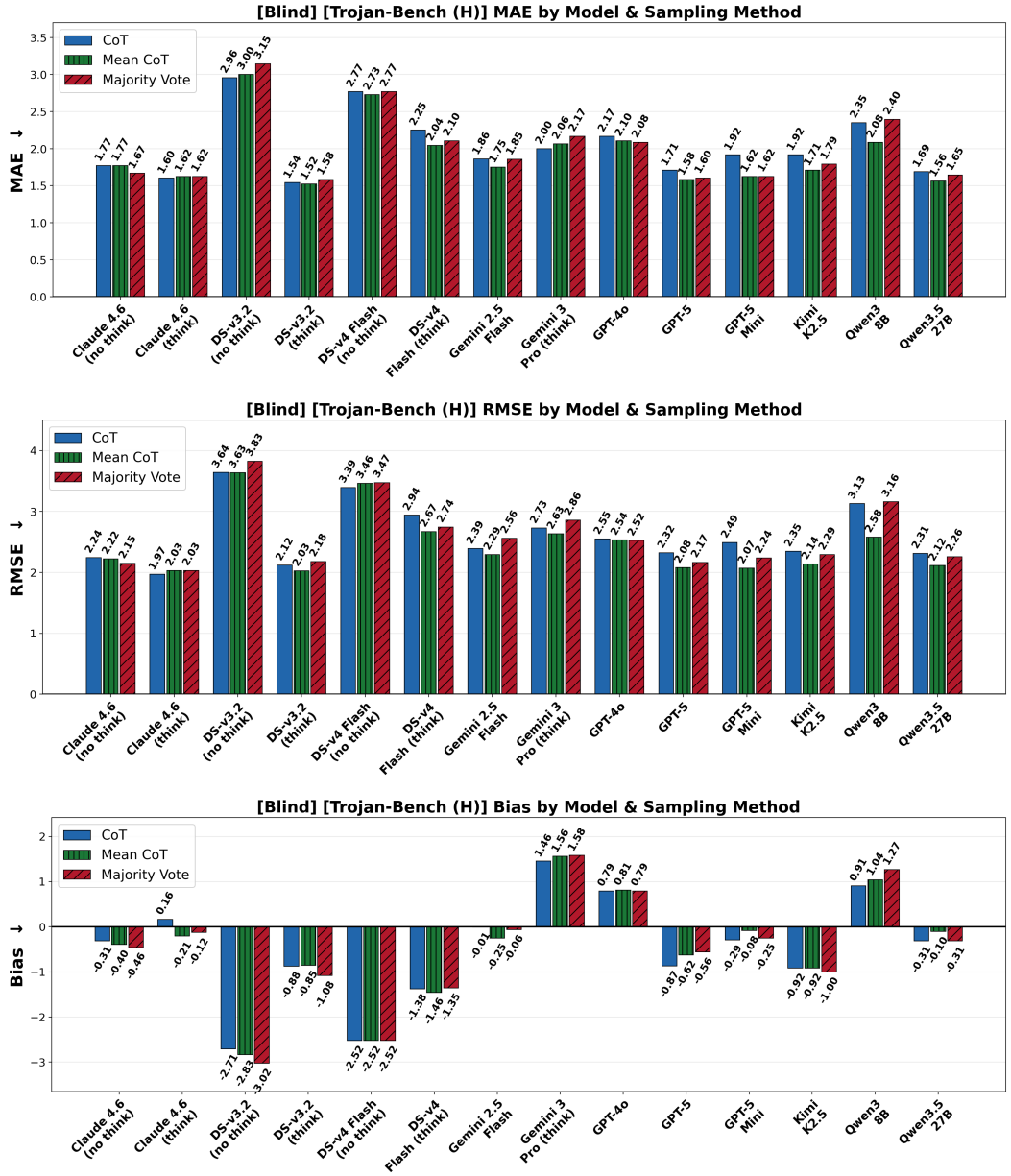

Figure 10: ***Trojan-Bench(H)*** (Blind): Error metrics (MAE, RMSE, Bias) by model and sampling method.

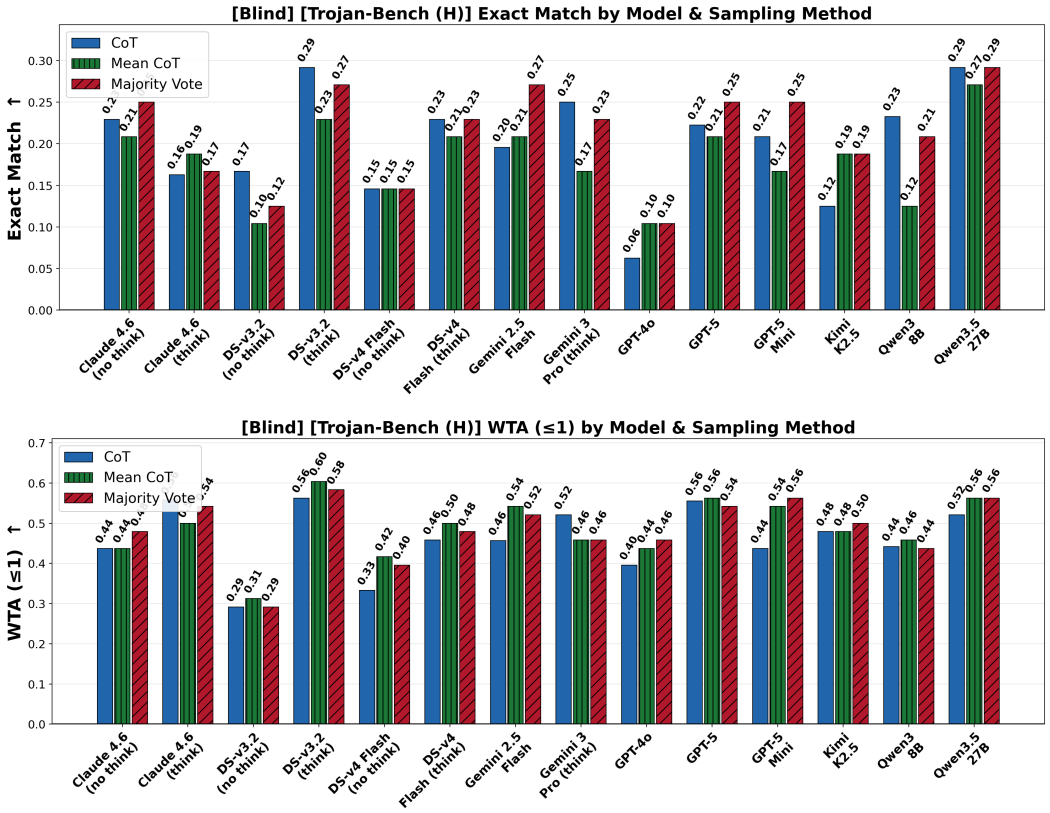

Figure 11: ***Trojan-Bench(H)*** (Blind): Agreement metrics (Exact Match, WTA ≤ 1) by model and sampling method.

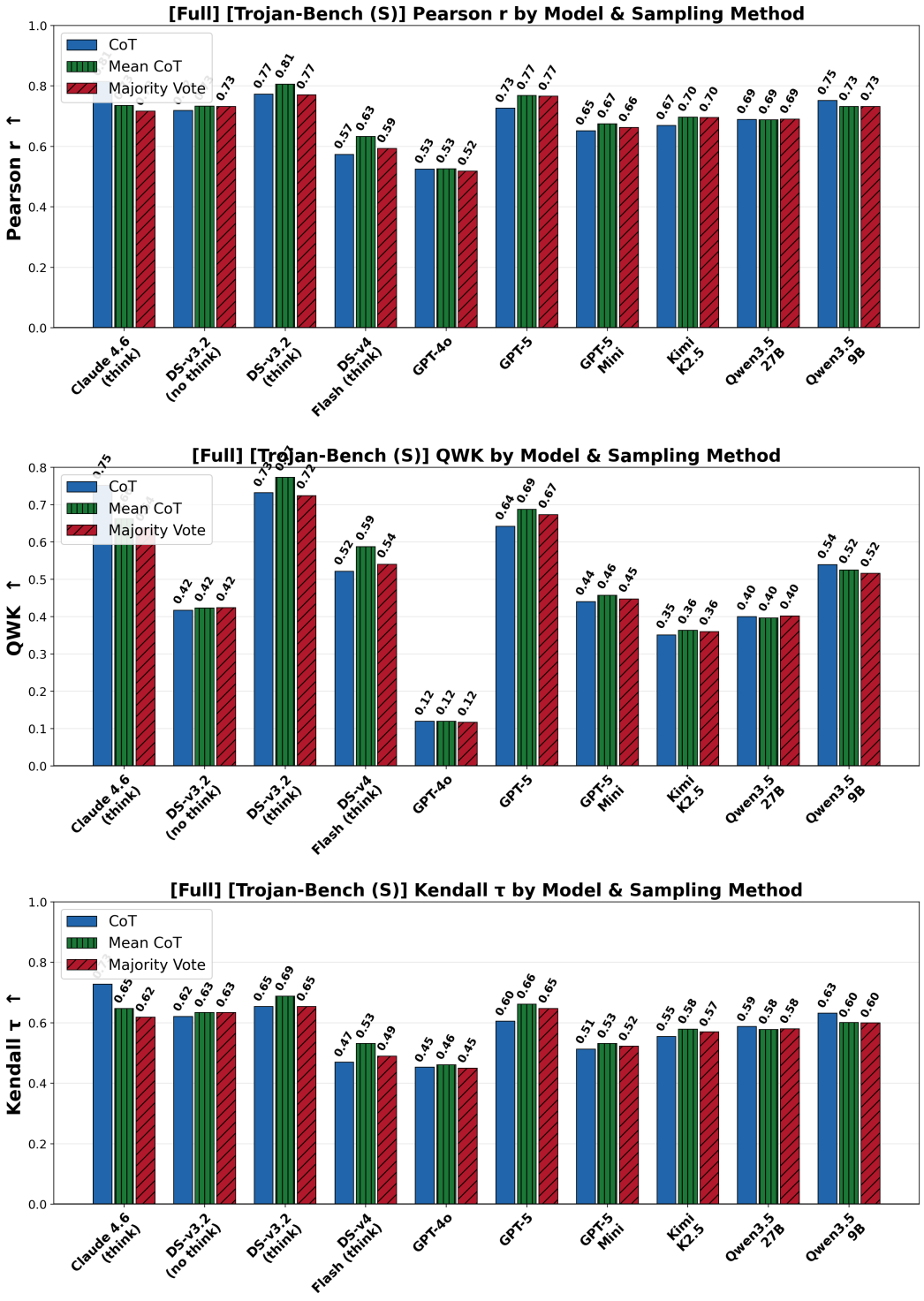

Figure 12: ***Trojan-Bench(S)*** (Full): Correlation metrics (Pearson r, QWK, Kendall τ) by model and sampling method.

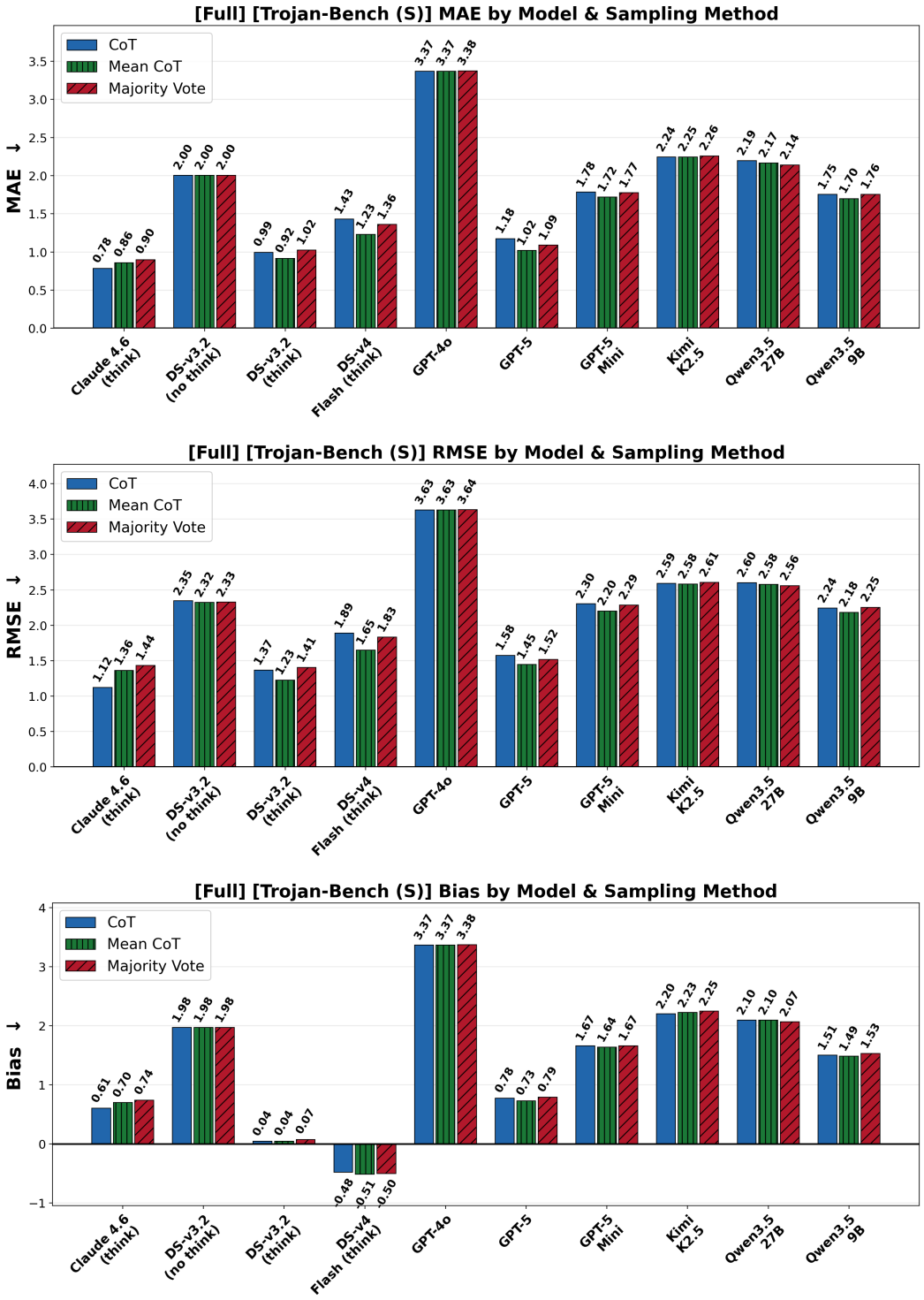

Figure 13: ***Trojan-Bench(S)*** (Full): Error metrics (MAE, RMSE, Bias) by model and sampling method.

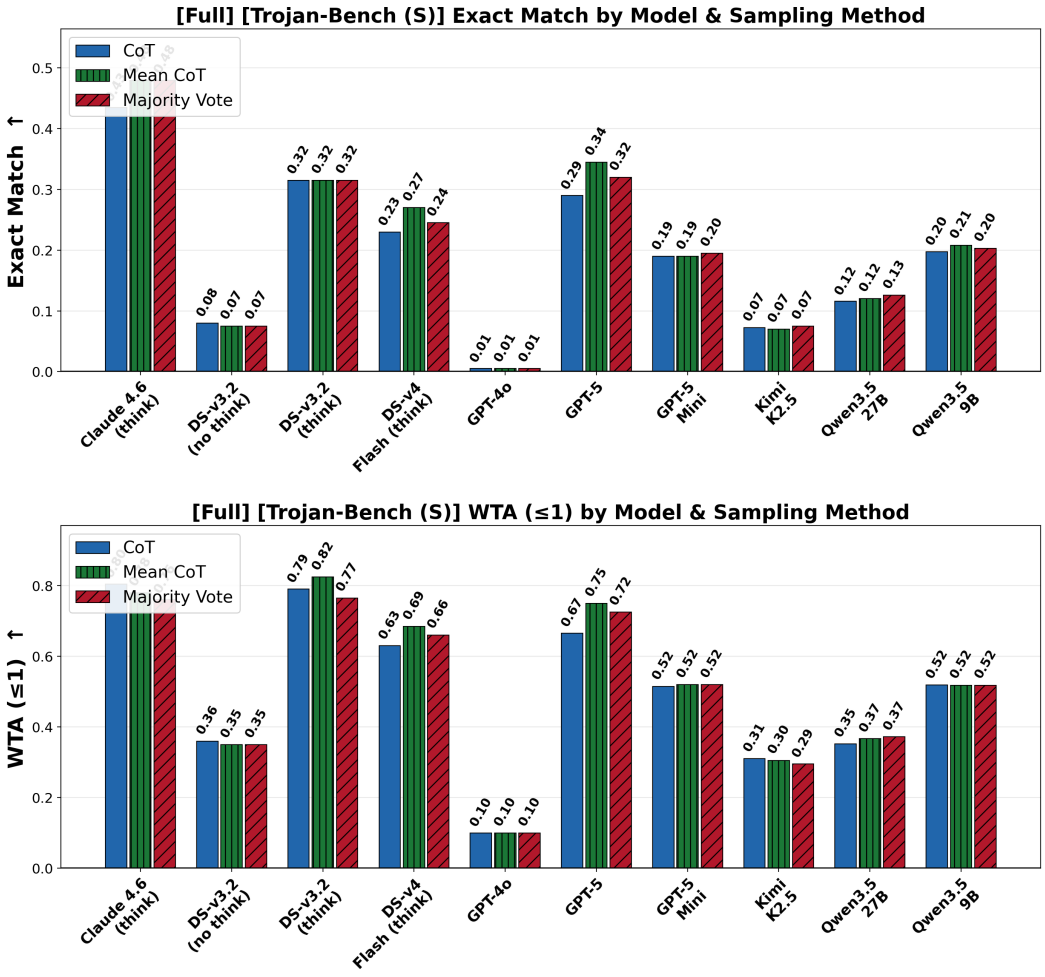

Figure 14: ***Trojan-Bench(S)*** (Full): Agreement metrics (Exact Match, WTA ≤ 1) by model and sampling method.

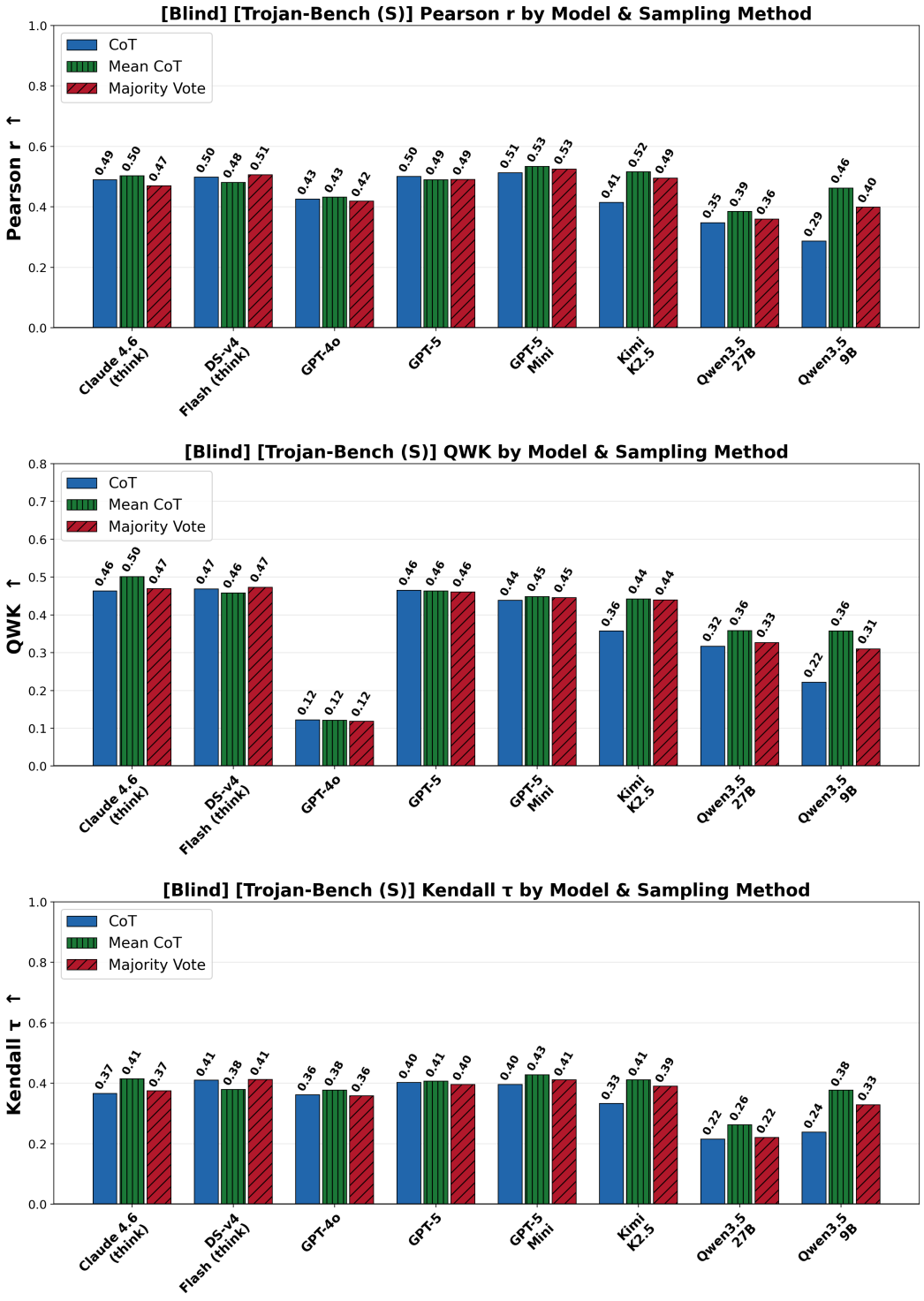

Figure 15: ***Trojan-Bench(S)*** (Blind): Correlation metrics (Pearson r, QWK, Kendall τ) by model and sampling method.

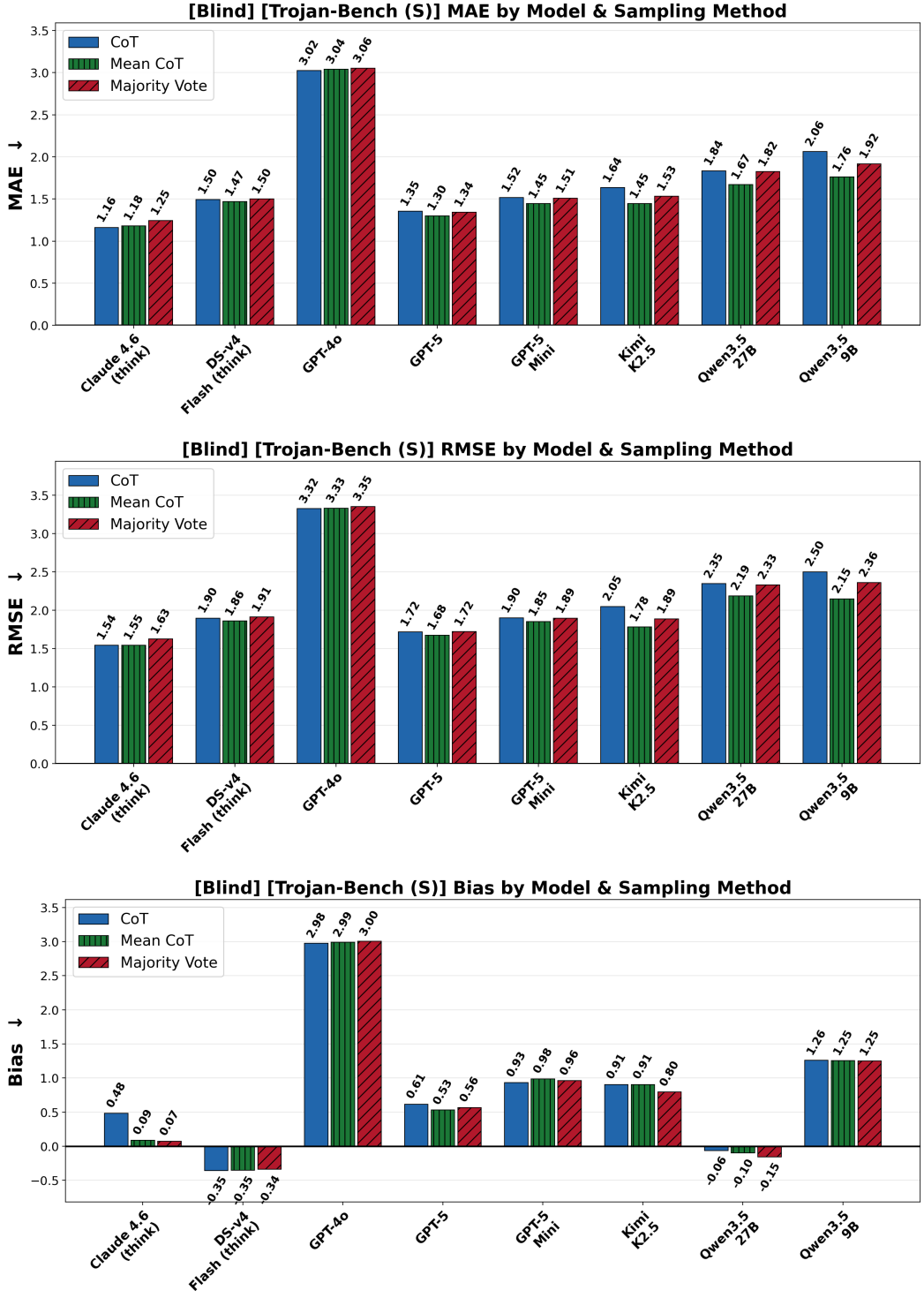

Figure 16: ***Trojan-Bench(S)*** (Blind): Error metrics (MAE, RMSE, Bias) by model and sampling method.

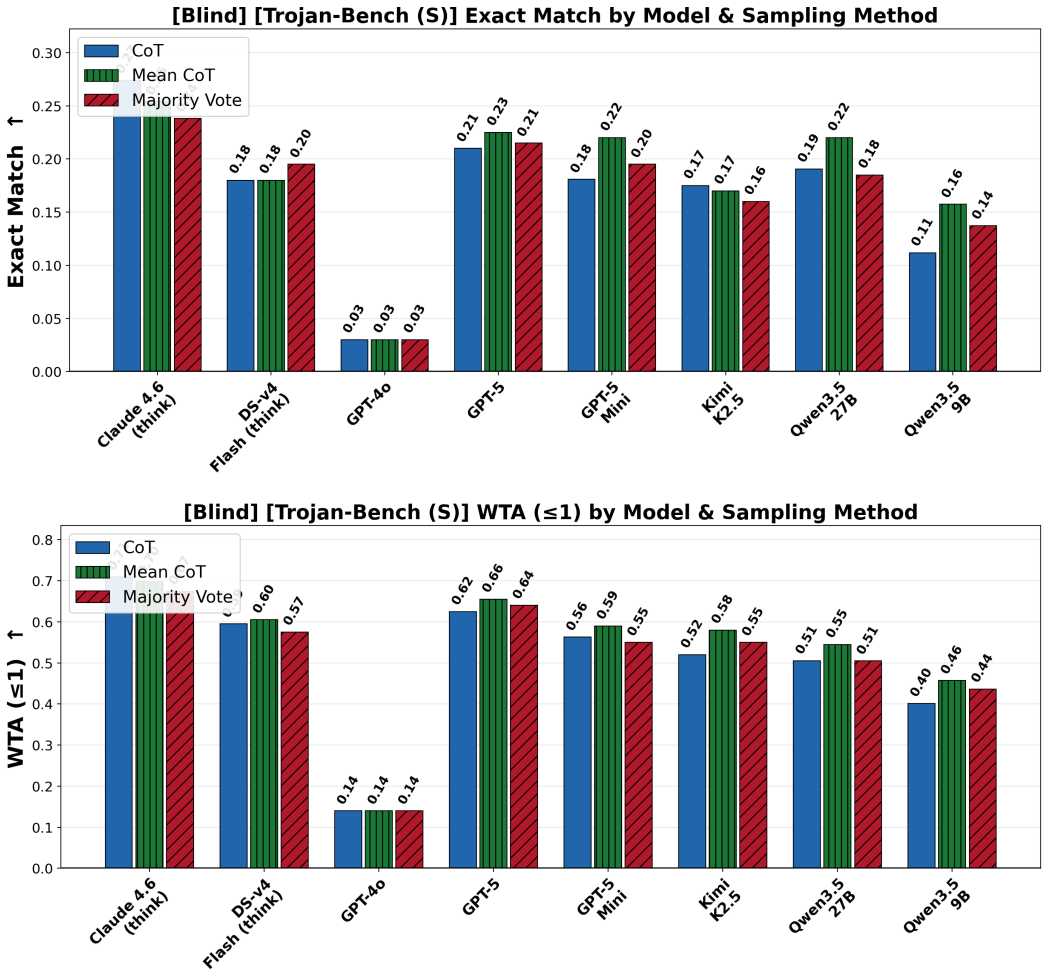

Figure 17: ***Trojan-Bench(S)*** (Blind): Agreement metrics (Exact Match, WTA ≤ 1) by model and sampling method.

F Evaluation of LLM correlation with human and other LLM annotators

The heatmaps in Figures 18–33 report pairwise agreement between every pair of graders in our suite (LLM-vs-LLM cells) together with their agreement with the human reference (LLM-vs-human row and column). Each figure covers one metric and one dataset, with the upper half reporting the Blind template and the lower half reporting the Full template. We include them so that the inter-grader correlation structure is visible at a glance, since a single number per (grader, metric) cell does not capture how strongly graders agree with each other rather than with the reference. All statistics cited in this section are recomputed from the MajorityVote@16 sampling that produced the heatmaps, and the discussion is restricted to the 14 graders run on *Trojan-Bench(H)* in both templates, and the 8 graders run on *Trojan-Bench(S)*.

The most striking pattern across both datasets is that hiding the markscheme degrades ordinal agreement with the human reference far more than it degrades the magnitude of grading error. On *Trojan-Bench(H)* the mean Human-vs-LLM Pearson r drops from 0.50 to 0.39 ($\Delta = -0.11$), Kendall τ from 0.42 to 0.30, and QWK from 0.41 to 0.33, yet mean absolute error is essentially unchanged (1.99 vs 2.00) and RMSE actually edges down slightly ($2.71 \rightarrow 2.60$). The *Trojan-Bench(S)* set shows the same dissociation more sharply: Pearson r collapses by 0.21 ($0.67 \rightarrow 0.46$) and Kendall τ by 0.20, while MAE improves under Blind ($1.83 \rightarrow 1.74$). Removing the markscheme does not cause judges to pick wildly different absolute numbers, since they still place most attempts within roughly one point of the reference, but it strips them of the ability to order attempts correctly.

A second pattern is a near-universal sign flip in bias on *Trojan-Bench(H)*, and a much subtler one on *Trojan-Bench(S)*. Under the Full setting the average judge is stricter than the human (mean bias $= -1.48$); under Blind that average flips to $+0.51$, a swing of almost two grade points, and 10 of 14 *Trojan-Bench(H)* judges individually change sign, including GPT-5 ($-0.98 \rightarrow +0.56$), Claude 4.6 thinking ($-1.23 \rightarrow +0.13$), DeepSeek v3.2 thinking ($-0.10 \rightarrow +1.08$), and DeepSeek v3.2 no-thinking ($-2.08 \rightarrow +3.02$). Only Gemini 3 Pro doubles down on harshness without the rubric ($-1.38 \rightarrow -1.58$), and GPT-4o together with Qwen3-8B remain stricter than the human in both regimes. On *Trojan-Bench(S)*, where the reference is itself an LLM-rubric consensus rather than a human grade, only 1 of 8 judges flips bias sign, which confirms that the flip on *Trojan-Bench(H)* is a real markscheme-versus-human-style effect rather than an artifact of model calibration. Read together with the previous observation, the markscheme behaves like a penalty list: when it is taken away, judges default to a more generous approach.

The LLM-vs-LLM blocks of the heatmaps reveal that consensus between judges itself depends heavily on the rubric and depends on it more strongly than human alignment does. Under Full, the frontier reasoning graders form a tight high-agreement block, with pairwise Pearson typically above 0.7 and pairwise QWK above 0.6, well above their mean agreement with the human reference. Hiding the rubric pulls this block apart: on *Trojan-Bench(H)* the mean inter-LLM Pearson r drops from 0.64 to 0.44 and its pairwise standard deviation almost doubles ($0.15 \rightarrow 0.28$), with DeepSeek v3.2 no-thinking becoming essentially decorrelated from every other judge ($|r| \leq 0.30$, including $r = -0.45$ against GPT-4o) and GPT-4o losing all coherence with the strong-reasoning cluster. The *Trojan-Bench(S)* inter-LLM block weakens less under rubric removal (mean r from 0.63 to 0.54, $\Delta = 0.09$), and there the gap between LLM-consensus and Human-agreement actually flips sign: with the rubric, LLMs match the synthetic ground truth slightly better than they match each other (gap -0.04), but without it they revert to agreeing with each other more than with the reference (gap $+0.09$). Per-model rubric gains are also heterogeneous: frontier reasoning models gain large boosts (DeepSeek v3.2 no-thinking $+0.35$ in r, Gemini 2.5 Flash $+0.29$, GPT-5 $+0.18$), whereas Qwen3-8B, Qwen3.5-27B, and Kimi-K2.5 actually lose a small amount of human alignment when given the rubric. Taken together, the markscheme is not a uniform injection of correct information but a structural scaffold whose benefit scales with the judge's reasoning capacity, and whose absence pushes weaker judges toward random consensus and stronger judges toward systematic leniency. The negative-rubric-gain pattern for small graders is examined further in Section J.1.

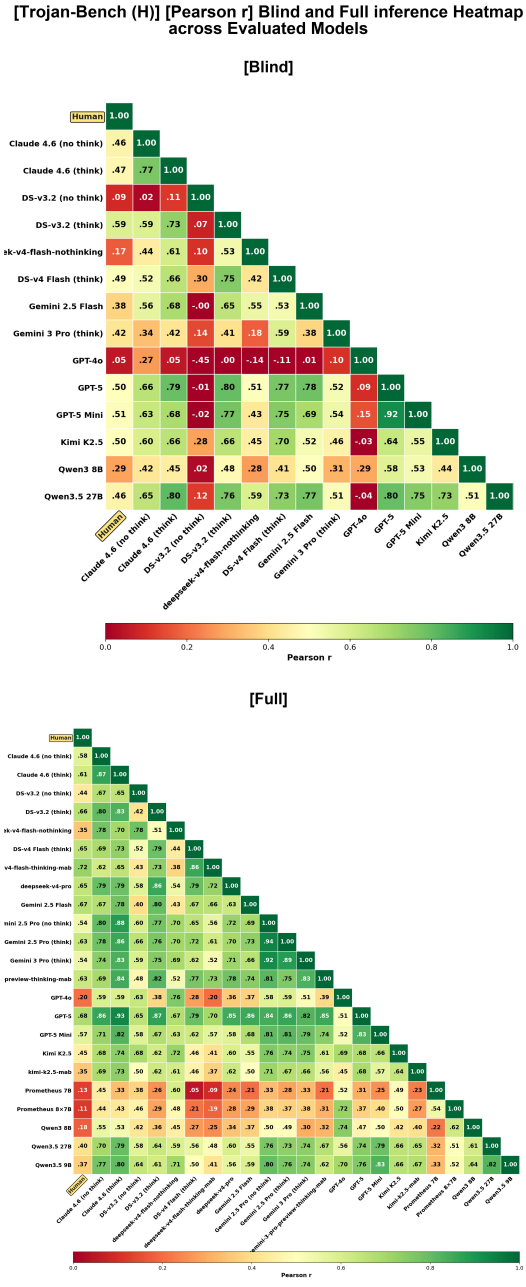

Figure 18: ***Trojan-Bench(H)*** : Pearson r blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full).

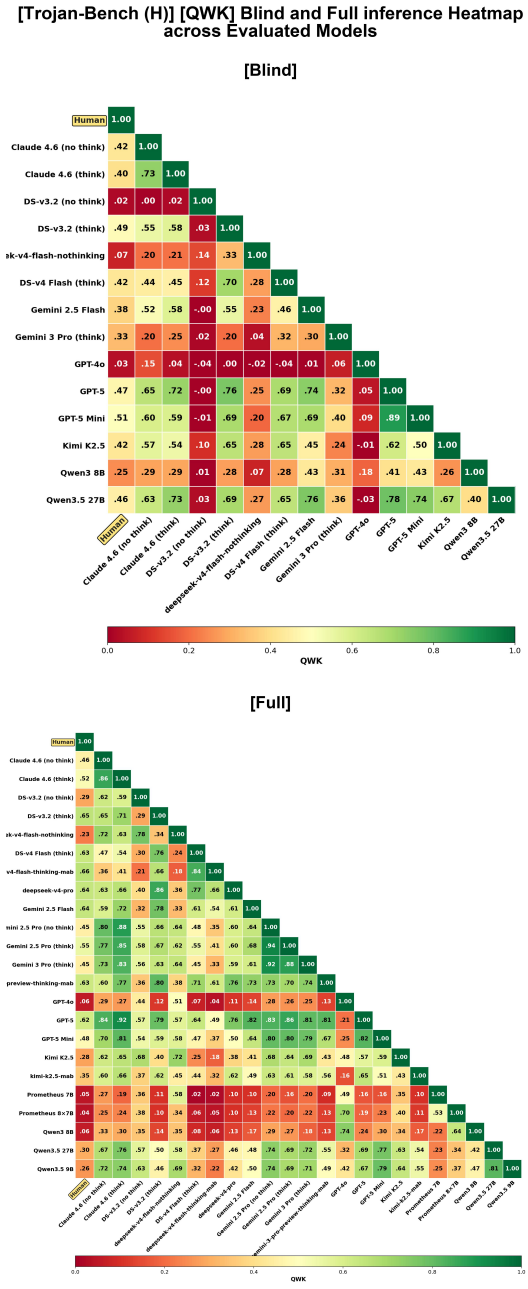

Figure 19: ***Trojan-Bench(H)*** : QWK blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full).

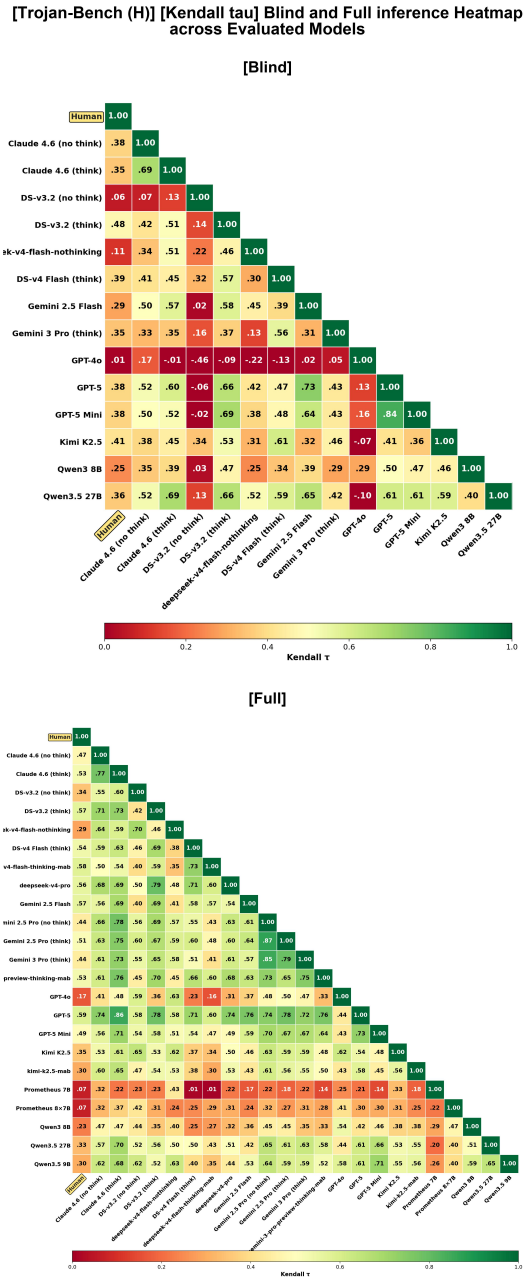

Figure 20: ***Trojan-Bench(H)*** : Kendall τ blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full).

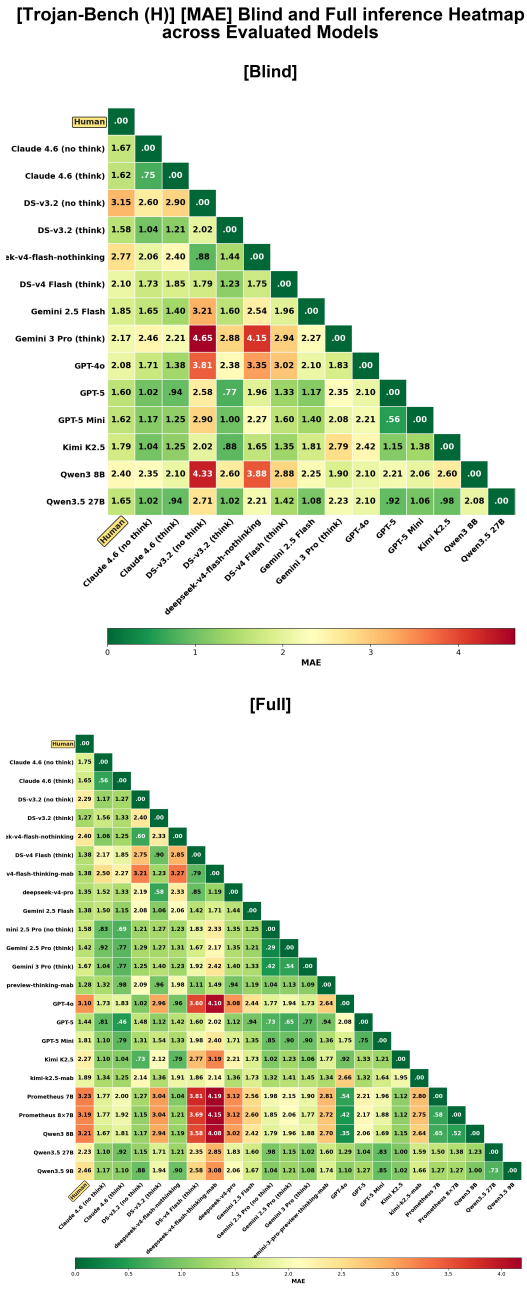

Figure 21: ***Trojan-Bench(H)*** : MAE blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full).

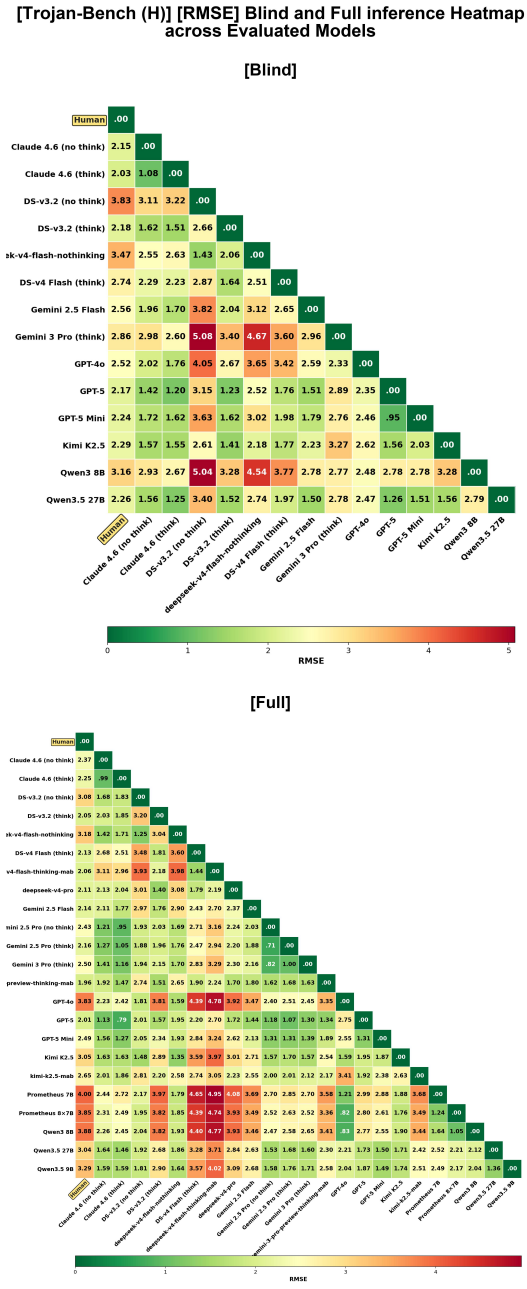

Figure 22: ***Trojan-Bench(H)*** : RMSE blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full).

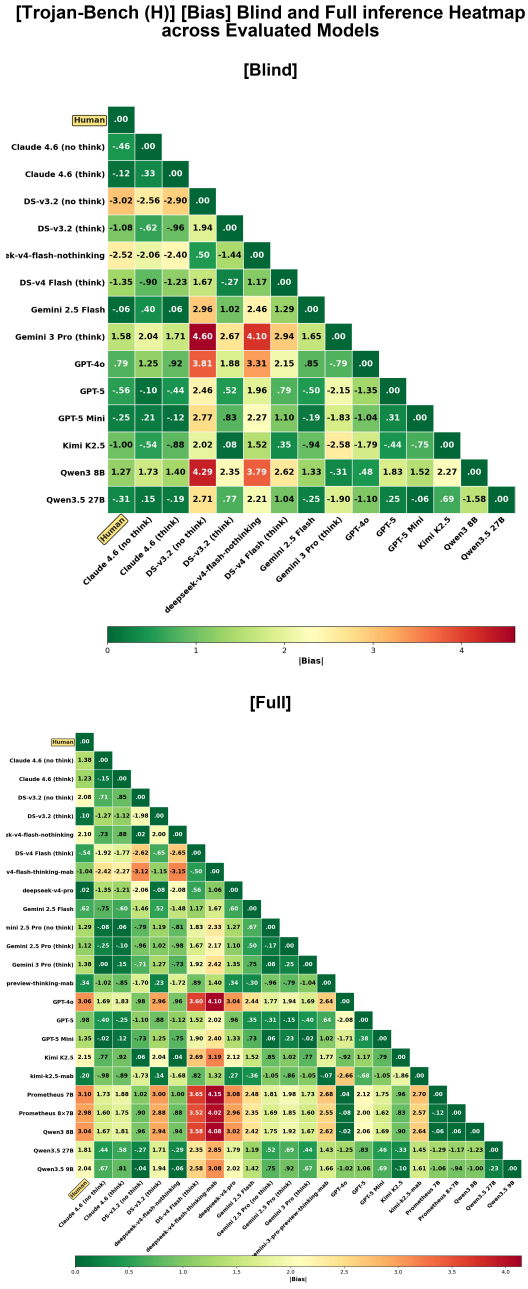

Figure 23: ***Trojan-Bench(H)*** : Bias blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full).

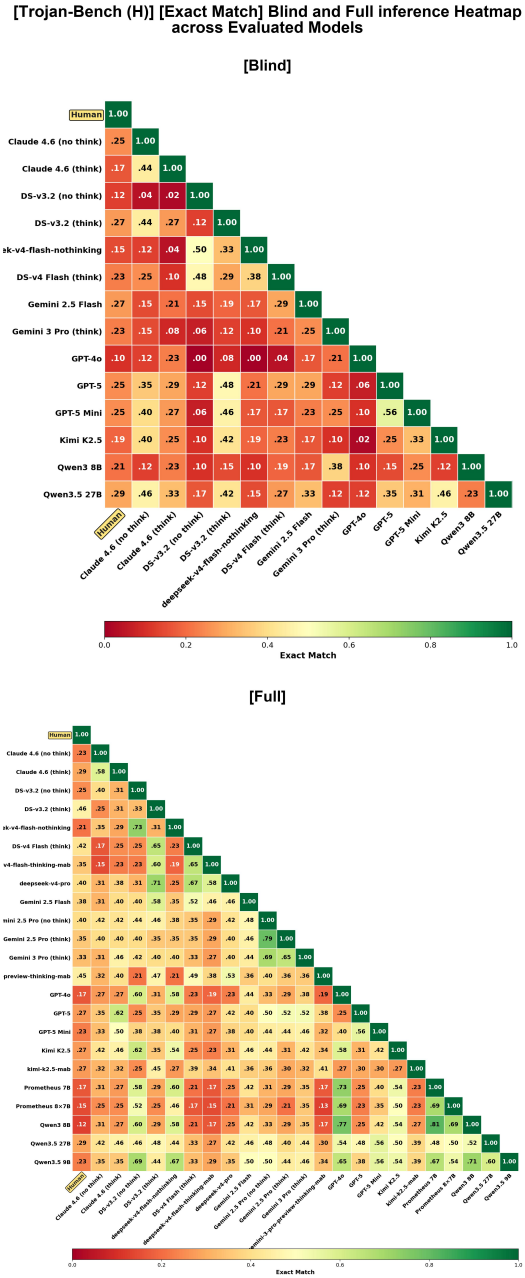

Figure 24: ***Trojan-Bench(H)*** : Exact Match blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full).

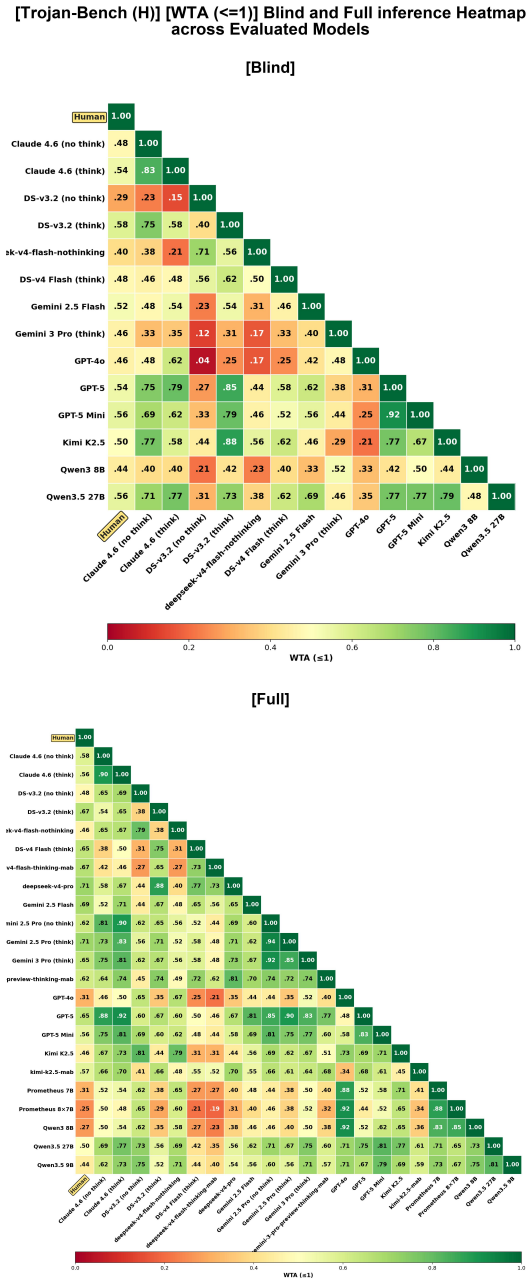

Figure 25: ***Trojan-Bench(H)*** : WTA (≤ 1) blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full).

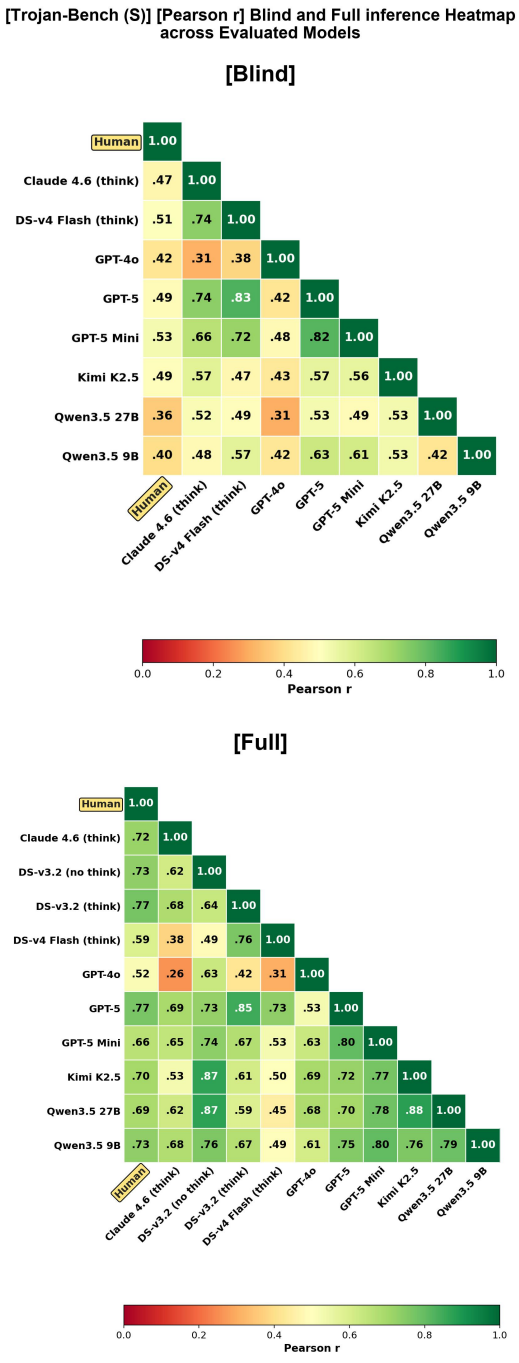

Figure 26: ***Trojan-Bench(S)*** : Pearson r blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full). Note that "human" in this heatmap indicates an LLM assigned grade during data generation with *TROJAN* and serves as a reference point when evaluating the LLMs.

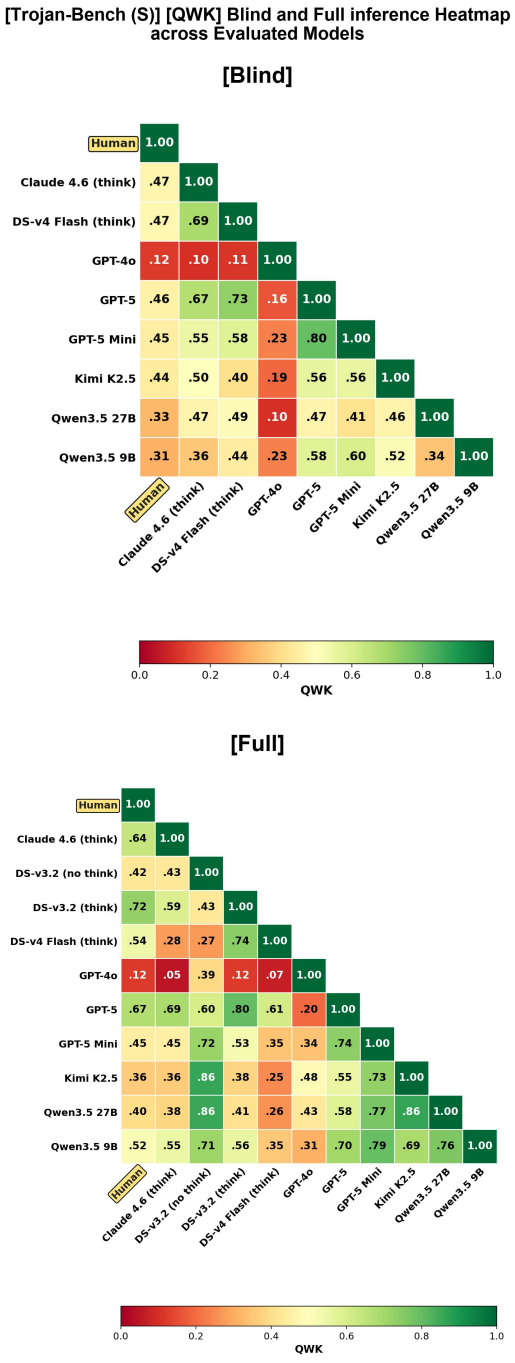

Figure 27: ***Trojan-Bench(S)*** : QWK blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full). Note that "human" in this heatmap indicates an LLM assigned grade during data generation with *TROJAN* and serves as a reference point when evaluating the LLMs.

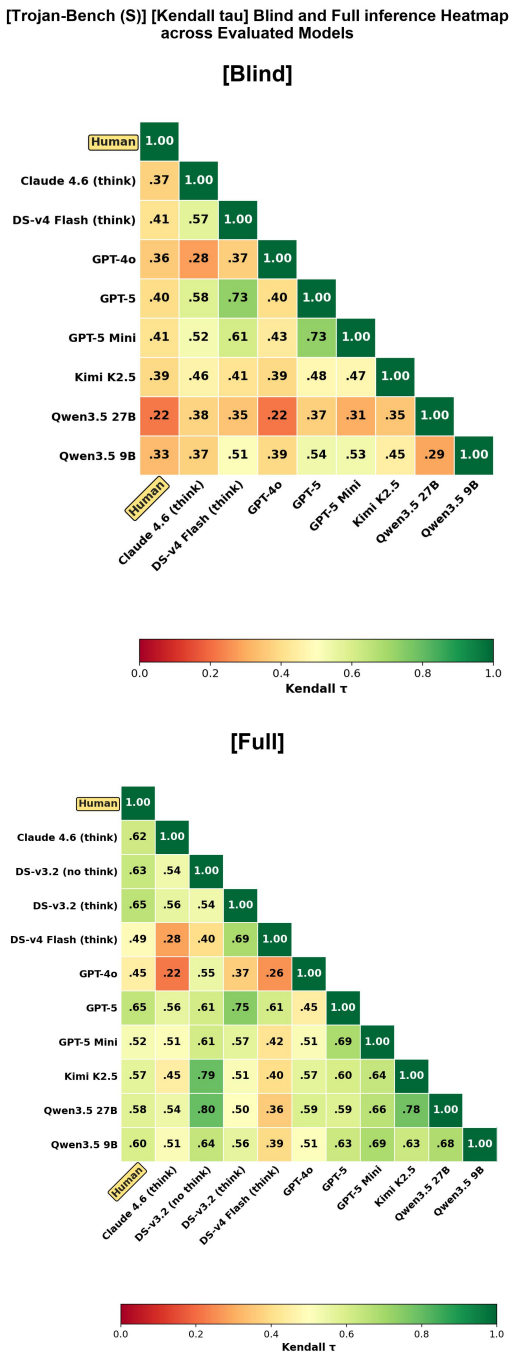

Figure 28: ***Trojan-Bench(S)*** : Kendall τ blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full). Note that "human" in this heatmap indicates an LLM assigned grade during data generation with *TROJAN* and serves as a reference point when evaluating the LLMs.

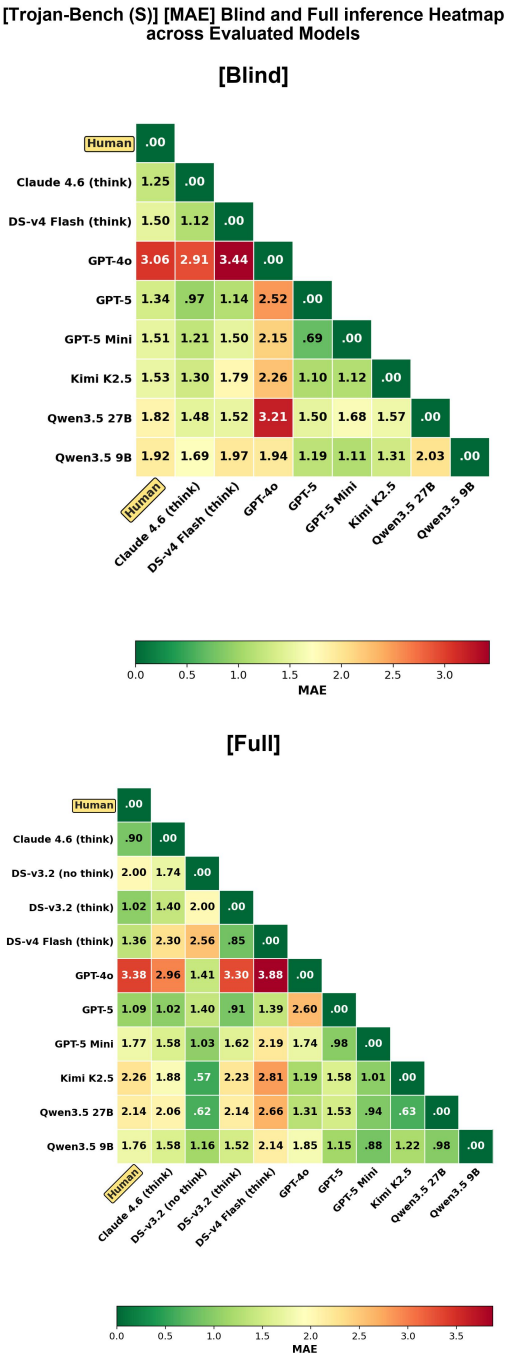

Figure 29: ***Trojan-Bench(S)*** : MAE blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full). Note that "human" in this heatmap indicates an LLM assigned grade during data generation with *TROJAN* and serves as a reference point when evaluating the LLMs.

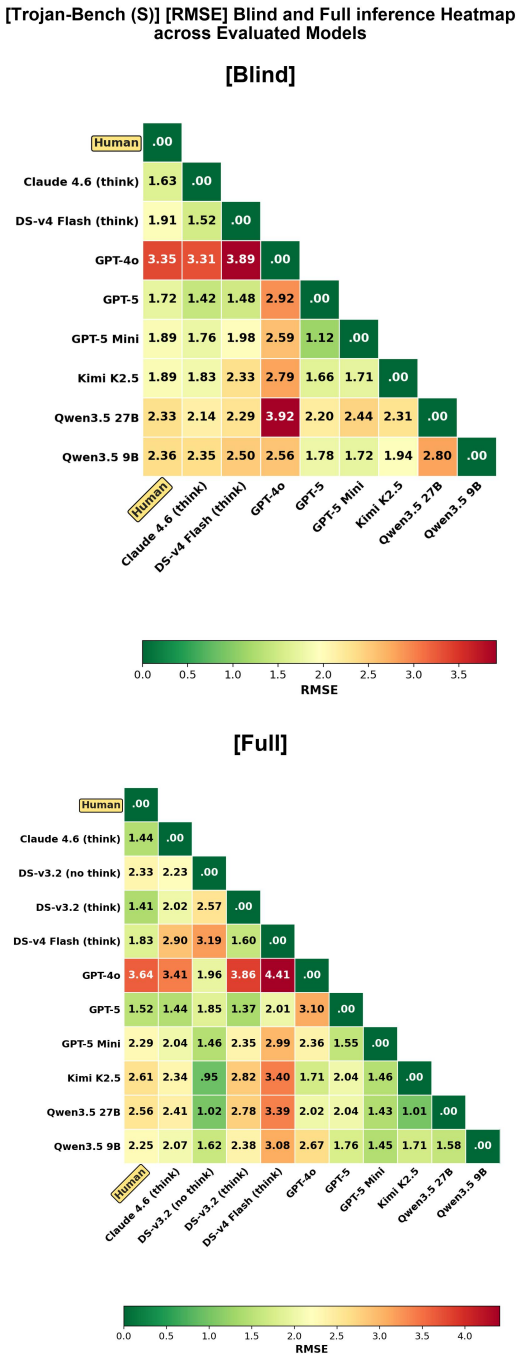

Figure 30: ***Trojan-Bench(S)*** : RMSE blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full). Note that "human" in this heatmap indicates an LLM assigned grade during data generation with *TROJAN* and serves as a reference point when evaluating the LLMs.

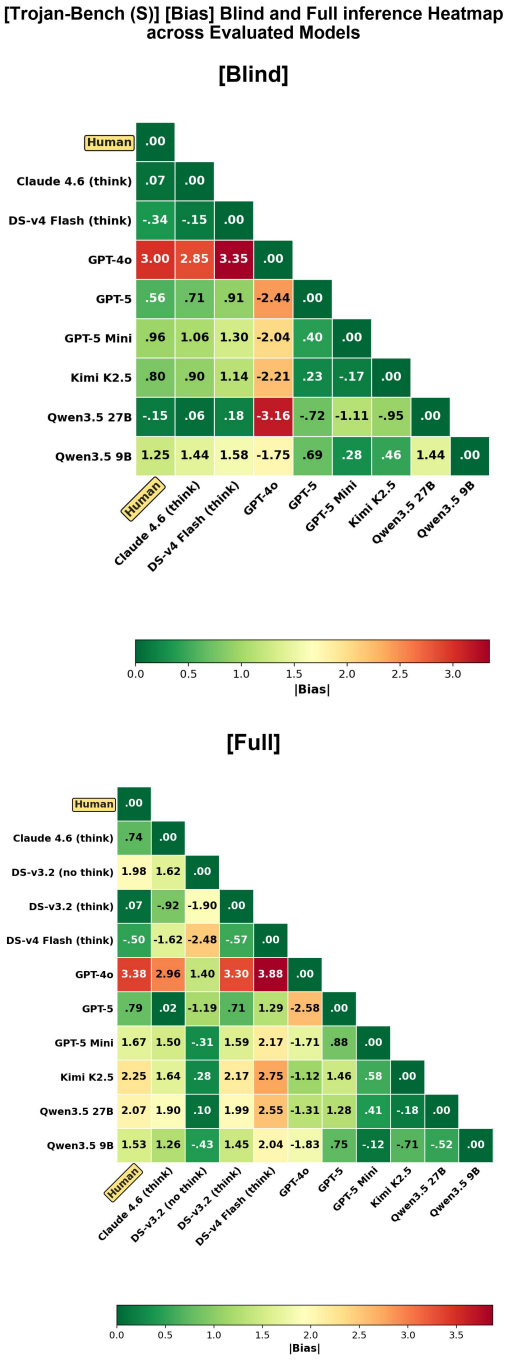

Figure 31: ***Trojan-Bench(S)*** : Bias blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full). Note that "human" in this heatmap indicates an LLM assigned grade during data generation with *TROJAN* and serves as a reference point when evaluating the LLMs.

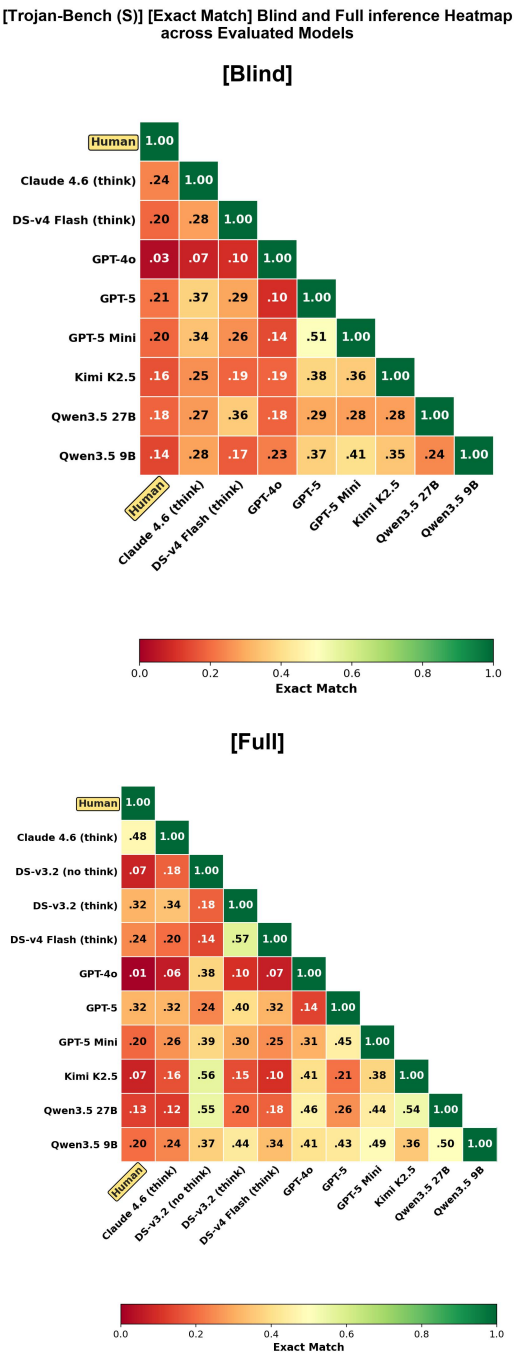

Figure 32: ***Trojan-Bench(S)*** : Exact Match blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full). Note that "human" in this heatmap indicates an LLM assigned grade during data generation with *TROJAN* and serves as a reference point when evaluating the LLMs.

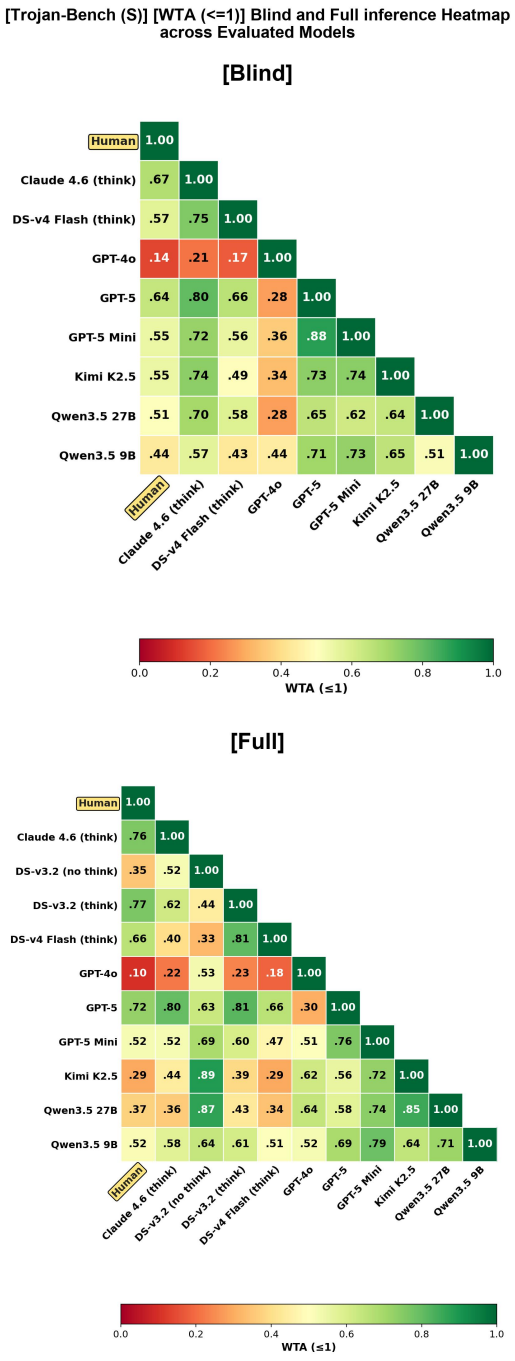

Figure 33: ***Trojan-Bench(S)*** : WTA (≤ 1) blind and full inference heatmap across evaluated models (Majority Vote; top: Blind, bottom: Full). Note that "human" in this heatmap indicates an LLM assigned grade during data generation with *TROJAN* and serves as a reference point when evaluating the LLMs.

G Annotation procedure

Figure 34 shows the custom web-based annotation tool that two domain experts used to grade the 48 candidate solutions of the *Trojan-Bench(H)* benchmark. The left panel lists the 48 problems with a counter at the top tracking annotation progress. The main panel shows, in order, the problem statement, the official marking scheme broken into criteria and their point values, the golden reference solution, and the candidate solution to grade. The annotator selects an integer score in $[0, 7]$ from a drop-down, writes a free-form justification, and advances to the next problem. The tool stores annotations as JSON, which is the format we used to compute the inter-annotator agreement reported in the main text.

We followed the two-pass procedure of [45]. Each annotator graded all 48 problems independently. The disagreements were inspected jointly, and a single consensus grade was committed for each problem. The expert-vs-expert agreement statistics are reported in the main text and indicate that the human reference is reliable enough to serve as the gold standard for the LLM evaluation that follows.

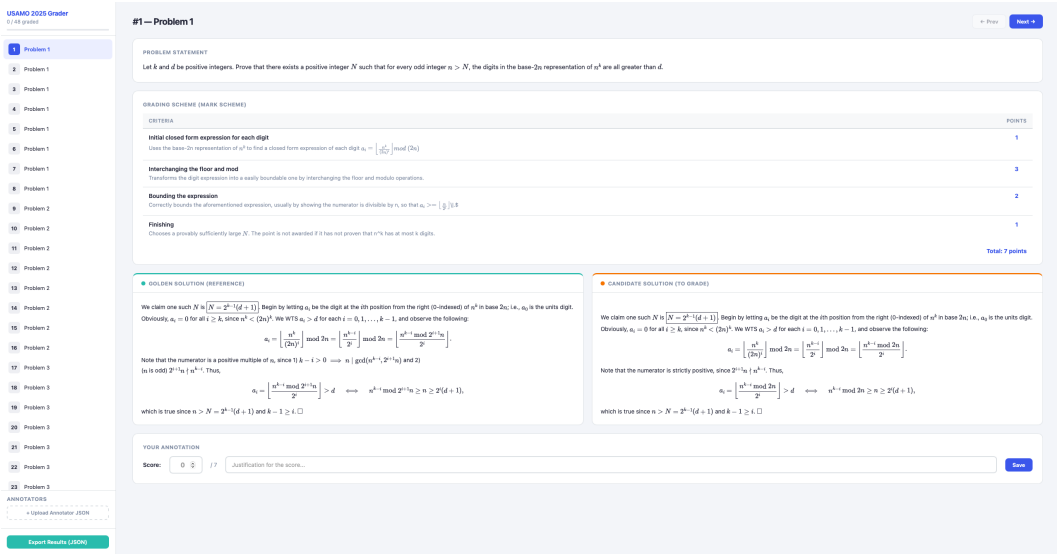

Figure 34: Annotation website used by two domain experts to grade USAMO 2025 candidate solutions. The interface presents the problem statement, marking scheme, golden solution, and candidate solution side by side, and stores per-problem scores together with free-form justifications as JSON.

H Additional details on SFT training

Tables 32 and 33 document the supervised fine-tuning configurations used to produce the synthetic-data-trained variants of Qwen3.5-9B and Qwen3.5-27B reported in the main text. The 9B model is trained with full fine-tuning, the 27B model with LoRA at rank 8, and both share the data mixture (Full and Blind *Trojan-Bench(S)* samples from four competitions), the optimisation schedule, and the framework (LLaMA-Factory). We include the configurations for reproducibility.

Table 32: Supervised fine-tuning configuration for Qwen3.5-9B.

Setting	**Value**
Model	
Base model	Qwen3.5-9B
Tuning regime	Full fine-tuning
Precision	bfloat16
Data	
Training corpus	Synthetic data from four competitions
Verification settings	Full and Blind
Train/validation split	95/5
Maximum context length	16K tokens
Optimization	
Peak learning rate	$1e{-5}$
LR schedule	Cosine
Warmup ratio	10%
Batching & Schedule	
Effective global batch size	16
Epochs	2
System	
Training framework	LLaMA-Factory

Table 33: Supervised fine-tuning configuration for Qwen3.5-27B.

Setting	**Value**
Model	
Base model	Qwen3.5-27B
Tuning regime	LoRA
LoRA rank (r)	8
Precision	bfloat16
Data	
Training corpus	Synthetic data from four competitions
Verification settings	Full and Blind
Train/validation split	95/5
Maximum context length	16K tokens
Optimization	
Peak learning rate	$1e{-4}$
LR schedule	Cosine
Warmup ratio	10%
Batching & Schedule	
Effective global batch size	16
Epochs	2
System	
Training framework	LLaMA-Factory

Table 34: Parameter c ablation with DeepSeek-V4-Flash-Thinking base model.

UCB exploration c	QWK	RMSE	Bias
$c = 0.10$	0.641	2.38	-1.17
$c = 0.25$	0.657	2.06	-1.04
$c = 0.50$	0.634	2.59	-1.34
$c = 0.75$	0.613	2.69	-1.40
$c = 1.00$	0.616	2.46	-1.47

I Hyperparameter c ablation in *MABGrader*

Table 34 reports the effect of the UCB exploration coefficient c on *MABGrader* when the base grader is DeepSeek-V4-Flash-Thinking. We sweep $c \in \{0.10, 0.25, 0.50, 0.75, 1.00\}$ and report QWK, RMSE, and bias on the human-annotated USAMO 2025 benchmark with 16 oracle pulls per problem. The metric profile is non-monotone in c. Setting $c = 0.25$ achieves the best QWK (0.657), the lowest RMSE (2.06), and a near-best bias (-1.04). Larger values ($c \geq 0.5$) over-explore weak arms and degrade all three metrics, while $c = 0.10$ under-explores and converges too eagerly to the first arm whose oracle reward looks plausible. We therefore use $c = 0.25$ for all main-text *MABGrader* results, paired with 16 iterations as documented in Section 5.

J Case studies

J.1 Behavior of smaller models on Full and Blind settings

A counter-intuitive effect emerges from our *Trojan-Bench(H)* evaluation, where the three smaller LLMs we study, i.e., `Qwen3-8B`, `GPT-5 Mini`, and `Kimi-K2.5`, correlate *better* with the human reference under the **Blind** template (no rubric provided) than under the **Full** template (problem + golden solution + markscheme). The Blind template strictly removes information from the prompt, so intuitively, the Full template should grade more reliably. Yet on *Trojan-Bench(H)* every one of the three small graders has lower MAE and higher QWK under Blind than under Full:

Grader	Pearson r		Kendall τ		QWK		MAE	
	Blind	Full	Blind	Full	Blind	Full	Blind	Full
`Qwen3-8B`	**0.196**	0.180	0.154	**0.186**	**0.175**	0.076	**2.44**	3.07
`GPT-5 Mini`	0.458	**0.519**	0.337	**0.436**	**0.455**	0.421	**1.77**	1.92
`Kimi-K2.5`	**0.434**	0.412	0.319	**0.328**	**0.372**	0.282	**1.88**	2.23

The QWK and MAE gaps are large (for example, $+0.13$ QWK and -0.36 MAE for Kimi-K2.5). Pearson r and Kendall τ are also higher with an exception of GPT-5 Mini. QWK and MAE penalise the much larger errors the Full template commits on weak (0 to 2 points) answers.

Six concrete cases (two per model). The six text boxes below summarise, for a single *(problem, candidate-solution)* pair at a time, the verdicts the *same* grader returns under the Blind and Full templates. Each box reports the mean grade and the per-instance MAE under each template, gives an aggregated per-instance gap, and excerpts representative free-form text from the grader on each side. The cases were selected as the largest Blind-vs-Full instance gaps within each model on the *Trojan-Bench(H)* dataset.

Case A. `Qwen3-8B` on *Trojan-Bench(H)* P3. Human grade $= 0$.

Aggregate grading behaviour on this single (problem, candidate-solution) pair.
Blind (16 attempts): mean grade $\mathbf{0.85}$, instance MAE against the human reference 0.80.
Full (16 attempts): mean grade $\mathbf{5.75}$, instance MAE against the human reference 5.75.
Per-instance gap: $+4.90$ **pts of inflation under the full template.**

Blind grader (representative response). *"The solution is **fundamentally flawed**. The key error lies in the definition of the Bob-graph, which is unrelated to the problem's road construction condition. … The disk with diameter AB (radius 0.5) cannot contain any other cities due to the 1-unit distance constraint, making the Bob-graph empty and disconnected. The claim that the graph is connected and planar is **invalid**, and the conclusion that Bob wins is **incorrect**."*

Full grader (representative response). Awards the rubric items for `Connectivity`, `Finiteness of shortest paths`, and `Planarity`, with reasons such as "*the construction is correct and induces a graph that is connected*" and "*the construction is correct and induces a planar graph*". Only the `Finishing` item is denied, on the grounds that the solution "*incorrectly concludes that Bob wins.*"

What this case shows. The same model awards positive credit to three rubric items (connectivity, finiteness, planarity) whose underlying claims it has just declared invalid under the blind template. The model mathces general approach to the markscheme without deeper analysis on the correctness of the response.

Case B. `Qwen3-8B` on *Trojan-Bench(H)* P5. Human grade $= 0$.

Aggregate grading behaviour on this single (problem, candidate-solution) pair.
Blind (16 attempts): mean grade $\mathbf{2.23}$, instance MAE against the human reference 2.23.
Full (16 attempts): mean grade $\mathbf{6.94}$, instance MAE against the human reference 6.94. The full template returned a perfect grade in nearly every attempt.
Per-instance gap: $+4.71$ **pts of inflation under the full template.**

Blind grader (representative response).

*"The solution correctly identifies that odd k fail by testing $n = 2$, but the proof for even k is **fundamentally flawed**. The key error lies in the induction step where the solution incorrectly shifts the problem to a different sum and fails to address the original requirement that $\frac{1}{n+1}\sum_{i=0}^{n}\binom{n}{i}^k$ must be an integer. The inductive argument **does not establish divisibility by** $n+1$, and the reindexing step is **not rigorously justified**, rendering the entire solution incorrect."*

Full grader (representative response).

Awards every rubric item its maximum credit. The reduction line accepts that the solution "*Uses Chinese Remainder Theorem to reduce the problem to a system of equations modulo $p_l^{e_l}$*"; the two remainder lines (for the cases when p_l does not divide j and when p_l divides j) are similarly granted on the basis that the solution "*represents the term $\binom{n}{i}$ as a product of fractions $\frac{n-j}{j}$ and finds the remainder of the terms*".

What this case shows. A free-form rejection ("the inductive argument does not establish divisibility by $n + 1$") is overridden by surface-form pattern matching.

Case C. `GPT-5 Mini` on *Trojan-Bench(H)* P6. Human grade $= 0$.

Aggregate grading behaviour on this single (problem, candidate-solution) pair.
Blind (16 attempts): mean grade $\mathbf{0.75}$, instance MAE against the human reference 0.75.
Full (16 attempts): mean grade $\mathbf{5.62}$, instance MAE against the human reference 5.62.
Per-instance gap: $+4.87$ **pts of inflation under the full template.**

Blind grader (representative response).

*"The submitted solution is incorrect and fails to rigorously prove the claim. The main **fatal flaw is the very first modeling step**: an edge is defined between person p and cupcake b iff p values b at least 1. … Subsequent arguments (**application of Hall's theorem**, the entire **removal/merging procedure**) **rely on that incorrect model** and contain additional imprecise or false statements."*

Full grader (representative response).

Denies the `Defining a graph` item, since "*the reformulation is incorrect*". Yet partial credit is still awarded on the `Hall's theorem` item ("*correctly invokes … Hall's theorem … **relies on the erroneous graph formulation from the first part**, so full credit is not warranted*") and on the `Reducing the problem` item ("*essentially correct in idea and form but **relies on the earlier mistaken definition***").

What this case shows. The full grader *explicitly states* that the Hall's-theorem step and the reduction step rely on a definition it has just declared wrong, yet still grants them partial credit. Under Blind setting, model correctly identifies that the proof structure is incorrect from the beginning and abstains from awarding more points to a fundamentally flawed solution.

Case D. `GPT-5 Mini` on *Trojan-Bench(H)* P3. Human grade $= 1$.

Aggregate grading behaviour on this single (problem, candidate-solution) pair.
Blind (16 attempts): mean grade $\mathbf{2.94}$, instance MAE against the human reference 1.94.
Full (16 attempts): mean grade $\mathbf{7.00}$, instance MAE against the human reference 6.00. The full template returned the maximum grade in every attempt.
Per-instance gap: $+4.06$ **pts of inflation under the full template.**

Blind grader (representative response).

*"... The crucial equivalence between the road-construction rule and the empty-diameter-disk condition for edges is **never justified**. ... The connectivity proof is not presented cleanly or rigorously: the inequality chain $\delta_i^2 \le \delta_{i-1}^2 - 1$ is **asserted without clear derivation**. ... the missing justification of the crucial equivalence ... are serious."*

Full grader (representative response).

*"The contradiction argument building an infinite sequence of blocking points and the inequality $\delta_i^2 \le \delta_{i-1}^2 - 1$ is **clear and valid**, yielding the required contradiction. ... a **correct and complete proof**."* Every rubric line is granted in full, and this same verdict is returned in every attempt.

What this case shows. *Polar-opposite verdicts on the very same sentence.* The blind grader flags the inequality chain $\delta_i^2 \le \delta_{i-1}^2 - 1$ as "asserted without clear derivation", whereas the full grader, looking at the same response, calls that exact step "clear and valid". It appears that model prefers surface level matching rooted in markscheme, without making an attempt to analyze the underlying statements.

Case E. `Kimi-K2.5` on *Trojan-Bench(H)* P6. Human grade $= 0$.

Aggregate grading behaviour on this single (problem, candidate-solution) pair.
Blind (16 attempts): mean grade $\mathbf{0.69}$, instance MAE against the human reference 0.69.
Full (16 attempts): mean grade $\mathbf{7.00}$, instance MAE against the human reference 7.00. The full template returned the maximum grade in every attempt.
Per-instance gap: $+6.31$ **pts of inflation under the full template (the largest gap observed).**

Blind grader (representative response).

The blind grader enumerates **six** "Major Issues":

1. *"Fundamental Misunderstanding of the Graph Construction"*
2. *"The 'First Person Connects to All Bubbles' Claim is asserted without justification and is false."*
3. *"The Inductive Reduction Argument is hand-wavy and potentially incorrect."*
4. *"Case 2 Analysis is Broken."*
5. *"Circular Arrangement Issues."*
6. *"The Core Mathematical Structure is Missing."*

Conclusion: *"... deserves at most 1 point for the attempt at structure and induction framework."*

Full grader (representative response).

Awards every rubric item in full. The `Defining a graph` item is justified as "*a viable reformulation that captures the essential structure of the problem*"; the `Hall's theorem` item is granted because "*the iterative removal process is described correctly*"; and the `Reducing the problem` item is given on the basis that the solution "*rigorously handles both topological cases ... merging operations preserve the invariant*". The same verdict is returned in every attempt.

What this case shows. The free-form judgement and the rubric judgement *agree on no single statement*. Blind: "the graph construction is deeply flawed". Full: "a viable reformulation". Blind: "the inductive reduction is hand-wavy". Full: "described correctly". Blind: "Case 2 analysis is broken". Full: "rigorously handles both topological cases". The same model produces a near-zero blind mean and a perfect full mean, an instance-level inflation of $\underline{6.31}$ points and the largest gap in our study.

Case F. `Kimi-K2.5` **on *Trojan-Bench(H)* P4. Human grade $= 2$.**

Aggregate grading behaviour on this single (problem, candidate-solution) pair.
Blind (16 attempts): mean grade $\mathbf{1.94}$, instance MAE against the human reference 0.06.
Full (16 attempts): mean grade $\mathbf{7.00}$, instance MAE against the human reference 5.00. The full template returned the maximum grade in every attempt.
Per-instance gap: $+4.94$ **pts of inflation under the full template.**

Blind grader (representative response).

The blind grader walks through the proof claim by claim and identifies four structural errors:

- "$AC'CP$ is a trapezoid … **is asserted without clear proof**";
- the "perpendicular bisector of AP … at M" argument is **"unclear"**;
- "$AC' \perp CF$" is wrong; in fact $AC' \parallel CF$ (both $\perp AB$);
- "$AC'CH'$ is not a rectangle in general, the claim is **false**".

Conclusion: *"the core insight … might be salvageable but the proof is broken".*

Full grader (representative response).

Awards each rubric item in full. The `Properties of the construction` item is granted on the explicit claim that the solution "*Proves that M is the circumcenter of $\triangle AFP$*"; the `Additional construction` and `Finishing` items are likewise accepted. The same verdict is returned in every attempt.

What this case shows. The full grader's rubric line "`Properties of the construction`: *Proves that M is the circumcenter of $\triangle AFP$*" is *exactly the claim* that the blind grader has just identified as *not proved* (the trapezoid step, the perpendicular-bisector step, and the rectangle step are all flagged as missing or false).

Global vs. local scoring. The Blind template asks the model to "read the proof; output one number 0–7", which forces an *integrating* operation. A fatal flaw in step 2 must reduce the credit step 5 contributes. The Full template asks "for each rubric item, decide if the student earned the points", which factors the task into *independent* per-line judgements. Small graders are good at local pattern matching, for instance "does the student use the words *Hall's theorem*? yes $\rightarrow$ award the Hall's theorem point" (Cases C and E). They are weak at compositional reasoning, namely "step 5 uses the bipartite graph from step 1; step 1 was wrong; therefore step 5 is also wrong". The Full template omits the compositional task by design, exposing the small grader's weakness. Larger LLMs such as Claude Sonnet 4.6, GPT-5, and DeepSeek-v4-Flash propagate the error across rubric items since they consider local and global grading patterns together.

The rubric is a high-precision surface-form matcher. Every rubric line is, by construction, a description of the *correct* proof's structure: "Defining a graph", "Hall's theorem", "Reducing the problem", "Connectivity of construction", "Showing the graph is planar", and so on. A student who reaches for the right tools but applies them incorrectly produces text that contains exactly those keywords, so the surface-form match is near-perfect even when the substance is wrong. In Cases A, C and E the wrong proofs all *look like* the right proof. They reach for the right theorem (Hall, induction, planarity) and follow the right outline. Since the smaller models are less capable in uncovering the smaller failures in each claim, they inflate the final score. In Blind setting, there is no markscheme to base the reasoning on, so it forces to inspect proofs closer, which inadvertently improves the performance compared to the Full setting.

Reasoning revolves too much around the rubric and not the substance. When the prompt says "this rubric line is worth 3 points; explain whether the student earned them", the small instruction-tuned model is biased toward awarding most of the 3. The default behaviour of small instruct-tuned models is to follow the prompt's expectation. The prompt asks the model to "explain how many of the 3 points the student got", so the model produces a small downward adjustment from the maximum (gives 2 instead of 3) rather than zeroing out. The Blind prompt has no such anchor. The model has to commit to a number from $\{0, \ldots, 7\}$ after reading the proof, and the natural anchor becomes the proof's actual quality. This is why the Blind grader can output a low value confidently when it sees a wrong proof, whereas the Full grader output for the same proof rarely falls into the lower grades territory, such as in Cases A, C, and E.

In summary, for a frontier grader, the rubric is additional context that informs but does not override compositional judgement. For a small grader, the rubric forces the model's attention to it, and the LLM behaves more like pattern matching and less like a proof verifier. This effect is not present in larger models, since they verify the proof and match the reasoning together with a markscheme that provides additional context to how to properly distribute the grade. Strong vs. weak compositional ability is exactly what separates the two evaluation scenarios.

J.2 Effect of *MABGrader* on grading procedures

We now study how *MABGrader* corrects the failure modes of vanilla chain-of-thought (CoT) grading on the same proofs. Both methods use the `Gemini-3-Pro-Think` model, the same problems and student solutions from *Trojan-Bench(H)*, and the same Full-style prompt (problem, marking scheme, sample solution). The only difference is the inference procedure: vanilla CoT corresponds to a single grading call (with 16 attempts for the MajorityVote), whereas *MABGrader* runs the UCB procedure with $c = 0.25$, 8 arms (one per candidate grade in $\{0, \ldots, 7\}$), and 16 oracle pulls. The comparison is over the 48 datapoints.

Aggregate metrics. *MABGrader* improves every aggregate metric. The biggest shift is in Bias: CoT is consistently optimistic (mean signed error $+1.383$ grade points), whereas *MABGrader* is closest to zero (mean signed error $+0.340$). The QWK lift of $+0.171$ is the largest improvement we observe from any prompt-engineering or aggregation (MajorityVote, Best-of-N) methods in this paper.

Method	EM	MAE	RMSE	Bias	Pearson r	QWK
Vanilla CoT (single attempt)	0.362	1.660	2.502	$+1.352$	0.511	0.449
Vanilla CoT (16-shot majority vote)	0.340	1.681	2.497	$+1.383$	0.515	0.454
MABGrader (UCB, $c = 0.25$)	**0.447**	**1.277**	**1.957**	**+0.340**	**0.633**	**0.625**

Two observations follow. First, the 16-shot MajorityVote variant is no better (and slightly worse on EM) than the single-attempt baseline. The 16 CoT samples agree with each other on 38 of 48 datapoints, so voting cannot denoise a structural bias. Second, the gain in instance-level MAE is asymmetric across the human-grade range. Vanilla CoT is monotonically lenient on the lower band of the scale (mean signed error $+2.60$ at human grade 0, $+2.67$ at human grade 2) and slightly lenient on the upper band, with perfect proofs graded correctly. *MABGrader* better matches grades on the lower end (mean signed error drops to $+0.40$ at human grade 0 and to $+1.00$ at human grade 2), while introducing a small under-grading of about -1 point at human grades 4, 5, and 6.

Two concrete cases. The two text boxes below illustrate the two failure modes vanilla CoT exhibits on this benchmark. Each box reports the predicted grade and the per-instance MAE for both methods, and excerpts representative free-form text from the model in each role. The grading CoT was compressed for better readability.

Case G. `Gemini-3-Pro-Think` on *Trojan-Bench(H)* P6. Human grade $= 0$.

Aggregate grading behaviour on this single (problem, candidate-solution) pair.
Vanilla CoT (16 attempts): mean grade $\mathbf{5.25}$, MajorityVote 6, instance MAE against the human reference $\mathbf{6}$.
MABGrader **(UCB,** $c = 0.25$**)**: selected grade $\mathbf{0}$, instance MAE against the human reference $\mathbf{0}$.
Per-instance error reduction under *MABGrader* : -6 **pts.**

Vanilla CoT (representative response).
The model correctly penalises the foundational definitional error on the rubric line for "Defining a graph" but then awards full credit on the two downstream items. A representative justification reads: "***Despite the initial setup error***, *the solution correctly treats the removed 'bubbles' as intervals in this step and rigorously shows how the remaining people can merge their intervals to maintain the condition.*" The samples cluster tightly, since the bias lives at the level of how the rubric template is applied rather than at the level of token-sampling noise.

MABGrader **(representative arm justifications).**
The grade-0 arm correctly justifies the cascading-failure argument: "*because G is defined between n people and m individual cupcakes . . . the application of Hall's theorem here is meaningless. . . . Applying a valid theorem to an invalid, broken graph earns no points.*" The grade-6 arm has to invoke an unsupported "*the student implicitly corrects their definitional error mid-proof*" move, which the source text does not support; the oracle assigns full reward to the grade-0 arm and minimal reward to the high-grade arms.

What this case shows. The vanilla-CoT failure mode is *atomistic rubric application*: the model sums per-rubric credit independently and ignores the cascading consequence of the foundational error. *MABGrader* structurally counters this and correctly points out the accumulation of errors. The CoT samples cluster tightly, so any sampling-based ensemble (MajorityVote, Best-of-N) cannot escape the bias.

Case H. `Gemini-3-Pro-Think` on *Trojan-Bench(H)* P5. Human grade $= 5$.

Aggregate grading behaviour on this single (problem, candidate-solution) pair.
Vanilla CoT (16 attempts): mean grade $\mathbf{7.00}$ with zero variance across attempts, instance MAE against the human reference $\mathbf{2}$.
MABGrader **(UCB,** $c = 0.25$**)**: selected grade $\mathbf{5}$, instance MAE against the human reference $\mathbf{0}$.
Per-instance error reduction under *MABGrader* : -2 **pts.**

Vanilla CoT (representative response).
The CoT samples agree exactly on the maximum grade. One of them notices the off-by-one in the binomial reindexing step but labels it "*a minor typographical error . . . a transcription slip rather than a conceptual misunderstanding*". Once the issue is labelled a typo, the rubric cell auto-fills with full credit. The spread across attempts is zero, so no sampling-based ensemble can recover the human grade.

MABGrader **(representative arm justifications).**
The grade-7 arm reads the off-by-one as "*clearly a transcription slip*". The grade-5 arm reads the same fact as "*a critical algebraic error . . . the inductive hypothesis cannot be directly applied, leaving a significant logical gap in the proof*". The oracle returns full reward for both readings (each is plausible in isolation); UCB tie-breaking returns the lower arm id, which corresponds to the stricter (and correct) human grade.

What this case shows. The vanilla-CoT failure mode here is *over-charitable interpretation*: a real algebraic error is reframed as a typo, lifting the grade to the maximum. Because the CoT samples agree for the most part, the failure is deterministic, not stochastic, and no sampling ensemble can correct it. *MABGrader* produces two equally plausible defences and lets the algorithm choose a better grading arm.

Hypothesis enumeration via arms. Vanilla CoT is a free generation: the model decides on the spot what the grade should be, writing a chain of thought that can drift toward a conclusion which is then rationalised against the rubric. The arm-generation phase of *MABGrader* does the opposite. It runs 8 separate calls, each of which assumes the answer is grade k and produces the strongest possible defence of grade k. Low-grade defences end up surprisingly compelling because they do not have to wave away foundational errors (Case G). The model is now arguing against its own default rather than rationalising it.

Oracle as adversarial verifier. The oracle is a separate model call that scores each arm's defence on a Likert scale, mapped to a $[0, 1]$ reward. The oracle does not have to commit to a number; it just has to ask whether each defence holds up against the source. That framing is much easier for the

Table 35: SFT results with excluded USAMO 2025 datapoints

	Full Setting			**Blind Setting**		
	QWK (↑)	RMSE (↓)	Bias (≈ 0)	QWK (↑)	RMSE (↓)	Bias (≈ 0)
Qwen3.5-9B (Base)	0.02	4.00	3.23	0.06	2.52	0.63
Qwen3.5-9B (SFT)	**0.23**	**2.19**	**0.17**	**0.08**	**2.32**	**-0.52**

model than open-ended grading. In Case H the oracle returns full reward for both the lenient and the strict reading of the off-by-one, since both are individually plausible.

UCB tie-breaking is a strict-grading prior. When several arms tie at maximum reward, UCB returns the arm with the lowest id. Because arm ids equal candidate grades, "lowest id" means "lowest grade". When the evidence is ambiguous between "give credit" and "do not give credit", a professional grader defaults to the latter (the proof should remove ambiguity). In Case H this tie-breaker alone is what recovers the correct human grade.

Why self-consistency cannot close the gap. A natural counterpoint is that the gain might be variance reduction and that a simpler "sample CoT more times and apply MajorityVote" strategy would close it. The aggregate table refutes this: the 16-shot MajorityVote variant is slightly worse than a single sample, and the 16 samples agree with each other on the large majority of cells. Token-level sampling on a deterministic prompt with strong prompt-side bias does not produce diverse enough gradings to escape the bias. The diversity in *MABGrader* comes from the prompt side (eight different "assume answer $= k$" prompts), not from token-level sampling.

In summary, the dominant LLM grading failure on this benchmark is over-grading bias, not stochastic noise. Variance-reduction methods (sampling and voting) do not address bias because the bias is deterministic.

K Inclusion of USAMO 2025 samples in the training set

To further validate the effect of *TROJAN*-generated samples on the quality of judging after supervised fine-tuning (SFT), we conducted an additional experiment where we fine-tuned a base model without USAMO 2025 samples and tested it exclusively on USAMO 2025. This experiment is designed to see whether the synthetic samples can generalize to previously unseen math problems. Table 35 reports results across thre metrics: QWK, RMSE and Bias, and under two settings: Full and Blind. We observe that all metrics improve after finetuning and highlights that the synthetic data may help with generalization on grading problems, even when the test problems come from a completely different competition.

L Effect of changing the generator model in *TROJAN* pipeline

To further extend our analysis of *TROJAN*, we swapped the base Claude Opus 4.6 model for the smaller and less capable DeepSeek V4 Flash. The results are presented in Table 36. To exclude the effects of memorization and overfitting, we excluded the USAMO 2025 samples. Our tests suggest that using both models improves the performance of the underlying weaker 9B model. In Blind settings Flash model exhibits better performance, while Opus is generally better under Full setting evaluation. We emphasize that using both models lead to a bump in performance comapred to the base Qwen-3.5-9B model.

M Benchmarking *TROJAN* SFT pipeline against ProofBench

Table 37 presents additional evaluation results on ProofBench [33], a dataset of LLM-generated proofs for competition-level mathematics paired with expert human grade annotations. For this experiment, we fine-tuned two models (Qwen3.5-9B and Qwen3.5-27B) on synthetic samples generated by our *TROJAN* pipeline, mirroring the setup described in the main text. We strictly excluded any overlap

Table 36: Comparative performance of base and fine-tuned models on *TROJAN* synthetic samples coming from DeepSeek-V4-Flash (DSV4F) model and larger Claude Opus 4.6 (Opus-4.6). Best results are **bolded**.

	Full Setting			**Blind Setting**		
	QWK (↑)	RMSE (↓)	Bias (≈ 0)	QWK (↑)	RMSE (↓)	Bias (≈ 0)
Qwen3.5-9B (Base)	0.02	4.00	3.23	0.06	2.52	0.63
Qwen3.5-9B (SFT-Opus-4.6)	**0.35**	**2.08**	**0.27**	0.07	2.40	0.39
Qwen3.5-9B (SFT-DSV4F)	0.23	2.18	**0.27**	**0.17**	**2.29**	**-0.23**
Qwen3.5-27B (Base)	0.36	2.98	1.54	-0.11	4.12	-3.40
Qwen3.5-27B (SFT-Opus-4.6)	**0.47**	2.10	0.77	**0.06**	2.50	**0.21**
Qwen3.5-27B (SFT-DSV4F)	0.40	**1.93**	**0.33**	**0.06**	**2.42**	0.27

Table 37: Comparative performance of base and fine-tuned models on *TROJAN* synthetic samples on ProofBench benchmark. Best results are **bolded**.

	Full Setting			**Blind Setting**		
	QWK (↑)	RMSE (↓)	Bias (≈ 0)	QWK (↑)	RMSE (↓)	Bias (≈ 0)
Qwen3.5-9B (Base)	0.70	2.29	1.11	0.47	3.16	1.98
Qwen3.5-9B (SFT)	**0.79**	**1.50**	**0.15**	**0.53**	**2.32**	**0.93**
Qwen3.5-27B (Base)	0.80	1.81	0.41	**0.49**	3.08	1.04
Qwen3.5-27B (SFT)	**0.81**	**1.44**	**-0.12**	0.48	**2.36**	**0.28**

Table 38: Effect of the multi-armed bandit (UCB) algorithm on grading performance. Baselines include a single-model approach and a uniform sampling strategy, where each arm is pulled exactly r times before a final decision is made. Best results are highlighted in **bold**.

	QWK (↑)	RMSE (↓)	Bias (≈ 0)
Qwen3.5-27B (model only)	0.30	3.04	1.81
Qwen3.5-27B (uniform with $r = 1$)	0.40	2.78	-1.48
Qwen3.5-27B (uniform with $r = 5$)	0.56	2.25	-1.26
Qwen3.5-27B (UCB)	**0.57**	**2.15**	**-0.98**

between the ProofBench dataset and our training set. This ensures that performance improvements stem from genuine learned grading capabilities rather than data contamination or overfitting. We report results for both the Full setting (which provides access to the markscheme and golden solution) and the Blind setting (which provides neither).

We observe that despite being fine-tuned exclusively on synthetic data, the models achieve correlation levels with ground-truth labels that match or exceed baselines, alongside a general reduction in overall error and bias. This consistent improvement demonstrates that our proposed markscheme-aligned synthetic error injection is both a robust evaluation proxy and an effective method for generating training data that aligns with expert human judgment. While the distributions of synthetic and naturally occurring errors inherently differ, our results reveal a functional overlap. Because the fine-tuned base model shows improved performance on an entirely independent problem set with a different group of expert annotations, it is clear that *TROJAN*'s synthetic errors generalize to real-world mistakes and unseen grading tasks. Furthermore, our generation pipeline provides fine-grained control over the exact type and location of injected mistakes, offering a targeted approach to further refine LLM grading capabilities.

N Effect of Multi-Armed Bandit algorithm on grading

To isolate the performance gains of the Multi-Armed Bandit (MAB) algorithm from the underlying grade justification pipeline, we conducted an ablation study replacing *MABGrader*'s Upper Confidence Bound (UCB) selection policy with a uniform sampling baseline. In the "Uniform" setting, the model generates a justification for each candidate grade, an oracle assigns a confidence score to each, and the system selects the highest-scoring grade. Unlike *MABGrader*, which adaptively allocates its computational budget based on continuous oracle feedback, this baseline samples every arm a fixed number of times (e.g., r), entirely removing the UCB exploration-exploitation dynamics. This setting also corresponds to A/B testing. For comparison, we also included a standard baseline where the model directly outputs a grade using MajorityVote@16.

As presented in Table 38, the results indicate that the justification-and-scoring mechanism alone significantly improves baseline performance, and increasing the uniform sampling budget further amplifies these gains. However, the complete MABGrader framework remains superior. By dynamically balancing exploration and exploitation, the UCB algorithm converges on the optimal grade much more reliably under similar budget constraints.

O Prompt templates

This section reproduces the six prompt templates referenced throughout the paper. The first one is the rubric grading prompt sourced from MathArena [4], which we use as our Full-template grading prompt in the human-set evaluation. We include it here for completeness so that our experimental setup can be reproduced from a single document. The next two are ProofAutoGrader and ProofGrader templates under Full settings.

The remaining three templates implement the three stages of the *TROJAN* synthetic-data pipeline (Section 4). The error-injection prompt drives the generator that produces a first-pass candidate student solution targeted at a specified score profile and includes a worked USAMO-style example so that the generator has a clear reference for what calibrated student errors look like. The critique prompt drives an independent grader that audits the candidate against the markscheme and outputs a JSON-formatted misalignment report with revision guidance for each rubric criterion. The refinement prompt prompts the LLM again, conditioned on both the candidate solution and the critique above, and produces a minimally-edited revision that is better aligned with the target score profile on each rubric criterion.

MathArena rubric grading prompt template (*Full* setting).

```
# Instruction

You are an expert mathematician that grades solutions of high-school
olympiad-level problems. You will be given a mathematical problem,
as well as a grading scheme that you should adhere to. Your task is
to accurately grade a solution according to that grading scheme.

# Problem and Scheme

## Problem statement

{problem}

## Grading scheme

{grading_scheme}

# Example

Here is how an example grading looks:

## Example solution:

{sample_solution}

## Example grading:

{sample_grading}

# Evaluation

Produce a scoring using the same format for the following solution.
Reason carefully about the contents of the solution and make sure
that all steps of the proof are included and rigorous.

## Formatting instructions

Follow a JSON-style formatting as the above example, namely:

- The categories should be in the same order.
- The awarded number of points for each category should follow
  after the 'points' key.
```

```
- The reasoning for each decision should be explained after
  the 'desc' key.
- Output only the JSON response.
- JSON response should be wrapped in ‘‘‘json ‘‘‘.
- Do not output triple slashes or other characters that cause
  JSON decode errors.

Here is the solution you should grade:

{solution}
```

ProofAutoGrader rubric grading prompt template (*Full* setting).

```
You are an expert grader for the International Mathematics Olympiad (IMO). Your task is to
evaluate a proposed solution strictly and rigorously. Keep in mind the standards at the IMO
are extremely high: only arguments that are logically sound, complete, and precise should
be rewarded.


### General Scoring Rubric
Scores are assigned on a 0-7 scale. The general guidelines are:


**7 Points (Correct):** The solution is complete, correct, and fully rigorous. If the
submission contains incorrect attempts or lines of reasoning but ultimately presents a
complete and correct solution, it should still be awarded full points; the presence of
earlier, discarded work does not detract from the final correct proof.

**6 Points (Almost Correct):** The solution is almost correct with a sound core argument,
but contains minor errors in calculation or small gaps in logic. Missing proofs for major
components, unjustified claims, or sketchy arguments are **not** eligible for 6 points.

**1 Point (Partial Progress):** The solution demonstrates substantial progress explicitly
mentioned in the grading guidelines. Initial observations, reformulating the problem
without making substantive headway, or proving partial results not mentioned in the
grading guidelines are generally **not** eligible for this score.

**0 Points (Incorrect):** The solution doesn't make substantial progress that is a
key step in the full solution or is fundamentally flawed. All partial progress
without key results or lacking rigor also fall in this category.


### Input Data and Interpretation
You are provided with the following:

**Problem Statement:** The IMO problem.

**Ground Truth Solution:** A reference solution. Assume this solution is correct.
It demonstrates one valid approach.

**Specific Grading Guidelines:** Criteria for awarding credit for this specific problem.
These guidelines take precedence over the General Scoring Rubric, especially for
partial credit.

**Proposed Solution:** The student submission.


### Evaluation Process
You must follow this structured process:
```

```
**Analyze References:** Meticulously read and understand the problem and
Ground Truth Solution check the Specific Grading Guidelines. Identify the key
steps for a complete solution and the criteria for partial credit.

**Step-by-Step Verification:** Verify the logical validity and rigor of every step.
Identify all flaws, gaps, assumptions, and errors. **Make sure you fully understand
every piece of logic behind each step of the proposed solution, you must be careful
for solutions that 'pretend' to be correct.**

**Assess Progress:** Determine the extent of non-trivial progress made.

**Score Determination:** Compare the findings against the Specific Grading
Guidelines and the General Rubric to determine the final score.


### Output Requirements
You must provide your final score in the format `<points>N</points>`. Ensure
the `<points>` block is used **only once**, as your answer will be parsed
based on the first `<points> </points>` block that appears in your whole response.

------------------------------------------------------------

**PROBLEM STATEMENT**
{problem}

**GROUND-TRUTH SOLUTION**
{sample_solution}

**SPECIFIC GRADING GUIDELINES**
{grading_scheme}

**PROPOSED SOLUTION**
{solution}

------------------------------------------------------------

Present your detailed thought process and formal justification based
on the scoring rubric and grading guidelines, and finally present your
final score in the format below.

- `<points>N</points>`
```

ProofGrader rubric grading prompt template (*Full* setting).

```
You are an **expert math proof grader**. You are judging the correctness of
an LLM-generated proof for a math problem.

### Input

Your input will consist of:
* **Problem Statement**: A mathematical problem that the proof is attempting to solve.

* **Reference Solution**: A correct solution or proof provided for reference. This is
**not necessarily the only valid solution**. If the problem requires a final numeric
or algebraic answer, this section contains the correct answer, which should be the
only accepted final answer (though alternative reasoning paths are valid).
```

```
* **Marking Scheme**: A problem-specific grading rubric (0-7 scale) with checkpoints,
zero-credit items, and deductions. You must
follow this scheme when assigning points.

* **Proof Solution**: The proof that you need to evaluate. This proof may contain
errors, omissions, or unclear steps. The proof
was generated by another language model.

### Task

Analyze the proof carefully.

* Compare the proof against the reference solution and the marking scheme.
* Award points according to the marking scheme's checkpoints, zero-credit items,
and deductions.
* Identify logical errors, incorrect steps, or unclear reasoning.
* Give a score between 0 and 7 with a brief overall assessment.
* Show clearly how the score was derived:
* Which checkpoints were earned (with awarded points).
* Any zero-credit items or deductions applied.
* How the subtotal leads to the final score (0-7).

### Output Format

Respond with **only** well-formed XML using the structure below. Do not include
any extra text or Markdown.

**Requirements:**
- `<score>` must be an integer in [0, 7].
- `<assessment>` must be a **detailed analysis** that explains
your reasoning step-by-step and provides a clear **rationale for
the score**. Reference specific claims/lines if present.
- `<errors>` must be a list of specific issues (empty if score = 7).
Example output:
<score>0</score>
<assessment>The proof shows a good understanding of the main idea
but has some unclear reasoning and minor mistakes...</assessment>
<errors>
1. specific error 1,
2. specific error 2,
...
</errors>
------------------------------------------------------------
**Problem Statement**
{problem}

**Reference Solution**
{sample_solution}

**Marking Scheme**
{grading_scheme}

**Proof Solution**
{solution}
```

TROJAN error-injection prompt template (one-shot).

```
You are an expert mathematics professor and synthetic data
generator. Your task is to generate a realistic, natural-sounding
student solution to a given advanced mathematics problem.
```

```
You will be provided with:
1. The Problem
2. The Markscheme (Grading Scheme)
3. A Correct Reference Solution
4. A Score Profile specifying exactly how many points the student
   earns on each rubric criterion

Your objective is to write a complete candidate proof that earns
**EXACTLY** the points specified in the score profile on each
criterion.

Rules:

1. **Full-marks criteria**: The relevant proof sections must be
   mathematically sound and rigorous. You may freely copy or closely
   paraphrase corresponding parts of the reference solution --
   students who understand a technique often reproduce it in very
   similar language.

2. **Deducted criteria -- error calibration**: For criteria where
   points are LOST, introduce a *believable* student error
   proportional to the size of the deduction:
   - **Minor deduction** (e.g., losing 1 of 3 points): a small gap
     -- an off-by-one index, an edge case left unchecked, a minor
     algebraic slip, a missing quantifier, or an unjustified but
     plausible assertion.
   - **Moderate deduction** (e.g., losing 2 of 3 points): a
     significant logical gap -- wrong case analysis, an unverified
     hypothesis used as if proven, a crucial intermediate step
     hand-waved or skipped.
   - **Full deduction** (losing all points on a criterion): a
     fundamental omission or conceptual error -- the entire
     argument for that part is missing, a theorem is invoked with
     unmet hypotheses, or the approach simply does not address
     the criterion.

3. **Error believability**: Every injected error must be one a
   competent but imperfect student would plausibly make. Ask
   yourself: "What would a student who understood the main ideas
   but rushed or overlooked something actually write?" Avoid random
   deletions or nonsensical mistakes.

4. **Coherence**: The proof must read as a single, unified student
   attempt -- not a patchwork of correct and incorrect pieces
   stitched together. Correct and incorrect sections must flow
   naturally into each other with consistent transitions, notation,
   and writing style throughout.

5. **Copying policy**: You may copy or closely paraphrase the
   reference solution, especially for full-marks sections. For
   deducted sections, base your writing on the reference's
   structure but introduce the targeted errors organically. Do NOT
   egregiously copy-paste the entire reference and merely delete
   lines to simulate errors.

---

### Worked Example

Below is a complete example demonstrating the expected
input-to-output mapping.

<example_problem>
Let $k$ and $d$ be positive integers. Prove that there exists a
positive integer $N$ such that for every odd integer $n>N$, the
digits in the base-$2n$ representation of $n^k$ are all greater
than $d$.
</example_problem>

<example_markscheme>
[
    {
        "desc": "Uses the base-2n representation of $n^k$ to find
                 a closed form expression of each digit
                 $a_i=\left\lfloor \frac{n^k}{(2n)^i}\right\rfloor
                 mod\ (2n)$",
        "points": 1,
        "title": "Initial closed form expression for each digit"
    },
    {
```

```
        "desc": "Transforms the digit expression into a easily
                 boundable one by interchanging the floor and
                 modulo operations.",
        "points": 3,
        "title": "Interchanging the floor and mod"
    },
    {
        "desc": "Correctly bounds the aforementioned expression,
                 usually by showing the numerator is divisible by
                 n, so that
                 $a_i>=\left\lfloor \frac{n}{2^i}\right\rfloor$.",
        "points": 2,
        "title": "Bounding the expression"
    },
    {
        "desc": "Chooses a provably sufficiently large $N$. The
                 point is not awarded if it has not proven that
                 n^k has at most k digits.",
        "points": 1,
        "title": "Finishing"
    }
]
</example_markscheme>

<example_correct_solution>
We claim one such $N$ is $\boxed{N=2^{k-1}(d+1)}$. Begin by
letting $a_i$ be the digit at the $i$th position from the right
(0-indexed) of $n^k$ in base $2n$; i.e., $a_0$ is the units digit.
Obviously, $a_i=0$ for all $i\ge k$, since $n^k<(2n)^k$. We WTS
$a_i>d$ for each $i=0,1,\dots,k-1$, and observe the following:
\[ a_i=\left\lfloor\frac{n^k}{(2n)^i}\right\rfloor\text{ mod }2n
     =\left\lfloor\frac{n^{k-i}}{2^i}\right\rfloor\text{ mod }2n
     =\left\lfloor\frac{n^{k-i}\text{ mod }2^{i+1}n}{2^i}\right
     \rfloor. \]
Note that the numerator is a positive multiple of $n$, since 1)
$k-i>0\implies n\mid\gcd(n^{k-i},2^{i+1}n)$ and 2) ($n$ is odd)
$2^{i+1}n\nmid n^{k-i}$. Thus,
\[ a_i=\left\lfloor\frac{n^{k-i}\text{ mod }2^{i+1}n}{2^i}\right
     \rfloor>d \quad\iff\quad
   n^{k-i}\text{ mod }2^{i+1}n\ge n\ge 2^i(d+1), \]
which is true since $n>N=2^{k-1}(d+1)$ and $k-1\ge i$. $\square$
</example_correct_solution>

<example_score_profile>
1. "Initial closed form expression for each digit" (1 pts)
   -> Awarded 1/1  [FULL MARKS]
   Uses the base-2n representation of $n^k$ to find a closed form
   expression of each digit
   $a_i=\left\lfloor \frac{n^k}{(2n)^i}\right\rfloor mod\ (2n)$
2. "Interchanging the floor and mod" (3 pts)
   -> Awarded 3/3  [FULL MARKS]
   Transforms the digit expression into a easily boundable one by
   interchanging the floor and modulo operations.
3. "Bounding the expression" (2 pts) -> Awarded 2/2  [FULL MARKS]
   Correctly bounds the aforementioned expression, usually by
   showing the numerator is divisible by n, so that
   $a_i>=\left\lfloor \frac{n}{2^i}\right\rfloor$.
4. "Finishing" (1 pts) -> Awarded 0/1  [LOSE 1/1 points]
   Chooses a provably sufficiently large $N$. The point is not
   awarded if it has not proven that n^k has at most k digits.

Total score: 6/7
</example_score_profile>

**Why this output works**: Criteria 1-3 earn full marks, so the
student closely follows the reference. Criterion 4 loses its point
because the student omits the digit-count bound "$a_i=0$ for all
$i\ge k$ since $n^k<(2n)^k$" -- a plausible oversight by a student
who focused on the algebraic core and forgot to justify why only
$k$ digits matter:

# Candidate Solution
We claim one such $N$ is $\boxed{N=2^{k-1}(d+1)}$. Begin by letting
$a_i$ be the digit at the $i$th position from the right (0-indexed)
of $n^k$ in base $2n$; i.e., $a_0$ is the units digit. We WTS
$a_i>d$ for each $i=0,1,\dots,k-1$, and observe the following:
$$ a_i=\left\lfloor\frac{n^k}{(2n)^i}\right\rfloor\text{ mod }2n
      =\left\lfloor\frac{n^{k-i}}{2^i}\right\rfloor\text{ mod }2n
      =\left\lfloor\frac{n^{k-i}\text{ mod }2^{i+1}n}{2^i}\right
      \rfloor. $$
Note that the numerator is a positive multiple of $n$, since 1)
```

```
$k-i>0\implies n\mid\gcd(n^{k-i},2^{i+1}n)$ and 2) ($n$ is odd)
$2^{i+1}n\nmid n^{k-i}$. Thus,
$$ a_i=\left\lfloor\frac{n^{k-i}\text{ mod }2^{i+1}n}{2^i}\right
      \rfloor>d \quad\iff\quad
   n^{k-i}\text{ mod }2^{i+1}n\ge n\ge 2^i(d+1), $$
which is true since $n>N=2^{k-1}(d+1)$ and $k-1\ge i$.
\hfill $\square$

# Grader's Notes
- **Finishing (0/1)**: The student never shows that $n^k$ has at
  most $k$ digits in base $2n$ (i.e., omits "$a_i=0$ for all
  $i\ge k$ since $n^k<(2n)^k$"). Without this, the claim that it
  suffices to check only $i=0,\dots,k-1$ is unjustified, so $N$ is
  not proven sufficiently large.

---

Now generate a candidate student solution for the following:

<problem>
{problem}
</problem>

<markscheme>
{grading_scheme}
</markscheme>

<correct_solution>
{correct_solution}
</correct_solution>

<score_profile>
{score_profile}
</score_profile>

Write the candidate proof below. Use standard LaTeX formatting for
all mathematics. After the proof, provide a brief "Grader's Notes"
section: for each deducted criterion, state the criterion name, the
score awarded, the specific error introduced, and why it warrants
that deduction.

# Candidate Solution
[The student's complete proof]

# Grader's Notes
[For each deducted criterion: criterion name (awarded/max), the
specific error, and why it matches the deduction]
```

TROJAN critique prompt template.

```
You are a rigorous mathematics examiner auditing a synthetic
student solution. A generator produced a candidate solution
designed to earn a specific TARGET score profile. Your task is
to independently grade the candidate and verify that a real
grader, using the markscheme below, would award exactly the
target scores.

You have access to the correct reference solution as the ground
truth for mathematical correctness.

<problem>
{problem}
</problem>

<markscheme>
{grading_scheme}
```

```
</markscheme>

<correct_solution>
{correct_solution}
</correct_solution>

<target_score_profile>
{score_profile}
</target_score_profile>

<candidate_solution>
{candidate_solution}
</candidate_solution>

Instructions:

1. **Grade each criterion independently.** For each rubric item,
   read the candidate solution and determine the ACTUAL points it
   would earn based solely on the markscheme and mathematical
   content. Use the reference solution as the correctness benchmark.

2. **Compare actual vs. target.** Set "aligned" to true only if
   actual_score == target_score for that criterion.

3. **For misaligned criteria, provide specific and actionable
   revision guidance:**
   - If the candidate earns **MORE** than the target (error too
     mild or absent): state what is currently correct that should
     be broken, and suggest a specific, *believable* student error
     -- something a competent student could plausibly make. Do not
     suggest random deletions.
   - If the candidate earns **LESS** than the target (error too
     severe): state what is currently wrong and how to partially
     or fully fix it so the student earns the target partial credit.

4. **Error calibration reference** (use this to judge whether error
   severity matches the deduction):
   - Minor deduction (e.g., 1 of 3 points lost): small gap --
     off-by-one, unchecked edge case, missing quantifier,
     unjustified assertion.
   - Moderate deduction (e.g., 2 of 3 points lost): significant
     logical gap -- skipped step, unverified hypothesis, wrong
     case analysis.
   - Full deduction (all points on a criterion lost): fundamental
     omission -- entire argument missing, theorem invoked with
     unmet hypotheses, approach doesn't address the criterion.

5. **Coherence check.** Also assess whether the proof reads as a
   natural, unified student attempt. If it feels like a patchwork
   of correct and incorrect pieces, or if errors seem artificially
   inserted rather than organically arising, flag this in the
   revision guidance.

6. **Copying policy.** Close paraphrasing of the reference solution
   is acceptable for full-marks sections -- do NOT flag similarity
   to the reference as a problem. Students who master a technique
   often reproduce it in very similar language.

Output ONLY a JSON object (no markdown fences, no surrounding text)
in this exact format:

{
  "criteria": [
    {
      "criterion_index": 1,
      "title": "criterion title",
      "target_score": 0,
      "actual_score": 0,
```

```
      "aligned": true,
      "feedback": "What the candidate does for this criterion,
                   what the target requires, and (if misaligned)
                   a specific revision to make."
    }
  ],
  "overall_aligned": true,
  "revision_guidance": "High-level summary of all revisions needed,
                        including any coherence issues. Empty
                        string if fully aligned."
}
```

TROJAN refinement prompt template.

```
You are an expert mathematics professor revising a synthetic
student solution. A critique found that some rubric criteria do
NOT earn the intended number of points. Your task is to revise
the solution so each criterion earns EXACTLY the target score.

<problem>
{problem}
</problem>

<markscheme>
{grading_scheme}
</markscheme>

<correct_solution>
{correct_solution}
</correct_solution>

<target_score_profile>
{score_profile}
</target_score_profile>

<current_candidate>
{candidate_solution}
</current_candidate>

<critique>
{critique}
</critique>

Revision rules:

1. **Preserve aligned criteria.** For criteria already matching the
   target score, keep those proof sections as close to unchanged as
   possible. Do not inadvertently break what already works.

2. **Address each critique point specifically.** For each misaligned
   criterion, follow the critique's feedback:
   - If the candidate earns **MORE** than the target: introduce a
     larger or more fundamental error to bring the score down. The
     error must be *believable* -- something a competent but
     imperfect student would plausibly write.
   - If the candidate earns **LESS** than the target: soften or
     partially fix the error so the student earns the intended
     partial credit. Use the reference solution to restore the
     relevant argument.
```

```
3. **Error calibration:**
   - Minor deduction (e.g., 1 of 3 points lost): small gap --
     off-by-one, unchecked edge case, missing quantifier,
     unjustified assertion.
   - Moderate deduction (e.g., 2 of 3 points lost): significant
     logical gap -- skipped step, unverified hypothesis, wrong
     case analysis.
   - Full deduction (all points on a criterion lost): fundamental
     omission -- entire argument missing, theorem misapplied,
     approach doesn't address the criterion.

4. **Coherence.** The revised proof must read as a single, unified
   student attempt. After changing one section, re-read the
   surrounding paragraphs to ensure transitions, notation, and
   writing style remain consistent throughout.

5. **Copying policy.** You may freely copy or closely paraphrase
   the reference solution for full-marks sections -- students who
   master a technique often reproduce it in similar language. For
   deducted sections, base your revision on the reference's
   structure but weave in errors organically. Do NOT simply delete
   lines from the reference to simulate mistakes.

6. **Minimal edits.** Make the smallest changes that fix the
   misalignments. Rewriting large correct sections increases the
   risk of introducing unintended new errors or breaking aligned
   criteria.

Output format:

# Candidate Solution
[The revised proof]

# Grader's Notes
[For each deducted criterion: criterion name (awarded/max), the
specific error, and why it matches the deduction]
```

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Each claim is supported by empirical evaluation across a broad range of language models.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide limitations and future work analysis of our work in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, we discuss all parameters for verification, grading, dataset construction, and inference, and provide training model cards for each SFT experiment.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The codebase is attached with the submission.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, we provide both Datasheet and model cards for evaluation, data collection and training.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report mean and variance of individual grading with 16 independent runs. Error bars are reported for each conducted experiment in the Appendix of the work.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We disclose every model accessed via API.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: All sourced data is in accordance with ethical standards.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our work discusses fairness in mathematical verification. We do not anticipate negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Every benchmark comes with a root source of the problem to direct the readers to original releases. Other than that we do not anticipate misuse risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite and explicitly mention every sourced data and templates in the main text and Appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, every datapoint comes with a marking scheme, candidate solutions and corresponding justifications.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not use crowd sourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not use crowd sourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We explicitly describe the LLM uses in grading mathemtical problems. Every experiment comes with a detailed explanation of base models, templates and other relevant parameters.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.