Large Language Models Struggle with Unreasonability in Math Problems

Jingyuan Ma[1], Damai Dai[1], Zihang Yuan[2], Rui Li[1], Weilin Luo[3], Bin Wang[3], Qun Liu[3], Lei Sha[2], Zhifang Sui[1]*

[1]State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University
[2]Institute of Artificial Intelligence, Beihang University
[3]Huawei Noah's Ark Lab, China
mjy@stu.pku.edu.cn

Abstract

Large Language Models (LLMs) have shown remarkable success across a wide range of math and reasoning benchmarks. However, we observe that they often struggle when faced with unreasonable math problems. Instead of recognizing these issues, the models frequently proceed as if the problems are well posed, producing incorrect answers or overthinking and producing verbose self-corrections. To systematically investigate this overlooked vulnerability, we propose the **Unreasonable Math Problems (UMP)** benchmark, designed to evaluate LLMs' ability to detect and respond to unreasonable math problems. Based on extensive experiments covering 19 LLMs, we find that even state-of-the-art general models such as GPT-4o struggle on UMP. Reasoning models such as DeepSeek-R1 demonstrate higher sensitivity to unreasonable inputs; however, this sensitivity often comes at the cost of generating overly long and meaningless responses that fail to converge. We further find that prompting and fine-tuning enhance the detection of unreasonable inputs with minor and acceptable trade-offs that make them practical solutions in this challenging setting.

1 Introduction

Large language models (LLMs) have recently shown impressive performance on advanced mathematical reasoning tasks, especially on benchmarks such as MATH (Hendrycks et al. 2021) and AIME24 (MAA 2024). However, we find that these models often fail to detect logical flaws or unreasonable assumptions in math problems, treating them as if they were well-posed. Instead of flagging such issues, they tend to generate confident yet nonsensical answers or fall into endless reasoning loops without reaching a valid conclusion. This counter-intuitive behavior raises serious concerns about their reliability in real-world applications such as automated tutoring (Kasneci et al. 2023), early education (Zhang et al. 2024) and open-domain problem solving (Lin and Chen 2023), where misleading answers to unreasonable questions can undermine trust and lead to negative outcomes.

To enable a comprehensive analysis of how LLMs behave when confronted with mathematically unreasonable inputs, we introduce the Unreasonable Math Problems (UMP) benchmark. We construct UMP by minimally editing questions from existing math datasets, including MATH (Hendrycks et al. 2021), AIME24 (MAA 2024), AMC23, and GSM8K (Cobbe et al. 2021a), to create unreasonable variants that contain logical inconsistencies, missing assumptions, or ill-defined objectives. These edits are guided by rule-based transformation templates and executed

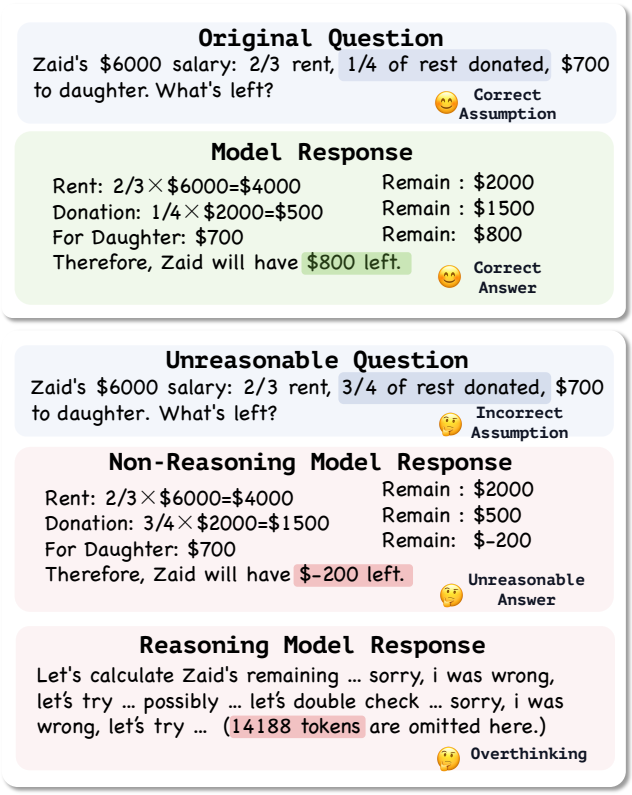

Figure 1: An example showing the contrast between a model's response to a well-posed question and its response to an unreasonable variant. While the model correctly solves the original problem, its response to the unreasonable version becomes less satisfactory in terms of clarity, coherence, and logical consistency.

*Corresponding author

with DeepSeek-R1, while preserving the original problem's structure, topic, and surface form. All generated questions are manually verified to ensure they are mathematically flawed yet still natural.

To evaluate model behavior, we present models with both the original and its corresponding unreasonable version. This paired setup enables us to directly attribute behavioral differences to the presence of unreasonableness, isolating it from other factors such as question length or difficulty.

Based on the UMP benchmark, we conduct extensive experiments on various models spanning three categories: general-purpose models (e.g., GPT-4o (OpenAI 2024)), reasoning models (e.g., DeepSeek-R1 (DeepSeek-AI 2025)), and math-specialized models (e.g., Qwen-Math (Yang et al. 2024a)). In addition, we analyze LLMs' failure patterns according to token repetition, reflection frequency, and token entropy. We find: 1) general-purpose models often proceed confidently without recognizing the unreasonableness of the question; 2) reasoning models tend to overthink and fall into excessive self-correction; 3) math-specialized models may fail to initiate reasoning when confronted with unreasonable premises. We further explore several mitigation strategies but find that none can robustly resolve these failure modes without introducing trade-offs, such as decreased performance on standard inputs, highlighting the need for future research into more principled and generalizable solutions.

Our main contributions are as follows:

- We propose the **Unreasonable Math Problems (UMP)** benchmark to more accurately and comprehensively evaluate how LLMs respond to mathematically unreasonable problem statements.
- We find that even high-performing models often fail to detect unreasonableness or produce overconfident and overly verbose responses to unreasonable questions.
- We show that simple prompting or fine-tuning can alleviate this challenging task while maintaining an acceptable trade-off.

2 Unreasonable Math Problems Benchmark

While large language models have demonstrated strong performance on standard mathematical reasoning benchmarks, they often produce inaccurate or confusing responses when presented with mathematically unreasonable problems—questions that contain flawed assumptions, undefined variables, or logical inconsistencies (as shown in Figure 1). To systematically evaluate model behavior under such conditions, we construct the **Unreasonable Math Problems (UMP)** benchmark, which consists of more than 1,000 unreasonable math problems, each paired with its corresponding original version, focusing on assessing LLMs' ability to detect and respond to irrational inputs.

2.1 Types of Unreasonableness

We identify five prevalent types of mathematical unreasonableness that frequently lead to LLM failures: (1) **Undefined Variables** (UV), where essential information is missing; (2) **Illogical Scenarios** (IS), involving situations that defy real-world logic; (3) **Incorrect Assumptions** (IA), based on mathematically invalid premises; (4) **Misinterpretation of Units** (MU), involving incorrect or inconsistent use of measurement units; and (5) **Inconsistent Conditions** (IC), where internal contradictions render the problem unsolvable. Each instance in our benchmark is formatted as a quintuple (q, a, q', t, e), where q is the original question, a is its answer, q' is the unreasonable variant, t denotes its unreasonableness type, and e provides an explanation of why q' is flawed. Further details and examples can be found in Appendices.

2.2 LLM-Guided Construction

Our data construction process is inspired by MetaMath (Yu et al. 2023), which leverages LLMs to produce problem variants under controlled transformations. As illustrated in Figure 2, we begin with test-set questions drawn from four widely used math benchmarks: GSM8K (Cobbe et al. 2021a), a collection of grade-school-level problems; MATH (Hendrycks et al. 2021), which covers formal secondary-school mathematics; and AIME24 (MAA 2024) and AMC23, both of which contain high-level competition problems with symbolic or abstract formats. We manually construct a set of transformation rules corresponding to the five types of mathematical unreasonableness and use them to guide an LLM in generating unreasonable variants for each original question, along with natural-language explanations of why the modified version is irrational. To ensure the unreasonable variant remains close in surface form to the original, we compute cosine similarity between the sentence embeddings of the original and generated questions using SimCSE (Gao, Yao, and Chen 2021), a contrastively trained BERT-based model (Devlin et al. 2019). Only variants whose similarity exceeds a predefined threshold k are retained for human verification.

2.3 Validation Checking

We manually verify each example to ensure the unreasonable variant introduces genuine mathematical unreasonableness while maintaining surface similarity with the original problem. Only examples satisfying both criteria are retained. Our transformation rules are carefully designed to embed logical flaws. We discard cases that either lack meaningful irrationality or make the flaw overly explicit (see the annotation protocol in Appendix). In such cases, we revise the examples by adjusting entities and numerical values to ensure the unreasonableness remains logically subtle yet plausible.

2.4 Benchmark Composition

Table 1 summarizes the distribution of unreasonableness types across different source datasets in the UMP benchmark. A large proportion of examples fall into the categories of *Incorrect Assumptions (IA)* and *Inconsistent Conditions (IC)*. This distribution emerges from the model generation and human filtering process. In particular, these two types of flaws tend to produce more plausible and contextually coherent questions, making them more likely to be retained during human review. Compared with GSM8K and MATH, the AMC23 and AIME24 datasets contain far fewer test questions (e.g., AIME24 has only 30), which inherently limits

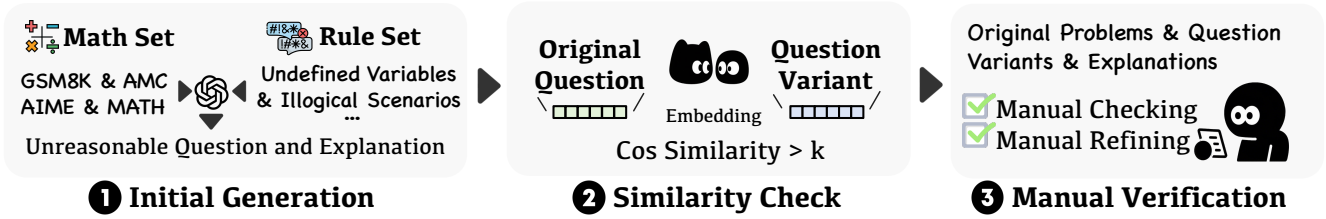

Figure 2: Overview of the UMP (Unreasonable Math Problems) generation pipeline. The process consists of three stages: (1) **Initial Generation**: original questions from GSM8K, MATH, AMC, and AIME datasets are paired with a rule set (e.g., Undefined Variables, Illogical Scenarios) to produce unreasonable variants via LLM prompts; (2) **Similarity Check**: generated questions are filtered based on cosine similarity to ensure surface closeness to the original; (3) **Manual Verification**: human annotators check and refine the generated variants and explanations to ensure clarity, coherence, and logical consistency.

Dataset	UV	IS	IA	MU	IC	Total
GSM8K	14.8%	7.3%	36.8%	16.7%	24.3%	682
MATH	7.2%	4.4%	41.7%	13.1%	33.6%	405
AMC23	0.0%	4.3%	43.5%	4.3%	47.8%	23
AIME24	10.5%	0.0%	31.6%	0.0%	57.9%	19
Total	8.1%	4.0%	38.4%	8.5%	40.9%	1129

Table 1: Joint distribution of unreasonableness types and datasets in the UMP benchmark. Abbreviations: UV = Undefined Variables, IS = Illogical Scenarios, IA = Incorrect Assumptions, MU = Misinterpreted Units, IC = Inconsistent Conditions.

the number of examples we can derive from them. In addition, these problems are often highly abstract and symbolic, making it difficult to apply natural, controlled perturbations without compromising their integrity. We include modified examples only when the unreasonableness can be introduced in a natural and plausible way. This selective inclusion ensures that the final benchmark remains both diverse and faithful to the original problem distributions.

3 Evaluating LLMs on UMP

3.1 Evaluation Setup

We evaluate a diverse set of LLMs spanning three major categories. **General-purpose models** are primarily trained for instruction following and open-domain tasks; this group includes models such as DeepSeek-V3 (DeepSeek-AI 2024), GPT-4o (OpenAI 2024), Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct (Qwen Team 2024), and Llama-3.1-8B-Instruct (MetaAI 2024). **Math-specialized models** are fine-tuned on mathematical corpora and optimized for numerical reasoning; we evaluate Qwen2.5-Math-1.5B-Instruct, Qwen2.5-Math-7B-Instruct (Yang et al. 2024a), and DeepSeek-Math-7B-Instruct (Shao et al. 2024). **Reasoning-enhanced models** are designed to support complex multi-step reasoning and include models such as DeepSeek-R1 and its distilled variants (e.g., DeepSeek-R1-Distill-Qwen-7B) (DeepSeek-AI 2025), as well as QwQ-32B-Preview (Team 2025) and Grok-3-Reasoning (xAI 2024). All models are evaluated on both the original problems and their unreasonable variants. All experiments were conducted on NVIDIA A100 GPUs (80 GB and 40 GB each). We ran smaller open-source models locally and queried larger models via API. Sampling settings followed common practice: Non-reasoning models used greedy decoding, whereas reasoning-enhanced models were sampled with temperature=0.6 and API models were sampled with their default temperature. [1]

3.2 Evaluation Metrics

Evaluating performance on unreasonable questions differs fundamentally from standard accuracy-based evaluations. Instead of checking for correct answers, the goal is to assess whether the model can recognize flawed premises and respond appropriately by rejecting or questioning, or flagging the input. To this end, we introduce two complementary metrics:

Absolute score measures the proportion of unreasonable questions for which the model explicitly identifies the problem as flawed or unanswerable.

Relative score conditions on the model's ability to correctly solve the original version of a question. It is defined as the proportion of corresponding unreasonable variants that are correctly recognized as unreasonable, among those for which the model answers the original (reasonable) version correctly. This design accounts for cases where the flaw only becomes apparent during intermediate reasoning steps. A model that lacks the necessary mathematical competence may never reach the point where the unreasonableness is revealed. By restricting the evaluation to questions the model can already solve in their original form, the relative score provides a fairer measure of its capacity to recognize flawed inputs.

Absolute Score Following the LLM-as-a-judge framework (Zheng et al. 2023), we use DeepSeek-V3 to label each answer as **A** (correctly identifies the main source of unreasonableness, explains it coherently, and proposes a valid fix without new errors), **B** (partially detects or vaguely justifies the flaw, with gaps, circular logic, or minor inconsistencies), or **C** (fails to spot the core flaw, misreads the reasoning,

[1]Details of model versions, inference configurations and standard errors for repeated runs are provided in Appendices.

Model	GSM			MATH			AMC+AIME			AVG		
	Acc	Abs	Rel	Acc	Abs	Rel	Acc	Abs	Rel	Acc	Abs	Rel
General Models												
Qwen2.5-3B-Instruct	0.893	0.323	0.339	0.706	0.370	0.405	0.333	0.202	0.286	0.794	0.335	0.360
Qwen2.5-7B-Instruct	0.929	0.440	0.440	0.819	0.428	0.465	0.303	0.250	0.178	0.858	0.429	0.445
Llama-3.1-8B-Instruct	0.872	0.227	0.238	0.548	0.205	0.238	0.182	0.191	0.340	0.714	0.218	0.239
DeepSeek-V3	0.965	0.560	0.562	0.933	0.658	**0.672**	0.576	0.536	**0.490**	0.935	0.594	0.600
GPT-4o	0.955	0.640	**0.657**	0.803	0.565	0.564	0.242	0.476	0.468	0.863	0.607	**0.624**
Claude-3.5-Sonnet	0.936	0.586	0.605	0.793	0.480	0.503	0.455	0.309	0.351	0.858	0.538	0.566
Reasoning Models												
R1-Distill-Qwen-1.5B	0.858	0.377	0.397	0.836	0.615	0.633	0.394	0.643	0.704	0.829	0.472	0.489
Marco-o1	0.893	0.539	0.546	0.736	0.554	0.567	0.273	0.429	0.503	0.804	0.540	0.553
R1-Distill-Qwen-7B	0.917	0.570	0.600	0.923	0.732	0.731	0.667	0.738	0.746	0.908	0.635	0.653
R1-Distill-Llama-8B	0.891	0.581	0.593	0.900	0.704	0.706	0.485	0.702	0.668	0.877	0.630	0.636
R1-Distill-Qwen-32B	0.967	0.725	0.734	0.920	0.757	0.768	0.576	0.762	0.802	0.931	0.738	0.748
QwQ-32B-Preview	0.957	0.550	0.560	0.900	0.693	0.697	0.606	0.643	0.672	0.919	0.605	0.612
DeepSeek-R1	0.972	0.830	0.844	0.950	0.806	0.810	0.879	0.667	0.609	0.959	0.815	0.824
Grok-3-Reasoning	0.967	0.875	**0.884**	0.913	0.918	**0.927**	0.879	0.893	**0.925**	0.942	0.891	**0.901**
Math Models												
Qwen2.5-Math-1.5B	0.844	0.326	**0.356**	0.783	0.385	**0.416**	0.394	0.202	0.268	0.800	0.343	**0.376**
MetaMath-Mistral-7B	0.770	0.125	0.146	0.304	0.089	0.128	0.061	0.036	0.000	0.566	0.110	0.143
DeepSeek-Math-7B	0.844	0.141	0.158	0.518	0.200	0.225	0.091	0.107	0.125	0.682	0.161	0.177
NuminaMath-7B-CoT	0.725	0.276	0.320	0.572	0.285	0.366	0.151	0.179	0.188	0.639	0.276	0.335
Qwen2.5-Math-7B	0.962	0.222	0.231	0.853	0.348	0.357	0.394	0.214	**0.272**	0.894	0.267	0.276

Table 2: Model performance metrics across datasets by category. Here, **Acc** denotes the accuracy of the model on original (well-posed) problems, while **Abs** and **Rel** refer to the Absolute Score and Relative Score on unreasonable problems, respectively. Among them, the **bold** ones are the models of each category with the highest **Rel** on different datasets.

or introduces new contradictions). Formally, let V denote the set of all responses to unreasonable questions, and let $E(v) \in \{\text{A}, \text{B}, \text{C}\}$ represent the evaluation label assigned to a response $v \in V$. We define a soft scoring function $\delta(E)$ that maps each label to a numeric value:

$$\delta(E) = \begin{cases} 1 & \text{if } E = \text{A}, \\ 0.5 & \text{if } E = \text{B}, \\ 0 & \text{if } E = \text{C}. \end{cases}$$

The final absolute score is computed as the average soft score across all unreasonable questions:

$$\text{Absolute Score} = \frac{1}{|V|} \sum_{v \in V} \delta(E(v)).$$

To ensure alignment with our evaluation criteria, we design an in-context learning (ICL) prompt that includes annotated examples for each rating level, following best practices for LLM-based evaluation (Dong et al. 2024).

Relative Score A model may fail to detect unreasonableness simply because it cannot solve the original problem. To control for this, we define the **relative score** as the probability of detecting unreasonableness conditioned on solving the original question (Yang et al. 2024b):

$$P(\text{Detect Unreasonable} \mid \text{Solve Original}).$$

Concretely, we generate k responses for each original question q and compute its average accuracy. Questions with accuracy above a threshold τ form the confidently solved set Q_τ^+. For each such question q, let $V(q) \subseteq V$ denote the set of unreasonable variants derived from q. The relative score S_{rel} is then computed by averaging the soft scores across all variants in $V(q)$ for all $q \in Q_\tau^+$:

$$S_{\text{rel}} = \frac{1}{|Q_\tau^+|} \sum_{q \in Q_\tau^+} \left(\frac{1}{|V(q)|} \sum_{v \in V(q)} \delta(E(v)) \right).$$

This formulation ensures that the evaluation focuses on cases where the model has already demonstrated sufficient problem-solving ability, providing a fairer measure of its capacity to recognize flawed inputs.

3.3 Experimental Results

Table 2 reports model performance on the UMP benchmark using three metrics: accuracy on original problems (Acc), and absolute and relative scores (Abs, Rel) on unreasonable variants. We categorize models into three groups. **General-purpose models** such as GPT-4o, Claude-3.5-Sonnet, and DeepSeek-V3 perform well on original problems (e.g., GPT-4o: 0.863 Acc) but exhibit only mediocre robustness on unreasonable questions (0.607 Abs). Smaller models like Qwen2.5-3B and Llama-3.1-8B perform poorly across all metrics. **Reasoning-enhanced models** (e.g., DeepSeek-R1, Grok-3-Reasoning) achieve the highest behavioral scores, with up to 0.891 Abs and 0.901 Rel, indicating strong detection of unreasonableness. **Math-specialized models** (e.g.,

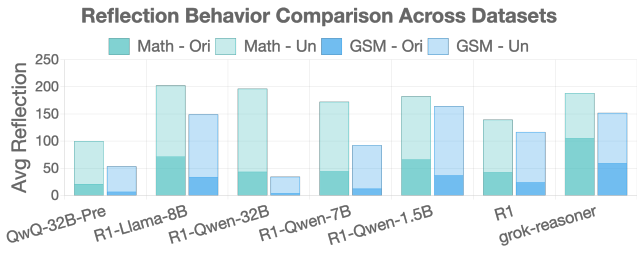

Figure 3: Average reflection counts for original and unreasonable problems. Reflections increase significantly on unreasonable problems, highlighted by light-colored segments. Ori and Un indicate the original and unreasonable problems, respectively.

Qwen2.5-Math-7B, MetaMath-Mistral-7B) excel in original accuracy (up to 0.894 Acc) but underperform on unreasonable variants (mostly below 0.30 Abs), highlighting a gap between solving and evaluating problem soundness.[2]

3.4 Overthinking on Reasoning Models

A prominent behavioral failure we observe is *overthinking*, where the model enters repeated cycles of reflection and revision without making meaningful progress (Sui et al. 2025; Cuadron et al. 2025). We analyze how often models revise their reasoning mid-generation by counting reflective phrases such as "rethink" or "I misunderstood" (see Appendix for keyword list). This frequency serves as a proxy for the model's self-correction behavior when faced with unreasonable inputs. Figure 3 shows that reasoning-enhanced models exhibit a large increase in reflection frequency on unreasonable questions compared to original ones. In contrast, this pattern is not observed in general-purpose or math-specialized models. This result suggests that reasoning models are more sensitive to flawed inputs, leading to more frequent internal reevaluation.[3]

3.5 Lexical Collapse and Redundancy in Response

In addition to analyzing reflection frequency, we further examine how model outputs deteriorate under unreasonable conditions by introducing two complementary indicators: **normalized token entropy** and **token-level repetition**. These metrics are designed to quantify lexical diversity and redundancy, offering additional perspectives on generation instability across model types.

Normalized Token Entropy. To assess lexical diversity, we compute normalized token entropy based on the empirical token frequency distribution in each response. For an output sequence of length T, let f_i be the count of token i, and define its empirical probability as $p_i = f_i/T$. Entropy is calculated as:

$$H = -\sum_i p_i \log_2 p_i, \quad H_{\text{norm}} = \frac{H}{\log_2 T}. \tag{1}$$

The normalization ensures comparability across varying output lengths. A lower H_{norm} indicates token concentration, which often signals collapsed or repetitive responses. Conversely, higher entropy suggests fluent and lexically diverse outputs (Yuan et al. 2024).

As shown in Figure 4, most models exhibit a decline in entropy on unreasonable questions. This drop is especially pronounced in reasoning-enhanced models, where entropy decreases by 0.1–0.15 on average. This semantic collapse often stems from repeated, ineffective reasoning loops, which reduce informativeness and mask the model's failure to recognize flawed assumptions.

Token-Level Repetition. We further quantify redundancy by computing the number of repeated n-grams (with $n = 10$) within each output. High repetition reflects local generation loops, where the model reiterates similar phrasing instead of progressing logically. Figure 4 shows that almost all models exhibit increased repetition under unreasonable inputs, with the most pronounced effects observed in reasoning-enhanced models. This highlights a clear tendency toward repetitive and collapsed outputs in the face of flawed questions. Notably, Qwen2.5-Math-7B-Instruct, a math-specialized model with strong performance on well-posed problems, also demonstrates a significant rise in repetition.

3.6 Summary of Findings

Our behavioral analysis uncovers several key patterns. First, general-purpose models often proceed with unwarranted confidence, failing to recognize flawed assumptions, a phenomenon we term *unconscious of unreasonableness*. Second, reasoning models are prone to *overthinking*, repeatedly revising their reasoning in response to irrational inputs, which lead to verbosity and incoherence. Third, both reasoning and math-specialized models exhibit *semantic collapse*, characterized by increased token repetition and reduced entropy under unreasonable conditions. Finally, despite strong accuracy on well-posed tasks, math-specialized models often fail to detect subtle logical flaws, highlighting a disconnect between mathematical proficiency and robustness to flawed inputs.

4 Can LLMs Detect Unreasonableness?

Through the experiments above, we observe that most models perform poorly at recognizing unreasonableness in mathematical problems. To explore possible ways to improve this behavior, we begin with a simple probing experiment to test whether models possess the basic ability to judge flawed inputs. Based on this, we further investigate two strategies, prompt-based intervention and supervised fine-tuning, to enhance the model's capacity to detect unreasonableness.

[2]We use DeepSeek-V3 as the evaluator for cost efficiency.

[3]Due to the limited number of AMC and AIME samples, we report detailed scores only for GSM and MATH datasets.

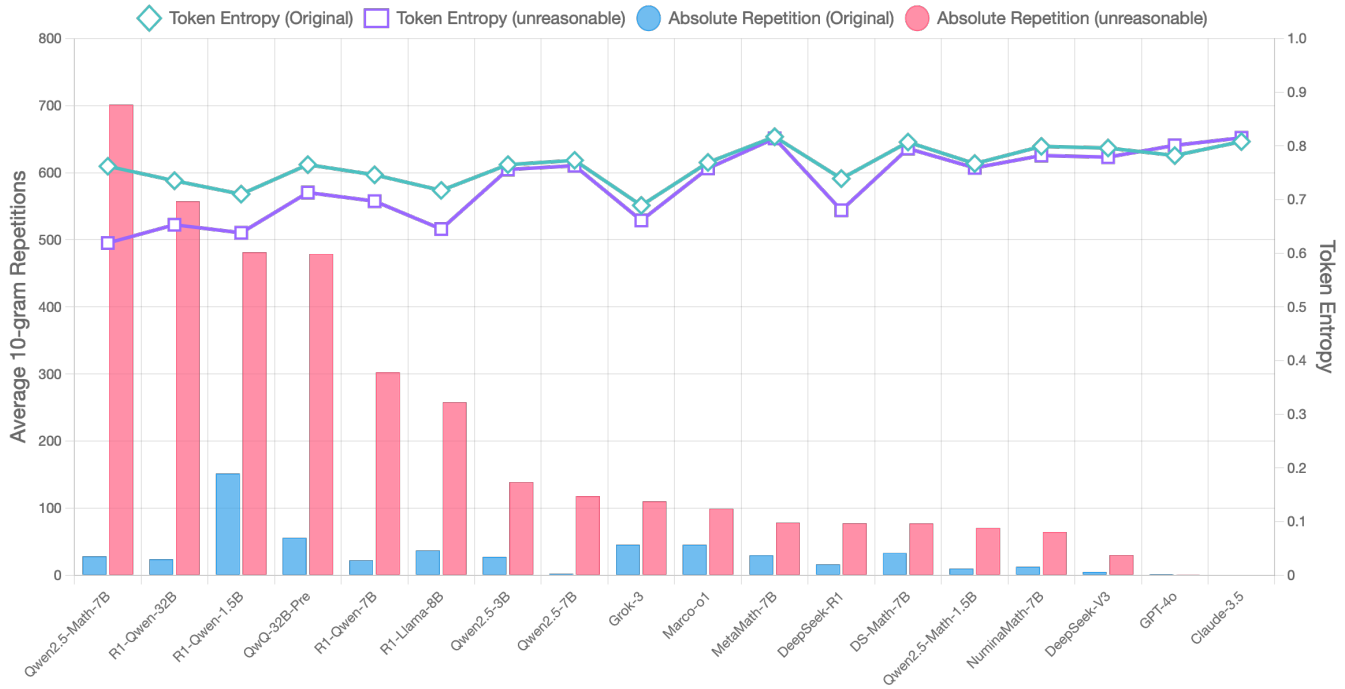

Figure 4: Comparison of 10-gram repetition and token entropy on the UMP dataset. The bar chart reflects the degree of repetition in model outputs (measured using 10-gram repetition), with blue representing original questions and pink for unreasonable ones. The line plot shows changes in normalized token entropy under original and unreasonable problems. A decrease in entropy and an increase in repetition under unreasonable inputs suggest output collapse and reduced lexical diversity.

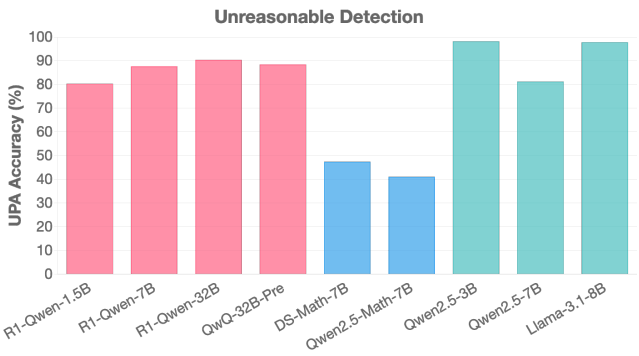

Figure 5: Unreasonable Phrase Accuracy (UPA) of different models. Higher scores indicate a stronger ability to identify unreasonableness in isolated expressions. Model categories are color-coded.

4.1 Probing Study

Before exploring how to improve model behavior on unreasonable problems, we first ask a fundamental question: do models even understand what counts as unreasonable? If a model lacks this awareness, it is unlikely to respond appropriately when encountering flawed questions. To test this, we design a simple probing task, where the model is presented with short, isolated expressions that are syntactically valid but semantically unreasonable, for example, "the voting result is -20 votes" or "there are 2.5 people." We refer to this metric as **Unreasonable Phrase Accuracy (UPA)**. For each expression, the model is asked to judge whether it makes sense. We construct this test set by extracting unreasonable phrases from the UMP benchmark and evaluate a representative subset of models used in our main experiments, including general-purpose, reasoning-enhanced, and math-specialized models. As shown in Figure 5, general-purpose and reasoning-enhanced models typically perform well on this task, suggesting that they retain basic commonsense priors. In contrast, several math-specialized models perform noticeably worse, indicating that domain-specific fine-tuning may come at the cost of general semantic awareness.

4.2 Prompting LLMs with a Critical Thinking Signal

To explore lightweight methods for improving flaw detection, we experiment with inserting a simple instruction, such as "Please solve these problems with criticism." into the prompt. This encourages the model to approach the task with a more skeptical and reflective reasoning style. We evaluate this strategy on Qwen2.5-7B-Instruct and R1-Distill-Qwen-7B, and observe a consistent improvement in identifying unreasonable questions. As shown in Table 3, both models achieve higher Absolute Scores on unreasonable inputs, while their accuracy on original questions remains largely unaffected. Notably, with the critical-thinking

prompt, the general-purpose Qwen2.5-7B-Instruct outperforms the much larger DeepSeek-V3 in detecting flawed assumptions, highlighting the potential of prompt-based interventions to activate latent reasoning ability without additional training.

Model	Orig Acc (%)	Abs Score (%)
Qwen2.5-7B	88.35	43.59
Qwen2.5-7B-Cri	86.96	66.88
R1-7B	91.96	63.06
R1-7B-Cri	89.88	69.50
DS-V3	95.15	59.66
Model (test set)		
Qwen2.5-7B	80.26	41.11
Qwen2.5-7B-SFT	79.93	49.01
Qwen2.5-3B	70.90	34.94
Qwen2.5-3B-SFT	67.55	46.30

Table 3: Original accuracy on reasonable problems and absolute score on unreasonable problems(GSM & MATH). SFT results are evaluated on a subset disjoint from the fine-tuning data.

4.3 Fine-Tuning Based Method

We further investigate whether the model's ability to detect unreasonableness can be enhanced through supervised fine-tuning. To construct the training data, we select a subset of questions from our UMP dataset and craft corresponding responses that correctly identify the embedded unreasonableness. These responses explicitly identify the flaws through concise, step-by-step reasoning, encouraging appropriate reflection without triggering overthinking. We then evaluate the fine-tuned model on an independent test set that was never seen during training. As shown in Table 3, the fine-tuned model exhibits clear improvement in detecting unreasonable inputs, with only a slight drop in original accuracy. Additional experimental details can be found in Appendix.

Discussion. We adopt several commonly used enhancement strategies, including prompting and fine-tuning, both of which improve a model's ability to detect unreasonable problems, though they may introduce trade-offs such as a slight drop in accuracy on well-posed inputs. However, we argue that in real-world settings, where the correctness of a problem cannot be assumed, such trade-offs are acceptable. Enhancing a model's reliability in the face of uncertainty helps prevent confidently incorrect responses and reduces the risk of misleading users. These findings suggest that incorporating even lightweight interventions can make LLMs more cautious and trustworthy when confronted with potentially flawed inputs.

5 Related Work

5.1 LLMs as Math Problem Solvers

Large language models have demonstrated impressive capabilities in solving math word problems and symbolic reasoning tasks. The most widely used benchmarks in this space include GSM8K (Cobbe et al. 2021b) and MATH (Hendrycks et al. 2021). To improve performance on these tasks, various data-centric approaches have been proposed. WizardMath (Luo et al. 2023a) and MetaMath (Yu et al. 2023) leverage self-instruct and verifier-guided generation to create high-quality training examples. These augmentations expose models to diverse problem formats and encourage generalization. In parallel, a wave of domain-specialized models has emerged, such as Qwen-Math (Yang et al. 2024a), DeepSeek-Math (Shao et al. 2024), and WizardMath-v2 (Luo et al. 2023b), which are fine-tuned on large-scale mathematical corpora and incorporate structured reasoning or reflection mechanisms. These models outperform general-purpose LLMs on math benchmarks.

5.2 Improving Models' Inference Ability

Recent efforts have focused on enhancing the reasoning capabilities of LLMs by structuring and extending their inference chains. The *chain-of-thought* prompting method (Wei et al. 2022) was among the earliest approaches to guide models toward step-by-step reasoning. Subsequent variants such as *Complex CoT* (Fu et al. 2023) and *Plan-and-Solve* (Wang et al. 2023) further emphasize intermediate planning and decompositional reasoning. More recently, large-scale models such as O1/R1 (DeepSeek-AI 2025) and QwQ-32B (Team 2025) have shown that extending the reasoning trajectory through structured plans, iterative self-correction, or reflective feedback can significantly improve performance on complex tasks. These models often adopt long-form generation, which mimics human deliberation. In addition, verifier-based strategies such as *Outcome-Supervised Learning* (Yu, Gao, and Wang 2023) and post-hoc reflection mechanisms have been introduced to scrutinize intermediate steps and promote robust decision-making. However, comparatively little work has explored how LLMs should respond when the mathematical problem itself is unreasonable, a gap our study seeks to address.

6 Conclusion

In this work, we investigate how LLMs behave when confronted with unreasonable mathematical problems. To facilitate this, we construct a benchmark comprising more than 1,000 questions containing hidden unreasonableness. Experimental results reveal that most models struggle to identify such unreasonableness. While reasoning models are more likely to uncover hidden inconsistencies during multi-step reasoning, they often fall into repetitive reflection and overthinking behavior that ultimately hinders clarity and usefulness. To further explore whether models possess latent capabilities for detecting flaws, we experiment with critical-thinking prompts and supervised fine-tuning. Our findings suggest that many models do have the potential to detect unreasonable content, but this ability requires explicit activation. We hope our work contributes to the evaluation of both the trustworthiness and behavioral robustness of LLMs in the presence of unreasonable inputs.

Acknowledgments

This paper is supported by NSFC project 62476009 and National Key Research and Development Program of China under Grant 2022ZD0116408.

Limitation

Our study examines how different models behave when facing unreasonable math problems and uses an LLM-as-a-Judge framework for evaluation. Due to cost constraints, we rely on DeepSeek-V3 as the judging model. While this approach is practical, it introduces some instability because the evaluation quality depends on the capability of the judge model itself and the prompt. We hope that our work provides a useful starting point for developing more stable and reliable evaluation methods in the future.

References

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021a. Training Verifiers to Solve Math Word Problems. *CoRR*, abs/2110.14168.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021b. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

Cuadron, A.; Li, D.; Ma, W.; Wang, X.; Wang, Y.; Zhuang, S.; Liu, S.; Schroeder, L. G.; Xia, T.; Mao, H.; Thumiger, N.; Desai, A.; Stoica, I.; Klimovic, A.; Neubig, G.; and Gonzalez, J. E. 2025. The Danger of Overthinking: Examining the Reasoning-Action Dilemma in Agentic Tasks. arXiv:2502.08235.

DeepSeek-AI. 2024. DeepSeek-V3 Technical Report. arXiv:2412.19437.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.

Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Ma, J.; Li, R.; Xia, H.; Xu, J.; Wu, Z.; Chang, B.; Sun, X.; and Sui, Z. 2024. A Survey on In-context Learning. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, 1107–1128. Association for Computational Linguistics.

Fu, Y.; Peng, H.; Sabharwal, A.; Clark, P.; and Khot, T. 2023. Complexity-Based Prompting for Multi-Step Reasoning. arXiv:2210.00720.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. *NeurIPS*.

Kasneci, E.; Seßler, K.; Küchemann, S.; Bannert, M.; Dementieva, D.; Fischer, F.; Gasser, U.; Groh, G.; Günnemann, S.; Hüllermeier, E.; et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and individual differences*, 103: 102274.

Lin, Y.; and Chen, Y. 2023. LLM-Eval: Unified Multi-Dimensional Automatic Evaluation for Open-Domain Conversations with Large Language Models. *CoRR*, abs/2305.13711.

Luo, H.; Sun, Q.; Xu, C.; Zhao, P.; Lou, J.; Tao, C.; Geng, X.; Lin, Q.; Chen, S.; and Zhang, D. 2023a. WizardMath: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evol-Instruct. arXiv:2308.09583.

Luo, H.; Sun, Q.; Xu, C.; Zhao, P.; Lou, J.; Tao, C.; Geng, X.; Lin, Q.; Chen, S.; and Zhang, D. 2023b. WizardMath: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evol-Instruct. *CoRR*, abs/2308.09583.

MAA. 2024. American Invitational Mathematics Examination (AIME). Accessed: 2024-04-27.

MetaAI. 2024. Llama 3 Model Card.

OpenAI. 2024. GPT-4o: OpenAI's New Multimodal Model. https://openai.com/index/hello-gpt-4o/. Accessed: 2024-05-20.

Qwen Team. 2024. Qwen2.5: A Party of Foundation Models.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y. K.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv:2402.03300.

Sui, Y.; Chuang, Y.-N.; Wang, G.; Zhang, J.; Zhang, T.; Yuan, J.; Liu, H.; Wen, A.; Zhong, S.; Chen, H.; and Hu, X. 2025. Stop Overthinking: A Survey on Efficient Reasoning for Large Language Models. arXiv:2503.16419.

Team, Q. 2025. QwQ-32B: Embracing the Power of Reinforcement Learning.

Wang, L.; Xu, W.; Lan, Y.; Hu, Z.; Lan, Y.; Lee, R. K.-W.; and Lim, E.-P. 2023. Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. *arXiv preprint arXiv:2305.04091*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; ichter, b.; Xia, F.; Chi, E.; Le, Q. V.; and Zhou, D. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 24824–24837. Curran Associates, Inc.

xAI. 2024. Grok-3 by xAI. https://x.ai/. Accessed: 2024-05-20.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; et al. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yang, Z.; Zhang, Y.; Liu, T.; Yang, J.; Lin, J.; Zhou, C.; and Sui, Z. 2024b. Can Large Language Models Always Solve Easy Problems if They Can Solve Harder Ones? In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings*

of the 2024 Conference on Empirical Methods in Natural Language Processing, 1531–1555. Miami, Florida, USA: Association for Computational Linguistics.

Yu, F.; Gao, A.; and Wang, B. 2023. Outcome-supervised Verifiers for Planning in Mathematical Reasoning. arXiv:2311.09724.

Yu, L.; Jiang, W.; Shi, H.; Yu, J.; Liu, Z.; Zhang, Y.; Kwok, J. T.; Li, Z.; Weller, A.; and Liu, W. 2023. MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models. arXiv:2309.12284.

Yuan, X.; Pang, T.; Du, C.; Chen, K.; Zhang, W.; and Lin, M. 2024. A Closer Look at Machine Unlearning for Large Language Models. *CoRR*, abs/2410.08109.

Zhang, Z.; Zhang-Li, D.; Yu, J.; Gong, L.; Zhou, J.; Liu, Z.; Hou, L.; and Li, J. 2024. Simulating Classroom Education with LLM-Empowered Agents. *CoRR*, abs/2406.19226.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36: 46595–46623.