

Are Your LLMs Capable of Stable Reasoning?

Junnan Liu, Hongwei Liu, Linchen Xiao, Ziyi Wang, Kuikun Liu, Songyang Gao
Wenwei Zhang, Songyang Zhang[†], Kai Chen[†]

Shanghai Artificial Intelligence Laboratory

to.liujn@outlook.com, zhangsongyang@pjlab.org.cn

Abstract

The rapid advancement of large language models (LLMs) has shown remarkable progress in complex reasoning tasks. However, a significant disparity exists between benchmark performances and real-world applications. We attribute this gap primarily to current evaluation protocols and metrics, which inadequately capture the full spectrum of LLM capabilities, especially in complex reasoning tasks where both accuracy and consistency are essential. In this paper, we introduce **G-Pass@ k** , a novel evaluation metric that continuously assesses model performance across multiple sampling attempts, quantifying both the model’s performance potential and its stability. Through extensive experiments on various public and newly constructed benchmarks, we employ **G-Pass@ k** in conjunction with state-of-the-art large language models to provide comprehensive insights into their potential capabilities and operational consistency. Our findings reveal a significant opportunity to enhance the realistic reasoning abilities of LLMs, underscoring the necessity for more robust evaluation metrics.

1 Introduction

Since the emergence of large language models (LLMs), complex reasoning, particularly in mathematical problem-solving, has been regarded as the *crown jewel* of LLM capabilities (Beeching et al., 2024; Shao et al., 2024). Numerous models have demonstrated remarkable performance on mathematical tasks, from general-purpose models like GPT-4o (OpenAI, 2024a), and the LLaMA series (AI, 2024) to specialized models such as DeepSeek-Math (Shao et al., 2024) and Qwen-Math (Yang et al., 2024c), which excel in complex and high-difficulty mathematical reasoning. More recently, long-chain-of-thought (Long-CoT) models like OpenAI-o1 (OpenAI, 2024b), QwQ (Team,

2024a), and DeepSeek-R1 (DeepSeek-AI et al., 2024) have further advanced the state-of-the-art in mathematical problem-solving.

In real-world applications, LLMs typically employ sampling with predefined decoding parameters (e.g., temperature, top-k, top-p, and repetition penalty) to maintain response diversity. Users often regenerate responses or initiate new sessions until obtaining satisfactory answers to specific questions. However, conventional evaluation metrics for LLMs, such as Greedy Accuracy, Pass@ k (Chen et al., 2021), Best-of-N (BoN), and Majority Voting, demonstrate significant limitations in measuring real-world performance, particularly regarding long-term consistency. While these metrics effectively capture either instantaneous accuracy or peak performance across multiple samples, they inadequately reflect **output stability**. Such instability poses significant challenges for applications requiring reliable and predictable outcomes, highlighting the need for evaluation metrics that effectively balance response diversity with consistent performance and comprehensive analysis of the reasoning stability of current LLMs.

To address these challenges, we introduce **G-Pass@ k** , a novel evaluation metric that simultaneously assesses both reasoning capability and performance consistency. The core principle of **G-Pass@ k** lies in evaluating model performance under varying thresholds of correctness, providing a nuanced understanding of model behavior across different stringency levels. By integrating measures of both stability and potential, **G-Pass@ k** delivers a comprehensive assessment of an LLM’s capabilities, particularly valuable for complex reasoning tasks.

To thoroughly study the reasoning stability of current LLMs using **G-Pass@ k** , we conduct comprehensive experiments to evaluate various models on multiple mathematical reasoning benchmarks, encompassing different categories of large

[†]Corresponding authors.

language models. These include general LLMs, mathematics-specialized models, and Chain-of-Thought (CoT) enhanced reasoning models, which are known for their step-by-step reasoning capabilities in complex mathematical problem-solving. We also introduce LiveMathBench, a challenging bilingual mathematics benchmark assembled from multiple competitions and examinations to mitigate the impact of data leakage.

Through systematic comparison between traditional metrics (Greedy Accuracy, Pass@ k) and G-Pass@ k , our findings reveal distinctive insights into model performance that conventional evaluation approaches may overlook and demonstrate significant limitations in traditional evaluation metrics regarding output stability assessment. We observe a substantial performance degradation across all models as the G-Pass@ k threshold becomes more stringent, a pattern consistent across both established open-source benchmarks (MATH (Lightman et al., 2024) and AIME (AIME2024, 2024; AIME2025, 2025)) and our proposed LiveMathBench. Specifically, the key observations include:

- **Instability in LLM Reasoning** (Section 3.2, Section 3.5): Both closed-source and open-source models exhibit significant instability when applied to challenging reasoning tasks. Performance drops exceed 50% in many cases, with the worst instances showing declines of up to 90%. This issue is particularly significant in high-difficulty questions, underscoring the necessity for further enhancement of current LLMs’ ability to address complex questions.
- **Inconsistency Between Expanded Model Scale and Enhanced Reasoning Ability** (Section 3.2): Merely scaling up model size does not necessarily enhance stable reasoning capabilities. The expected improvements in performance and stability are not consistently observed.
- **Discrepancy Between Potential and Stability** (Section 3.2): There is a noticeable gap between the models’ potential capabilities, as measured by their G-Pass@ $k_{\tau \rightarrow 0}$, and their actual stability, reflected in G-Pass@ k_{τ} . This disparity highlights the need for further research into developing methods that can better harness the theoretical capabilities of these models in practical, stable applications.

We also investigate the robustness of the newly proposed metric G-Pass@ k , demonstrating that

G-Pass@ k consistently evaluates model performance across different hyperparameters, guaranteeing its reliability and applicability (Section 3.3, Section 3.4). Furthermore, we conduct a preliminary investigation into enhancing the reasoning stability of the model and identifying the reasons behind strong stability. Our investigation reveals two key findings: 1) merely relying on SFT does not achieve significant improvement in reasoning stability (Section 4.1), and 2) the incorporation of backtracking and self-reflection mechanisms in o1-like models improves the stability of their reasoning capabilities (Section 4.2).

2 Generalized Metric for LLM Reasoning

2.1 Preliminary: Pass@ k

Pass@ k was initially proposed to evaluate the functional correctness of code generated by models (Kulal et al., 2019; Chen et al., 2021). With the growing application of large language models across diverse reasoning tasks (Rajani et al., 2019; Imani et al., 2023; Giadikiaroglou et al., 2024), the Pass@ k metric has gained increasing recognition (Luo et al., 2023; Yu et al., 2024). It provides an effective measure of a model’s **potential performance** in solving complex questions. Pass@ k represents the probability of generating **at least one correct solution** within k attempts for each question, as defined by the formula:

$$\text{Pass@}k = \mathbb{E}_{\text{Questions}} \left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}} \right], \quad (1)$$

where n denotes the total number of generations in the reasoning task, and c is the number of correct solutions among them. Intuitively, Equation (1) calculates the expected proportion of questions for which at least one correct solution is obtained. In practice, n is typically set equal to k , primarily to minimize inference costs.

2.2 Generalized Metric: G-Pass@ k

While Pass@ k provides an indication of a model’s performance potential, it does not consider the stability of the model’s reasoning across diverse attempts. To evaluate both the potential and stability of a model, we propose a generalized metric called **G-Pass@ k** . Simply put, G-Pass@ k assesses the stability of a reasoning model by measuring the probability of consistently generating correct solutions across all attempts.

Definition of G-Pass@ k and G-Pass@ k_τ . Let p^* denote the latent success probability of a model generating correct solutions. Assuming each generation is independent and identically distributed (i.i.d.), the probability of obtaining m correct solutions follows a *binomial distribution*:

$$m \sim B(n, p^*). \quad (2)$$

Since p^* is typically inaccessible, we use a *hypergeometric distribution* to approximate the *binomial distribution*:

$$\lim_{n \rightarrow \infty} H(m; k, c, n) \rightarrow B(m; n, p^*). \quad (3)$$

Therefore, G-Pass@ k can be defined as (Yao et al., 2024):

$$\text{G-Pass@}k = \mathbb{E}_{\text{Questions}} \left[\frac{\binom{c}{k}}{\binom{n}{k}} \right]. \quad (4)$$

Here, n denotes the total number of generations per question, and c signifies the number of generations that yield correct solutions. Considering the stringent requirement in Equation (4), we draw inspiration from the mean Average Precision (mAP) metric (Everingham et al., 2010) in the object detection task to introduce a tolerance threshold $\tau \in (0.0, 1.0]$, leading to the definition of G-Pass@ k_τ :

$$\text{G-Pass@}k_\tau = \mathbb{E}_{\text{Questions}} \left[\sum_{j=\lceil \tau \cdot k \rceil}^c \frac{\binom{c}{j} \cdot \binom{n-c}{k-j}}{\binom{n}{k}} \right], \quad (5)$$

where $\lceil \tau \cdot k \rceil$ denotes the smallest integer greater than or equal to $\tau \cdot k$. Conceptually, for $\tau < 1.0$, there is flexibility to allow up to $k - \lceil \tau \cdot k \rceil$ incorrect solutions within the k generations. In conclusion, we leverage the hypergeometric distribution for sampling without replacement to approximate the binomial distribution for sampling with replacement. Such approximations tend to have smaller errors when n is sufficiently large ($k \ll n$). Further details are provided in Appendix B.

Pass@ k as a Special Case of G-Pass@ k_τ .

Pass@ k can be regarded as a special case of G-Pass@ k_τ , as formalized in the following theorem:

Theorem 2.1. *Pass@ k is a special case of G-Pass@ k as τ approaches 0, formally expressed as:*

$$\lim_{\tau \rightarrow 0} \sum_{j=\lceil \tau \cdot k \rceil}^c \frac{\binom{c}{j} \cdot \binom{n-c}{k-j}}{\binom{n}{k}} = 1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}. \quad (6)$$

The proof is provided in Appendix C.

Definition of mG-Pass@ k . Recall that when the threshold τ is low, G-Pass@ k_τ primarily reflects the model’s performance potential. Conversely, at higher τ values, G-Pass@ k_τ evaluates the model’s stability, *i.e.*, its level of mastery over the question. Thus, G-Pass@ k_τ enables the continuous assessment of both performance potential and stability. We further define **mG-Pass@ k** as follows:

$$\begin{aligned} \text{mG-Pass@}k_\tau &= 2 \int_{0.5}^{1.0} \text{G-Pass@}k_\tau d\tau \\ &= \frac{2}{k} \sum_{i=\lceil 0.5 \cdot k \rceil + 1}^k \text{G-Pass@}k_{\frac{i}{k}}. \end{aligned} \quad (7)$$

Intuitively, mG-Pass@ k provides an interpolated estimate of the area under the curve of G-Pass@ k_τ for $\tau \in [0.5, 1.0]$, serving as a comprehensive metric that integrates all G-Pass@ k_τ values within a certain range. For models that are both optimal and stable, the mG-Pass@ k value should approach 1.

3 Stability of LLM Reasoning

In this section, we perform extensive experiments on the mathematical reasoning task to evaluate and analyze the reasoning stability of current LLMs, given the prevalence and ease of verification of the mathematical reasoning task.

3.1 Setup

3.1.1 LLMs

We evaluate various LLMs recognized for their strong mathematical reasoning capabilities, including three categories: general LLMs, mathematical LLMs, and o1-like LLMs. For the details, please refer to Appendix E.

3.1.2 Data

In all experiments, we involve several benchmarks, which include the newly constructed benchmark named **LiveMathBench** and publicly available benchmarks, MATH500-L5, AIME2024-45, and AIME2025. For more detailed information about benchmarks, please refer to Appendix F.

3.2 Main Performance

Table 1 demonstrates the performance on LiveMathBench and Table 2 demonstrates the performance on MATH500-L5 and AIME2024-45. More experimental results can be found in Appendix I.

Table 1: Performance of models on LiveMathBench. We perform 48 runs and report results of greedy accuracy, and G-Pass@16_{0.5,0.75,1.0} and mG-Pass@16. A more detailed performance can be found in Table 6 at Appendix I.1.

LLMs	Greedy	G-Pass@16 (Equation (5)) / %			
		G-Pass@16 _{0.5}	G-Pass@16 _{0.75}	G-Pass@16 _{1.0}	mG-Pass@16
General LLMs					
Llama-3.1-8B-Instruct	24.0	18.2	11.3	4.5	10.4
Yi-1.5-34B-Chat	24.8	18.6	11.3	6.0	11.0
Gemma-2-27b-it	26.9	23.5	17.8	12.7	17.3
Llama-3.1-70B-Instruct	29.8	30.0	22.2	12.5	20.8
Qwen2.5-7B-Instruct	37.0	36.5	27.2	16.0	25.8
DeepSeek-V2.5-1210	38.7	38.9	27.9	17.3	26.7
Llama-3.3-70B-Instruct	40.3	36.2	28.9	19.1	27.5
GPT-4o-2024-11-20 [†]	44.8	41.9	32.9	22.2	31.6
InternLM3-8B-Instruct	44.5	43.0	35.4	23.0	33.6
Claude-3.5-Sonnet [†]	46.7	44.1	36.2	26.6	35.3
Mistral-Large-Instruct-2411	41.6	39.4	37.1	32.9	36.4
Qwen2.5-72B-Instruct	51.7	47.3	39.6	29.0	37.8
Qwen2.5-32B-Instruct	50.8	48.3	39.5	28.6	38.1
Gemini-1.5-Pro-Latest [†]	59.1	55.9	47.3	31.0	44.3
DeepSeek-V3.0-Chat [†]	55.0	59.5	49.9	35.0	47.9
Mathematical Reasoning LLMs					
DeepSeek-Math-7B-RL	23.5	19.8	14.0	9.7	13.7
NuminaMath-72B-CoT	40.8	34.0	27.1	14.2	25.0
Qwen2.5-Math-7B-Instruct	44.1	44.1	38.3	28.1	36.6
Qwen2.5-Math-72B-Instruct	57.6	52.7	45.4	27.9	42.3
O1-like Reasoning LLMs					
Skywork-o1	45.4	39.3	31.9	21.7	30.4
QwQ-32B-Preview	72.7	74.9	65.8	40.1	61.2
OpenAI o1-mini [†]	74.1 [‡]	76.3	67.3	48.3	64.8
DeepSeek Distill Qwen-32B	67.7	81.2	72.3	54.5	69.7
DeepSeek Distill Llama-70B	74.8	80.8	73.0	53.0	69.7
DeepSeek R1 [†]	81.1	83.6	79.1	69.5	77.6

[†] API-based LLMs.

[‡] OpenAI o1 series model does not provide an optional temperature parameter, so we chose the average accuracy of 20 generations as the proxy for greedy accuracy.

From the results, we derive the following observations:

1) Reasoning Ability Still Needs to be Properly Evaluated. While most models demonstrate relatively strong performance in terms of Greedy Accuracy and Pass@16, their effectiveness significantly declines when assessed using the G-Pass@ k metric. Specifically, when τ is set to 1.0, indicating that the model must generate accurate solutions in all 16 attempts, almost all models experience a drastic decline in performance. Notably, even the most robust model in Table 1, DeepSeek R1, displays a 14.3% decline, diminishing from 81.1% to 69.5%. Even when τ is relaxed to 0.5, which requires only half of the generated solutions to be correct, General LLMs, Mathematical Reasoning LLMs, and

o1-like Reasoning LLMs still experience a decrease in performance. This indicates that, under demanding conditions, most models struggle to maintain consistency in their reasoning abilities across multiple samples, irrespective of whether the criteria are strict or lenient. These findings underscore the necessity for more rigorous evaluations of models' reasoning capabilities, particularly in scenarios that require consistent and reliable performance across multiple instances. The current evaluation metrics, often reliant on single-shot greedy decoding, may not fully capture the real robustness and stability of these models in real-world applications.

2) Increasing Model Size May Not Significantly Enhance Robustness. A comparison of models within the same series, such as Qwen2.5-32B-

Table 2: Performance of models on MATH500 and AIME2024. Aligning with experiments on LiveMathBench, we also perform 48 runs and report results of greedy accuracy, $G\text{-Pass}@16_{\{0.5,0.75,1.0\}}$, and $mG\text{-Pass}@16$. More detailed results are available in Table 8 at Appendix I.3.

LLMs	Greedy	$G\text{-Pass}@16$ (Equation (5)) / %			
		$G\text{-Pass}@16_{0.5}$	$G\text{-Pass}@16_{0.75}$	$G\text{-Pass}@16_{1.0}$	$mG\text{-Pass}@16$
MATH500-L5					
Qwen2.5-7B-Instruct	56.0	54.8	43.3	28.0	41.5
Llama-3.3-70B-Instruct	54.5	55.4	49.5	35.0	47.3
Mistral-Large-Instruct-2411	55.2	52.3	51.2	45.6	50.1
Qwen2.5-72B-Instruct	63.4	62.5	54.4	44.9	53.1
Qwen2.5-32B-Instruct	64.2	66.6	59.4	41.0	55.6
Qwen2.5-Math-7B-Instruct	65.7	65.0	62.2	57.6	61.5
Qwen2.5-Math-72B-Instruct	71.6	64.9	59.4	46.0	57.6
Skywork-o1	61.2	56.5	52.2	42.9	50.7
QwQ-32B-Preview	83.6	87.2	78.8	57.4	75.6
DeepSeek Distill Qwen-32B	83.6	89.9	83.8	70.4	81.9
AIME2024-45					
Qwen2.5-32B-Instruct	11.1	7.1	3.4	2.2	3.7
Qwen2.5-7B-Instruct	11.1	8.9	8.1	4.7	7.5
Mistral-Large-Instruct-2411	13.3	10.4	6.8	2.4	6.1
Qwen2.5-72B-Instruct	13.3	13.7	12.9	7.5	11.7
Llama-3.3-70B-Instruct	22.2	25.3	18.2	6.9	16.4
Qwen2.5-Math-7B-Instruct	11.1	4.6	2.6	2.2	2.8
Qwen2.5-Math-72B-Instruct	20.0	18.7	16.2	6.7	14.1
Skywork-o1	11.1	11.2	10.3	1.5	8.2
QwQ-32B-Preview	44.4	41.0	28.6	8.1	24.7
OpenAI o1-mini [†]	60.3 [‡]	62.2	53.3	15.6	43.1
DeepSeek Distill Qwen-32B	62.2	77.0	66.5	31.3	59.3

[†] API-based LLMs.

[‡] OpenAI o1 series model does not provide an optional temperature parameter, so we chose the average accuracy of 20 generations as greedy accuracy.

Instruct and Qwen2.5-72B-Instruct, reveals that despite a more than twofold difference in model size, their performance is similar across various metrics and datasets. For example, on both LiveMathBench and existing open-source datasets, the difference in *Greedy Accuracy* and $mG\text{-Pass}@k$ between these two models is within two percentage points. Additionally, in the larger LLMs like Mistral-Large-Instruct-2411 (123B), although the scale has increased further, performance and stability have actually declined compared to Qwen2.5-72B-Instruct. This suggests that for certain tasks, particularly those requiring deep understanding and logical reasoning, mere parameter expansion may not yield the expected gains in performance or stability. Conversely, the current base model still holds untapped potential, and improved training paradigms or test-time scaling methods should be proposed to enhance its reasoning performance and ability, rather than merely increasing the model

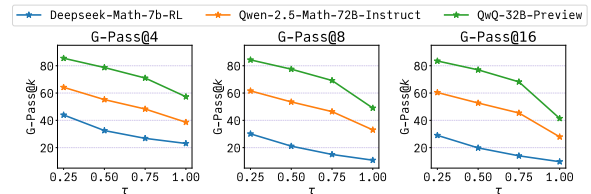


Figure 1: Illustration of $G\text{-Pass}@k$ w.r.t. different values of k , where $k = \{4, 8, 16\}$ on MATH500-L5.

size (Snell et al., 2024).

3) Significant Gap Between Theoretical Performance Potential and Actual Stability. In evaluating model performance, we observe a notable gap between the theoretical upper limit ($G\text{-Pass}@16_{\tau \rightarrow 0}$), the actual performance (*Greedy Accuracy*), and the stability across multiple samples ($G\text{-Pass}@16_{\tau=1.0}$). As evident from the main performance, while models theoretically possess high potential performance, their actual performance in practical applications falls short of this optimal

Table 3: Performance on CCEE and WLPMC. The table shows the decreasing trend of **Greedy** *w.r.t.* **Pass@16** and **G-Pass@16_{1.0}** *w.r.t.* **Greedy**, which are marked with colors of different transparency.

LLMs	CCEE			WLPMC		
	G-Pass@16 _{→0}	↘ Greedy	↘ G-Pass@16 _{1.0}	G-Pass@16 _{→0}	↘ Greedy	↘ G-Pass@16 _{1.0}
Llama-3.3-70B-Instruct	75.9	59.0 _{↓22.2}	35.2 _{↓40.3}	42.1	9.1 _{↓78.3}	~0.0 _{↓100.0}
Mistral-Large-Instruct-2411	71.5	63.6 _{↓11.0}	52.3 _{↓17.7}	18.2	9.1 _{↓49.8}	6.1 _{↓32.9}
DeepSeek-V3.0-Chat	84.6	68.2 _{↓19.3}	53.3 _{↓21.8}	56.8	18.2 _{↓68.0}	6.6 _{↓63.7}
Qwen2.5-72B-Instruct	80.3	72.7 _{↓9.4}	56.9 _{↓21.7}	50.4	18.2 _{↓63.9}	4.1 _{↓77.5}
Gemini-1.5-Pro-Latest	81.8	68.2 _{↓16.6}	53.1 _{↓22.1}	60.0	36.4 _{↓40.0}	4.3 _{↓88.2}
GPT-4o	79.4	61.7 _{↓22.3}	41.8 _{↓32.3}	29.9	18.2 _{↓39.1}	4.0 _{↓78.0}
Qwen2.5-Math-7B-Instruct	80.5	63.6 _{↓21.0}	49.3 _{↓22.5}	43.8	18.2 _{↓58.5}	0.7 _{↓96.2}
Qwen2.5-Math-72B-Instruct	84.1	77.3 _{↓8.0}	53.8 _{↓30.4}	47.0	27.3 _{↓41.9}	18.2 _{↓33.3}
QwQ-32B-Preview	92.3	86.4 _{↓6.3}	55.0 _{↓36.3}	88.4	27.3 _{↓69.1}	11.7 _{↓66.9}

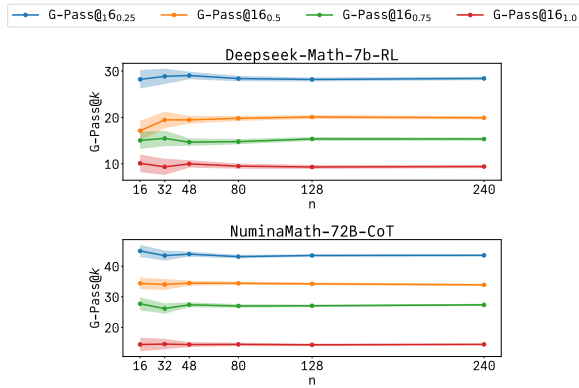


Figure 2: Illustration of G-Pass@ k *w.r.t.* different values of n for DeepSeek-Math-7b-RL and NuminaMath-72B-CoT on LiveMathBench.

level, particularly in terms of output stability.

3.3 Performance *w.r.t.* G-Pass@ k_τ Settings

Performance *w.r.t.* k . Figure 1 presents the results of selected models for G-Pass@4, G-Pass@8, and G-Pass@16. From the experiments, G-Pass@ k can achieve consistent evaluation results under different k values. In addition, for advanced reasoning models with strong performance, a larger value of k has better differentiation.

Performance *w.r.t.* n . As previously noted, the number of attempts n is crucial for the accuracy of the estimation. We select two models, DeepSeek-Math-7b-RL and NuminaMath-72B-CoT, to conduct experiments with $n = \{16\} \times \{1, 2, 3, 5, 8, 15\} = \{16, 32, 48, 128, 240\}$, and report G-Pass@16 $_\tau$. The results are illustrated in Figure 2. When n is small, the estimation deviation is large, as shown by the significant fluctuations in the G-Pass@16 $_\tau$ values for both models. Conversely, for larger n , G-Pass@16 $_\tau$ tends to stabilize, indicating a more consistent and reliable performance. Empirically, we recommend mak-

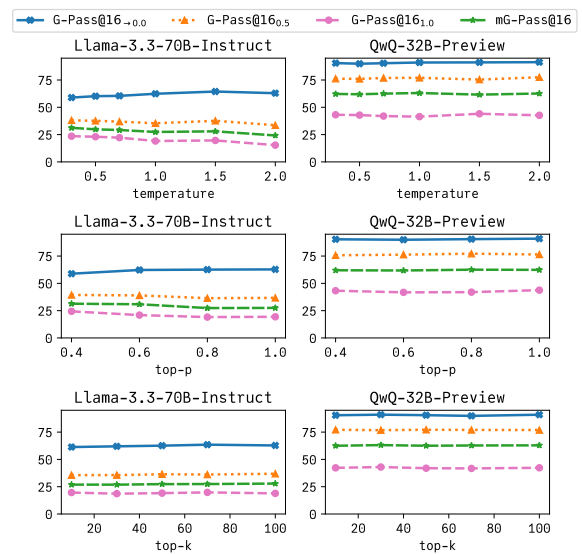


Figure 3: G-Pass@ k_τ performance of LLMs *w.r.t.* sampling parameters, *i.e.*, top- p , top- k , and temperature.

ing at least $n = 3k$ generations when calculating G-Pass@ k to ensure estimation accuracy.

3.4 Impact of Sampling Parameters

The configuration of sampling parameters influences the diversity of model generation and, consequently, its reasoning stability. To investigate the effects of various sampling parameters, we evaluate models across different combinations of these parameters. Specifically, we focus on 3 key parameters: *temperature*, *top-p*, and *top-k*. For each evaluation, we adjust one parameter while holding the others constant. Figure 3 illustrates the experimental results for the Llama-3.3-70B-Instruct, Qwen2.5-72B-Instruct, and QwQ-32B-Preview models. Additional details can be found in Appendix I.4, Appendix I.5, and Appendix I.6. Our findings can be summarized as follows:

1) G-Pass@ k_τ Demonstrates Robustness Across Sampling Parameters. The proposed metric, G-Pass@ k_τ , demonstrates stability across various models under different sampling parameters. This indicates that G-Pass@ k_τ serves as a robust metric, accurately reflecting model performance.

2) Sensitivity of Different Models to Sampling Parameters Varies Greatly. Experimental results show that some models, including Llama-3.3-70B-Instruct, Mistral-Large-Instruct-2411, and Qwen2.5-72B-Instruct, maintain stable performance across various sampling parameters. We hypothesize that models with more parameters exhibit greater resistance to perturbations after adequate pre-training. In contrast, certain models, such as Qwen2.5-Math-72B-Instruct, experience significant performance degradation under atypical sampling parameters. This degradation may be attributed to the effects of post-training, making these models more susceptible to perturbations.

3) Long CoT Can Help to Enhance the Robustness of the Model. An intriguing observation is that several o1-like models exhibit considerable robustness to sampled parameters. For instance, the performance of QwQ remains consistently stable across all experiments. We believe that long CoT reasoning aids the model in rectifying errors introduced by random sampling. This finding underscores the importance of backtracking and self-reflection in addressing complex reasoning problems.

3.5 Impact of Question Difficulty

We also examine the performance of models with respect to questions of varying difficulty levels. We analyze CCEE and WLPMC splits from LiveMathBench. CCEE is a college entrance examination that primarily involves fundamental high school mathematics knowledge, whereas WLPMC is a prestigious collegiate mathematics competition that presents significantly greater challenges. Table 3 shows the experimental results. The findings indicate that models struggle more with generalizing this ability to challenging questions. The sharp decline in performance on WLPMC, compared to CCEE, suggests a higher level of difficulty models face when grasping these questions. For instance, on WLPMC, the Greedy Accuracy of QwQ-32B-Preview declines by 69.1% compared to its Pass@16 performance. In contrast, on CCEE, the Greedy Accuracy of QwQ-32B-Preview only

declines by 6.3% compared to its Pass@16 performance.

From the observation, we conjecture that models tend to learn superficial patterns from training data, which is reflected in the marked improvement of the Pass@ k metric. However, this increase does not necessarily translate into an enhancement of the model’s real reasoning capabilities.

4 Enhancing the Stability of Reasoning

In this section, we aim to explore factors that can improve the reasoning stability. Initially, we assess whether straightforward supervised fine-tuning on specific corpora enhances reasoning stability. Subsequently, we explore the reasons for the pronounced stability of o1-like models, positing that the characteristics of their reasoning paths contribute to this improved stability.

4.1 Can SFT Enhance Model Stability? An Empirical Analysis of Data Contamination

Currently, most large language models are trained on massive pre-training corpora which may lead to data contamination. Data contamination arises when the test data is mixed into training data, also referred to as data leakage (Dickson, 2024; Dong et al., 2024). Additionally, supervised fine-tuning on specific datasets is a common post-training method to enhance model performance. In this section, we explore whether training a model on a particular corpus significantly affects its reasoning stability.

To investigate the effect of SFT, we train the Qwen2.5-7B model on a previously unseen corpus and evaluate its reasoning performance and stability throughout the training process to ascertain whether mere SFT (which we also refer to as overfitting or contamination) significantly impacts reasoning stability. To be specific, the training process begins with a base set of 200,000 randomly sampled instructions from the Numina-Math-CoT corpus (LI et al., 2024), which serves as the uncontaminated training set. Subsequently, we introduce incremental rounds of data contamination, consisting of 0, 6, 8, 10, and 16 rounds, where a round of 0 indicates the absence of contamination, i.e., training exclusively on the original NuminaMath data. The model’s efficacy was assessed across these five conditions, as illustrated in Figure 4.

Despite the observed increase in greedy score

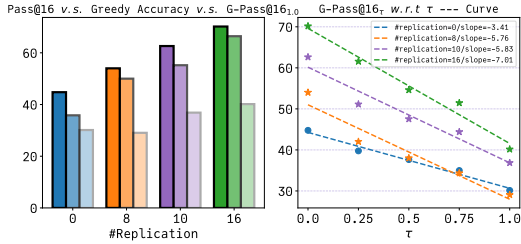


Figure 4: The data contamination experiment involves different contamination rounds, where $\#Replication$ represents the number of these rounds. The term $Slope$ denotes the slope value of the $G-Pass@16_{\tau}$ curve with respect to τ .

with escalating rounds of contamination, the stability, as quantified by the $G-Pass@k$ metric, did not exhibit a corresponding enhancement. Specifically, in Figure 4 left part, the disparity between actual performance (Greedy Accuracy) and stability across multiple samples ($G-Pass@k@16_{\tau=1.0}$) at each contamination round—6, 8, 10, and 16—was 22, 20, 18, and 26, respectively. In contrast, this gap for a non-contaminated model was only 5, which indicates that the discrepancy between performance and stability in contaminated models is more than three times greater. Additionally, as shown in Figure 4 right part, the slope becomes increasingly steep as the rounds of contamination increase, showing a deterioration in model stability with each additional round of contamination. This phenomenon is particularly significant in certain downstream training scenarios where overfitting becomes necessary, such as in contexts characterized by data scarcity. In these cases, while an increase in greedy accuracy might be achieved, it often comes at the cost of reduced stability. Notably, the aforementioned performance-stability gap may not necessarily narrow proportionally with increased levels of contamination.

From the experimental results, we can contend that simple SFT does not significantly enhance the model’s reasoning stability, or that LLMs are more inclined to memorize during SFT, rather than improving genuine reasoning ability (Chu et al., 2025).

4.2 Mechanistic Analysis of Long CoT in Enhancing Model Stability

From the experimental results presented in Table 1 and Table 2, it is evident that o1-like LLMs demonstrate greater stability when compared to general LLMs (e.g., Llama-3.1-8B-Instruct & Skywork-o1, and Qwen2.5-32B-Instruct & QwQ-32B-Preview).

In this section, we explore the reasons behind the superior inference performance and stability of o1-like LLMs.

We select Qwen2.5-32B-Instruct and QwQ-32B-Preview as the foundational models. Our first observation is that o1-like LLMs significantly surpass general LLMs in terms of the length of reasoning paths. This leads us to hypothesize whether extending the reasoning path length of general LLMs can enhance their reasoning performance and stability. To verify this hypothesis, we designed the following experiments: 1) leverage Qwen2.5-32B-Instruct to perform parallel reasoning m times, treating m reasoning paths combined as a single reasoning process; and 2) leverage Qwen2.5-32B-Instruct to execute serial reasoning m times, inserting a break-link (e.g., *Wait, let’s re-evaluate the solution steps.*) between each reasoning path. We subsequently compared the performance of these two enhanced reasoning models with that of QwQ-32B-Preview, and the results are illustrated in Figure 5. We observe that: 1) while parallel reasoning in general LLMs, combined with majority voting, can improve reasoning performance and stability to some extent, this improvement is limited. Such performance and stability remain significantly inferior to those of o1-like LLMs when considering an equivalent number of reasoning tokens at the same scale. While parallel reasoning is anticipated to outperform certain models with more parameters as the number of tokens rises, it unavoidably incurs substantial reasoning overhead. 2) a robust backbone model possesses the capability to reflect and backtrack. Introducing specialized break-links can enhance this capability to some extent, thereby improving the performance and stability of reasoning. However, this capability requires further stimulation compared to models similar to o1-like LLMs.

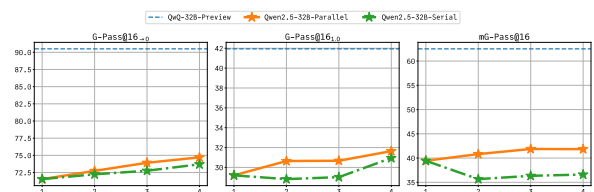


Figure 5: Comparison between Qwen2.5-32B-Instruct and QwQ-32B-Preview.

Therefore, we propose a preliminary hypothesis: **o1-like LLMs exhibit superior reasoning performance and stability due to their reflective and backtracking capabilities, which enable them to**

correct errors in the reasoning path. The reasoning path can be conceptualized as a gradient descent process in which general LLMs may easily become trapped in local optima when optimizing in a single direction. In contrast, o1-like LLMs, through their backtracking and reflection abilities, can escape local optima and more effectively identify global optima.

To validate our hypothesis, we investigate the hidden states of LLMs during the reasoning process, which reflects the decision process of LLMs (Ren et al., 2023; Wang et al., 2024). Specifically, we utilize the CoE-Score to quantitatively analyze the variation tendencies of hidden states across different layers of LLMs. For detailed information, please refer to the original paper by Wang et al. (2024). We choose several questions from LiveMathBench and prompted Qwen2.5-32B-Instruct and QwQ-32B-Preview to perform multiple reasoning. Subsequently, we gather all candidate answers generated by Qwen2.5-32B-Instruct and QwQ-32B-Preview and calculate the CoE-Scores with respect to candidate answers at each position of the reasoning paths. The curves are depicted in Figure 11, Figure 12, Figure 13, and Figure 14. These figures reveal that o1-like LLMs exhibit multiple decision changes during the reasoning process, as indicated by frequent peaks and troughs in their CoE-Score curves. In contrast, the decision process of general LLMs is more linear, resulting in a notably flat CoE-Score curve. This linearity makes general LLMs more susceptible to noise and incorrect intermediate steps, thereby reducing their inference performance and stability.

5 Related Work

Stability of LLM Reasoning. Large language models (LLMs) exhibit remarkable performance in reasoning tasks, encompassing question answering, programming, and mathematical problem-solving. Despite their prowess, the output stability of LLMs poses a significant challenge, whereby the model’s outputs can vary for the same input due to random sampling or hallucinations, impacting the model’s reliability and predictability (Gupta et al., 2024; Xu et al., 2024; Atil et al., 2024; Zhuo et al., 2024). Atil et al. (2024) introduced two new metrics: TARr@N for the total agreement rate at N runs over raw output and TARa@N for total agreement over parsed-out answers. However, TARr@N and TARa@N focus solely on measuring output

consistency, our work introduces a novel evaluation metric $G\text{-Pass}@k$ for evaluating the mathematical reasoning proficiency of LLMs. This metric aims to assess the model’s true reasoning ability by not only considering output consistency but also emphasizing correctness. Additionally, Yao et al. (2024) propose pass^k , an approach to assess the reliability and consistency of real-world agent tasks, sharing the same formulation as $G\text{-Pass}@k$. We relax the overly restrictive conditions of $G\text{-Pass}@k$ and further introduce $G\text{-Pass}@k_\tau$ and $mG\text{-Pass}@k$. In comparison to the original $G\text{-Pass}@k$ and pass^k , our methods offer enhanced flexibility and more comprehensive evaluation capabilities.

Mathematical Reasoning Benchmarks for LLMs. The assessment of large language models (LLMs) in mathematical reasoning has led to the development of specialized benchmarks focusing on different aspects of an LLM’s mathematical proficiency. GSM8K (Cobbe et al., 2021) presents a dataset composed of elementary-level math word problems, segregated into training and testing sets, that demand multi-step reasoning and detailed solution paths. MATH (Hendrycks et al., 2021) encompasses 12,500 problems derived from high school math competitions, challenging LLMs with advanced topics like calculus and algebra, and providing step-by-step solutions to facilitate coherent training. MathBench (Liu et al., 2024) is a hierarchical benchmark that assesses both theoretical and applied mathematical abilities, consisting of 3,709 questions spanning basic arithmetic to university level, structured across five educational tiers. Omni-Math (Gao et al., 2024) focuses on Olympic-level mathematical reasoning, featuring 4,428 competition-level problems categorized into over 33 subfields and 10 difficulty levels, spanning from entry-level to professional international competitions.

6 Conclusion

In this work, we propose $G\text{-Pass}@k_\tau$ and $mG\text{-Pass}@k$, novel evaluation metrics that assess both the reasoning capability and performance consistency of LLMs across varying correctness thresholds. Through detailed evaluations conducted on mathematical reasoning benchmarks, we find that current LLMs struggle with consistent reasoning. Additionally, we demonstrate $G\text{-Pass}@k$ ’s robustness and preliminarily explore strategies for enhancing the models’ reasoning stability.

Limitations

In this study, we analyze the stability of large language models in reasoning tasks, propose a new metric, and conduct experiments on various mathematical reasoning benchmarks. Due to constraints in space and resources, our work has the following limitations:

- our experiments are not extended to broader reasoning tasks. Nevertheless, we assert that mathematical reasoning, as a representative task, aptly supports our experiments and conclusions. Future research will address a wider range of reasoning tasks.
- We attempted to run experiments on as many large language models (LLMs) as possible; however, due to hardware limitations, API call overhead, and the service instability of some APIs, a few experimental results were missing. However, we believe that the experimental results presented in this paper are sufficient to support our conclusions.
- The benchmarks in this paper include only English and Chinese, however, the conclusions are generally applicable to large language models of all languages.

Acknowledgements

This work was supported by National Key R&D Program of China 2022ZD0161600, and Shanghai Oriental Talents Project BJZH2024070.

References

- Meta AI. 2024. Llama 3.3 - 70b parameters instruct: A large language model supporting multilinguality, coding, reasoning, and tool usage. <https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct>. Accessed: 2024-12-13.
- AIME2024. 2024. Aime problems and solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.
- AIME2025. 2025. Aime problems and solutions. https://artofproblemsolving.com/wiki/index.php/2025_AIME_I.
- Anthropic Inc. 2024. Claude 3.5 Sonnet: A large language model for advanced text generation. <https://claude.ai/chats>. Accessed: 2024-12-10.
- Berk Atil, Alexa Chittams, Liseng Fu, Ferhan Ture, Lixinyu Xu, and Breck Baldwin. 2024. LLM stability: A detailed analysis with some surprises. *CoRR*, abs/2408.04667.
- Edward Beeching, Shengyi Costa Huang, Albert Jiang, Jia Li, Benjamin Lipkin, Zihan Qina, Kashif Rasul, Ziju Shen, Roman Soletskyi, and Lewis Tunstall. 2024. Numinamath 72b cot. <https://huggingface.co/AI-M0/NuminaMath-72B-CoT>.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.
- Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *CoRR*, abs/2501.17161.
- 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. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.
- OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. <https://github.com/open-compass/opencompass>.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L.

- Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yudian Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yang, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2024. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.
- DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. 2024. Deepseek-v3 technical report. *CoRR*, abs/2412.19437.
- Ben Dickson. 2024. Why data contamination is a big issue for llms. <https://api-docs.deepseek.com/news/news1120>. Accessed: 2023-7-17.
- Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. 2024. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *ACL (Findings)*, pages 12039–12050. Association for Computational Linguistics.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Al-lonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.
- Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. 2010. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.*, 88(2):303–338.
- 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. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *CoRR*, abs/2410.07985.
- Panagiotis Giadikiaroglou, Maria Lymperaioi, Giorgos Filandrianos, and Giorgos Stamou. 2024. Puzzle

- solving using reasoning of large language models: A survey. In *EMNLP*, pages 11574–11591. Association for Computational Linguistics.
- Vipul Gupta, David Pantoja, Candace Ross, Adina Williams, and Megan Ung. 2024. Changing answer order can decrease MMLU accuracy. *CoRR*, abs/2406.19470.
- 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. In *NeurIPS Datasets and Benchmarks*.
- Hugging Face. 2024. InternLM 3-8B Instruct Model. <https://huggingface.co/internlm/internlm3-8b-instruct>. Accessed: 2025-01-15.
- Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. In *ACL (industry)*, pages 37–42. Association for Computational Linguistics.
- Sumith Kulal, Panupong Pasupat, Kartik Chandra, Mina Lee, Oded Padon, Alex Aiken, and Percy Liang. 2019. Spoc: Search-based pseudocode to code. In *NeurIPS*, pages 11883–11894.
- Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024. Numinamath. <https://huggingface.co/AI-M0/NuminaMath-CoT>.
- Yucheng Li, Yunhao Guo, Frank Guerin, and Chenghua Lin. 2024. An open-source data contamination report for large language models. In *EMNLP (Findings)*, pages 528–541. Association for Computational Linguistics.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let’s verify step by step. In *ICLR*. OpenReview.net.
- Hongwei Liu, Zilong Zheng, Yuxuan Qiao, Haodong Duan, Zhiwei Fei, Fengzhe Zhou, Wenwei Zhang, Songyang Zhang, Dahua Lin, and Kai Chen. 2024. Mathbench: Evaluating the theory and application proficiency of llms with a hierarchical mathematics benchmark. In *ACL (Findings)*, pages 6884–6915. Association for Computational Linguistics.
- Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *CoRR*, abs/2308.09583.
- Shiwen Ni, Xiangtao Kong, Chengming Li, Xiping Hu, Ruifeng Xu, Jia Zhu, and Min Yang. 2024. Training on the benchmark is not all you need. *CoRR*, abs/2409.01790.
- Skywork o1 Team. 2024. *Skywork-o1 open series*. <https://huggingface.co/Skywork>.
- OpenAI. 2024a. Gpt-4o: Advancements in text and vision integration. <https://www.openai.com/blog/gpt-4o-integration/>. Accessed: 2024-12-10.
- OpenAI. 2024b. Learning to reason with llms. <https://openai.com/index/learning-to-reason-with-llms/>. Accessed: 2024-12-10.
- OpenAI. 2025. Openai o3-mini. <https://openai.com/index/openai-o3-mini/>. Accessed: 2025-1-32.
- Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *ACL*, pages 4932–4942. Association for Computational Linguistics.
- Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J. Liu. 2023. Out-of-distribution detection and selective generation for conditional language models. In *ICLR*. OpenReview.net.
- Google Research. 2024. Gemini 1.5 pro: A large language model by google. <https://ai.google.dev/>. Accessed: 2024-12-12.
- Google Research. 2025. Gemini 2.0 flash exp: A large language model by google. <https://ai.google.dev/>. Accessed: 2025-2-12.
- Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozinska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucinska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin

- McDonnell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjösund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *CoRR*, abs/2408.03314.
- Guijin Son, Hyunwoo Ko, Hoyoung Lee, Yewon Kim, and Seunghyeok Hong. 2024. Llm-as-a-judge & reward model: What they can and cannot do. *CoRR*, abs/2409.11239.
- Qwen Team. 2024a. **Qwq: Reflect deeply on the boundaries of the unknown.**
- The Mistral AI Team. 2024b. Mistral-Large Instruct 2411: A large language model supporting advanced instructions. <https://huggingface.co/mistralai/Mistral-Large-Instruct-2411>. Accessed: 2024-12-13.
- A. T. Vandermonde. 1772. *Mém. Acad. Roy. Sciences Paris*.
- Yiming Wang, Pei Zhang, Baosong Yang, Derek F. Wong, and Rui Wang. 2024. Latent space chain-of-embedding enables output-free LLM self-evaluation. *CoRR*, abs/2410.13640.
- Hanzi Xu, Renze Lou, Jiangshu Du, Vahid Mahzoon, Elmira Talebianaraki, Zhuoan Zhou, Elizabeth Garrison, Slobodan Vucetic, and Wengpeng Yin. 2024. Llms’ classification performance is overclaimed. *CoRR*, abs/2406.16203.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024a. Qwen2 technical report. *CoRR*, abs/2407.10671.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Wang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024b. Qwen2.5 technical report. *CoRR*, abs/2412.15115.
- An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024c. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *CoRR*, abs/2409.12122.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ -bench: A benchmark for tool-agent-user interaction in real-world domains. *CoRR*, abs/2406.12045.
- Huaiyuan Ying, Shuo Zhang, Linyang Li, Zhejian Zhou, Yunfan Shao, Zhaoye Fei, Yichuan Ma, Jiawei Hong, Kuikun Liu, Ziyi Wang, Yudong Wang, Zijian Wu, Shuaibin Li, Fengzhe Zhou, Hongwei Liu, Songyang Zhang, Wenwei Zhang, Hang Yan, Xipeng Qiu, Jiayu Wang, Kai Chen, and Dahua Lin. 2024. Internlm-math: Open math large language models toward verifiable reasoning. *CoRR*, abs/2402.06332.
- Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. *CoRR*, abs/2403.04652.
- Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhengguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *ICLR*. OpenReview.net.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *NeurIPS*.
- Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. 2023. Don’t make your LLM an evaluation benchmark cheater. *CoRR*, abs/2311.01964.
- Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. 2024. Prosa: Assessing and understanding the prompt sensitivity of llms. In *EMNLP (Findings)*, pages 1950–1976. Association for Computational Linguistics.

A Detailed Related Work

B Estimation of G-Pass@k

To demonstrate the unbiasedness of Equation (5), we conduct the simulation experiment illustrated in Figure 6. Specifically, we assume the probability of a model providing the correct solution in a single run is $p^* = 0.4$. For each n , we perform several random Bernoulli samplings to obtain different values of c to calculate $\text{G-Pass}@k_\tau$, and then compute the mean and variance to generate the figure. From Figure 6, it can be observed that Equation (5) is an unbiased estimator, facilitating fair comparison across different values of n .

C Proof of Theorem 2.1

Proof. Since j starts iterating at the upward rounding of $\lceil \tau \cdot k \rceil$ and $\tau \in (0, 1]$, so we have:

$$\lim_{\tau \rightarrow 0} \sum_{j=\lceil \tau \cdot k \rceil}^c \frac{\binom{c}{j} \cdot \binom{n-c}{k-j}}{\binom{n}{k}} = \sum_{j=1}^c \frac{\binom{c}{j} \cdot \binom{n-c}{k-j}}{\binom{n}{k}}. \quad (8)$$

According to the Vandermonde’s Identity (Vandermonde, 1772), the numerator term on the right side of Equation (8) can be written as

$$\begin{aligned} & \sum_{j=1}^c \binom{c}{j} \cdot \binom{n-c}{k-j} \\ &= \sum_{j=0}^c \binom{c}{j} \cdot \binom{n-c}{k-j} - \binom{n-c}{k} \\ &= \binom{n}{k} - \binom{n-c}{k}. \end{aligned} \quad (9)$$

So we conclude that:

$$\begin{aligned} \lim_{\tau \rightarrow 0} \sum_{j=\lceil \tau \cdot k \rceil}^c \frac{\binom{c}{j} \cdot \binom{n-c}{k-j}}{\binom{n}{k}} &= \sum_{j=1}^c \frac{\binom{c}{j} \cdot \binom{n-c}{k-j}}{\binom{n}{k}} \\ &= \frac{\binom{n}{k} - \binom{n-c}{k}}{\binom{n}{k}} \\ &= 1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}. \end{aligned} \quad (10)$$

D Pass@k v.s. G-Pass@k

To facilitate an intuitive comparison between $\text{Pass}@k$ and $\text{G-Pass}@k$, Figure 7 presents the metric values for various c values with $n = 80$. The figure illustrates that, while $\text{Pass}@k$ offers insights into the model’s capabilities, relying solely on it may lead to an overestimation of the model’s actual performance. For instance, as shown in the upper left of Figure 7, even when the model solves the question correctly only 8 times out of 80 runs, $\text{Pass}@k$ produces a notably high score ($\text{Pass}@k > 0.8$ for $k \geq 16$). Additionally, as c increases beyond a certain threshold, differentiating performance based on $\text{Pass}@k$ becomes increasingly challenging.

In contrast, $\text{G-Pass}@k$ provides a more accurate and nuanced evaluation of the model’s performance, as depicted in Figure 7. Across varying c values, $\text{G-Pass}@k$ demonstrates clear distinctions between performance levels. Moreover, by adjusting the threshold parameter τ , $\text{G-Pass}@k$ can highlight different aspects of the model’s performance: a lower threshold emphasizes the model’s potential, while a higher threshold underscores its stability and mastery of the problem.

In summary, $\text{G-Pass}@k$ not only delivers a more comprehensive performance assessment compared to $\text{Pass}@k$, but also, through flexible threshold configurations, effectively balances the evaluation of the model’s potential and stability.

E Evaluated LLMs

In this paper, we conduct experiments on LLMs including: InternLM3-8B-Instruct (Ying et al., 2024; Hugging Face, 2024), DeepSeek-Math-7b-RL (Shao et al., 2024), DeepSeek-V3.0-Chat (DeepSeek-AI et al., 2024), Gemma-2-27b-it (Rivière et al., 2024), Llama-3.1-8B-Instruct (Dubey et al., 2024), Llama-3.1-70B-Instruct (Dubey et al., 2024), Yi-1.5-34B-Chat (Young et al., 2024), Llama-3.3-70B-Instruct (Dubey et al., 2024; AI, 2024), NuminaMath-72B-CoT (Beeching et al., 2024), Mistral-Large-Instruct-2411 (Team, 2024b), Qwen2.5-7B-Instruct (Yang et al., 2024b), Qwen2.5-Math-7B-Instruct (Yang et al., 2024c), Qwen2.5-32B-Instruct (Yang et al., 2024b), Qwen2.5-72B-Instruct (Yang et al., 2024b), Qwen2.5-Math-72B-Instruct (Yang et al., 2024c), Qwen2.5-Max (Yang et al., 2024b), Claude-3.5-Sonnet (Anthropic Inc., 2024), Gemini-1.5-

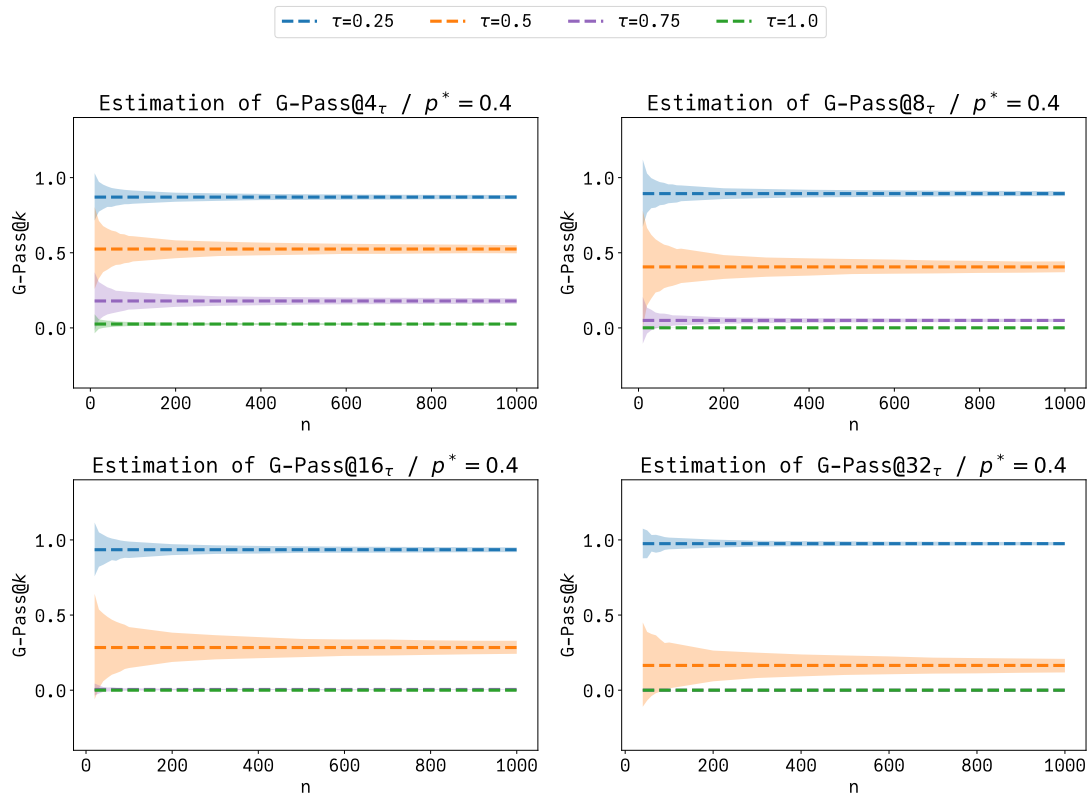


Figure 6: Illustration of estimation and the true value of G-Pass@ k_τ .

Pro (Research, 2024), Gemini-2.0-Flash-Exp (Research, 2025), and GPT-4o-2024-11-20 (OpenAI, 2024a). Additionally, we include several o1-like LLMs, such as QwQ-32B-Preview (Team, 2024a), Skywork-o1-Open-Llama-3.1-8B (o1 Team, 2024), DeepSeek-R1 series models (DeepSeek-AI et al., 2024), OpenAI o1-mini (OpenAI, 2024b), and OpenAI o3-mini (OpenAI, 2025).

F Data Details

F.1 LiveMathBench

To effectively analyze the reasoning stability of large language models, we construct a new benchmark named LiveMathBench. LiveMathBench consists of the latest complex mathematical questions from various examinations and competitions, ensuring minimal risk of data contamination. LiveMathBench will undergo ongoing updates with new questions to continuously evaluate the mathematical reasoning performance of models.

F.1.1 Benchmark Construction

LiveMathBench is specifically designed to include out-of-domain question sets with different difficulty spans from various mathematical exams and competitions, aiming to avoid data contamination

issues in existing LLMs and public math benchmarks (Zhou et al., 2023; Li et al., 2024; Ni et al., 2024). LiveMathBench (version of 202412) incorporates the latest problems from the China National Mathematical Olympiad (CNMO), China’s College Entrance Examination (CCEE), American Mathematics Competition (AMC), and William Lowell Putnam Mathematical Competition (WLPMC). These datasets encompass diverse levels of difficulty and linguistic variations and have low overlap with publicly available datasets, ensuring a comprehensive evaluation of the generalization capabilities of LLMs across various mathematical scenarios.

F.1.2 Data Sources

LiveMathBench (version of 202412) is composed of 4 parts including CNMO, CCEE, AMC, and WLPMC.

CNMO. The CNMO section features curated questions from the latest Chinese National Mathematics Olympiad. To enhance the difficulty level, single-choice questions are transformed into problem-solving tasks by concealing answer options, necessitating models to reason independently and provide solutions.

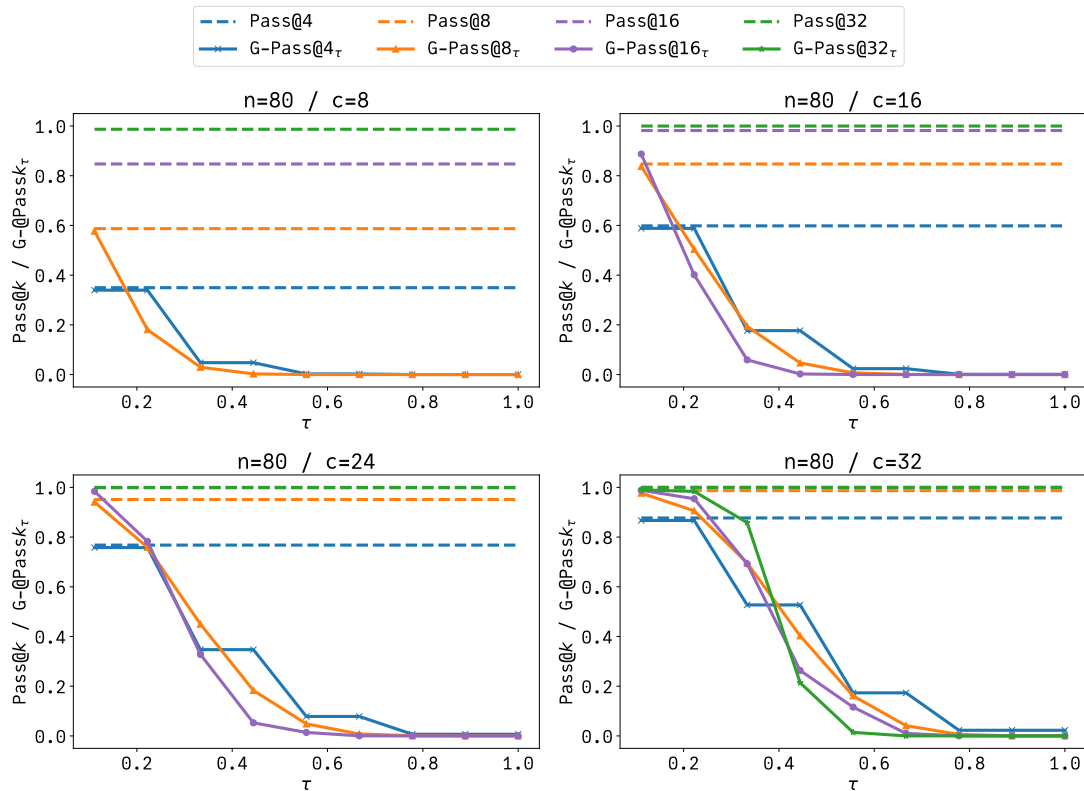


Figure 7: Comparison of Pass@ k and G-Pass@ k . In our simulation setup, we set $n = 10$ and $c = \{8, 16, 24, 32\}$, and then calculate Pass@ k and G-Pass@ k .

Table 4: Statistics of LiveMathBench

Dataset	Language	#Fill-In-the-Blank	#Problem-Solving	#Questions
CNMO	en & cn	-	18 _{×2}	18 _{×2}
CCEE	en & cn	13 _{×2}	31 _{×2}	44 _{×2}
AMC	en & cn	-	46 _{×2}	46 _{×2}
WLPMC	en & cn	-	11 _{×2}	11 _{×2}
ALL	en & cn	13 _{×2}	106 _{×2}	119 _{×2}

CCEE. In the CCEE segment, we have selected questions from recent mock exams of China’s College Entrance Examination, excluding multi-modal proof problems. We have excluded multiple-choice questions and converted single-choice items into problem-solving questions, removing provided answer choices to assess the models’ ability to generate solutions autonomously.

AMC. The AMC section includes questions from the latest American Mathematics Competition, where each original question typically offers five possible answers labeled A through E, with only one correct option. Consistent with our approach in other sections, we convert these single-choice questions into problem-solving cues,

encouraging models to deduce solutions without the aid of provided options.

WLPMC. We also include questions from the latest William Lowell Putnam Mathematical Competition (WLPMC). Regarded as one of the most prestigious university-level mathematics competitions globally, the WLPMC challenges participants with problems that span a broad spectrum of mathematical disciplines. These include geometry, algebra, trigonometry, calculus, linear algebra, combinatorics, probability theory, number theory, complex numbers, and differential equations.

F.1.3 Benchmark Statistics

Table 4 presents comprehensive statistics for the LiveMathBench. In order to enhance benchmark diversity and assess the performance of LLMs in multilingual settings, both English and Chinese versions of the questions are included.

F.1.4 Data Samples

Here we provide some samples in LiveMathBench.

Example in CNMO

[Question]

设复数 z, w 满足 $z + w = 2$, 求 $S = |z^2 - 2w| + |w^2 - 2z|$ 的最小可能值。

[Answer]

$8\sqrt{5} - 16$

[Question Type]

问答

Example in CCEE

[Question]

函数 $f(x) = \frac{x^3 e^{3x} - 3 \ln x - 1}{x} (x > 0)$ 的最小值是_____

[Answer]

3

[Question Type]

填空

Example in AMC

[Question]

The graph of $y = e^{x+1} + e^{-x} - 2$ has an axis of symmetry. What is the reflection of the point $(-1, \frac{1}{2})$ over this axis?

[Answer]

$(0, \frac{1}{2})$

[Question Type]

Problem-Solving

F.1.5 Hard Split

To achieve better differentiation, we also implemented a hard split for evaluating models with strong inference performance. Specifically, we selected subsets where QwQ-32B-Preview and OpenAI o1-mini models demonstrated lower accuracy,

Example in WLPMC

[Question]

A sequence y_1, y_2, \dots, y_k of real numbers is called *zigzag* if $k = 1$, or if $y_2 - y_1, y_3 - y_2, \dots, y_k - y_{k-1}$ are nonzero and alternate in sign. Let X_1, X_2, \dots, X_n be chosen in-

dependently from the uniform distribution on $[0, 1]$. Let $a(X_1, X_2, \dots, X_n)$ be the largest value of k for which there exists an increasing sequence of integers i_1, i_2, \dots, i_k such that $X_{i_1}, X_{i_2}, \dots, X_{i_k}$ is zigzag. Find the expected value of

$a(X_1, X_2, \dots, X_n)$ for $n \geq 2$.

[Answer]

$\frac{2n+2}{3}$

[Question Type]

Problem-Solving

consisting of 21 questions in English and 24 questions in Chinese.

F.2 MATH500-L5

MATH500 (Lightman et al., 2024) dataset is a curated subset of a larger collection MATH (Hendrycks et al., 2021), intended to challenge LLMs with complex mathematical problems. It encompasses a variety of advanced questions from multiple domains including algebra, geometry, probability, and number theory, thereby providing a comprehensive assessment of a model's proficiency in mathematical reasoning. We select all questions with difficulty 5, resulting in **MATH500-L5**, which contains 134 questions.

F.3 AIME2024-45

Tailored for evaluating LLM performance at the American Invitational Mathematics Examination (AIME) level, the AIME question set (AIME2024, 2024; Yang et al., 2024c) presents a series of intricate tasks that test logical thinking, abstract reasoning, and accurate calculation skills. This resource aims to push the boundaries of what LLMs can achieve in solving sophisticated mathematical problems. We combine the part 1 and the part 2 of the American Invitational Mathematics Examination 2024, resulting in 45 questions, called **AIME2024-45**.

F.4 AIME2025

We also incorporate Part 1 of the American Invitational Mathematics Examination 2025, referred to as **AIME2025** (AIME2025, 2025), as our evaluation benchmark, comprising 15 questions.

G Judge Details

G.1 Configurations of Judge Model

Inspired by previous works (Zheng et al., 2023; Son et al., 2024), we leverage Qwen2.5-72B-Instruct (Yang et al., 2024a) to judge if the answers generated by the models are consistent with the golden answers, consider the high inference cost of the closed source models such as OpenAI models. We set the temperature to 0.0, and maximum output tokens to 8,192.

G.2 Prompt for Judge

We leverage the prompts shown in Box Chinese Version of Judge Prompt and Box English Version of Judge Prompt to judge the consistency between candidate answers and reference answers.

G.3 Evaluation of Judge Model

To evaluate the effectiveness of our judge model, we compared the agreement rate between Qwen2.5-72B-as-Judge and GPT4o-as-Judge (OpenAI, 2024a). Specifically, we randomly selected 300 samples from the generations of five different models and used the judgments from GPT4o as the ground truth. We then calculated the agreement rate between the judgments made by our model and those by GPT4o. Table 5 presents the results, demonstrating that Qwen2.5-72B-as-Judge achieves high consistency with GPT4o-as-Judge across different models. These findings validate the feasibility of Qwen2.5-72B-as-Judge.

H Implementation Details

In all experiments, we set the number of generations, n , to $16 \times 3 = 48$ and report the greedy accuracy, $\text{Pass}@k$ ($\text{G-Pass}@k \rightarrow 0$), and $\text{G-Pass}@k$ values, where $k \in \{4, 8, 16\}$ and $\tau \in \{0.25, 0.5, 0.75, 1.0\}$. For the sampling parameters of open-source models, we configure the temperature to 1.0, top-p to 0.8, top-k to 50, and repetition-penalty to 1.0. For open-source models, the maximum number of tokens is set to 8,192 for non-o1 LLMs and 32,768 for o1-like LLMs. For

closed-source models, due to constraints of inference costs, we configured the maximum completion tokens to 4,096 for non-o1 LLMs and 8,192 for o1-like LLMs. We use the OpenCompass (Contributors, 2023) platform to evaluate all LLMs.

Due to the diverse formats of the final answers produced by models in complex mathematical questions, we leverage Qwen-2.5-72B-Instruct (Yang et al., 2024a) to judge whether the content generated by the tested model aligns with the standard answer. In our judge pipeline, we provide the original question, reference answer, and model-generated answer, prompting Qwen-2.5-72B-Instruct to determine whether the candidate solution is consistent with the reference answer. The details of the judging process can be found in Appendix G.

All experiments were conducted on clusters equipped with NVIDIA A800 GPUs and Intel(R) Xeon(R) Platinum 8336C CPUs, requiring hundreds of GPU hours to complete.

I Additional Experimental Results

I.1 Full Performance on LiveMathBench

Table 6 presents the comprehensive performance results on LiveMathBench.

I.2 Full Performance on LiveMathBench Hard Split

Table 7 presents the comprehensive performance results on LiveMathBench hard split.

I.3 Full Performance on MATH500-L5 & AIME2024-45

Table 8, Table 9 presents the comprehensive performance results for MATH500-L5, AIME2024-45, and AIME2025.

I.4 Full Performance *w.r.t.* Different temperatures

Figure 8 demonstrates the performance of models across different sampling temperatures. The values for the other sampling parameters, namely top-p (0.8), top-k (50), and repetition-penalty (1.0), remain constant. It can be observed that most models exhibit relatively stable performance when the temperature values are set to $\{0.3, 0.5, 0.7, 1.0\}$. This indicates that the $\text{G-Pass}@k_\tau$ metric is robust for commonly used temperature parameters. However, it is worth noting that certain models, such as Qwen2.5-Math-7B-Instruct and Qwen2.5-Math-72B-Instruct, display an unusual sensitivity

Chinese Version of Judge Prompt

请你作为一个数学阅卷专家，判断下面的答案是否与标准答案一致，即考生是否回答正确。下面是一些评判标准：

1. 有些答案可能包含多项内容，可能有单选题，多选题，填空题和问答题，只要答案与标准答案一致即可，对于多选题和多个空的填空题，需要考生对应的选项或空都回答正确才算正确。

2. 有些答案可能通过不同的方式表达，比如有些答案可能是一个数学表达式，有些答案可能是一个文字描述，只要表达的意思一致即可。且有些公式通过不同的方式表达，但等价，也是正确的。

3. 你不需要重新计算问题答案，因为标准答案已经给出，只需要根据问题形式来判断考生的答案是否与标准答案一致，是否正确即可。

请你根据上述标准，判断下面的答案是否与标准答案一致，如果一致，请在最后输出 $\boxed{\text{yes}}$ ，否则输出 $\boxed{\text{no}}$ ，如果难以判断，请输出 $\boxed{\text{no}}$ 。

原问题：{question}

标准答案：{reference_answer}

考生答案：{candidate_answer}

分析：

English Version of Judge Prompt

Please act as an expert in grading mathematics exam papers, and judge whether the following answers match the standard answers, i.e., whether the examinee answered correctly. Here are some evaluation criteria:

1. Some answers may contain multiple parts, such as single-choice questions, multiple-choice questions, fill-in-the-blank questions, and problem-solving questions. As long as the answer matches the standard answer, it is considered correct. For multiple-choice questions and fill-in-the-blank questions with multiple blanks, the examinee must answer all corresponding options or blanks correctly to be considered correct.

2. Some answers may be expressed in different ways; for example, some answers may be mathematical expressions, while others may be textual descriptions. As long as the meaning conveyed is consistent, it is considered correct. Additionally, some formulas may be expressed differently but are equivalent, which is also considered correct.

3. You do not need to recalculate the problem answers, as the standard answers are already provided. You only need to judge whether the examinee's answer matches the standard answer based on the form of the question and whether it is correct.

Please judge whether the following answer matches the standard answer according to the above criteria. If they match, output $\boxed{\text{yes}}$, otherwise output $\boxed{\text{no}}$. If it is difficult to judge, also output $\boxed{\text{no}}$.

Original Question: {question}

Standard Answer: {reference_answer}

Examinee's Answer: {candidate_answer}

Analysis:

Table 5: Agreement rates between Qwen2.5-72B-as-Judge and GPT4o-as-judge.

Models Need to Judge	Agreement	Disagreement	Accuracy (%)
Deepseek-Math-7B-RL	296	4	98.7
Qwen2.5-32B-Instruct	282	18	94.0
Qwen2.5-Math-72B-Instruct	287	13	95.7
Mistral-Large-Instruct-2411	285	15	95.0
QwQ-32B-Preview	290	10	96.7

to larger temperature coefficients. This discrepancy may reflect the impact of different training strategies on the model distribution. Intriguingly, we discover that the model QwQ-32B-Preview shows exceptional stability when it comes to sampling temperatures. We speculate that models similar to o1, which rely on long COT and reflection, are capable of consistently identifying the correct answers through continuous self-examination and backtracking. This characteristic contributes to the high reasoning stability observed in these models.

I.5 Full Performance *w.r.t.* Different top-p

Figure 9 illustrates the performance of models across various sampling top-p parameters. The values for the other sampling parameters, namely, temperature (1.0), top-k (50), and repetition-penalty (1.0), remain constant. Similar phenomena are also observed in the experiments conducted with varying temperatures. Most models exhibit stable performance within the range of commonly used parameters, demonstrating the effectiveness of $G\text{-Pass}@k_\tau$. Furthermore, QwQ-32B-Preview also exhibits stability *w.r.t.* different top-p values.

I.6 Full Performance *w.r.t.* Different top-k

Figure 10 illustrates the performance of models across various sampling top-k parameters. The values for the other sampling parameters, specifically, temperature (1.0), top-p (0.8), and repetition-penalty (1.0), remain constant. Our analysis indicates that all models exhibit stable performance, suggesting that the top-k parameter has a minimal effect on sampling compared to temperature and top-p.

Table 6: Full performance of models on LiveMathBench. We report results of greedy decoding, Pass@16 (G-Pass@16 \rightarrow 0), G-Pass@16 $_{\{0.25,0.5,0.75,1.0\}}$, and mG-Pass@16.

LLMs	Greedy	G-Pass@16 (Equation (5)) / %					
		G-Pass@16 \rightarrow 0	G-Pass@16 $_{0.25}$	G-Pass@16 $_{0.5}$	G-Pass@16 $_{0.75}$	G-Pass@16 $_{1.0}$	mG-Pass@16
General LLMs							
Llama-3.1-8B-Instruct	24.0	53.3	29.9	18.2	11.3	4.5	10.4
Yi-1.5-34B-Chat	24.8	58.7	31.4	18.6	11.3	6.0	11.0
Gemma-2-27b-it	26.9	54.3	33.6	23.5	17.8	12.7	17.3
Llama-3.1-70B-Instruct	29.8	59.2	38.6	30.0	22.2	12.5	20.8
Qwen2.5-7B-Instruct	37.0	66.5	47.3	36.5	27.2	16.0	25.8
Llama-3.3-70B-Instruct	40.3	62.0	45.8	36.2	28.9	19.1	27.5
GPT-4o-2024-11-20 [†]	44.8	70.8	54.6	41.9	32.9	22.2	31.6
InternLM3-8B-Instruct	44.5	69.5	50.8	43.0	35.4	23.0	33.6
Claude-3.5-Sonnet [†]	46.7	71.2	54.3	44.1	36.2	26.6	35.3
Mistral-Large-Instruct-2411	41.6	47.3	42.2	39.4	37.1	32.9	36.4
Qwen2.5-72B-Instruct	51.7	69.6	55.6	47.3	39.6	29.0	37.8
Qwen2.5-32B-Instruct	50.8	72.0	57.6	48.3	39.5	28.6	38.1
Qwen2.5-Max [†]	52.9	74.8	62.8	52.7	44.3	31.1	42.2
Gemini-1.5-Pro-Latest [†]	59.1	78.8	65.7	55.9	47.3	31.0	44.3
DeepSeek-V3.0-Chat [†]	55.0	80.7	69.7	59.5	49.9	35.0	47.9
Mathematical Reasoning LLMs							
DeepSeek-Math-7B-RL	23.5	45.0	29.0	19.8	14.0	9.7	13.7
NuminaMath-72B-CoT	40.8	63.3	43.5	34.0	27.1	14.2	25.0
Qwen2.5-Math-7B-Instruct	44.1	68.4	53.0	44.1	38.3	28.1	36.6
Qwen2.5-Math-72B-Instruct	57.6	74.2	60.4	52.7	45.4	27.9	42.3
O1-like Reasoning LLMs							
Skywork-o1	45.4	61.1	47.7	39.3	31.9	21.7	30.4
QwQ-32B-Preview	72.7	89.0	81.8	74.9	65.8	40.1	61.2
DeepSeek Distill Qwen-1.5B	42.4	83.0	71.7	61.9	48.8	25.6	45.1
DeepSeek Distill LLama-8B	58.4	88.8	78.4	67.8	56.8	31.9	52.2
DeepSeek Distill Qwen-7B	65.6	88.5	81.8	73.0	66.4	48.4	63.1
OpenAI o1-mini [†]	74.1 [‡]	89.5	82.4	76.3	67.3	48.3	64.8
DeepSeek Distill Qwen-14B	69.8	90.5	85.9	79.9	71.2	51.6	68.0
DeepSeek Distill Qwen-32B	67.7	90.1	86.3	81.2	72.3	54.5	69.7
DeepSeek Distill LLama-70B	74.8	89.6	86.1	80.8	73.0	53.0	69.7
OpenAI o3-mini [†]	84.7 [‡]	94.3	90.2	85.7	78.8	65.3	76.8
DeepSeek R1 [†]	81.1	91.8	86.9	83.6	79.1	69.5	77.6

[†] API-based close-source LLMs.

[‡] OpenAI o1-like series model does not provide an optional temperature parameter, so we chose the average accuracy of 20 generations as greedy accuracy.

Table 7: Full performance of models on LiveMathBench hard split. We report results of greedy decoding, Pass@16 (G-Pass@16 \rightarrow 0), G-Pass@16 $_{\{0.25,0.5,0.75,1.0\}}$, and mG-Pass@16.

LLMs	Greedy	G-Pass@16 (Equation (5)) / %					
		G-Pass@16 \rightarrow 0	G-Pass@16 $_{0.25}$	G-Pass@16 $_{0.5}$	G-Pass@16 $_{0.75}$	G-Pass@16 $_{1.0}$	mG-Pass@16
General LLMs							
Llama-3.1-8B-Instruct	2.2	35.5	11.5	0.8	0.0	0.0	0.0
Qwen2.5-7B-Instruct	13.3	23.6	11.4	6.2	3.2	2.2	3.3
Llama-3.3-70B-Instruct	4.4	23.1	11.0	8.0	4.7	2.3	4.4
Llama-3.1-70B-Instruct	4.4	25.0	17.5	12.2	7.5	2.7	7.0
InternLM3-8B-Instruct	11.1	40.0	19.3	10.7	8.2	2.7	7.1
Qwen2.5-32B-Instruct	13.3	30.0	19.1	14.1	10.5	3.5	9.1
Qwen2.5-72B-Instruct	17.8	29.0	18.1	15.3	11.3	5.4	10.5
Mathematical Reasoning LLMs							
DeepSeek-Math-7B-RL	8.9	23.4	6.9	3.9	2.3	0.6	2.2
Qwen2.5-Math-7B-Instruct	15.6	21.3	13.1	8.2	3.3	2.2	3.8
NuminaMath-72B-CoT	11.1	28.9	10.3	8.8	7.3	5.9	7.3
Qwen2.5-Math-72B-Instruct	11.1	32.2	23.2	11.8	7.9	5.9	7.9
O1-like Reasoning LLMs							
QwQ-32B-Preview	15.6	54.3	22.1	5.9	4.4	2.4	4.0
DeepSeek Distill Qwen-1.5B	6.7	33.9	15.8	9.8	4.5	2.2	4.6
DeepSeek Distill LLama-8B	8.9	56.5	30.7	16.1	5.6	2.4	6.2
DeepSeek Distill Qwen-7B	17.8	57.4	32.9	13.9	8.8	3.3	8.1
OpenAI o1-mini [†]	18.4 [‡]	68.3	45.6	21.0	10.1	0.5	8.5
DeepSeek Distill Qwen-14B	15.6	59.8	43.0	26.9	15.9	8.1	15.5
DeepSeek Distill Qwen-32B	22.2	59.1	43.8	29.9	16.9	3.3	15.1
DeepSeek Distill LLama-70B	35.6	61.2	50.1	33.1	19.0	5.8	17.3
OpenAI o3-mini [†]	43.3 [‡]	72.3	57.1	47.4	32.5	7.7	28.6
DeepSeek R1 [†]	42.2	71.5	55.9	46.6	33.6	9.8	29.6

[†] API-based close-source LLMs.

[‡] OpenAI o1-like series model does not provide an optional temperature parameter, so we chose the average accuracy of 20 generations as greedy accuracy.

Table 8: Full performance of models on MATH500-L5 and AIME2024-45. Results of greedy decoding, Pass@16 (G-Pass@16 \rightarrow 0), G-Pass@16 $_{\{0.25,0.5,0.75,1.0\}}$, and mG-Pass@16 are reported.

LLMs	Greedy	G-Pass@16 (Equation (5)) / %					
		G-Pass@16 \rightarrow 0	G-Pass@16 $_{0.25}$	G-Pass@16 $_{0.5}$	G-Pass@16 $_{0.75}$	G-Pass@16 $_{1.0}$	mG-Pass@16
MATH500-L5							
Llama-3.1-8B-Instruct	26.1	62.2	29.9	17.8	10.7	3.5	9.7
Yi-1.5-34B-Chat	26.1	66.4	38.8	25.7	16.7	9.5	16.2
Gemma-2-27b-it	24.6	50.5	32.6	23.7	18.4	13.0	17.6
Llama-3.1-70B-Instruct	39.6	73.8	53.7	41.8	32.1	16.1	29.3
InternLM-3-8B-Instruct	51.5	72.3	56.8	49.9	40.3	26.9	38.3
Qwen2.5-7B-Instruct	56.0	76.1	66.6	54.9	43.3	28.0	41.5
Llama-3.3-70B-Instruct	54.5	73.1	63.1	55.4	49.5	35.0	47.3
Mistral-Large-Instruct-2411	55.2	58.4	54.4	52.3	51.2	45.6	50.1
Qwen2.5-72B-Instruct	63.4	78.9	69.5	62.5	54.4	44.9	53.1
Qwen2.5-Max [†]	63.4	87.1	73.5	65.8	57.3	38.9	54.5
Qwen2.5-32B-Instruct	64.2	79.9	71.1	66.6	59.4	41.0	55.6
Gemini-1.5-Pro-Latest [†]	72.4	92.7	82.7	74.4	64.9	45.3	61.8
DeepSeek-Math-7b-RL	15.7	44.0	26.3	15.5	8.7	5.7	9.0
NuminaMath-72B-CoT	41.0	67.3	51.1	36.8	26.8	16.8	25.6
Qwen2.5-Math-72B-Instruct	71.6	77.8	71.1	64.9	59.4	46.0	57.4
Qwen2.5-Math-7B-Instruct	65.7	78.9	71.4	65.0	62.2	57.6	61.5
Skywork-o1	61.2	70.9	60.0	56.5	52.2	42.9	50.7
DeepSeek Distill Qwen-1.5B	53.0	89.5	82.3	72.1	62.1	34.5	57.0
QwQ-32B-Preview	82.8	95.9	92.5	87.2	78.8	57.4	75.6
DeepSeek Distill LLama-8B	65.7	92.2	86.6	79.5	70.0	39.5	64.5
DeepSeek Distill Qwen-7B	78.4	96.8	94.2	87.9	80.5	62.6	77.6
DeepSeek Distill Qwen-14B	76.1	97.1	93.4	91.1	85.9	67.7	82.6
DeepSeek Distill LLama-70B	87.3	96.5	93.5	89.6	85.5	66.8	81.9
DeepSeek Distill Qwen-32B	83.6	96.1	93.6	89.9	83.8	70.4	81.9
AIME2024-45							
Yi-1.5-34B-Chat	2.2	20.5	5.9	0.5	0.0	0.0	0.0
Llama-3.1-8B-Instruct	4.4	28.1	4.9	2.2	1.6	0.0	1.2
Gemma-2-27b-it	6.7	21.0	8.3	5.2	1.8	0.0	1.8
InternLM-3-8B-Instruct	11.1	20.5	13.3	7.2	4.3	1.0	3.7
Qwen2.5-32B-Instruct	11.1	32.0	14.9	7.1	3.4	2.2	3.7
Mistral-Large-Instruct-2411	13.3	15.4	11.1	10.4	6.8	2.4	6.1
Qwen2.5-7B-Instruct	11.1	26.3	11.2	8.9	8.1	4.7	7.5
Llama-3.1-70B-Instruct	15.6	41.2	23.5	15.0	8.1	3.0	8.0
Gemini-1.5-Pro-Latest [†]	13.3	45.7	26.5	16.8	8.7	2.5	8.1
Qwen2.5-72B-Instruct	13.3	33.7	16.3	13.7	12.9	7.5	11.7
Llama-3.3-70B-Instruct	22.2	37.1	28.7	25.3	18.2	6.9	16.4
Qwen2.5-Max [†]	22.2	44.4	25.2	15.5	9.9	5.3	9.8
DeepSeek-Math-7b-RL	2.2	16.3	4.4	1.5	0.1	0.0	0.1
NuminaMath-72B-CoT	2.2	21.3	4.8	2.9	2.2	0.1	1.6
Qwen2.5-Math-7B-Instruct	11.1	20.8	8.5	4.6	2.6	2.2	2.8
Qwen2.5-Math-72B-Instruct	20.0	35.2	24.8	18.7	16.2	6.7	14.1
Skywork-o1	11.1	22.1	13.6	11.2	10.3	1.5	8.2
DeepSeek Distill Qwen-1.5B	17.8	68.7	41.6	23.9	14.9	2.8	13.5
QwQ-32B-Preview	44.4	74.3	59.3	41.0	28.6	8.1	24.7
DeepSeek Distill LLama-8B	44.4	82.1	72.6	53.9	30.4	9.0	28.0
DeepSeek Distill Qwen-7B	44.4	79.6	73.9	56.3	35.4	17.5	33.8
OpenAI o1-mini [‡]	60.3 [‡]	86.7	80.0	62.2	53.3	15.6	43.1
DeepSeek Distill Qwen-14B	62.2	86.5	79.3	75.8	62.9	26.5	56.0
DeepSeek Distill LLama-70B	62.2	84.4	76.9	72.9	63.4	32.2	57.6
DeepSeek Distill Qwen-32B	62.2	86.3	79.7	77.0	66.5	31.3	59.3

[†] API-based LLMs.

[‡] OpenAI o1 series model does not provide an optional temperature parameter, so we chose the average accuracy of 20 generations as greedy accuracy.

Table 9: Full performance of models on AIME2025. We report results of greedy decoding, Pass@16 (G-Pass@16 \rightarrow 0), G-Pass@16 $_{\{0.25,0.5,0.75,1.0\}}$, and mG-Pass@16.

LLMs	Greedy	G-Pass@16 (Equation (5)) / %					
		G-Pass@16 \rightarrow 0	G-Pass@16 $_{0.25}$	G-Pass@16 $_{0.5}$	G-Pass@16 $_{0.75}$	G-Pass@16 $_{1.0}$	mG-Pass@16
General LLMs							
Llama-3.1-8B-Instruct	0.0	8.9	0.0	0.0	0.0	0.0	0.0
Gemma-2-27b-it	0.0	9.5	0.0	0.0	0.0	0.0	0.0
Yi-1.5-34B-Chat	0.0	14.8	4.8	0.1	0.0	0.0	0.0
GPT-4o-2024-11-20 [†]	0.0	25.5	7.4	0.1	0.0	0.0	0.0
Llama-3.1-70B-Instruct	6.7	21.3	8.3	4.6	0.2	0.0	0.7
InternLM3-8B-Instruct	13.3	30.3	16.2	6.7	0.1	0.0	0.8
Qwen2.5-32B-Instruct	20.0	33.3	28.1	11.5	0.2	0.0	1.4
Claude-3.5-Sonnet [†]	13.3	34.4	16.9	6.4	1.2	0.0	1.7
Qwen2.5-7B-Instruct	6.7	25.2	13.5	9.7	6.2	0.2	4.7
Qwen2.5-72B-Instruct	20.0	33.2	23.4	12.2	5.8	0.1	4.9
Llama-3.3-70B-Instruct	6.7	13.6	6.7	6.7	6.6	0.5	5.0
Gemini-1.5-Pro-Latest [†]	20.0	40.9	25.8	10.8	6.7	4.4	6.8
Qwen2.5-Max [†]	13.3	39.9	24.0	11.9	6.8	2.9	6.8
Mistral-Large-Instruct-2411	13.3	19.7	14.4	10.8	6.8	6.7	7.2
Gemini-2.0-Flash-Exp [†]	26.7	44.7	30.3	26.5	21.5	14.0	21.2
Mathematical Reasoning LLMs							
DeepSeek-Math-7B-RL	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NuminaMath-72B-CoT	0.0	21.0	10.3	6.7	6.7	4.4	6.4
Qwen2.5-Math-7B-Instruct	20.0	36.8	16.2	8.7	6.7	6.7	6.8
Qwen2.5-Math-72B-Instruct	13.3	30.1	15.7	13.3	13.3	13.3	13.3
O1-like Reasoning LLMs							
Skywork-o1	13.3	31.2	21.5	15.3	13.3	7.2	11.8
DeepSeek Distill Qwen-1.5B	26.7	54.6	38.1	31.9	23.8	1.4	18.6
DeepSeek Distill LLama-8B	40.0	62.2	55.3	40.4	21.2	7.9	21.0
QwQ-32B-Preview	26.7	60.5	43.5	34.5	32.4	15.6	28.1
OpenAI o1-mini [†]	46.7 [‡]	62.0	41.0	39.9	32.5	14.0	28.4
DeepSeek Distill Qwen-7B	46.7	62.1	53.2	46.6	38.3	22.7	36.1
DeepSeek Distill Qwen-14B	46.7	67.7	60.1	58.8	41.1	25.2	40.8
DeepSeek Distill LLama-70B	46.7	75.1	54.9	52.5	38.6	26.8	37.4
DeepSeek R1 [†]	66.7	68.9	63.7	52.6	46.8	24.3	42.5
OpenAI o3-mini [†]	53.3 [‡]	80.0	77.7	59.0	46.5	29.4	43.6
DeepSeek Distill Qwen-32B	46.7	72.6	63.6	59.7	50.2	29.5	47.3

[†] API-based close-source LLMs.

[‡] OpenAI o1-like series model does not provide an optional temperature parameter, so we chose the average accuracy of 20 generations as greedy accuracy.

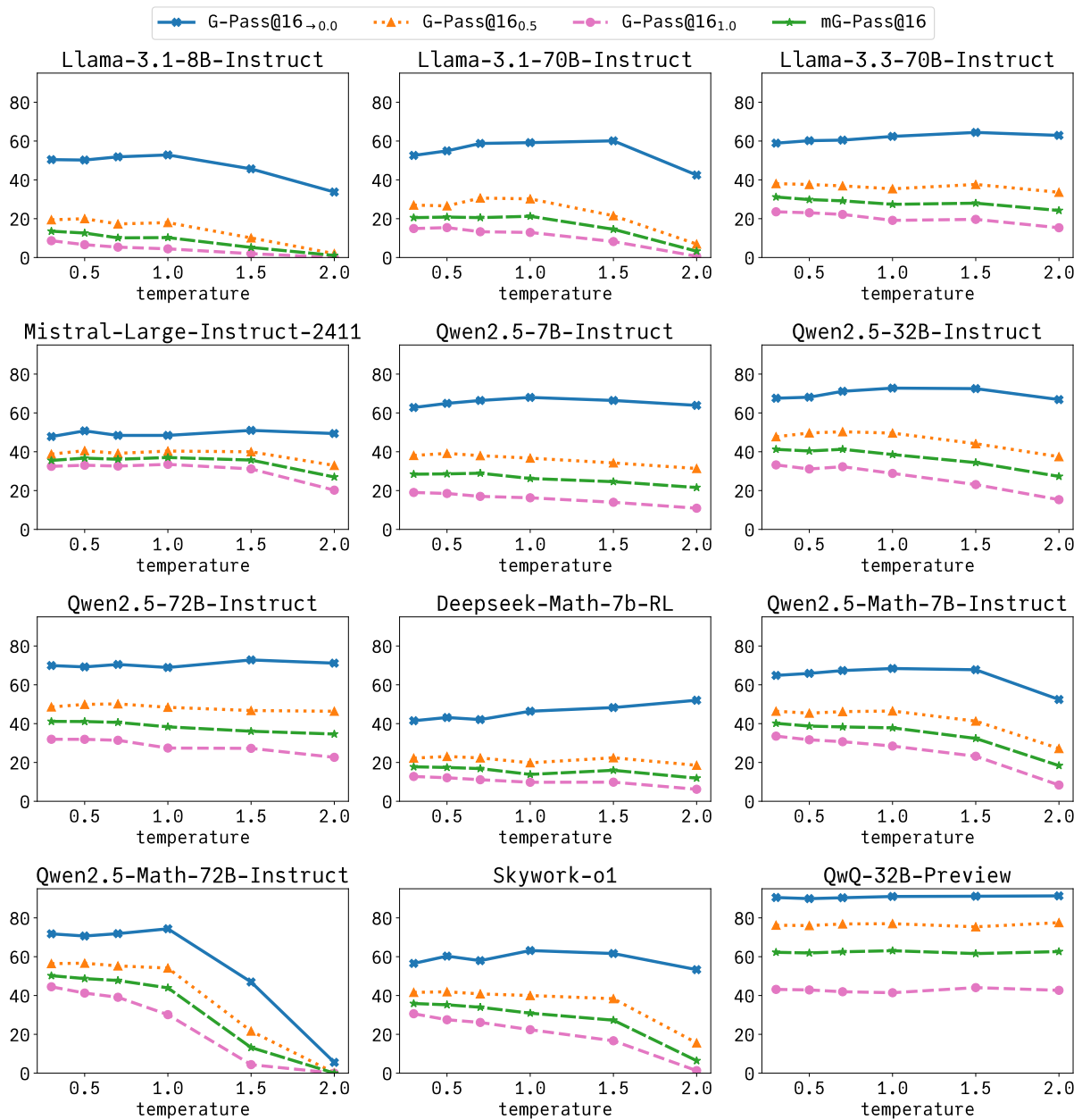


Figure 8: G-Pass@ k_T performance of LLMs *w.r.t.* different temperatures.

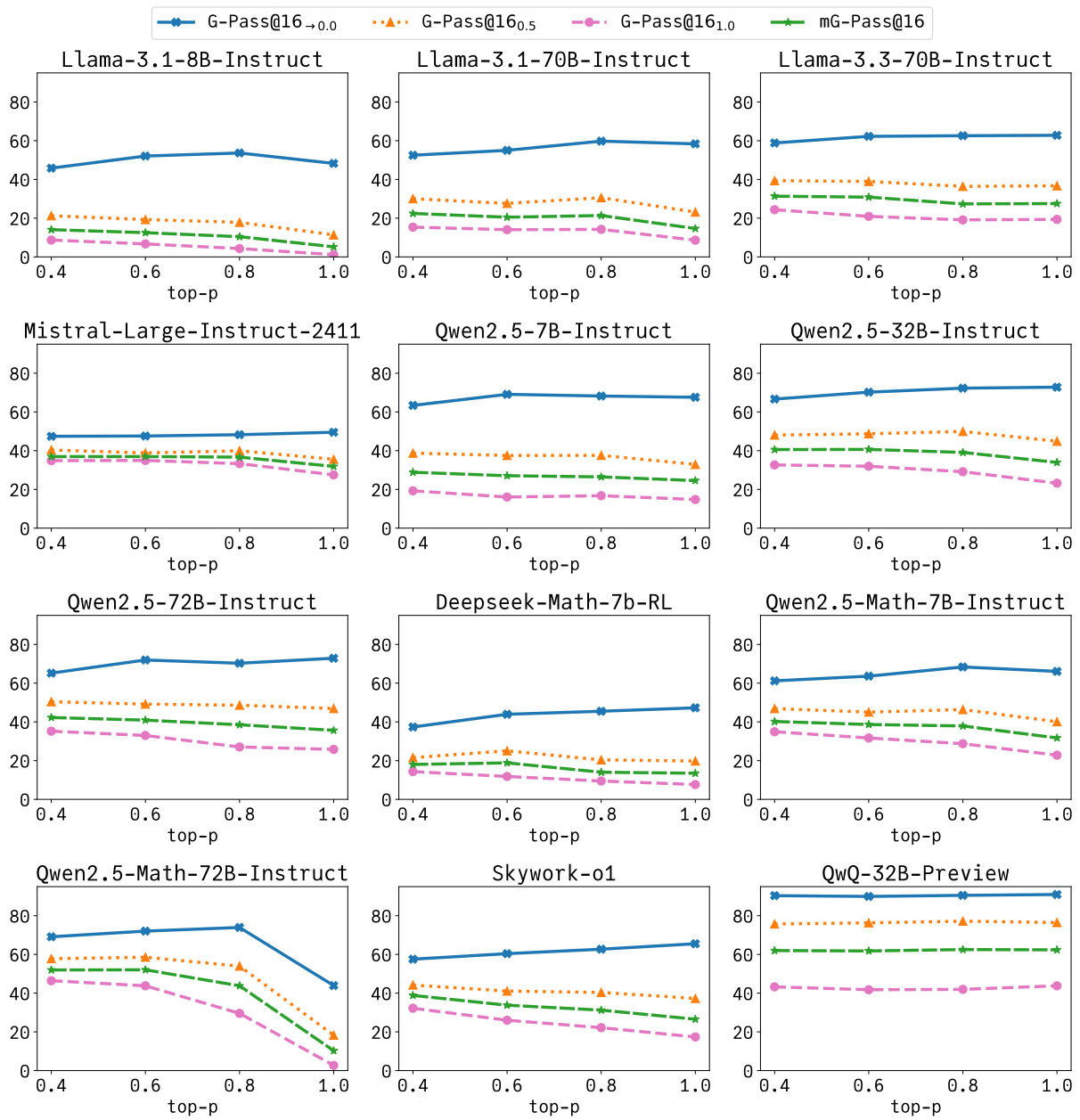


Figure 9: G-Pass@ k_τ performance of LLMs *w.r.t.* different top-p.

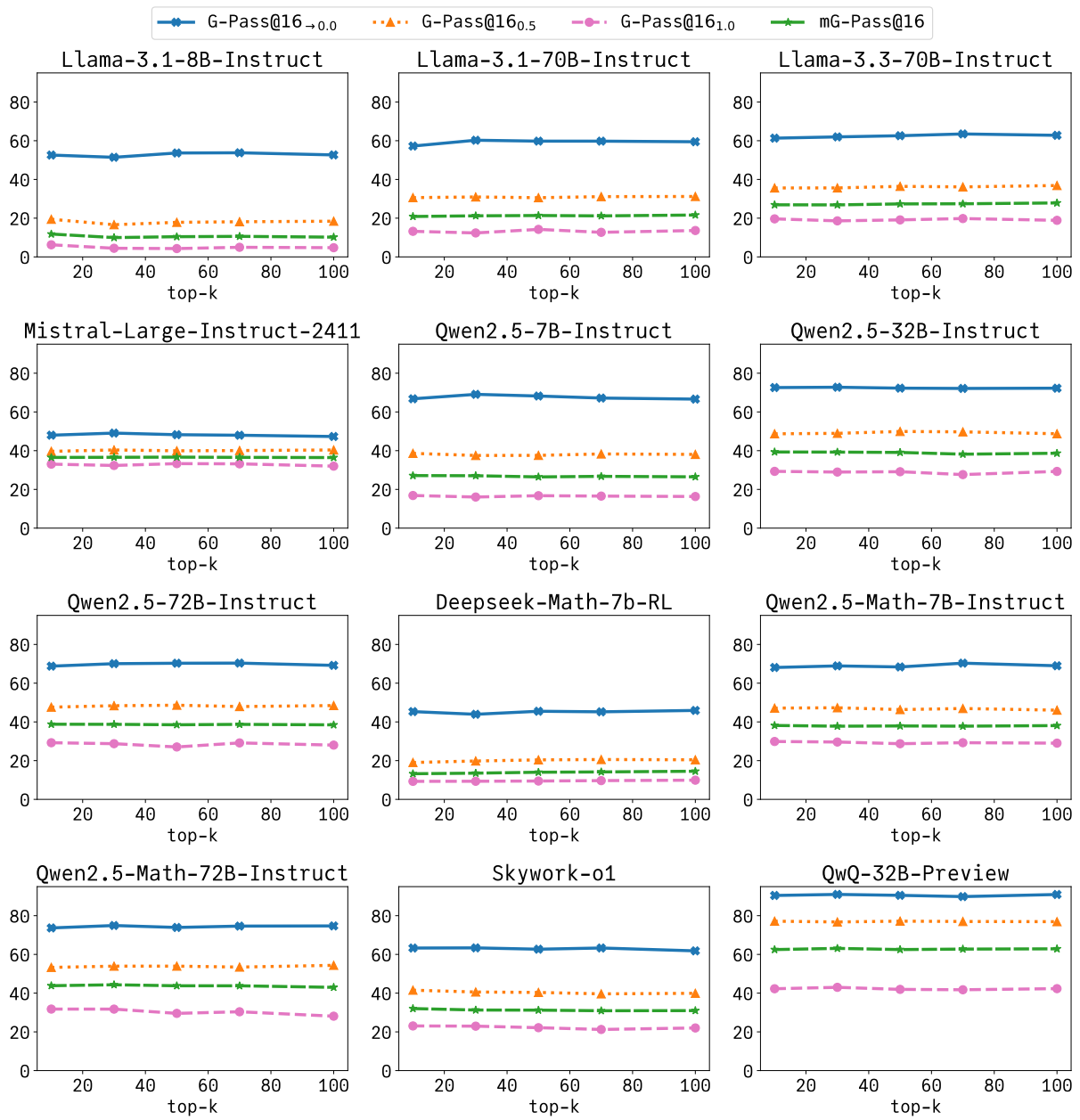


Figure 10: G-Pass@ k_{τ} performance of LLMs *w.r.t.* different top-k.

Question 1

In $\triangle ABC$, given $\cos C = \frac{\sin A + \cos A}{2} = \frac{\sin B + \cos B}{2}$, find the value of $\cos C$.

QwQ-32B-Preview Response for Question 1

Let's tackle this problem step by step. So, we have a triangle ABC, and we're given that $\cos C$ equals both $(\sin A + \cos A)/2$ and $(\sin B + \cos B)/2$. Our goal is to find the value of $\cos C$. First, let's write down what we know:

$$\cos C = \frac{\sin A + \cos A}{2} = \frac{\sin B + \cos B}{2}$$

.....

Therefore, $\cos C = (\sqrt{7})/4$.

Wait, but earlier steps led to $\cos C = -\cos 2A$, and now $\cos C = (\sqrt{7})/4$.

Let me verify if this makes sense.

We have $\cos C = (\sqrt{7})/4$, and from $\cos C = -\cos 2A$.

So, $-\cos 2A = (\sqrt{7})/4 \rightarrow \cos 2A = -\sqrt{7}/4$.

But $\cos 2A = 2\cos^2 A - 1$.

Substitute $\cos A = (\sqrt{7} - 1)/4$:

$$\begin{aligned} 2\left(\frac{\sqrt{7}-1}{4}\right)^2 - 1 &= 2\left(\frac{7-2\sqrt{7}+1}{16}\right) - 1 = 2\left(\frac{8-2\sqrt{7}}{16}\right) - 1 = \frac{8-2\sqrt{7}}{8} - 1 \\ &= \frac{8-2\sqrt{7}-8}{8} = -\frac{\sqrt{7}}{4} \end{aligned}$$

This matches $\cos C = -\cos 2A = -(-\sqrt{7}/4) = \sqrt{7}/4$.

Therefore, $\cos C = \sqrt{7}/4$.

Wait, but I thought $\cos C = (\sqrt{7})/4$, and $\cos 2A = -\sqrt{7}/4$, which satisfies $\cos C = -\cos 2A$.

Therefore, the value of $\cos C$ is $\sqrt{7}/4$.

So, the answer should be $\frac{\sqrt{7}}{4}$.

Final Answer

$$\frac{\sqrt{7}}{4}$$

Qwen2.5-32B-Instruct-Preview Response Question 1

Given the equation in $\triangle ABC$:

$$\cos C = \frac{\sin A + \cos A}{2} = \frac{\sin B + \cos B}{2}$$

We will denote the common value of these expressions by x :

$$\cos C = x$$

$$\frac{\sin A + \cos A}{2} = x$$

$$\frac{\sin B + \cos B}{2} = x$$

From these equations, we derive:

$$\sin A + \cos A = 2x$$

$$\sin B + \cos B = 2x$$

1. Square both sides of each equation:

$$(\sin A + \cos A)^2 = (2x)^2$$

$$(\sin B + \cos B)^2 = (2x)^2$$

2. Expand the squares:

$$\sin^2 A + \cos^2 A + 2 \sin A \cos A = 4x^2$$

$$\sin^2 B + \cos^2 B + 2 \sin B \cos B = 4x^2$$

3. Use the Pythagorean identity $\sin^2 \theta + \cos^2 \theta = 1$:

$$1 + 2 \sin A \cos A = 4x^2$$

$$1 + 2 \sin B \cos B = 4x^2$$

4. Subtract 1 from both sides:

$$2 \sin A \cos A = 4x^2 - 1$$

$$2 \sin B \cos B = 4x^2 - 1$$

.....
16. Since $\sin A = \cos A$ when $A = B$:

$$\sin A = \cos A \implies \sin A = \frac{\sqrt{2}}{2}$$

$$2 \sin^2 A = 1$$

17. Substitute back to find x :

$$x = -1 + 1 = 0$$

Thus, the value of $\cos C$ is:

$$\boxed{\frac{\sqrt{2}}{2}}$$

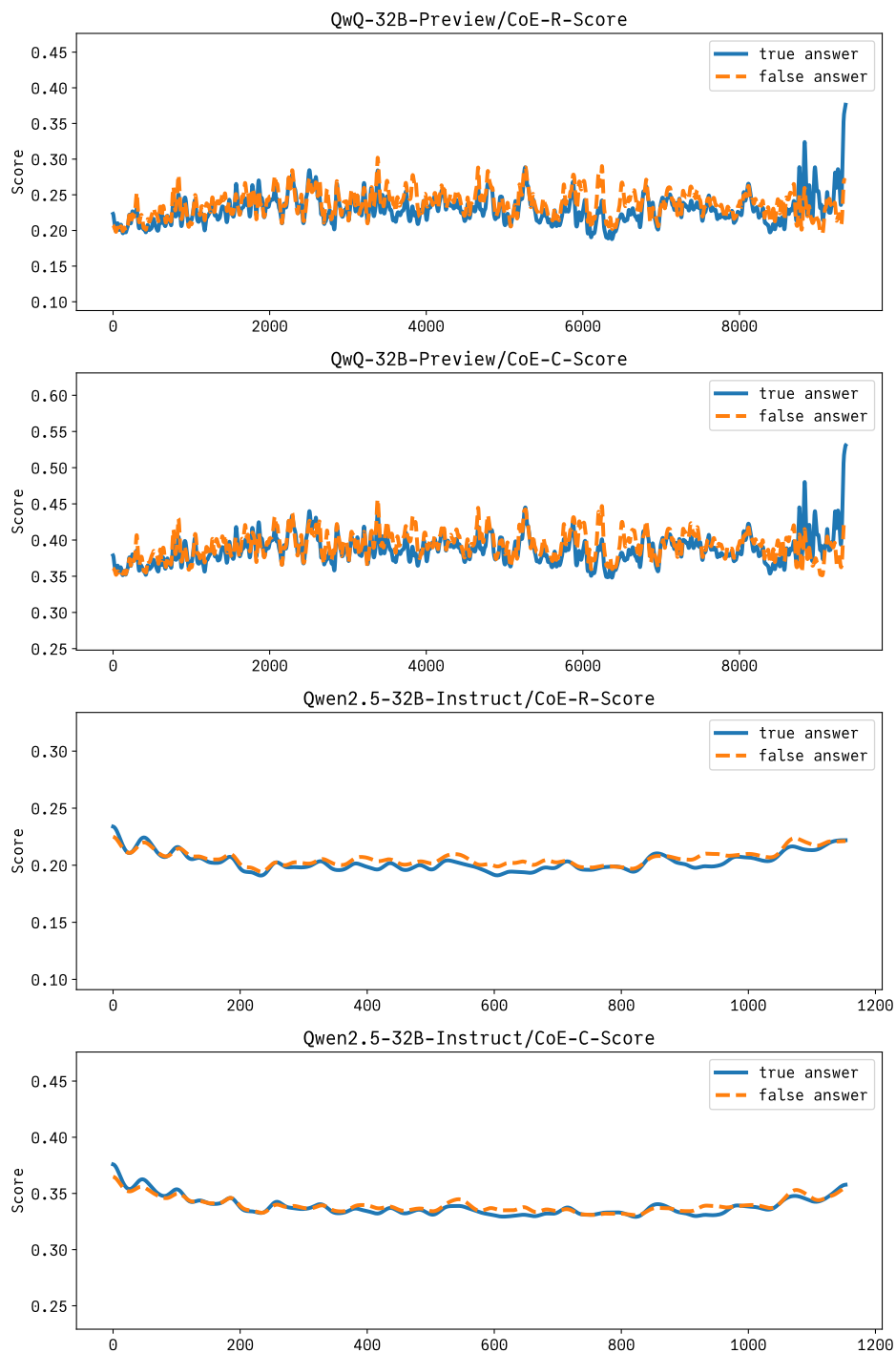


Figure 11: CoE-Score of QwQ and Qwen on Question 1.

Question 2

The sequence $\{a_n\}$ satisfies $a_1 = 1$, and for any positive integer n , we have $a_{n+1} = 10^n a_n^2$. What is the general term formula for $\{a_n\}$?

QwQ-32B-Preview Response for Question 2

Let's tackle this problem step by step. We have a sequence defined by a recursive formula, and we need to find a general term for it. The sequence is given by $a_1 = 1$, and for any positive integer n , $a_{n+1} = 10^n a_n^2$. Our goal is to find a formula for a_n in terms of n .

First, I'll try to compute the first few terms of the sequence to see if there's a pattern.

Given:

$$a_1 = 1$$

For $n = 1$:

$$a_2 = 10^1 \cdot a_1^2 = 10 \cdot (1)^2 = 10$$

For $n = 2$:

$$a_3 = 10^2 \cdot a_2^2 = 100 \cdot (10)^2 = 100 \cdot 100 = 10,000$$

For $n = 3$:

$$a_4 = 10^3 \cdot a_3^2 = 1,000 \cdot (10,000)^2 = 1,000 \cdot 100,000,000 = 10^{12}$$

.....

Wait a minute, let's check this with the earlier terms to verify.

For $n = 1$:

$$a_1 = 10^{2^1-1-1} = 10^{2-1-1} = 10^0 = 1$$

Correct.

For $n = 2$:

$$a_2 = 10^{2^2-2-1} = 10^{4-2-1} = 10^1 = 10$$

Correct.

For $n = 3$:

$$a_3 = 10^{2^3-3-1} = 10^{8-3-1} = 10^4 = 10,000$$

Correct.

For $n = 4$:

$$a_4 = 10^{2^4-4-1} = 10^{16-4-1} = 10^{11} = 100,000,000,000$$

Wait, but earlier I calculated $a_4 = 10^{12}$. Hmm, there's a discrepancy here.

Wait, let me double-check my earlier calculation for a_4 :

Given $a_3 = 10^4$, then:

$$a_4 = 10^3 \cdot (10^4)^2 = 10^3 \cdot 10^8 = 10^{11}$$

Ah, I made a mistake earlier. I thought $a_4 = 10^{12}$, but actually, it's 10^{11} . So, my general formula $a_n = 10^{2^n - n - 1}$ gives $a_4 = 10^{11}$, which matches the correct calculation. Great!

Therefore, the general term formula for the sequence $\{a_n\}$ is:

$$a_n = 10^{2^n - n - 1}$$

Final Answer

$$a_n = 10^{2^n - n - 1}$$

Qwen2.5-32B-Instruct-Preview Response Question 2

To find the general term formula for the sequence $\{a_n\}$ defined by $a_1 = 1$ and $a_{n+1} = 10^n a_n^2$ for any positive integer n , we will proceed step by step.

1. **Initial Condition:** $a_1 = 1$

2. **First Iteration:** $a_2 = 10^1 a_1^2 = 10 \cdot 1^2 = 10$

3. **Second Iteration:** $a_3 = 10^2 a_2^2 = 10^2 \cdot 10^2 = 10^4$

4. **Third Iteration:** $a_4 = 10^3 a_3^2 = 10^3 \cdot (10^4)^2 = 10^3 \cdot 10^8 = 10^{11}$

From these calculations, we observe a pattern in the exponents of 10. Let's hypothesize the general form of a_n and then prove it by induction.

5. **Hypothesis:** Suppose $a_n = 10^{b_n}$, where b_n is some function of n .

6. **Base Case:** For $n = 1$,

$$a_1 = 1 = 10^0 \implies b_1 = 0$$

7. **Inductive Step:** Assume $a_k = 10^{b_k}$ for some $k \geq 1$. We need to show that $a_{k+1} = 10^{b_{k+1}}$.

By the recurrence relation,

$$a_{k+1} = 10^k a_k^2 = 10^k (10^{b_k})^2 = 10^k \cdot 10^{2b_k} = 10^{k+2b_k}$$

Thus, if $a_k = 10^{b_k}$, then $a_{k+1} = 10^{b_{k+1}}$ with $b_{k+1} = k + 2b_k$.

8. **Solving the Recurrence for b_n :** We have the recurrence relation:

$$b_{k+1} = k + 2b_k$$

with the initial condition $b_1 = 0$.

To solve this, we will find a pattern by calculating the first few terms:

$$b_1 = 0$$

$$b_2 = 1 + 2b_1 = 1 + 2 \cdot 0 = 1$$

$$b_3 = 2 + 2b_2 = 2 + 2 \cdot 1 = 4$$

$$b_4 = 3 + 2b_3 = 3 + 2 \cdot 4 = 11$$

Observing the pattern, we hypothesize:

$$b_n = \frac{2^n - n - 2}{3}$$

9. **Verification:** We verify by induction: - Base case $n = 1$:

$$b_1 = \frac{2^1 - 1 - 2}{3} = \frac{2 - 1 - 2}{3} = \frac{-1}{3} = 0$$

- Inductive step: Assume $b_k = \frac{2^k - k - 2}{3}$. Then,

$$b_{k+1} = k + 2b_k = k + 2 \left(\frac{2^k - k - 2}{3} \right) = \frac{3k + 2^{k+1} - 2k - 4}{3} = \frac{2^{k+1} + k - 4}{3}$$

which matches our hypothesis.

Therefore, the general term of the sequence is:

$$a_n = 10^{b_n} = 10^{\frac{2^n - n - 2}{3}}$$

The final answer is:

$$\boxed{10^{\frac{2^n - n - 2}{3}}}$$

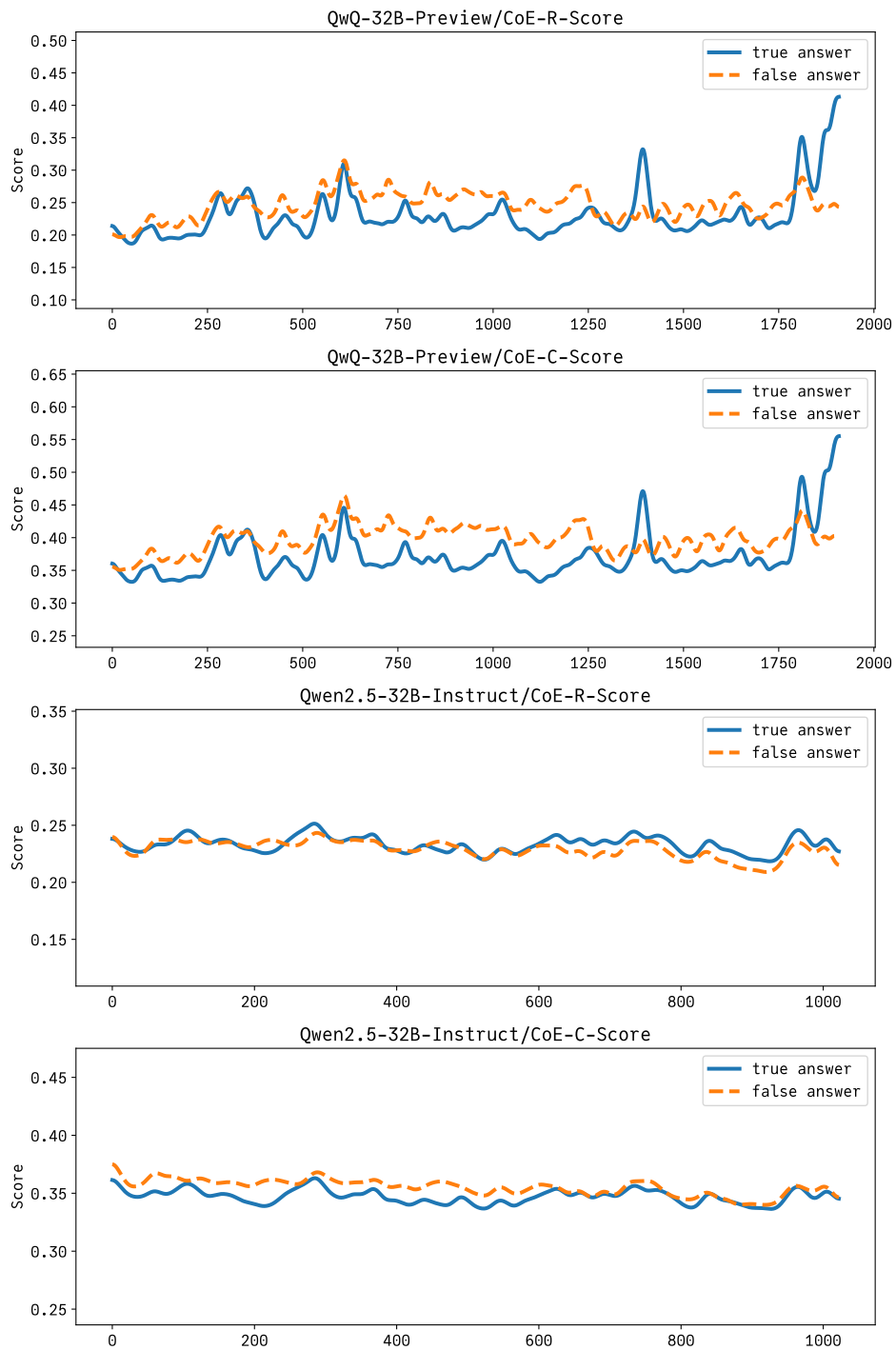


Figure 12: CoE-Score of QwQ and Qwen on Question 2.

Question 3

If the real part of the complex number z is greater than 0, and $\bar{z}(z + 1) = \frac{20}{3+i}$, then $z = ?$

QwQ-32B-Preview Response for Question 3

Let's tackle this problem step by step. So, we have a complex number z with a positive real part, and it satisfies the equation $\bar{z}(z + 1) = \frac{20}{3+i}$. We need to find what z is.

First, let's recall that for any complex number $z = a + bi$, where a and b are real numbers, the complex conjugate \bar{z} is $a - bi$.

So, if $z = a + bi$, then $\bar{z} = a - bi$.

Given that, the equation becomes:

$$(a - bi)((a + bi) + 1) = \frac{20}{3+i}$$

Simplify the left side:

First, compute $z + 1 = (a + bi) + 1 = (a + 1) + bi$.

Then, $\bar{z}(z + 1) = (a - bi)((a + 1) + bi)$.

Let's multiply these two complex numbers:

$$(a - bi)((a + 1) + bi) = a(a + 1) + a(bi) - bi(a + 1) - bi(bi) \\ = a(a + 1) + abi - bai - bi + b^2i^2$$

Wait, that seems messy. Maybe there's a better way to expand this.

Actually, let's use the distributive property more carefully:

$$(a - bi)((a + 1) + bi) = a(a + 1) + a(bi) - bi(a + 1) - bi(bi)$$

Now, compute each term:

$$a(a + 1) = a^2 + a$$

$$a(bi) = abi$$

$$-bi(a + 1) = -bai - bi$$

$$-bi(bi) = -b^2i^2 = -b^2(-1) = b^2$$

Now, combine all these:

$$a^2 + a + abi - bai - bi + b^2$$

Notice that $abi - bai = 0$, since they cancel each other out.

So, we're left with:

$$a^2 + a + b^2 - bi$$

Wait, but that doesn't seem right because the imaginary parts should cancel out completely since the right side is a real number (as we'll see when we simplify it).

Wait, actually, no. Let's check the right side:

$$\frac{20}{3+i}$$

.....

However, the problem states that the real part of z is greater than 0. So, $a > 0$.

Therefore, $a = 1$ (since $a = -2$ is less than 0 and doesn't satisfy the condition).

So, $z = a + bi = 1 + 2i$.

Let me verify this solution by plugging back into the original equation.

Compute $\bar{z}(z + 1)$:

$$\bar{z} = 1 - 2i$$

$$z + 1 = (1 + 2i) + 1 = 2 + 2i$$

$$\text{Now, } \bar{z}(z + 1) = (1 - 2i)(2 + 2i)$$

Compute this product:

$$(1)(2) + (1)(2i) + (-2i)(2) + (-2i)(2i) = 2 + 2i - 4i - 4i^2 = 2 - 2i - 4(-1) = 2 - 2i + 4 = 6 - 2i$$

And the right side is $\frac{20}{3+i} = 6 - 2i$, as we calculated earlier.

So, both sides match, confirming that $z = 1 + 2i$ is indeed the correct solution.

Final Answer

$$1 + 2i$$

Qwen2.5-32B-Instruct-Preview Response Question 3

To solve the problem, let's first set up the equation and use the given conditions to find the complex number z .

Given: $\bar{z}(z + 1) = \frac{20}{3+i}$

Let's start by simplifying the right-hand side of the equation:

$$\frac{20}{3+i} = \frac{20(3-i)}{(3+i)(3-i)} = \frac{20(3-i)}{3^2 - i^2} = \frac{20(3-i)}{9+1} = \frac{20(3-i)}{10} = 2(3-i) = 6 - 2i$$

So, the equation becomes:

$$\bar{z}(z + 1) = 6 - 2i$$

Let $z = x + yi$ where $x, y \in \mathbb{R}$ and $x > 0$. Then, $\bar{z} = x - yi$.

Substituting z and \bar{z} into the equation:

$$(x - yi)((x + yi) + 1) = 6 - 2i$$

Simplifying the left-hand side:

$$\begin{aligned} (x - yi)(x + 1 + yi) &= (x - yi)(x + 1 + yi) \\ &= x(x + 1) + xyi - yix - yi^2 + (-yi)(yi) \\ &= x(x + 1) + y^2 + xyi - xyi \end{aligned}$$

Since $i^2 = -1$, we get:

$$x(x + 1) + y^2 + xyi - xyi = x^2 + x + y^2$$

So, the equation becomes:

$$x^2 + x + y^2 = 6 - 2i$$

Separating the real and imaginary parts, we have:

$$x^2 + x + y^2 = 6 \quad \text{and} \quad 0 = -2i$$

Since the imaginary part of the left-hand side is zero and the imaginary part of the right-hand side is $-2i$, it must be that the imaginary part of z is such that it cancels out. This means $y = -2$.

Substituting $y = -2$ into the real part equation:

$$x^2 + x + (-2)^2 = 6$$

$$x^2 + x + 4 = 6$$

$$x^2 + x - 2 = 0$$

Solving the quadratic equation $x^2 + x - 2 = 0$:

$$x = \frac{-1 \pm \sqrt{1+8}}{2} = \frac{-1 \pm 3}{2}$$

So, $x = 1$ or $x = -2$. Since the real part of z is greater than 0, we have $x = 1$.

Thus, $z = 1 - 2i$.

The solution is:

$$\boxed{1 - 2i}$$

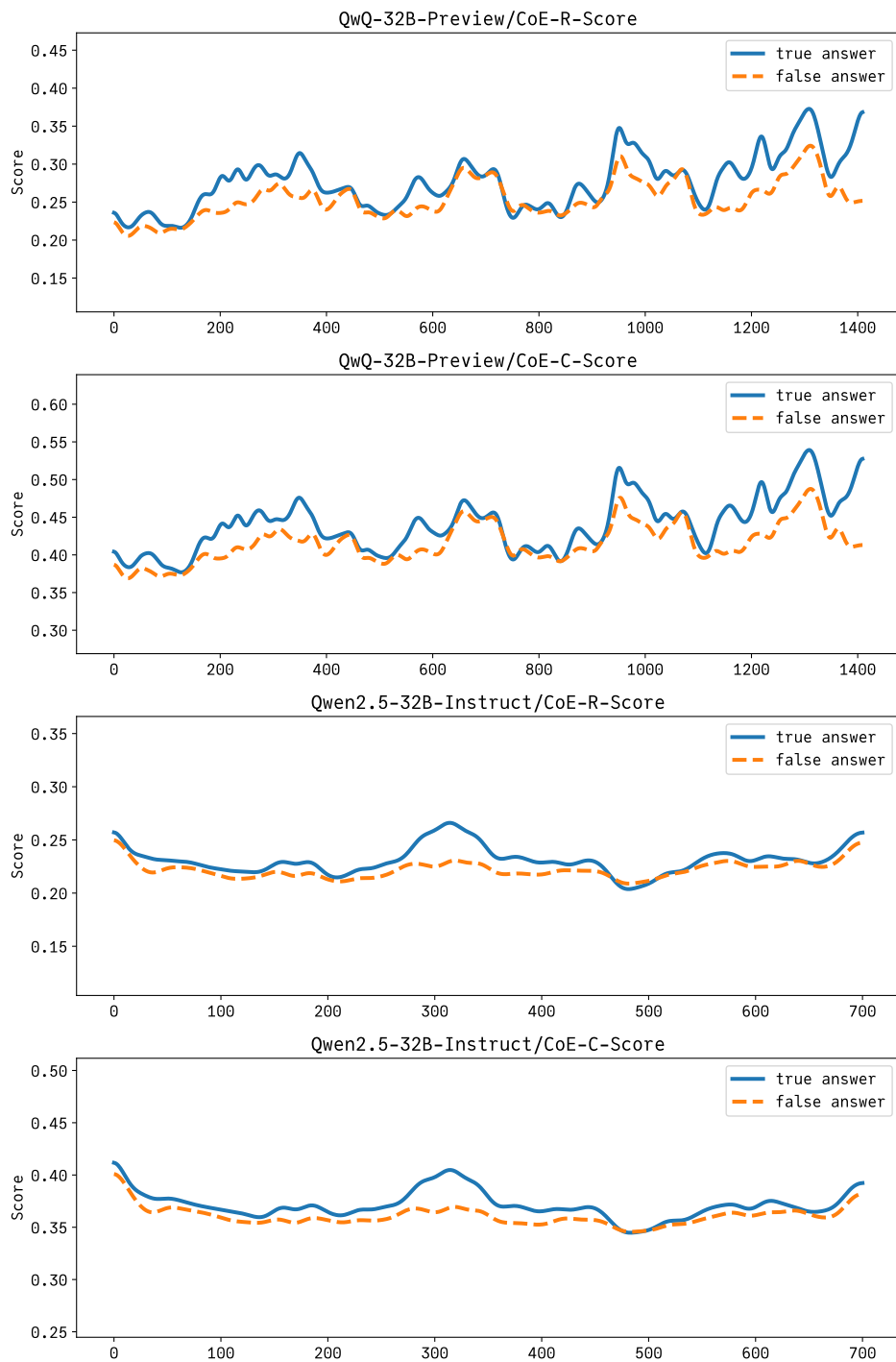


Figure 13: CoE-Score of QwQ and Qwen on Question 3.

Question 4

Integers a and b are randomly chosen without replacement from the set of integers with absolute value not exceeding 10. What is the probability that the polynomial $x^3 + ax^2 + bx + 6$ has 3 distinct integer roots?

QwQ-32B-Preview Response for Question 4

So I've got this probability problem here. It says that integers a and b are randomly chosen without replacement from the set of integers with absolute value not exceeding 10. So, first things first, I need to understand what that set is. If the absolute value of the integers doesn't exceed 10, that means they can range from -10 to 10, inclusive. So the set is $\{-10, -9, -8, \dots, 9, 10\}$. That's 21 integers in total.

Now, we're choosing a and b without replacement from this set, meaning that a and b are distinct integers from this set. So the total number of possible pairs (a, b) is $21 \times 20 = 420$, since there are 21 choices for a and then 20 remaining choices for b .

The problem is asking for the probability that the polynomial $x^3 + ax^2 + bx + 6$ has 3 distinct integer roots. So, for the polynomial to have 3 distinct integer roots, it must be factorable into $(x - r)(x - s)(x - t)$, where r, s, t are distinct integers.

Let's expand $(x - r)(x - s)(x - t)$ to see how the coefficients relate to a and b :

$$(x - r)(x - s)(x - t) = x^3 - (r + s + t)x^2 + (rs + rt + st)x - rst$$

Comparing this to the given polynomial $x^3 + ax^2 + bx + 6$, we can see that:

$$\begin{cases} -(r + s + t) = a \\ rs + rt + st = b \\ -rst = 6 \end{cases}$$

.....

All these calculations seem correct.

Now, is there any possibility that different sets of roots could lead to the same (a, b) pair? For example, if two different triplets of roots result in the same a and b , then that would mean that pair is counted only once in the favorable outcomes.

But in our case, each set of roots gives a distinct (a, b) pair, as we can see:

- $(0, -7)$
- $(-2, -5)$
- $(-6, -1)$
- $(-4, 1)$

No overlaps here.

Therefore, the number of favorable outcomes is indeed 4.

Total possible outcomes are $21 \times 20 = 420$.

Thus, the probability is $\frac{4}{420} = \frac{1}{105}$.

Wait, but perhaps I should consider that a and b are chosen without replacement, meaning that (a, b) and (b, a) are different unless $a = b$, which isn't the case here. But in our favorable outcomes, each (a, b) is unique and distinct from (b, a) , since none of the a and b pairs have $a = b$. Therefore, no need to adjust for that.

Alternatively, if the problem considered (a, b) and (b, a) as the same, but since it's about choosing without replacement and the polynomial has distinct roots, I think the ordered pairs are appropriate here.

So, I think $\frac{1}{105}$ is the correct probability.

Final Answer

$$\frac{1}{105}$$

Qwen2.5-32B-Instruct-Preview Response Question 4

To solve this problem, we need to analyze the conditions under which the polynomial $x^3 + ax^2 + bx + 6$ has three distinct integer roots.

1. **Identify the Set of Integers:** The set of integers with absolute value not exceeding 10 is $\{-10, -9, -8, \dots, 9, 10\}$. There are 21 integers in this set.

2. **Polynomial Roots:** Let the roots of the polynomial $x^3 + ax^2 + bx + 6$ be $r, s,$ and t . By Vieta's formulas, we know:

$$r + s + t = -a, \quad rs + rt + st = b, \quad rst = -6$$

3. **Identifying Integer Roots:** The integer roots $r, s,$ and t must satisfy $rst = -6$. We list the possible sets of integer roots:

$$\{1, 2, -3\}, \{1, -2, 3\}, \{-1, 2, 3\}, \{-1, -2, -3\}$$

4. **Counting Valid Combinations:** We need to count the number of ways to choose a and b such that the polynomial has these roots. Each set of roots $\{r, s, t\}$ must be chosen from the set $\{-10, -9, \dots, 9, 10\}$ without replacement.

5. **Calculate a and b for Each Set:** For each set of roots, calculate a and b : - For $\{1, 2, -3\}$:

$$r + s + t = 1 + 2 - 3 = 0 \implies a = 0$$

$$rs + rt + st = 1 \cdot 2 + 1 \cdot (-3) + 2 \cdot (-3) = 2 - 3 - 6 = -7 \implies b = -7$$

- For $\{1, -2, 3\}$:

$$r + s + t = 1 - 2 + 3 = 2 \implies a = -2$$

$$rs + rt + st = 1 \cdot (-2) + 1 \cdot 3 + (-2) \cdot 3 = -2 + 3 - 6 = -5 \implies b = -5$$

- For $\{-1, 2, 3\}$:

$$r + s + t = -1 + 2 + 3 = 4 \implies a = -4$$

$$rs + rt + st = (-1) \cdot 2 + (-1) \cdot 3 + 2 \cdot 3 = -2 - 3 + 6 = 1 \implies b = 1$$

- For $\{-1, -2, -3\}$:

$$r + s + t = -1 - 2 - 3 = -6 \implies a = 6$$

$$rs + rt + st = (-1) \cdot (-2) + (-1) \cdot (-3) + (-2) \cdot (-3) = 2 + 3 + 6 = 11 \implies b = 11$$

6. **Total Possible Choices for a and b :** The total number of ways to choose a and b from the set of integers $\{-10, -9, \dots, 9, 10\}$ without replacement is:

$$\binom{21}{2} = \frac{21 \times 20}{2} = 210$$

7. **Probability Calculation:** There are 4 valid combinations of (a, b) that make the polynomial have three distinct integer roots. Therefore, the probability is:

$$\frac{4}{210} = \frac{2}{105}$$

Thus, the probability that the polynomial $x^3 + ax^2 + bx + 6$ has three distinct integer roots is

$$\boxed{\frac{2}{105}}$$

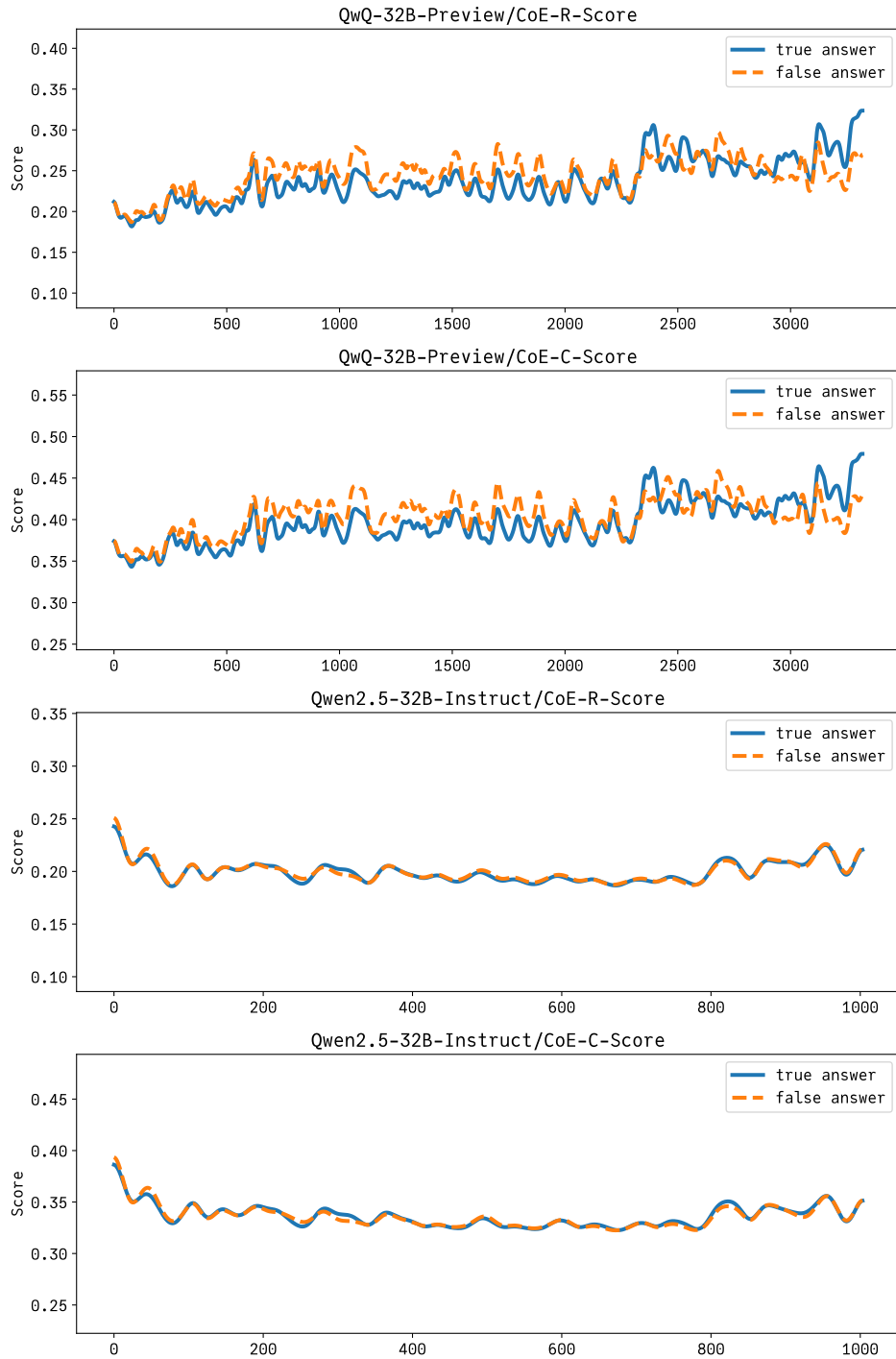


Figure 14: CoE-Score of QwQ and Qwen on Question 4.