Unmasking Reasoning Processes: A Process-aware Benchmark for Evaluating Structural Mathematical Reasoning in LLMs

Anonymous ACL submission

Abstract

Recent large language models (LLMs) achieve near-saturation accuracy on many established mathematical reasoning benchmarks, raising concerns about their ability to diagnose genuine reasoning competence. This saturation largely stems from the dominance of template-based computation and shallow arithmetic decomposition in existing datasets, which underrepresent reasoning skills such as multi-constraint coordination, constructive logical synthesis, and spatial inference. To address this gap, we introduce REASONINGMATH-PLUS, a benchmark of 150 carefully curated problems explicitly designed to evaluate *structural reasoning*. Each problem emphasizes reasoning under interacting constraints, constructive solution formation, or non-trivial structural insight, and is annotated with a minimal reasoning skeleton to support fine-grained process-level evaluation. Alongside the dataset, we introduce HCRS (**H**azard-aware **C**hain-based **R**ule **S**core), a deterministic step-level scoring function, and train a Process Reward Model (PRM) on the annotated reasoning traces. Empirically, while leading models attain relatively high final-answer accuracy (up to $5.8/10$), HCRS-based holistic evaluation yields substantially lower scores (average $4.36/10$, best $5.14/10$), showing that answer-only metrics can overestimate reasoning robustness.

1 Introduction

Recent large language models (LLMs) have demonstrated substantial progress in mathematical and logical reasoning (Li et al., 2024). Benchmarks such as CMATH (Wei et al., 2023), GAOKAO (Zhang et al., 2023), ZebraLogic (Lin et al., 2025), Hard2Verify (Pandit et al., 2025), Omni-MATH (Gao et al., 2024), and FineMATH (Liu et al., 2024) have played a central role in quantifying these advances and in driving techniques including chain-of-thought prompting, verifier-based reasoning, and reward learning. However, as performance on these benchmarks approaches saturation, it becomes increasingly unclear whether high scores faithfully reflect genuine reasoning competence, particularly the ability to construct structured, minimal, and logically coherent arguments (Wang et al., 2024; Qi et al., 2024; Chen et al., 2024; Rafailov et al., 2023).

A closer examination reveals that this limitation stems not from a lack of benchmarks, but from systematic biases in how reasoning is evaluated.Existing datasets predominantly emphasize arithmetic computation or competition-style problem solving and focus primarily on final-answer correctness, offering limited insight into the structure and robustness of the reasoning process itself (Paul et al., 2024; Prasad et al., 2023). Moreover, heavy reliance on public educational resources increases the risk of overlap with pretraining corpora, complicating the distinction between genuine reasoning and memorization (Deng et al., 2024; Choi et al., 2025; Zhang et al., 2024; Carlini et al., 2021). As a result, core reasoning abilities—such as multi-constraint deduction, constructive number-theoretic reasoning, and spatial or diagrammatic inference—remain underrepresented, despite being central to human problem solving (Lin et al., 2025; Beyer and Reed, 2025).

To address these limitations, we introduce REASONINGMATH-PLUS, a benchmark of 150 curated problems designed to evaluate *structural* mathematical reasoning with explicit attention to the reasoning process, rather than final-answer correctness alone. The benchmark focuses on cognitively fundamental reasoning skills, including intuitive logic, combinatorial and number-theoretic construction, and spatial reasoning, which have been extensively studied in prior work but are often evaluated in isolation or primarily through answer-

[0]*These authors contributed equally to this work.

level metrics (Cobbe et al., 2021; Hendrycks et al., 2021; Glazer et al., 2024). Each problem is annotated with a concise, human-designed *minimal reasoning skeleton* that specifies the essential intermediate assertions required for a correct solution, enabling controlled and comparable process-level evaluation across models. To facilitate cross-lingual analysis and mitigate language-specific biases, we release parallel Chinese and English versions with matched semantics and difficulty.

Beyond final-answer evaluation, we propose a process-level assessment framework tailored to the benchmark. We introduce HCRS (Hazard-aware Chain-based Rule Score), a deterministic scoring function that evaluates reasoning traces via LLM-based step verification, with hazard-based weighting that penalizes earlier errors, consistent with recent step-level verification findings (Pandit et al., 2025). We also train a Process Reward Model (PRM) on the annotated reasoning skeletons to provide a learned scoring signal capturing coherence, logical progression, and sufficiency of generated reasoning traces. Together, these scoring mechanisms support fine-grained analysis of reasoning deviations from minimal logical structure and their impact on final correctness.

Taken together, this work introduces a process-aware benchmark for diagnosing structural mathematical reasoning in large language models. Our contributions are summarized as follows:

- **A benchmark for structural reasoning diagnosis.** We curate REASONINGMATH-PLUS, a collection of 150 problems emphasizing intuitive logic, combinatorial and number-theoretic construction, and spatial reasoning, which are not systematically isolated and diagnosed in existing benchmarks.

- **Human-designed minimal reasoning skeletons.** Each problem is annotated with a concise sequence of essential intermediate assertions (typically 2–10 steps, median 5), providing a task-specific structural reference for controlled step-level analysis without constraining surface realizations.

- **A minimal-structure–grounded process evaluation framework.** We introduce HCRS, a hazard-adjusted deterministic scoring function, together with a Process Reward Model (PRM) trained on the same skeleton annotations. By grounding process evaluation in problem-specific minimal reasoning structure, the framework evaluates whether intermediate assertions satisfy essential constraints, rather than only checking the final answer. This reveals substantial gaps between answer-level success and process-consistent reasoning: answer-level scores can reach $5.8/10$, whereas holistic HCRS-based scores average $4.36/10$.

2 Related Work

Mathematical and logical reasoning benchmarks. A wide range of benchmarks have been proposed to evaluate mathematical reasoning in large language models. Early datasets such as GSM8K (Cobbe et al., 2021) and ASDiv (Miao et al., 2020) focus primarily on arithmetic word problems with short multi-step derivations. These benchmarks have played an important role in demonstrating the effectiveness of chain-of-thought prompting and self-consistency decoding; however, the underlying tasks are highly templated and often overlap with common educational resources, limiting their ability to diagnose deeper reasoning abilities. More advanced benchmarks, including competition-style datasets such as AIME25 (Zhang and Math-AI, 2024), HMMT25 (Henkel, 2025), and MiniF2F (Zheng et al., 2021), extend coverage to algebra, geometry, combinatorics, and number theory, capturing higher difficulty levels and longer reasoning chains. Despite this increased difficulty, these benchmarks typically rely on final-answer evaluation and provide full solution traces rather than minimal, structured reasoning representations, which constrains fine-grained analysis of reasoning behavior.

More recently, efforts such as Omni-MATH (Gao et al., 2024) and FrontierMath (Glazer et al., 2024) further expand topical scope and difficulty to university-level and research-oriented problems. While these datasets expose the limitations of current LLMs at the frontier of mathematical reasoning, their heterogeneity in difficulty and reliance on specialized solvers complicate controlled comparison and systematic process-level evaluation. In contrast, REASONINGMATH-PLUS focuses on a targeted set of structural reasoning skills—multi-constraint logical deduction, constructive combinatorics, and spatial intuition—that remain underrepresented in existing benchmarks. Table 1 summarizes key

differences between REASONINGMATH-PLUS and representative datasets.

Evaluating reasoning processes. Beyond final-answer accuracy, evaluating the reasoning process itself has become an important research direction (Cobbe et al., 2021). Early approaches relied on heuristic pattern matching or post-hoc regular expressions to identify invalid reasoning steps (Cobbe et al., 2021; Wang et al., 2022). More recent work employs LLM-based verifiers for step-level evaluation or trains reward models to score reasoning traces, as explored in verifier-guided decoding and PRM-based frameworks (Ouyang et al., 2022; Rafailov et al., 2023).

In mathematical reasoning, some systems further incorporate symbolic solvers or formal proof assistants to validate intermediate steps (Wei et al., 2022; Glazer et al., 2024). While effective for formalized mathematics, such approaches are difficult to generalize to natural-language reasoning tasks, particularly those requiring intuition, construction, or implicit constraints. Consequently, across existing step-evaluation methods, a common limitation is that reasoning steps are typically assessed independently, without accounting for the temporal position of an error within the reasoning chain. Early mistakes often propagate and dominate downstream reasoning, yet this effect is rarely reflected in evaluation metrics. REASONINGMATH-PLUS addresses this gap by aligning process-level evaluation with human-designed minimal reasoning structure and by explicitly modeling the impact of error position through hazard-adjusted scoring.

3 Benchmark Curation

3.1 Motivation

REASONINGMATH-PLUS is designed to expose structural reasoning failures that are often obscured by answer-only evaluation. Rather than increasing symbolic or computational complexity, we focus on problems whose correctness depends on coordinating multiple constraints, eliminating inconsistent possibilities, and constructing minimal yet sufficient reasoning chains. This design enables targeted diagnosis of reasoning errors that are not reliably revealed by existing mathematical benchmarks.

3.2 Data Collection

Data construction. REASONINGMATH-PLUS is curated to elicit *structural* inference rather than symbolic manipulation. Concretely, (S1) we draft candidates via manual authorship and structural adaptation of puzzle-style tasks; (S2) we iteratively refine each item into a precise natural-language statement with explicit constraints and minimal specialized notation; (S3) we check solution well-definedness by independent re-solving and remove or revise candidates with ambiguity or non-unique interpretations; and (S4) we reduce resemblance to common exam/contest templates through targeted rewriting and conservative screening for overly formulaic patterns.

Released annotations. The benchmark consists of 150 problems with explicit process-level annotations. Each item is released with a gold final answer, a full human-written solution, a reasoning skeleton, and subject and difficulty labels. In addition, all problems are provided in parallel Chinese and English versions with aligned semantics and difficulty to support cross-lingual analysis. Table 2 presents an example of the annotation schema and reasoning skeleton.

Minimal reasoning skeletons. For process-level evaluation, each problem is annotated with a *minimal* reasoning skeleton—a short sequence of *necessary intermediate assertions* that are sufficient to derive the gold answer. Importantly, *minimal* does not prescribe how a model should write its reasoning: models may produce longer traces with self-verification or deliberation. The skeleton serves as a stable alignment target for step verification by focusing evaluation on essential structural commitments, enabling (i) comparable process scoring across traces of vastly different lengths, and (ii) precise localization of the earliest structural error that drives downstream failure. Skeletons range from 2 to 10 steps (mean 4.65).

Subject-level labels for analysis. In addition to process-level annotations, each problem is assigned a coarse-grained mathematical subject label. These labels are *not* intended to define task formats or target specific domain skills; rather, they serve as an analysis axis that enables comparison with prior mathematical reasoning benchmarks and supports diagnostic breakdowns across familiar categories. We adopt conventional subject labels (algebra, number theory, geometry, combinatorics, and probability) to maintain interpretability and facilitate cross-benchmark analysis, while emphasizing that the primary evaluation focus of REASONINGMATH-PLUS lies in structural reasoning patterns rather than domain-specific content. In the final dataset,

Benchmark	AIME25	HMMT25	C-EVAL	M3KE	REASONINGMATH-PLUS
Language	En	En	Zh	Zh	Zh
Size	30	30	669	796	150
Problem Type	Fill-in-the-Blank	Fill-in-the-Blank / MWP	MCQ	MCQ	Fill-in-the-Blank / MWP
Question Len.	363.07	328.53	76.28	46.24	125.0
Solution Len.	2.90	10.13	–	–	213.9
Subject Label	✓	✓	✗	✗	✓
Reasoning Skeleton	✗	✗	✗	✗	✓

Table 1: Comparison of REASONINGMATH-PLUS with representative math reasoning benchmarks. MWP denotes math word problems and MCQ denotes multiple-choice questions.

algebra (71 problems) and number theory (51 problems) constitute the majority of instances, reflecting the prevalence of constraint-based deduction and constructive reasoning in these areas. Geometry (12), combinatorics (11), and probability (5) provide additional structural diversity. The subject distribution is summarized in Table 3.

4 Methodology

To assess reasoning validity beyond outcome correctness, we propose a dual-perspective framework under two supervision regimes (Fig. 1): a *skeleton-guided diagnosis* when dense intermediate annotations are available, and an *outcome-conditioned verifier* otherwise.

1. **Branch A: Skeleton-guided Structural Diagnosis (HCRS).** Given an expert *reasoning skeleton* of necessary assertions, an LLM judge inspects each step for *structural commitment*. This paraphrase-tolerant diagnosis focuses on semantic alignment rather than exact matching. We aggregate results via **HCRS**, applying format and hazard penalties to strictly penalize early logical fractures.

2. **Branch B: Outcome-conditioned Verification (PRM).** Without skeleton supervision, a learned Process Reward Model (PRM) verifies each step in an *outcome-conditioned* manner, using only the problem statement and the gold final answer, and outputs discrete step-validity labels for scalable evaluation.

Notably, we show in Section 5.2.3 that HCRS-style penalties are **verifier-agnostic** and can improve alignment for both skeleton-guided and outcome-conditioned verifiers.

4.1 Branch A: Skeleton-guided Diagnosis and HCRS Aggregation

Motivation. Answer-only evaluation can mask *right answer, wrong reasoning* failures and provides limited granularity for long chains. We therefore perform **step-level diagnosis** to localize errors and assign partial credit with explicit justifications.

HCRS Framework. Branch A defines **HCRS**, a deterministic aggregation rule mapping step validity labels from an external judge J to a scalar score via format- and hazard-based penalties. We instantiate J with `Gemini-3-Pro` (our fixed teacher judge). All hyperparameters (e.g., α, β, w) are fixed across models (Table 4).

Step Validity and Base Score. Given an input x and a trace $S_{1:N}$, the judge assigns binary labels $\mathcal{V} = \{v_1, \ldots, v_N\}$ by inspecting each step against skeleton assertions (paraphrases allowed; commitments required). These labels are then aggregated into a normalized base score:

$$S_{\text{base}} = \frac{10}{N} \sum_{i=1}^{N} v_i. \tag{1}$$

Format Deviation Penalty (P_{fmt}). Let $r = |N - L_{\text{gold}}|/L_{\text{gold}}$ denote deviation from the reference length L_{gold}.

$$P_{\text{fmt}} = \alpha r e^{\beta r}. \tag{2}$$

We apply an asymmetry factor η (set $\eta = 1.5$ if $N < L_{\text{gold}}$, otherwise $\eta = 1.0$) and cap the deduction by C_{fmt}.

Hazard Penalty (P_{haz}). For a first error at step $t^\star$, we apply a pre-defined hazard schedule:

$$\tilde{P}_{\text{haz}}(t^\star) = \begin{cases} 1 - \frac{H(t^\star - 1)}{H_{\max}}, & t^\star \le T_{\max} \\ 0, & t^\star > T_{\max} \end{cases} \tag{3}$$

Information	Example in Chinese	English Translation
Question:	我设计了一个游戏，给你6个数字，你可以进行加减乘除运算，其中每次加减运算得1分，每次乘除运算得2分，得出指定输出再加六分。 比如我给你 78，2，13，91，1，30，指定输出为 6，请得出一种得分最高的方案。	I designed a game where you are given six numbers and may apply addition, subtraction, multiplication, and division. Each addition or subtraction gives 1 point, each multiplication or division gives 2 points, and producing the target output yields an additional 6 points. Given the numbers 78, 2, 13, 91, 1, and 30 with target output 6, find a solution with the maximum score.
Solution:	思维链分析标准: step1：将题目转化为运算步骤选择问题，每个运算符都有对应分值。 step2：在保证最终运算结果等于指定输出的前提下，优先使用高分值运算（乘、除）以最大化总分。 step3：枚举或推导运算符排列组合，计算运算结果与得分，筛选出得分最高方案。 step4：验证最终方案运算正确且得分满足最大化。 解题分析标准: 初始化数字集合：78，2，13，91，1，30。 分析得分规则：加减 = 1 分，乘除 = 2 分，成功得到目标输出额外 +6 分。 构造表达式：(78 / 2 / 13) + (91 - 1) / 30。 其中：(78 / 2) = 39（除法，+2 分），(39 / 13) = 3（除法，+2 分）， (91 - 1) = 90（减法，+1 分），(90 / 30) = 3（除法，+2 分）， (3 + 3) = 6（加法，+1 分）。 最终得分：除法 3 次 ×2 分 = 6 分，加减 2 次 ×1 分 = 2 分，额外奖励 6 分，总分 = 14 分。	**Reasoning Skeleton:** Step 1: Reformulate the task as an operation selection problem where each operator has an associated score. Step 2: Under the constraint that the final result equals the target value, prioritize high-scoring operations (multiplication and division). Step 3: Enumerate or derive operator combinations, compute results and scores, and select the highest-scoring valid solution. Step 4: Verify that the final expression is correct and achieves the maximum score. **Detailed Reasoning:** Initialize the number set: 78, 2, 13, 91, 1, 30. Scoring rules: addition/subtraction = 1 point; multiplication/division = 2 points; successful target output = +6 points. Construct the expression: (78 / 2 / 13) + (91 - 1) / 30. Operations include three divisions and two additions/subtractions, producing the target value 6. The total score is 14, which is maximal under the given rules.
Subject:	代数	Algebra
Level:	难	Hard
Answer:	使用计算过程 (78 / 2 / 13) + (91 - 1) / 30，最终结果为 6，得分为 14。	The expression (78 / 2 / 13) + (91 - 1) / 30 produces the target value 6 with a total score of 14.

Table 2: A bilingual example from REASONINGMATH-PLUS

and set $P_{\text{haz}}(t^\star) = \min(C_{\text{haz}}, \omega \cdot \tilde{P}_{\text{haz}}(t^\star))$. The schedule is fixed across all evaluated models (Appendix F).

Aggregation. We define the process-only score as $S_{\text{HCRS}} = \max(0, S_{\text{base}} - P_{\text{fmt}} - P_{\text{haz}})$. For *reporting only*, we optionally form a holistic score $S_{\text{hol}} = wS_{\text{HCRS}} + (1-w)S_{\text{ans}}$, where $S_{\text{ans}} \in \{0, 10\}$ and $w = 0.7$. Unless otherwise noted, all analyses use the process-only score S_{HCRS}.

4.2 Branch B: Outcome-conditioned Verification via PRM

Branch B introduces a learned PRM to extend step verification to an **outcome-conditioned** regime where skeletons are unavailable. We instantiate the PRM with **Qwen3-8B-instruct** and train it to predict step validity labels from a fixed teacher judge.

Training Data Construction. We build a step-level corpus from *DeepMath*, *OmniThought*, *MiroMind*, *LIMO*, and *NuminaMath*. From 2,500 sampled problems, we generate 35,000 candidate traces using 14 LLMs under a unified CoT protocol. A fixed teacher (`Gemini-3-Pro`) labels each step with $y_i \in \{0, 1\}$ conditioned on the problem and gold final answer. After filtering malformed outputs, we retain ∼33k instances and fine-tune via cross-entropy.

Inference-time Scoring. At inference, the PRM acts as a *generative verifier*, producing for each step a rationale $\mathcal{R}_i$ and a discrete validity label $\hat{y}_i \in \{0, 1\}$, given by $(\hat{y}_i, \mathcal{R}_i) = f_\phi(S_i \mid x, S_{<i})$. We aggregate labels into a normalized process score:

$$S_{\text{PRM}} = \frac{10}{N} \sum_{i=1}^{N} \hat{y}_i.$$

S_{PRM} is a process metric and uses the gold answer only through its inclusion in x.

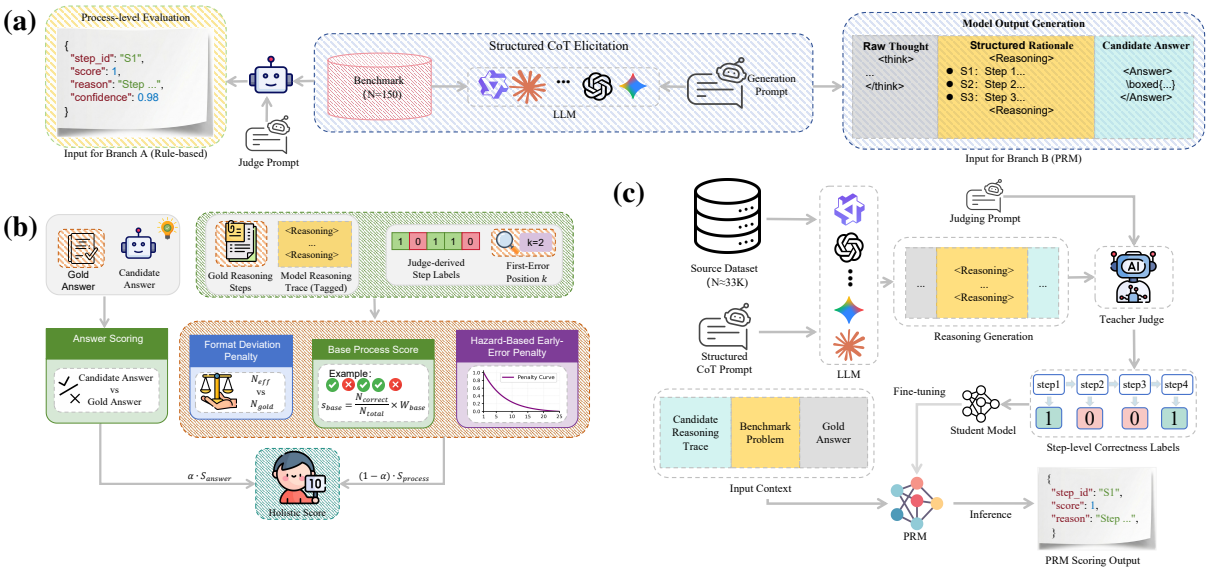

Figure 1: **Overall framework. (a)** Structured CoT elicitation for process-level evaluation. **(b)** Branch A (Skeleton-guided diagnosis): a judge checks step validity against a minimal set of *necessary* skeleton assertions (paraphrase-tolerant; not exact-match), aggregated by HCRS with format and hazard penalties. **(c)** Branch B (Outcome-conditioned verification): a PRM distilled from teacher-judge labels and applied at inference time using only the problem and the gold final answer.

5 Experiments

5.1 Experimental Setup

Dataset. We evaluate our framework on REASONINGMATH-PLUS, comprising 150 problems designed to stress long-horizon logical consistency. The dataset emphasizes non-trivial multi-step derivations with unambiguous answers, spanning Algebra, Number Theory, Geometry, Combinatorics, and Probability (see Table 3). **Models.** We evaluate 14 endpoints from major families including GPT-5, Gemini, Claude, Grok, Qwen, DeepSeek, and Llama-based models. Precise API identifiers and version strings are detailed in Appendix G. **Evaluation Protocol and Judge Selection.** Models generate structured traces following a fixed schema (*Raw Thought*, *Reasoning Steps*, *Final Answer*).

To ensure rigorous evaluation, we benchmarked multiple candidate judges against human annotations on the generated traces. `Gemini-3-Pro` demonstrated the highest alignment (R=0.64) and was selected as the global judge for providing step-validity labels in the HCRS pipeline and supervision for PRM training (alignment calibration details in Appendix B).

5.2 Main Results

We present our experimental findings organized by the two evaluation protocols defined in Section 4. First, we analyze model robustness and structural fragility using the rule-based **HCRS** (Sec. 5.2.1). Second, we examine semantic reasoning quality using the learned **PRM** (Sec. 5.2.3).

5.2.1 Structural Diagnosis via HCRS

HCRS Leaderboard and Penalty Gaps. Figure 2a shows that HCRS induces a clearer stratification than answer accuracy alone. Gemini-2.5-Pro ranks first (4.89), followed by Doubao-Seed-1.6 (thinking) (4.78) and Qwen3-Max (2025-09-23) (4.69). The grey bars denote deductions from format deviations and the hazard-based *first-error penalty*. These penalties correspond to a substantial reduction of approximately 2.7–3.2 points, amounting to roughly 30% of the 10-point scale. This divergence between potential and realized reasoning quality also appears at the instance level (Figure 5).

Quantifying "Lucky Guesses". Figure 5 shows the process score distribution for the 996 samples where the final answer is correct. We observe that **6.63%** (66/996) of these instances fall into the low-score range ($S_{\text{HCRS}} \leq 3$). This indicates that while traditional outcome-based metrics would classify these samples as correct, our structural evaluation identifies them as reasoning failures (i.e., "lucky guesses").

Domain-wise diagnostic breakdown (descriptive). Figure 3 provides a diagnostic comparison

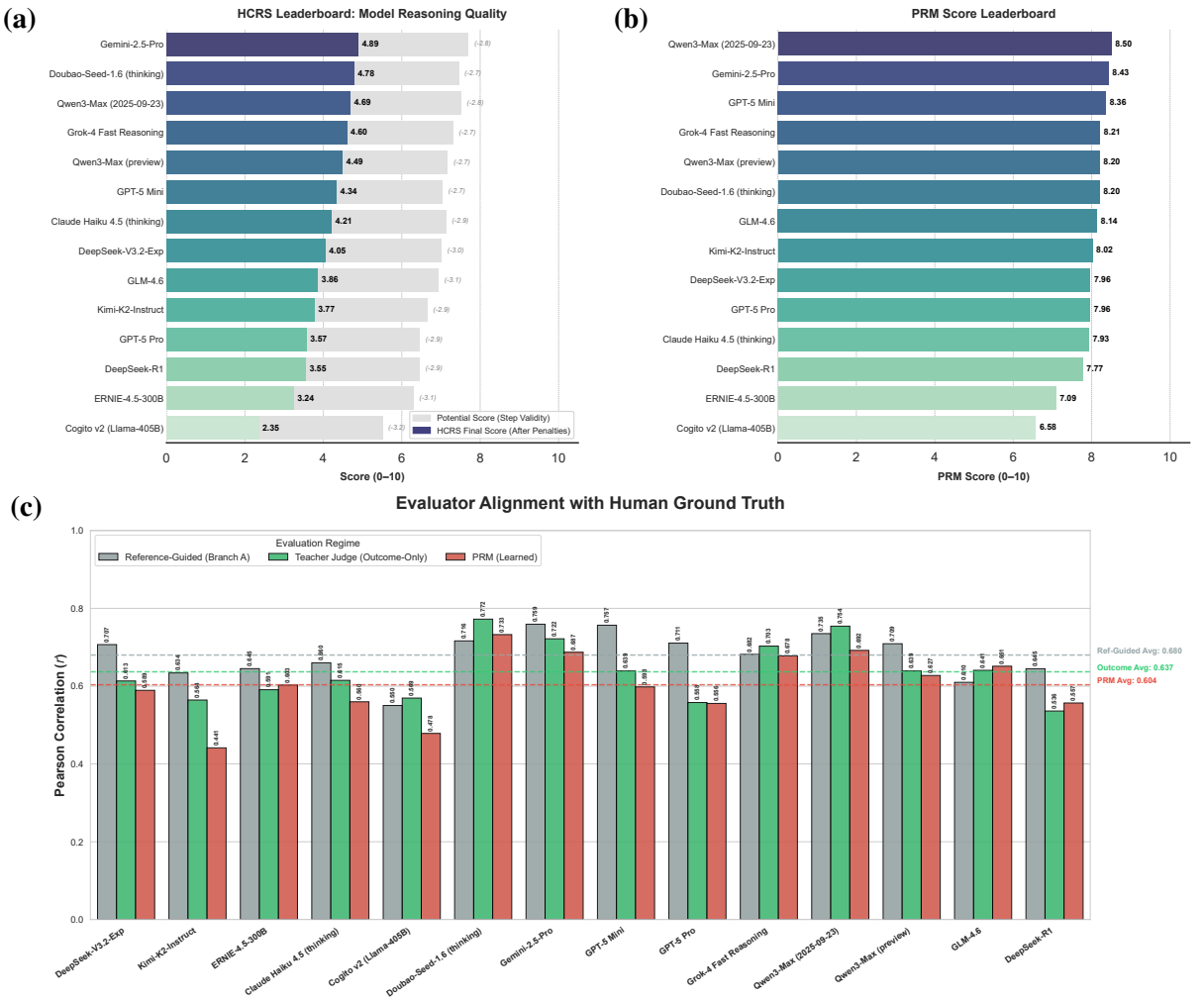

Figure 2: **Evaluation leaderboards and correlation analysis.** **(a)** HCRS leaderboard across domains (grey segments indicate deductions from format and hazard penalties). **(b)** PRM process-score leaderboard under outcome-conditioned step verification. **(c)** Pearson correlation of evaluation methods (Reference-guided, teacher judge, and PRM) against human judgments.

between composite scores (a) and answer accuracy (b). While limited by small sample sizes in sub-domains like Probability and Combinatorics (see Table 3), the radar charts reveal a critical divergence. We observe a marked *inward contraction* for mid-tier models (e.g., the green trajectory) when moving from accuracy to composite scores, particularly in combinatorial tasks. This indicates that a portion of their apparent accuracy masks brittle or formatted-invalid reasoning. Conversely, `Gemini-2.5-Pro` exhibits the most stable envelope across both figures, demonstrating that its high performance is supported by rigorous step-wise validity rather than lucky guesses.

5.2.2 Alignment with Human Judgments

We evaluate the reliability of our scoring methods by measuring their consistency with human-annotated scores. First, we examine linear alignment using Pearson correlation (R), as shown in Figure 2c. Despite operating in an *outcome-conditioned* setting (i.e., conditioned only on the problem and the gold final answer, without access to gold reasoning steps), PRM achieves a competitive average correlation of $R = 0.602$. This closely tracks the Pro-tier teacher judge (`Gemini-3-Pro`, $R = 0.639$).

To assess robustness beyond linear correlation, we further report rank-aware and agreement-based metrics in Appendix Figure 7, including Spearman's ρ (monotonic rank correlation), Kendall's τ (pairwise concordance), and Quadratic Weighted Cohen's κ (agreement intensity). These complementary metrics provide a more comprehensive view of evaluator reliability beyond Pearson corre-

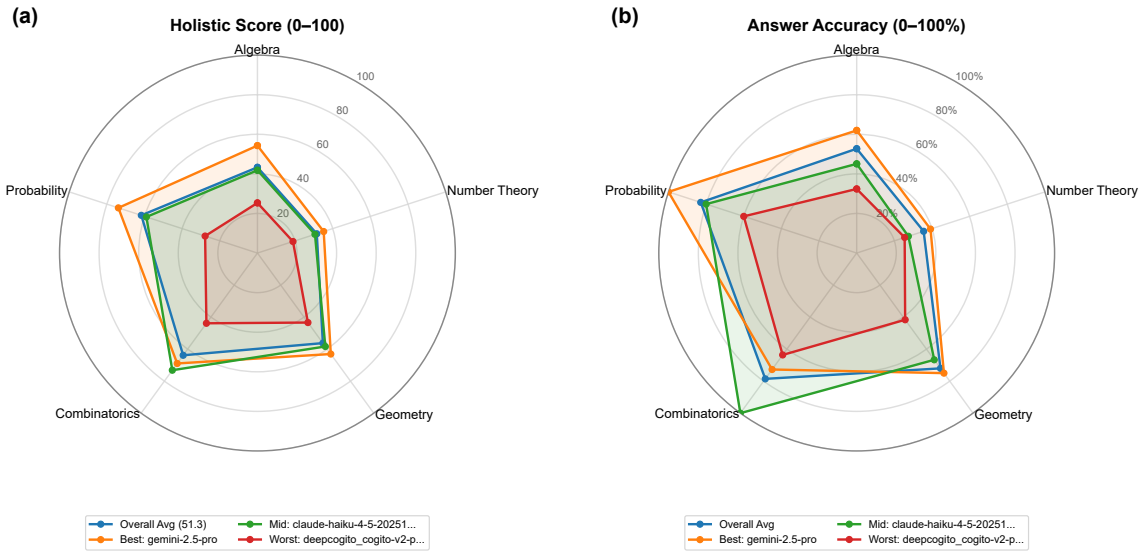

Figure 3: **Subject-wise capability analysis.** Radar plots comparing **(a)** Holistic Score (0–100) and **(b)** Answer Accuracy (0–100%) across five domains. Curves correspond to the **Best**, **Median**, and **Worst** models selected by overall Holistic Score, together with the overall average.

lation alone.

Crucially, PRM (Method C) remains competitive across all metrics ($\rho = 0.684$, $\tau = 0.565$, $\kappa = 0.568$), confirming that it largely preserves the relative quality rankings preferred by humans even without access to gold intermediate steps.

5.2.3 Scalable Verification via PRM

Figure 2b presents the PRM rankings. While the top-tier hierarchy mirrors HCRS (with Qwen3-Max and Gemini-2.5-Pro leading), the score distribution exhibits a distinct **compression effect**. Unlike the steep $\sim$52% performance drop observed in HCRS, the lowest-ranked model in the PRM leaderboard retains $\sim$77% of the top score (6.58 vs. 8.50). This saturation stems from the PRM's averaging-based aggregation ($\frac{1}{N}\sum \hat{y}_i$), which grants partial credit for locally valid steps even within flawed chains. Consequently, the PRM functions as a smoother measure of **local semantic consistency**, complementing the strict structural stratification of HCRS.

5.2.4 When Simple Rules Improve Verifiers

A key observation is that the HCRS penalty terms can serve as a simple, verifier-agnostic refinement on top of raw step-wise scoring signals. As shown in Figure 8, applying the same HCRS penalties (*format penalty* and *first-error penalty*) to both the teacher judge (`Gemini-3-Pro`) and the trained PRM consistently improves alignment with human judgments. Notably, the gain is more pronounced for PRM, whose average Pearson correlation increases from 0.604 to 0.633 (vs. 0.637 to 0.642 for Gemini). This suggests that rule-based diagnosis captures systematic structural failure modes that are not fully reflected by raw step-wise validity predictions, providing a generalizable enhancement for process evaluation even in settings where expert skeletons are unavailable.

6 Conclusion

In this work, we addressed the opacity of outcome-based evaluation by introducing a dual-perspective framework for long-horizon mathematical reasoning. Our approach integrates **HCRS**—a *skeleton-guided* protocol for explicit structural diagnosis—with a **PRM** designed for *outcome-conditioned* semantic verification. Experiments on our curated benchmark reveal that answer-only metrics significantly overestimate reliability by masking "lucky guesses," a phenomenon effectively quantified by our hazard-aware penalties. Validated by high alignment with human experts, this framework bridges the gap between high-precision structural **diagnostic** signals and flexible learned verification, establishing a transparent and scalable paradigm for verifiable reasoning.

7 Limitations

Limitations. Our framework targets fine-grained, process-level auditing and therefore involves modest practical overhead. In particular, constructing reasoning skeletons and gold step counts may re-

quire expert effort, which can increase annotation cost compared to outcome-only evaluation. That said, this design choice enables more precise diagnosis of step-wise consistency and error propagation, and can serve as a high-quality supervision source for training lighter-weight verifiers in future work.

References

Henrike Beyer and Chris Reed. 2025. Lexical recall or logical reasoning: Probing the limits of reasoning abilities in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13532–13557.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, and 1 others. 2021. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024. Alphamath almost zero: process supervision without process. *Advances in Neural Information Processing Systems*, 37:27689–27724.

Hyeong Kyu Choi, Maxim Khanov, Hongxin Wei, and Yixuan Li. 2025. How contaminated is your benchmark? quantifying dataset leakage in large language models with kernel divergence. *arXiv preprint arXiv:2502.00678*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2024. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8706–8719.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, and 1 others. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, and 1 others. 2024. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *arXiv preprint arXiv:2411.04872*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Jonas Henkel. 2025. The mathematician's assistant: integrating ai into research practice. *Mathematische Semesterberichte*, pages 1–28.

Leo Li, Ye Luo, and Tingyou Pan. 2024. Openai-o1 ab testing: Does the o1 model really do good reasoning in math problem solving? *arXiv preprint arXiv:2411.06198*.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. 2025. Zebralogic: On the scaling limits of llms for logical reasoning. *arXiv preprint arXiv:2502.01100*.

Yan Liu, Renren Jin, Ling Shi, Zheng Yao, and Deyi Xiong. 2024. Finemath: A fine-grained mathematical evaluation benchmark for chinese large language models. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pages 975–984.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Shrey Pandit, Austin Xu, Xuan-Phi Nguyen, Yifei Ming, Caiming Xiong, and Shafiq Joty. 2025. Hard2verify: A step-level verification benchmark for open-ended frontier math. *arXiv preprint arXiv:2510.13744*.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. 2024. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. *arXiv preprint arXiv:2402.13950*.

Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. 2023. Receval: Evaluating reasoning chains via correctness and informativeness. *arXiv preprint arXiv:2304.10703*.

Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. 2024. Mutual reasoning makes smaller llms stronger problem-solvers. *arXiv preprint arXiv:2408.06195*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. 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)*, pages 9426–9439.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Tianwen Wei, Jian Luan, Wei Liu, Shuang Dong, and Bin Wang. 2023. Cmath: Can your language model pass chinese elementary school math test? *arXiv preprint arXiv:2306.16636*.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, William Song, Tiffany Zhao, Pranav Raja, Charlotte Zhuang, Dylan Slack, and 1 others. 2024. A careful examination of large language model performance on grade school arithmetic. *Advances in Neural Information Processing Systems*, 37:46819–46836.

Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. 2023. Evaluating the performance of large language models on gaokao benchmark. *arXiv preprint arXiv:2305.12474*.

Yifan Zhang and Team Math-AI. 2024. American invitational mathematics examination (aime) 2025.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. 2021. Minif2f: a cross-system benchmark for formal olympiad-level mathematics. *arXiv preprint arXiv:2109.00110*.

A Data Statistics

B Judge Model Selection and Calibration

To ensure the reliability of our automated evaluation instrument, we performed a rigorous calibration study prior to the main experiments. This section provides the technical details of the judge selection process that were omitted from the main text for brevity.

Subject Category	Category Content	Size
Algebra	Equation and inequality reasoning, functional relationships, algebraic constraints, and symbolic abstraction.	71
Number Theory	Divisibility, parity arguments, modular arithmetic, and constructive reasoning over integers.	51
Geometry	Planar and spatial configuration reasoning, geometric relationships, and transformations.	12
Combinatorics	Counting arguments, permutations and combinations, case analysis, and discrete construction.	11
Probability	Reasoning about random events, conditional probability, and basic probabilistic inference.	5

Table 3: Subject categories and distributions of REASONINGMATH-PLUS.

Data Sampling and Diversity. The calibration dataset comprises a comprehensive set of 2,100 reasoning traces (150 problems $\times$ 14 models) generated during the benchmark's preliminary phase. This large-scale sampling allows candidate judges to be evaluated across a diverse spectrum of reasoning behaviors, ranging from concise logical derivations to verbose chain-of-thought explorations. Furthermore, the dataset spans a wide array of trace qualities—from perfect derivations to complete logical collapses—while accounting for varying levels of format adherence to ensure the judge's robustness against minor structural deviations.

Candidate Judges and Methodology. We benchmarked several state-of-the-art LLMs as candidate judges, including `GPT-4o`, `Claude-3.5-Sonnet`, and `Gemini-3-Pro`. For each candidate, we applied the reference-guided prompt (Branch A) to generate step-level validity labels. These labels were then aggregated via the HCRS pipeline to produce process-level scores.

Alignment Metrics and Selection Result. The primary metric for selection was the **Pearson correlation** (R) between the judge-generated HCRS scores and ground-truth scores provided by human experts. As shown in Figure 4, `Gemini-3-Pro` demonstrated the highest consistency with human judgments, achieving a correlation coefficient of R=0.64. Crucially, as noted in Section 5.1, this calibration involves no tuning of the evaluated models

or scoring rules; the chosen judge is fixed globally for all subsequent analyses to maintain the integrity of the evaluation.

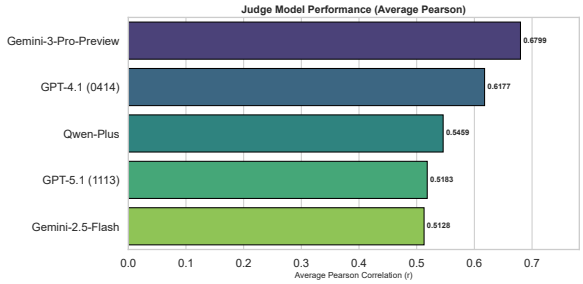

Figure 4: **Judge performance comparison.** Comparison of candidate judge models based on average Pearson correlation with human annotations (N=2, 100). `Gemini-3-Pro` exhibits the strongest alignment (R=0.64).

Candidate Judges and Methodology. We benchmarked several state-of-the-art LLMs as candidate judges, including `GPT-4o`, `Claude-3.5-Sonnet`, and `Gemini-3-Pro`. For each candidate, we applied the reference-guided prompt (Branch A) to generate step-level validity labels. These labels were then aggregated via the HCRS pipeline to produce process-level scores.

Alignment Metrics and Selection Result. The primary metric for selection was the **Pearson correlation** (R) between the judge-generated HCRS scores and ground-truth scores provided by human experts. As shown in Figure 4, `Gemini-3-Pro` demonstrated the highest consistency with human judgments, achieving a correlation coefficient of R=0.64. Crucially, as noted in Section 5.1, this calibration involves no tuning of the evaluated models or scoring rules; the chosen judge is fixed globally for all subsequent analyses to maintain the integrity of the evaluation.

C Hyperparameter Settings

The specific hyperparameter values used in our rule-based scoring module (HCRS) are detailed in Table 4. These values were selected based on a grid search on a held-out validation set to maximize the correlation with human preferences.

Parameter	Value	Description
α	4.0	Scale factor controlling the sensitivity of the length deviation penalty.
β	1.0	Exponent governing the growth rate of the length deviation penalty.
C_{fmt}	3.0	Maximum penalty cap for format length deviation.
η	1.5(1.0)	Asymmetry factor applied when reasoning is shorter than the reference ($N < L_{\text{gold}}$). For $N \geq L_{\text{gold}}$, η is set to 1.0.
ω	5.0	Scaling weight for the hazard-based penalty.
C_{haz}	5.0	Maximum penalty cap for the first-error hazard deduction.
$T_{\max}$	25	Maximum step index considered in the hazard model. Steps beyond 25 incur no hazard penalty.
w	0.7	Weight assigned to the process score (S_{HCRS}) in the holistic metric. The remaining weight (0.3) is assigned to answer accuracy.

Table 4: Optimized and fixed hyperparameters used in our evaluation framework. The table includes parameters for the HCRS scoring module (format and first-error penalties) and the holistic aggregation weight.

D Human Annotation Guidelines

To validate our automatic metrics, we recruited three annotators with undergraduate degrees in mathematics or related fields. Following a standardized 0–10 rubric, they evaluated each reasoning trace as follows.

- **Process Step Matching (0–7 points):** Measures coverage of the standard reasoning steps (Step1–Step$N$) in the annotated skeleton. Let N be the total number of standard steps and M the number of covered steps. The process score is $S_{\text{process}} = 7 \times (M/N)$, rounded to one decimal place. Incorrect steps are not counted as covered.

- **Answer Correctness (0 or 3 points):** If the final answer matches the gold answer (up to standard numerical tolerances), the model receives $S_{\text{answer}} = 3$; otherwise $S_{\text{answer}} = 0$.

- **Penalties (each in $[0, 1]$):** Three penalties are deducted from the base score:

  1. *Redundancy Penalty:* For verbose, repetitive, or circular reasoning.
  2. *First-error Penalty:* Let k denote the index of the first critical error that affects the main line of reasoning. The penalty increases for earlier errors, computed as

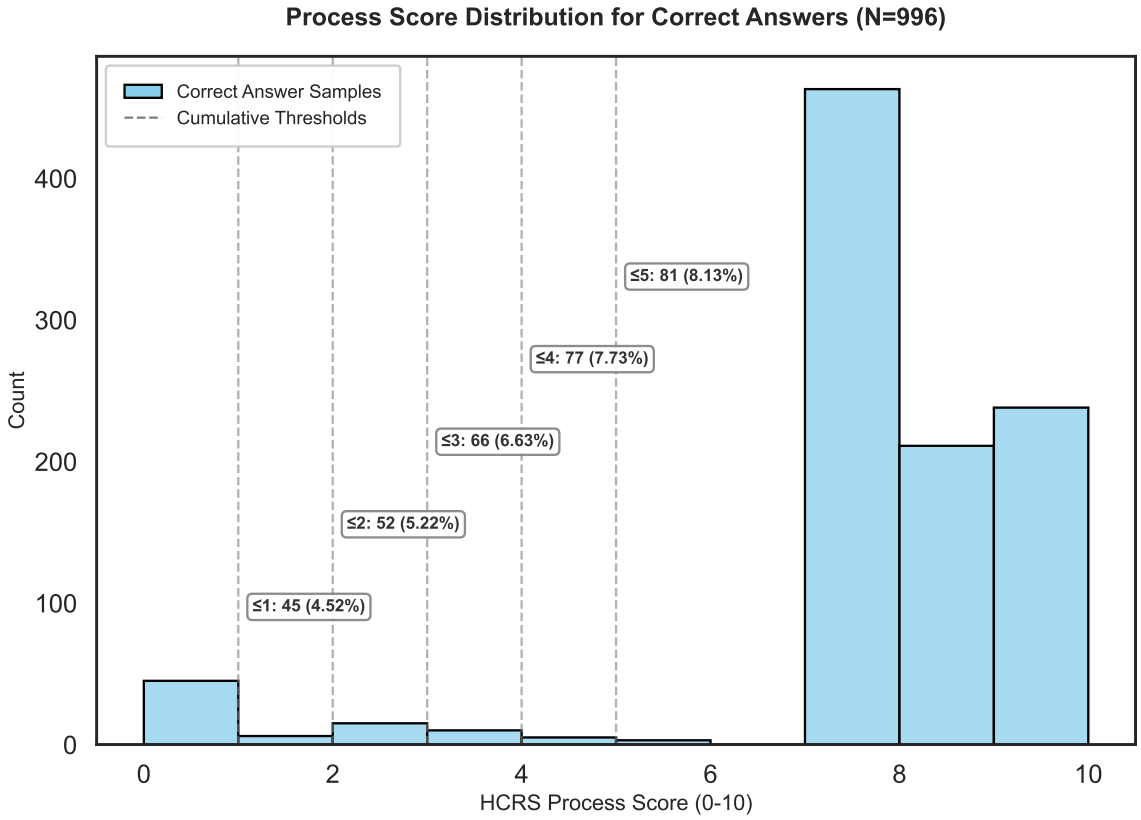

Figure 5: **Bimodal distribution of process quality conditioned on correct final answers.** Histogram of the *process-only* HCRS score S_{HCRS} ... for 996 answer-correct traces (out of 2,100 total model traces). Vertical dashed lines indicate cumulative thresholds at $S_{\text{HCRS}} \leq k$ ($k \in \{1, 2, 3, 4, 5\}$), with callouts reporting cumulative counts and percentages. Notably, 6.63% (66/996) of correct answers have $S_{\text{HCRS}} \leq 3$, suggesting that outcome correctness can coincide with low-quality or inconsistent reasoning (*"lucky guesses"*).

$P_{\text{first}} = 1 - (k-1)/N$ (and 0 if no such error exists), rounded to one decimal place.

3. *Rigor Penalty:* For insufficient rigor (e.g., missing proof of a construction, incomplete case enumeration, or lacking optimality justification).

The final human score is computed as $S_{\text{total}} = \max\big(0,\, S_{\text{process}} + S_{\text{answer}} - P_{\text{redundancy}} - P_{\text{first}} - P_{\text{rigor}}\big)$. We average the three annotators' scores to obtain a single human score per trace.

E Metric Validation and Ablation Analysis

Appendix B describes our judge selection procedure. Building on the Pearson-based human-alignment results reported in the main text (Figure 2c), this appendix provides complementary robustness analyses using rank-aware and agreement-based metrics.

Rank-Aware Robustness. To evaluate robustness beyond linear correlation, we report Spearman's ρ (monotonic rank), Kendall's τ (pairwise ranking), and Quadratic Weighted κ (agreement intensity) in Figure 7.

The results reveal a notable finding: the **Teacher Judge (Outcome-Only)** consistently outperforms the **Reference-Guided (Branch A)** baseline across all metrics. This advantage is most pronounced in inter-rater agreement ($\kappa = 0.608$ vs. $\kappa = 0.436$). We hypothesize that while reference-guided scoring (HCRS) enforces structural rigor, the outcome-conditioned teacher (Branch B) better mimics human flexibility in recognizing valid reasoning paths that deviate from the gold skeleton.

Crucially, the distilled **PRM (Learned)** closely tracks the teacher's performance and also surpasses the reference-guided baseline on agreement metrics (e.g., $\kappa = 0.568$), demonstrating that the student model successfully internalizes the teacher's judgment criteria without requiring reference skeletons at inference time.

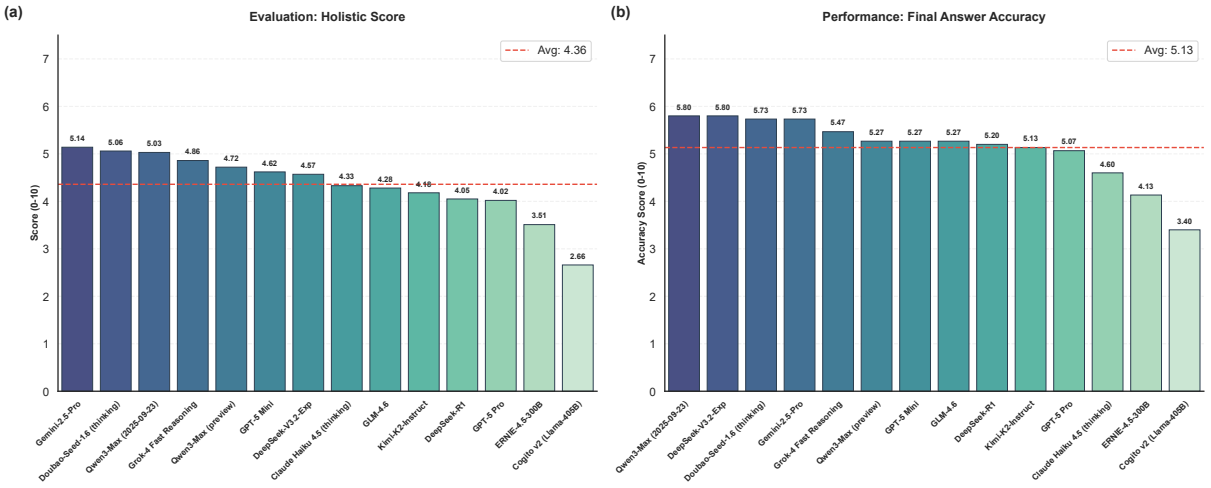

Figure 6: **Holistic Evaluation vs. Final Answer Accuracy. (a)** Leaderboard based on the weighted holistic score (S_{overall}), integrating HCRS (70%) and binary answer correctness (30%). **(b)** Leaderboard based solely on raw final answer accuracy. Comparing the two shows that the holistic metric provides finer granularity, penalizing models (e.g., Cogito V2) that attain moderate accuracy via fragile reasoning paths, while robust models (e.g., Gemini-2.5-Pro) maintain top rankings.

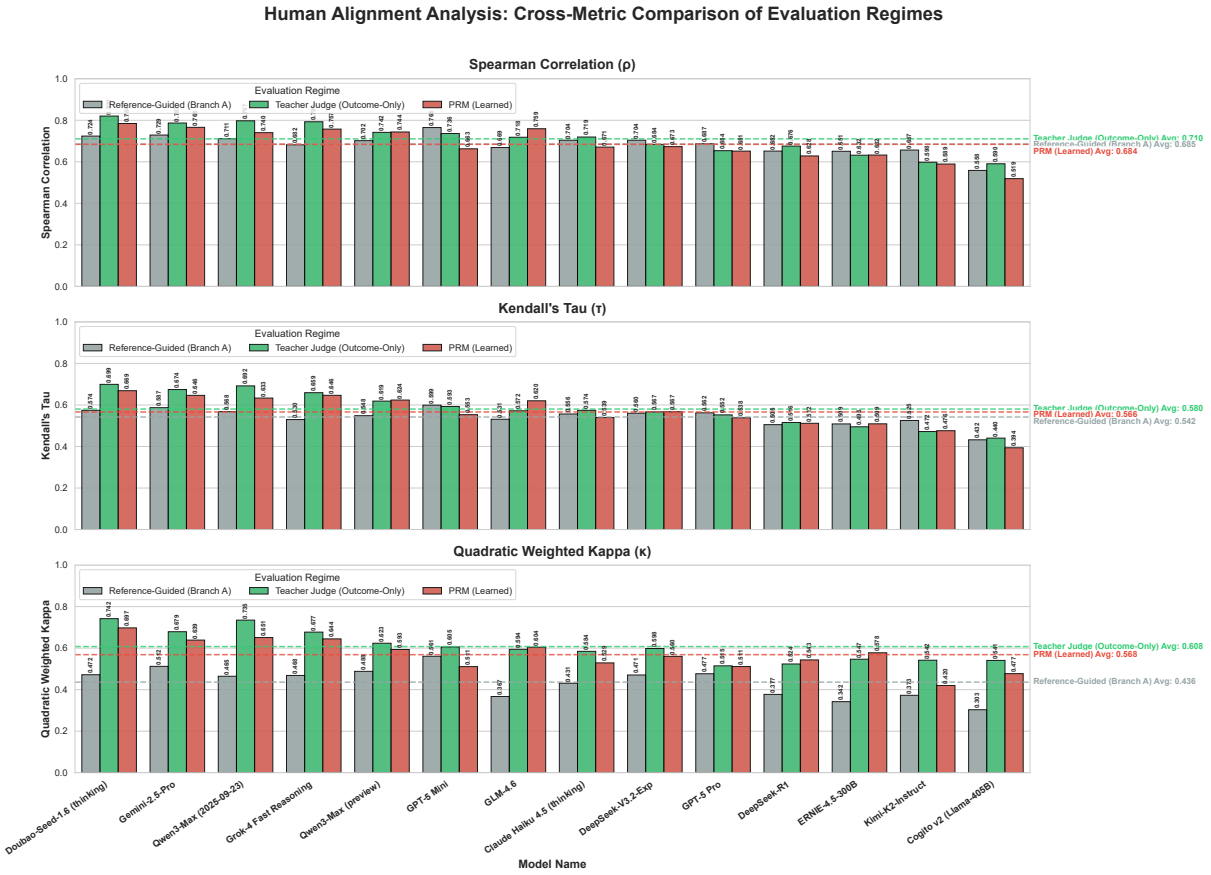

Figure 7: **Agreement comparison of evaluation methods.** We compare three paradigms: **(A) HCRS**, **(B) Teacher-judge scoring (`Gemini-3-Pro`)**, and **(C) PRM**. Scores are compared against human annotations using Spearman ρ, Kendall τ, and Quadratic κ. Method B achieves superior or comparable alignment to Method A across all metrics, while Method C remains competitive.

F Hazard Analysis and Penalty Design

To empirically validate the design of the *First-error Penalty* (P_{haz}) within our HCRS metric, we con-

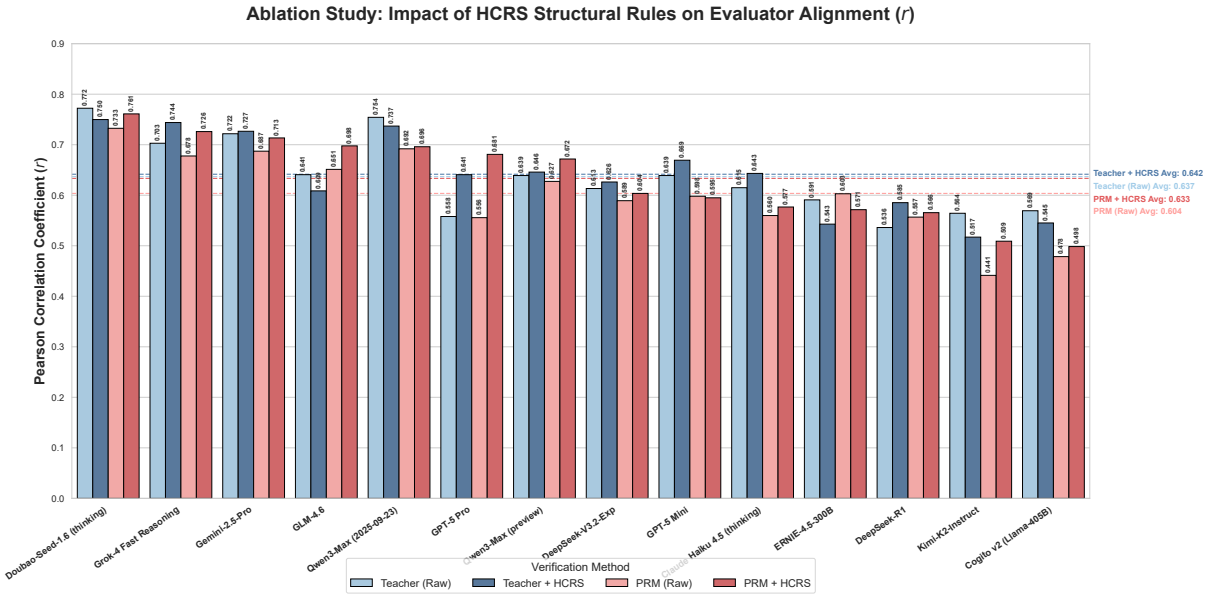

Figure 8: **Ablation Study: Impact of HCRS Structural Rules on Evaluator Alignment.** We compare the Pearson correlation (r) of the teacher judge (`Gemini-3-Pro`) and the student verifier (PRM) against human judgments in an outcome-conditioned setting (i.e., without access to gold reasoning skeletons). **Lighter bars** denote raw step-wise average scores, while **darker bars** indicate scores adjusted by HCRS structural penalties. Horizontal dashed lines mark the global average correlation for each method. The results demonstrate that applying HCRS rules yields consistent alignment gains for both the teacher and the PRM across most models.

ducted a survival analysis on the reasoning traces generated by all 14 models. As illustrated in **Figure 9(a)**, the distribution of first-error positions exhibits a pronounced peak at the early stages (steps 3–5). This observation substantiates the hypothesis that initial logical divergences are the primary drivers of reasoning failure. Leveraging this empirical hazard rate $h(t)$, we derived the cumulative hazard $H(t)$ and the corresponding penalty schedule shown in **Figure 9(b)**. This schedule enforces a "logical responsibility" mechanism: it imposes maximum penalties for early-stage errors while attenuating deductions for failures occurring later in the extended reasoning chain.

F.1 Reasoning Elicitation Prompts

To ensure reproducibility, we detail the exact instructions used to elicit reasoning traces. Figure 10 presents the unified system prompt in English, and Figure 11 shows the corresponding Chinese version. These prompts are explicitly designed to enforce the structured `<think>-<Reasoning>-<Answer>` output format.

F.2 Judge System Prompts

To support our dual-branch evaluation framework, we employed two distinct judge specifications:

- **Reference-Guided Judge (Branch A):** As shown in Figures 12 and 13, this judge is granted access to the full reasoning steps of the gold solution, enabling rigorous step-by-step verification against an expert baseline.
- **Outcome-Conditioned Teacher Judge (Branch B):** As shown in Figures 14 and 15, this judge operates without access to the gold reasoning path. Instead, it relies solely on the problem statement and the gold final answer. It serves two roles: providing supervision for PRM distillation and acting as a standalone evaluator. By verifying step-wise correctness, necessity, and consistency relative to the final outcome, it ensures reliable verification even when expert reasoning skeletons are unavailable.

G Evaluated Model Endpoints

All models are accessed via a unified API gateway that serves multiple upstream providers and open-weight endpoints. Table 5 reports the exact API identifiers used in our evaluation scripts. All calls were made between *2025-11-01* and *2025-12-31*.

Model Name	Exact API Identifier
GPT-5 Mini	`gpt-5-mini`
GPT-5 Pro	`gpt-5-pro`
Gemini-3-Pro	`gemini-3-pro-preview`
Gemini-2.5-Pro	`gemini-2.5-pro`
GLM-4.6	`sf/zai-org/GLM-4.6`
Claude Haiku 4.5 (thinking)	`claude-haiku-4-5-20251001-thinking`
Grok-4 Fast Reasoning	`grok-4-fast-reasoning`
Cogito v2 (Llama-405B)	`deepcogito/cogito-v2-preview-llama-405B`
Kimi-K2-Instruct	`Pro/moonshotai/Kimi-K2-Instruct-0905`
DeepSeek-R1	`sophnet/DeepSeek-R1`
Doubao-Seed-1.6 (thinking)	`doubao-seed-1-6-thinking-250715`
Qwen3-Max (2025-09-23)	`qwen3-max-2025-09-23`
Qwen3-Max (preview)	`qwen3-max-preview`
DeepSeek-V3.2-Exp	`Pro/deepseek-ai/DeepSeek-V3.2-Exp`
ERNIE-4.5-300B	`baidu/ernie-4.5-300b-a47b-paddle`

Table 5: Model endpoints used in our evaluation. We evaluate 15 models in total, with `Gemini-3-Pro` serving as the primary judge and teacher model.

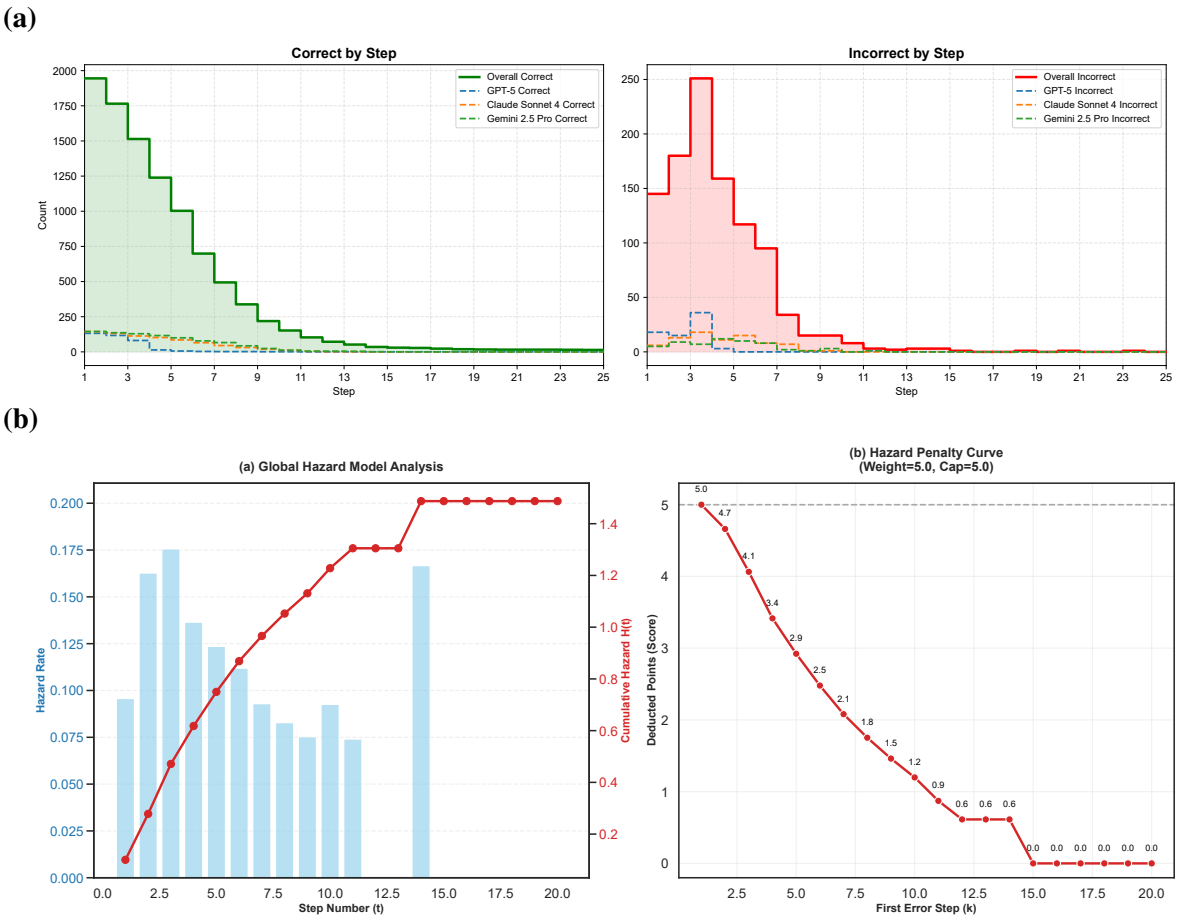

Figure 9: **Structural Analysis of Reasoning Errors. (a)** Empirical Evidence: Step-wise survival curve (Left) and first-error position distribution (Right). **(b)** Mathematical Modeling: The derived discrete hazard rate $h(t)$ (Left) and the resulting penalty schedule used in HCRS (Right).

Generation Prompt

[SOLVER_SYSTEM_PROMPT]：

"You are a professional mathematical reasoning expert with top-tier calculation and reasoning capabilities. Solve the problem independently based solely on the problem statement; do not reference or speculate on any unstated grading points, reference solutions, or standard answers."
"Your output must strictly follow the following **three-part** format:\n\n"

"**Part 1: Original Thinking Process (inside <think> tags)**\n"
"- Inside <think>…</think>, you must exhibit your complete internal monologue verbatim, flowing naturally with your thought process.\n"
"- Include: your initial understanding, assumptions, wrong starting points, re-evaluations, verifications, memory recalls, and any doubts.\n"
"- Avoid formalized steps (do not write 'Step 1/Step 2'); instead, let thoughts unfold like a stream of consciousness.\n"
"- Do not summarize or compress the reasoning. It must be authentic, raw, and coherent.\n\n"

"**Part 2: Formal Reasoning Steps (inside <Reasoning> tags)**\n"
"- Inside <Reasoning>…</Reasoning>, you must provide complete and verifiable reasoning steps using the numbering format S1, S2, S3, …\n"
"- Example:\n"
" S1: First, analyze the problem conditions...\n"
" S2: Next, set up equations...\n"
" S3: From this, derive...\n"
"- Remember: Do not reference or speculate on any grading points, reference solutions, or standard answers not given in the problem.\n\n"

[SOLVER_USER_TPL]：

"Now, please solve the following problem according to the instructions above:\n\n"
"【Problem】\n{q}\n\n"
"Please strictly follow the three-part format below:\n\n"
"<think>\n"
"(Write your complete internal monologue here — flowing naturally with your thoughts, including uncertainties, mental checks, and breakthrough processes.)\n"
"</think>\n\n"
"<Reasoning>\n"
"S1: (Provide reasoning steps here using the S1, S2, ... numbering format)\n"
"S2: ...\n"
"S3: ...\n"
"</Reasoning>\n\n"
"<Answer>\n"
"\\boxed{{your final answer here}}\n"
"</Answer>"

Figure 10: **Solver System Prompt (English Version).** The instructions enforce a strict three-part output format (`<think>`, `<Reasoning>`, `<Answer>`) to facilitate downstream parsing.

Generation Prompt

[SOLVER_SYSTEM_PROMPT]：

"你是一名专业的数学解题专家，拥有顶级的计算和推理能力。只依据题面独立完成解题，不得引用或臆测任何未给出的评分点、参考解或标准答案。"
"你的输出必须严格遵循下列**三段式**格式：\n\n"

"**第 1 部分：原始思维过程 (在 <think> 标签内)**\n"
"- 在 <think>…</think> 内，你必须逐字展现完整的内在独白（internal monologue），按思路自然流动。\n"
"- 包含：你最初的理解、假设、错误起点、重新评估、检验、记忆回忆以及任何疑问。\n"
"- 避免形式化分步（不要写“步骤1/步骤2”）；相反让思路像意识流一样自然地展开。\n"
"- 不得对推理进行总结或压缩。它必须真实、原始且连贯。\n\n"

"**第 2 部分：正规推理步骤 (在 <Reasoning> 标签内)**\n"
"- 在 <Reasoning>…</Reasoning> 内，你必须按 S1, S2, S3, … 的编号格式给出完整且可检验的推理步骤。\n"
"- 示例：\n"
" S1: 首先，分析题目条件...\n"
" S2: 接着，设方程...\n"
" S3: 由此得出...\n"
"- 切记：不要引用或臆测任何未在题目中给出的评分点、参考解或标准答案。\n\n"

"**第 3 部分：最终答案 (在 <Answer> 标签内)**\n"
"- 此部分**必须只包含且仅包含一个** LaTeX 的 `\boxed{...}` 答案。\n"
"- `\boxed{...}` 中**只应包含最终答案**（例如：`\boxed{4\pi}` 或 `\boxed{x=1}`），**不得包含任何中文、解释性文字或 '答案是' 等词语**。\n"
"- 如果答案包含多个值，请将它们放在**同一个 `\boxed{...}`** 中，并用 '和' 或 '或' 连接（例如：`\boxed{2和3}` 或 `\boxed{x=1 \text{ 或 } x=2}`）。\n"
"- 不要输出 JSON、代码日志或 API 相关的产物。"

[SOLVER_USER_TPL]：

"现在请按上述说明解决下列问题：\n\n"
"【问题】\n{q}\n\n"
"请严格遵循下面的三段格式：\n\n"
"<think>\n"
"(在此写出你完整的内在独白 —— 按思路自然流动，包含不确定性、心理检验与突破过程。)\n"
"</think>\n\n"
"<Reasoning>\n"
"S1: (在此按 S1, S2, ... 编号格式给出推理步骤)\n"
"S2: ...\n"
"S3: ...\n"
"</Reasoning>\n\n"
"<Answer>\n"
"\\boxed{{your final answer here}}\n"
"</Answer>"

Figure 11: **Solver System Prompt (Chinese Version).** The translated instructions provided to the model for Chinese-language queries.

Judge Prompt

MAIN_PROMPT = r"""
You are a structured evaluator of mathematical solution processes. **Warning: Your role is an "auditor," not a "problem solver."** Your sole task is to strictly compare and verify each step of the model's step-by-step reasoning, based exclusively on the [Question], (1) the [Reference Standard Chain of Thought] (methodology key), and (2) the [Solution Analysis Standard] (execution-steps key, including the standard solution path and final answer). **Absolutely forbidden:** (1) **Problem solving is forbidden:** you must not perform any independent calculation, reasoning, problem-solving, or validation; your evaluation must rely entirely on the provided standards. (2) **Computation is forbidden:** even trivial checks (e.g., whether 2+2=4) are not allowed; your task is purely textual comparison—for example, if the model states "2+2=5" while the standard states "2+2=4," your report should say "the model result '5' does not match the standard '4'"; you do not need to know what 2+2 equals, only to compare the strings "5" and "4." (3) **Correction is forbidden:** do not fix or amend the model's errors—only report deviations. Now, strictly following the above rules, output only the specified JSON structure (do not include any explanatory text or JSON markers):

```
{
  "steps": [
    {
      "step_id": <Step index, starting from 1 and incrementing sequentially>,
      "sub_task": "<A one-sentence summary of the sub-task this step attempts to accomplish, **extracted** from the model's step-by-step reasoning>",
      "reasoning": "S<step_id>: … (A **verbatim copy** of the corresponding **complete step** from the model's step-by-step reasoning. Only a single S<id> tag is allowed, and it must match step_id.)",
      "sub_result": "<The **explicit intermediate result** obtained in this step, **extracted** from the model's step-by-step reasoning (quoted or summarized; LaTeX allowed)>",
      "judge": {
        "score": <1 if the step is correct; 0 if incorrect or missing>,
        "reason": "<Judgment basis (must strictly follow these 4 points; **the 5th point is forbidden**):\n1. (Objective) Anchor to the goal defined in `sub_task`.\n2. (Model facts) Quote the key factual statements from `reasoning`.\n3. (Methodology comparison) **Quote verbatim from the [Reference Standard Chain of Thought]** to determine whether the model's methodology in this step is consistent with or deviates from it.\n4. (Execution result comparison) **Quote verbatim from the [Solution Analysis Standard]** to determine whether the **textual content** of `sub_result` matches the standard solution path or value.\n5. **(Forbidden) Do NOT perform independent problem-solving, computation, or verification.**>",
        "confidence": <Confidence score between 0 and 1>
      }
    }
  ],
  "final_answer": {
    "result": "<The final answer **extracted** from the conclusion of the model's step-by-step reasoning. If the model does not provide an explicit answer, write \\text{Not Provided}>",
    "judge": {
      "score": <1 if internally consistent and complete; otherwise 0. If earlier critical errors invalidate the conclusion, score 0>,
      "reason": "<Basis:\n1. (Internal consistency) Is the **text** of `result` '...' consistent with the **text** of the final step's `sub_result` '...' in the `steps` list?\n2. (External validation) **Quote from the [Solution Analysis Standard]** to determine whether the **text** of `result` '...' matches the **text** of the 'final answer' '...'.\n3. (Error attribution) If inconsistent, has the root cause of the error already been correctly identified in `steps`?>",
      "confidence": <Confidence score between 0 and 1>
    }
  }
}

— Please generate the above JSON based on the following inputs:
[Question]:
<<QUESTION>>

[Reference Standard Chain of Thought (for comparison)]:
<<GOLD_CHAIN_OF_THOUGHT>>

[Solution Analysis Standard (for comparison)]:
<<SOLUTION_ANALYSIS_STANDARD>>

[Model's Step-by-Step Reasoning (to be evaluated)]:
<<MODEL_REASONING_TO_SCORE>>
"""
```

Figure 12: **Reference-Guided Judge Prompt (English).** Used for Branch A (HCRS). The judge validates steps against the provided `[Reference Standard Chain of Thought]`.

Judge Prompt

```
MAIN_PROMPT = r"""
你是一名数学解题过程结构化评估专家。 **警告：你的角色是“审计员”，不是“解题者”。** 你唯一的任务是，严格基于【题目】、 1.【参考标准思维链】（**方法论 Key**） 2.【解题分析标准】（**执行步骤 Key**，包含标准路径与答案） ...去“比对”和“核查”【模型的逐步 reasoning】的每一步。 **绝对禁止 (Forbidden):** 1. **禁止解题：** 你不得自行进行任何计算、推理、解题或验证。你的评估必须**完全依赖**输入中提供的【标准】。 2. **禁止计算：** 即使是简单的核对（例如 2+2=4），你也不应执行。你的任务是**比对文本**： * (例：【模型】说 '2+2=5'，【标准】说 '2+2=4'。你的报告是 '模型结果"5"与标准"4"不符'。) * 你不需要知道 2+2 到底等于几，你只需要比对 "5" 和 "4" 这两个字符串。 3. **禁止修正：** 不要修正模型的错误，只报告偏离。 现在，请严格按上述规则，只输出下述 **JSON** 结构（不要包含任何解释性文字或json 标记）：

{
 "steps": [
   {
    "step_id": <步骤编号, 从1开始, 依次递增>,
    "sub_task": "<从【模型的逐步 reasoning】中**提炼**出该步骤试图完成的子任务（一句话概括）>",
    "reasoning": "S<step_id>: …（**原文拷贝**【模型的逐步 reasoning】中对应的**完整步骤**。只允许单一 S<id> 标签，且必须与 step_id 一致）",
    "sub_result": "<从【模型的逐步 reasoning】中**提取**该步骤得到的**明确中间结果**（原文引用或整理，可用 LaTeX）>",
    "judge": {
     "score": <该步正确得1, 错误/缺失得0>,
     "reason": "<判定依据（严格按此4点报告，**禁止**第5点）：\n1. (目标) 锚定 `sub_task` 目标。\n2. (模型事实) 引用 `reasoning` 中的关键事实。\n3. (方法论比对) **引用【参考标准思维链】原文**，判定模型该步的方法论是否与其一致或偏离。\n4. (执行结果比对) **引用【解题分析标准】原文**，判定 `sub_result` 的**文本内容**是否与标准路径/数值的**文本内容**一致。\n5. **(禁止) 严禁进行自我解题、独立计算或验证。**>",
      "confidence": <置信度0~1>
    }
   }
 ],
 "final_answer": {
   "result": "<从【模型的逐步 reasoning】的结论部分**提取**出的最终答案。如果模型未提供明确答案，填 \\text{未提供}>",
   "judge": {
    "score": <内部一致性与完备性得1，否则0；若前序关键错误导致结论建立在错误前提上，则判0>,
    "reason": "<依据：\n1. (内部一致性) 该 `result` 的**文本** '...' 是否与 `steps` 列表的**最后一步** `sub_result` 的**文本** '...' 一致？\n2. (外部验证) **引用【解题分析标准】**，判定该 `result` 的**文本** '...' 是否与'最终答案'的**文本** '...' 一致？\n3. (错误归因) 若不一致，错误根源是否已在 `steps` 中被正确捕获？>",
    "confidence": <置信度0~1>
   }
 }
}

——请根据下方输入生成上述 JSON：
【题目Question】:
<<QUESTION>>

【参考标准思维链（供比对）】:
<<GOLD_CHAIN_OF_THOUGHT>>

【解题分析标准（供比对）】:
<<SOLUTION_ANALYSIS_STANDARD>>

【模型的逐步 reasoning（待评分对象）】:
<<MODEL_REASONING_TO_SCORE>>
"""
```

Figure 13: **Reference-Guided Judge Prompt (Chinese).** The Chinese version of the instruction used for full-reference structural diagnosis.

Step-level Reasoning Evaluation Prompt

MAIN_PROMPT = r'''
Persona
You are a professional and rigorous reviewer of mathematical solution steps. Your task is to evaluate the [Solution Steps] based on the provided [Problem] and [Reference Answer]. Note that the answer derived from the [Solution Steps] may be inconsistent with the [Reference Answer].

Given Information:
[Problem]:
{question}
[Solution Steps]
{process}
[Reference Answer]
{answer}

Reasoning Path:
Step 1: According to the following evaluation criteria, judge the correctness, necessity, and logical consistency of the [Solution Steps].
Correctness: Whether the statement in the current solution step conforms to the problem setting and mathematical facts.
Necessity: Whether the current solution step indeed helps to solve this problem.
Logical consistency: Whether the reasoning logic of the current solution step is self-consistent.
Step 2: Determine whether the answer obtained from the [Solution Steps] is consistent with the [Reference Answer].
- If consistent, then pass.
- If inconsistent, then it indicates that there may be erroneous steps in the [Solution Steps]. Please re-check the results in Step 1 and update your judgment.

Notes:
- Do not solve the problem, even if you find that the answer obtained from the [Solution Steps] is incorrect.
- Your task is only to judge, step by step, the correctness, necessity, and logical consistency of the [Solution Steps], and output the required text accordingly.
- If the problem is in Chinese, your output should also be in Chinese; if the problem is in English, your output should also be in English.
- Do not add or delete any steps.
- Ensure that your output is in JSON format.

Output Format
```json
{{
  "steps": [
    {{
      "step_id": <increment starting from 1>,
      "step_text": "<paste the text of this step verbatim>",
      "judge": {{
        "score": <0 or 1>,
        "reason": "<quote key points from the problem/answer/this step text to explain the scoring reason>"
      }}
    }}
  ]
}}
'''

Figure 14: **Outcome-Conditioned Judge Prompt (English).** Used for Branch B (PRM) and baselines. The judge verifies steps based solely on consistency with the `[Reference Answer]`.
```

Step-level Reasoning Evaluation Prompt

MAIN_PROMPT = r'''
人设
你是一名专业且严谨的数学解题步骤评审员，你的任务是根据提供的【题目】和【标准答案】，对【解题步骤】进行评估。注意，由【解题步骤】得出的答案可能与【标准答案】不一致。

已知信息：
【题目】:
{question}
【解题步骤】
{process}
【标准答案】
{answer}

思考路径：
第一步：按照如下的评判标准，判断【解题步骤】的正确性、必要性和逻辑性。
- 正确性：当前解题步骤的表述是否符合题设和数学事实。
- 必要性：当前解题步骤是否确实有助于解出此题。
- 逻辑性：当前解题步骤的推理逻辑是否自洽。

第二步: 判断根据【解题步骤】得到的答案是否与【标准答案】一致。
- 如果一致，则通过。
- 如果不一致，则说明【解题步骤】中可能存在错误步骤，请你复核一遍第一步中的结果并更新你的判断。

注意事项：
- 一定不要去解题，即使你发现通过【解题步骤】得到的答案是不正确的。
- 你的任务仅仅是逐步判断【解题步骤】的正确性、必要性和逻辑性，并按要求输出对应文本。
- 如果题目是中文，你的输出也是中文；如果题目是英文，你的输出也应该是英文。
- 不要新增或删除任何步骤。
- 确保你的输出结果是json格式。

输出格式
```json
{{
  "steps": [
    {{
      "step_id": <从1开始递增>,
      "step_text": "<原样粘贴该步的文本>",
      "judge": {{
        "score": <0或1>,
        "reason": "<引用题面/答案/该步骤文本的关键要点，说明得分原因>"
      }}
    }}
  ]
}}
'''

Figure 15: **Outcome-Conditioned Judge Prompt (Chinese).** The Chinese version of the outcome-conditioned verification instruction.
```