# Large Language Models Struggle with Unreasonability in Math Problems

#### **Anonymous ACL submission**

#### Abstract

Large Language Models (LLMs) have shown 001 remarkable success on 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, models frequently pro-007 ceed as if the problem is well-posed, producing incorrect answers or falling into overthinking and verbose self-correction. To systematically investigate this overlooked vulnerability, we propose the Unreasonable Math Problems (UMP) benchmark, designed to evaluate LLMs<sup>3</sup> ability to detect and respond to unreasonable math problem statements. Based on extensive 015 experiments covering 19 LLMs, we find that even state-of-the-art general models like GPT-017 40 achieve only a score of 0.6 on UMP. While reasoning models such as DeepSeek-R1 demonstrate a higher sensitivity to unreasonable in-019 puts, this often comes at the cost of generating overly long and meaningless responses that fail to converge. We further explore prompting and fine-tuning methods, which offer partial improvements but also introduce trade-offs, shedding light on both the potential and limitations of LLMs in this challenging setting.

#### 1 Introduction

027

037

041

Large language models (LLMs) have recently shown impressive performance on advanced mathematical reasoning tasks, especially on benchmarks like 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



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, or logical consistency.

et al., 2023), early education (Zhang et al., 2024b), 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)

056

090

100

101

102

104

to create unreasonable variants that contain logical inconsistencies, missing assumptions, or illdefined objectives. These edits are guided by rulebased transformation templates and executed using 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-40 (OpenAI, 2024)), reasoning models (e.g., DeepSeek-R1 (DeepSeek-AI, 2025)), and mathspecialized models (e.g., Qwen-Math (Yang et al., 2024a)). In addition, we analyze LLMs' failure patterns according to tokens repetition, reflection frequency, and token entropy. We find: 1) generalpurpose 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 could 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 partially mitigate these issues, but often introduce trade-offs such as decreased accuracy on well-posed problems, posing new challenges for future research.

#### 2 **Unreasonable Math Problems(UMP) 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 1000+ 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 commonly found in LLM failure cases: (1) undefined variables, (2) illogical scenarios, (3) incorrect assumptions, (4) misinterpretation of units, and (5) inconsistent conditions. Each instance in our benchmark is represented as (q, a, q', t, e), consisting of the original question q and its answer a, the unreasonable variant q' with its assigned type t, and an explanation e describing why q' is unreasonable. Detailed definitions and examples for each category are provided in Appendix A.

# 2.2 LLM-Guided Construction of **Unreasonable Variants**

Our data construction process is inspired by Meta-136 Math (Yu et al., 2023b), which leverages LLMs to 137 produce problem variants under controlled transformations. As illustrated in Figure 2, we begin 139 with test set questions drawn from four widely-140 used math benchmarks: GSM8K (Cobbe et al., 141 2021a), a collection of grade-school level prob-142 lems; MATH (Hendrycks et al., 2021), which 143 covers formal secondary school mathematics; and 144 AIME24 (MAA, 2024) and AMC23, both of which 145 contain high level competition problems with sym-146 bolic or abstract formats. We manually construct 147 a set of transformation rules corresponding to five 148 types of mathematical unreasonableness and use 149 them to guide an LLM in generating unreasonable 150 variants for each original question, along with nat-151 ural language explanations of why the modified 152 version is irrational. To ensure the unreasonable 153

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135



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 prompting; (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, correctness, and alignment with error categories.

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 et al., 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

154

155

156

157

158

159

160

161

162

163

165

166

167

168

169

170

172

174

175

176

We manually verify each example to ensure that 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 H). In such cases, we revise the examples by adjusting entities, numerical values, or phrasing to ensure the unreasonableness remains logically subtle yet plausible.

## 2.4 Benchmark Composition

Table 1 summarizes the distribution of unreason-177 ableness types across different source datasets in 178 the UMP benchmark. A large proportion of exam-179 ples fall into the categories of *Incorrect Assump-*180 tions (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 186 more likely to be retained during human recheck. Compared to GSM8K and MATH, the AMC23 and 187 AIME24 datasets contain far fewer test questions (e.g., AIME24 has only 30), which inherently limits the number of examples we can derive from 190

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.

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

## 3 Evaluating LLMs on Unreasonable Math Problems

#### 3.1 Evaluation Setup

3

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 DeepSeek-V3 (DeepSeek-AI, 2024), GPT-40 (Ope, 2023), Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct (Qwen Team, 2024), and LLaMA3.1-8B-Instruct (AI, 2024). **Math-specialized models** are fine-tuned on mathematical corpora and optimized for numerical reasoning; we include 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-** 195 196 197

191

192

193

194

200

202

203 204 205

206

207

209

210

211

212

213

214

enhanced models are designed to support complex multi-step reasoning and include DeepSeek-R1
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.<sup>1</sup>

#### **3.2 Evaluation Metrics**

224

238

241

242

245

246

247

253

257

258

262

263

Evaluating model performance on unreasonable questions poses a fundamentally different challenge compared to standard accuracy-based evaluation. In conventional settings, the goal is to assess whether the model arrives at the correct answer. However, for unreasonable questions, the objective is to evaluate whether the model can recognize the unreasonability embedded in the problem and respond appropriately by rejecting, questioning, or flagging the input as flawed. To address this, 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 unreasonability is revealed. By restricting the evaluation to questions the model can already solve in their original form, relative score isolates the model's ability to detect unreasonableness from its general problem-solving skill.

#### 3.2.1 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, or introduces new contradictions). Each model response receives a label  $E(v) \in \{A, B, C\}$ , and we define a soft scoring function  $\delta(E)$  to assign partial credit:

$$\delta(E) = \begin{cases} 1 & \text{if } E = \mathbf{A} \\ 0.5 & \text{if } E = \mathbf{B} \\ 0 & \text{if } E = \mathbf{C}. \end{cases}$$
(1) 26

264

265

266

268

269

271

272

273

274

275

276

277

278

279

280

281

283

286

289

290

291

293

294

295

296

297

298

299

300

301

302

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

Absolute Score = 
$$\frac{1}{|V|} \sum_{v \in V} \delta(E(v)).$$
 27

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).

## 3.2.2 Relative Score

If a model lacks the skill to solve the original problem, it may never reach the point where the flaw is revealed. To control this factor, we introduce the relative score, which measures a model's ability to detect unreasonableness on the subset of questions it has already solved correctly in their original form. Formally, following Yang et al. (2024b), we define the relative score as:

 $P(\text{Detect Unreasonable} \mid \text{Solve Original})$  (2)

Under a multi-sample evaluation setting, we generate k responses for each original question q and compute its average accuracy:

$$\bar{M}(q) = \frac{1}{k} \sum_{i=1}^{k} m_{q,i}.$$
 (3)

where  $m_{q,i} \in \{0, 1\}$  indicates whether the *i*-th sampled response to question q is correct (1 if correct, 0 otherwise), and  $\overline{M}(q)$  denotes the proportion of correct responses over k samples.

We then define a set of confidently solved questions  $Q_{\tau}^+$  by applying a threshold  $\tau \in [0, 1]$  to the average accuracy:

$$Q_{\tau}^{+} = \left\{ q \in Q \mid \overline{M}(q) \ge \tau \right\}.$$

$$\tag{4}$$

This ensures that only questions the model has solved reliably in their original (reasonable) form are considered.

Let V(q) be the set of unreasonable variants derived from question q. Each variant  $v \in V(q)$  is

<sup>&</sup>lt;sup>1</sup>Details of model versions and inference configurations are provided in Appendix D.

Model		GSM			MATH		A	MC+AIM	1E		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	0.707	0.370	0.402	0.333	0.202	0.273	0.795	0.336	0.361
Qwen2.5-7B-Instruct	0.929	0.440	0.440	0.820	0.324	0.362	0.303	0.250	0.111	0.858	0.386	0.403
Llama-3.1-8B-Instruct	0.872	0.227	0.238	0.549	0.205	0.238	0.182	0.190	0.391	0.714	0.217	0.245
DeepSeek-v3	0.965	0.560	0.562	0.933	0.658	0.671	0.576	0.536	0.442	0.935	0.598	0.600
GPT-40	0.955	0.640	0.657	0.803	0.565	0.564	0.242	0.476	0.417	0.863	0.603	0.610
Claude-3.5-Sonnet	0.936	0.586	0.605	0.793	0.480	0.503	0.455	0.309	0.359	0.858	0.532	0.554
Reasoning Models												
R1-Distill-Qwen-1.5B	0.858	0.377	0.397	0.836	0.619	0.634	0.394	0.643	0.818	0.829	0.484	0.510
Marco-o1	0.893	0.539	0.546	0.736	0.554	0.567	0.273	0.428	0.591	0.804	0.540	0.557
R1-Distill-Qwen-7B	0.917	0.570	0.600	0.923	0.732	0.731	0.667	0.738	0.747	0.909	0.642	0.659
R1-Distill-Llama-8B	0.891	0.581	0.593	0.900	0.704	0.706	0.485	0.702	0.698	0.877	0.635	0.642
R1-Distill-Qwen-32B	0.967	0.725	0.734	0.920	0.757	0.768	0.576	0.762	0.796	0.931	0.739	0.750
QwQ-32B-Preview	0.957	0.550	0.560	0.900	0.692	0.697	0.606	0.643	0.679	0.919	0.610	0.620
DeepSeek-R1	0.972	0.830	0.844	0.950	0.806	0.810	0.879	0.667	0.621	0.959	0.813	0.821
Grok3-Reasoning	0.967	0.875	0.884	0.913	0.918	0.927	0.879	0.892	0.918	0.941	0.893	0.903
Math Models												
Qwen2.5-Math-1.5B-Instruct	0.844	0.326	0.356	0.783	0.385	0.416	0.394	0.202	0.262	0.800	0.344	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	0.109	0.139
DeepSeek-Math-7B-Instruct	0.844	0.141	0.158	0.520	0.200	0.225	0.091	0.107	0.136	0.682	0.163	0.188
NuminaMath-7B-CoT	0.725	0.276	0.320	0.572	0.285	0.366	0.152	0.178	0.136	0.639	0.276	0.337
Qwen2.5-Math-7B-Instruct	0.962	0.222	0.231	0.853	0.348	0.355	0.394	0.214	0.458	0.894	0.271	0.290

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. The definitions and evaluation methodology for these two behavioral metrics are detailed in Section 3.2. Among them, the **bold** ones are the models of each category with the highest **Rel** on different datasets.

scored with a rating  $E(v) \in \{A, B, C\}$ , which is mapped to a soft score via the function  $\delta(E(v))$ defined earlier.

The final relative score  $S_{rel}$  is computed by averaging the soft scores over all unreasonable variants associated with confidently solved questions:

$$S_{\rm rel} = \frac{1}{|Q_{\tau}^+|} \sum_{q \in Q_{\tau}^+} \left( \frac{1}{|V(q)|} \sum_{v \in V(q)} \delta(E(v)) \right).$$
(5)

#### **3.3 Experimental Results**

Table 2 reports model performance on UMP benchmark and three evaluation metrics: accuracy on original problems (Acc), and absolute and relative scores (Abs, Rel) on unreasonable variants. As defined in Section 3.2, these behavioral scores capture a model's ability to detect and respond appropriately to flawed problem setups.

We categorize evaluated models into three groups: general-purpose, reasoning-enhanced, and math-specialized. General models such as GPT-40, Claude-3.5-Sonnet, and DeepSeek-v3 achieve strong accuracy on well-posed problems (e.g., GPT-40: 0.863 Acc) and moderate robustness to unreasonableness (0.603 Abs), while smaller models like Qwen2.5-3B and LLaMA-3.1-8B struggle across behavioral metrics. Reasoning models (e.g., DeepSeek-R1, grok-3-reasoning) perform best on UMP, reaching 0.813 Abs and 0.903 Rel, benefiting from structured verification steps, but often generating overly verbose responses (see Section 4.1). In contrast, math-specialized models (e.g., Qwen2.5-Math-7B, MetaMath-Mistral-7B) excel in original task accuracy (up to 0.894 Acc) but perform poorly on UMP (mostly below 0.30), indicating a gap between precise reasoning and unreasonability detection. 326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

350

351

These results highlight a trade-off between accuracy on well-posed problems and sensitivity to unreasonableness, with reasoning models offering better detection at the cost of increased verbosity.

#### 4 Analysis

# 4.1 Reasoning Models Tend to Overthink on Unreasonable Problems

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 quantify overthinking by counting occurrences of reflective phrases such as "rethink" or "I misunderstood," which indicate mid-generation self-correction (see Appendix F for keyword de-

309

310

311

312

313

314

315

317

319

321

324

426

427

428

429

430

431

432

433

434

385

386

387

390



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

tails). Although we evaluate reflection frequency across all model types, only reasoning-enhanced models exhibit consistent and substantial increases on unreasonable problems.

354

356

367

371

372

373 374

377

378

379

382

384

As shown in Figure 3, reasoning models show an average increase in reflective behavior by a factor of three to four when moving from original to unreasonable questions. Instead of terminating early, these models tend to reevaluate their reasoning repeatedly, leading to longer and less coherent responses. This pattern is largely absent in generalpurpose and math-specialized models, suggesting that stronger reasoning capacity also increases susceptibility to overthinking when faced with unreasonable inputs.<sup>2</sup>

# 4.2 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:

$$H = -\sum_{i} p_i \log_2 p_i, \quad H_{\text{norm}} = \frac{H}{\log_2 T}.$$
 (6)

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 reasoningenhanced models such as R1-Distill-Qwen-32B, where entropy decreases by 0.1–0.15 on average. These semantic collapse often stems from repeated, ineffective reasoning loops, which reduce informativeness and mask the model's failure to recognize flawed assumptions. Examples of such behavior are presented in Appendix L.1.

**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 10-gram repetition under unreasonable inputs, with the most substantial jumps observed in reasoning-enhanced models. In some cases, repetition increases by a factor of 5 to 10 compared to original problems, reinforcing the pattern of collapsed outputs. Interestingly, Qwen2.5-Math-7B-Instruct, a math-specialized model with strong performance on well-posed questions, also shows a marked rise in repetition, suggesting that semantic instability is not limited to reasoning-heavy models. Appendix L.2 provides detailed case studies of this failure pattern.

#### 4.3 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, reasoningenhanced models are prone to *overthinking*, repeatedly revising their reasoning in response to irrational inputs, which leads 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,

<sup>&</sup>lt;sup>2</sup>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.

highlighting a disconnect between mathematical proficiency and robustness to flawed inputs.



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

## 5 Can LLMs Detect unreasonability?

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 unreasonability.

#### 5.1 **Probing Study**

Before exploring how to improve model behav-449 ior on unreasonable problems, we first ask a fun-450 damental question: do models even understand 451 what counts as unreasonable? If a model lacks this 452 awareness, it is unlikely to respond appropriately 453 when encountering flawed questions. To test this, 454 we design a simple probing task, where the model 455 is presented with short, isolated expressions that 456 are syntactically valid but semantically unreason-457 able-for example, "the voting result is -20 votes" 458 or "there are 2.5 people." We refer to this metric as 459 Unreasonable Phrase Accuracy (UPA). For each 460 expression, the model is asked to judge whether it 461 makes sense. We construct this test set by extract-462 ing unreasonable phrases from the UMP bench-463 mark and evaluate a representative subset of mod-464 els used in our main experiments (Section 3), in-465 cluding general-purpose, reasoning-enhanced, and 466 math-specialized models. As shown in Figure 5, 467 general-purpose and reasoning-enhanced models 468 typically perform well on this task, suggesting that 469 they retain basic commonsense priors. In contrast, 470 several math-specialized models perform notice-471 ably worse, indicating that domain-specific fine-472 tuning may come at the cost of general semantic 473

447

448

442

443

444

445

446

474 awareness.

475

476

477

481

483

484

487

490

491

493

495

#### 5.2 Prompting LLMs with a Critical Thinking Signal

To explore lightweight methods for improving flaw detection, we experiment with inserting a sim-478 ple instruction—such as "Please solve these prob-479 lems with criticism."-into the prompt. This en-480 courages the model to approach the task with a more skeptical and reflective reasoning style. We 482 evaluate this strategy on Qwen2.5-7B-Instruct and R1-Qwen-7B, and observe a consistent improvement in identifying unreasonable questions. As 485 shown in Figure 6, both models achieve signifi-486 cantly higher Absolute Scores on unreasonable inputs, while their accuracy on original questions re-488 mains largely unaffected. Notably, with the critical-489 thinking prompt, the general-purpose Qwen2.5-7B-Instruct nearly matches the performance of the much larger DeepSeek-Chat in detecting flawed 492 assumptions-highlighting the potential of promptbased interventions to activate latent reasoning abil-494 ity without additional training.



Figure 6: Performance comparison of models with and without critical-thinking prompts. Original Accuracy refers to performance on well-posed questions, and Absolute Score on their unreasonable variants. Prompting improves flaw detection with minor accuracy loss; fine-tuning yields larger gains but affects original performance more.

496 497

498

499

503

504

507

#### **Fine-Tuning Based Method** 5.3

We further investigate whether the model's ability to detect unreasonability can be enhanced through supervised fine-tuning. To construct the training data, we pair unreasonable questions from GSM8K with carefully edited responses based on DeepSeek-R1 outputs. These responses explicitly identify the flaws through concise, step-by-step reasoning, encouraging appropriate reflection without triggering overthinking. We also include original questions paired with similarly edited answers to maintain general reasoning competence and prevent the model from overgeneralizing flaw detection to well-posed inputs. This balanced training setup provides the model with positive demonstrations of how to respond to both flawed and valid inputs. We fine-tune on a subset of GSM8K and evaluate the resulting model on MATH to assess generalization beyond the training distribution. As shown in Figure 6, the fine-tuned model exhibits clear improvement in detecting unreasonable inputs. However, this gain comes at a cost: accuracy on original problems drops noticeably. This suggests a trade-off between flaw sensitivity and general problem-solving ability. Further training details are shown in Appendix I.

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

**Discussion.** We adopt several commonly used enhancement strategies, including prompting and fine-tuning, both of which yield promising gains in unreasonable problem detection. However, these improvements consistently come at the cost of reduced accuracy on original, well-posed inputs. This trade-off highlights a core challenge: simple prompting or fine-tuning alone is insufficient to robustly address unreasonable inputs without introducing new side effects. The limitations of these methods underscore the inherent difficulty of the UMP benchmark and point to the need for more principled, generalizable flaw detection mechanisms that can integrate critical reasoning while preserving performance on standard tasks.

#### 6 Conclusion

In this work, we investigate how LLMs behave when confronted with unreasonable mathematical problems. To facilitate this, we construct a benchmark comprising over 1,000 questions containing hidden unreasonability. Experimental results reveal that most models struggle to identify such unreasonableness. While reasoning-enhanced 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 criticalthinking 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 that our benchmark serves as a valuable tool for evaluating both the trustworthiness and behavioral robustness of LLMs in the presence of unreasonable inputs.

## Limitations

558

This study has several limitations. First, we do not include proprietary reasoning models such as those from OpenAI in our analysis of reasoning-561 enhanced systems. Since these APIs do not expose 562 intermediate reasoning chains, comparing their be-563 564 havior to open-source models that explicitly generate multi-step reasoning would introduce unfairness in both behavioral and metric-based evaluations. Second, due to hardware constraints, our intervention experiments such as prompt engineer-568 569 ing and supervised fine-tuning are conducted only on 7B-scale models. While these experiments provide encouraging insights, scaling these methods to larger models and exploring more effective so-572 lutions for mitigating failure behaviors remain im-573 portant directions for future work. 574

## Ethical Considerations

Our paper explores how LLMs perform in the face of unreasonable mathematical problems. Sometimes the model may not realize the unreasonability in the math problem, which may result in incorrect answers that mislead users.

#### References

- 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.
- Meta AI. 2024. Llama 3 model card.
  - 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. 2021a. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.
  - 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. 2021b. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.
  - Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, Nicholas Thumiger, Aditya Desai, Ion Stoica, Ana Klimovic, Graham Neubig, and Joseph E. Gonzalez. 2025. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *Preprint*, arXiv:2502.08235.
  - Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. 2024. Or-bench: An over-refusal

benchmark for large language models. Preprint, 607 arXiv:2405.20947. 608 DeepSeek-AI. 2024. Deepseek-v3 technical report. 609 Preprint, arXiv:2412.19437. 610 DeepSeek-AI. 2025. Deepseek-r1: Incentivizing rea-611 soning capability in llms via reinforcement learning. 612 Preprint, arXiv:2501.12948. 613 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and 614 Kristina Toutanova. 2019. Bert: Pre-training of deep 615 bidirectional transformers for language understand-616 ing. Preprint, arXiv:1810.04805. 617 Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan 618 Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, 619 Baobao Chang, Xu Sun, and Zhifang Sui. 2024. A 620 survey on in-context learning. In Proceedings of 621 the 2024 Conference on Empirical Methods in Nat-622 ural Language Processing, EMNLP 2024, Miami, 623 FL, USA, November 12-16, 2024, pages 1107–1128. 624 Association for Computational Linguistics. 625 Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and 626 Tushar Khot. 2023. Complexity-based prompting for 627 multi-step reasoning. Preprint, arXiv:2210.00720. 628 Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. 629 SimCSE: Simple contrastive learning of sentence 630 embeddings. In Empirical Methods in Natural Lan-631

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. *NeurIPS*. 632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

guage Processing (EMNLP).

- Fabian Hildebrandt, Andreas Maier, Patrick Krauss, and Achim Schilling. 2025. Refusal behavior in large language models: A nonlinear perspective. *Preprint*, arXiv:2501.08145.
- Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274.
- Yen-Ting Lin and Yun-Nung Chen. 2023. Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. *CoRR*, abs/2305.13711.
- Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023a. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *Preprint*, arXiv:2308.09583.
- Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei

- 578 579 580
- 583
- 58
- 58 50

59

5

596

597

603

710

711

712

713

Lin, Shifeng Chen, and Dongmei Zhang. 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.
  - Atsuyuki Miyai, Jingkang Yang, Jingyang Zhang, Yifei Ming, Qing Yu, Go Irie, Yixuan Li, Hai Li, Ziwei Liu, and Kiyoharu Aizawa. 2025. Unsolvable problem detection: Robust understanding evaluation for large multimodal models. *Preprint*, arXiv:2403.20331.
- OpenAI. 2024. Gpt-4o: Openai's new multimodal model. https://openai.com/index/ hello-gpt-4o/. Accessed: 2024-05-20.
  - Stefan Pasch. 2025. Llm content moderation and user satisfaction: Evidence from response refusals in chatbot arena. *Preprint*, arXiv:2501.03266.
  - Qwen Team. 2024. Qwen2.5: A party of foundation models.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024.
  Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.
- Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *Preprint*, arXiv:2503.16419.
- Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.
- Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Planand-solve prompting: Improving zero-shot chain-ofthought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.
- xAI. 2024. Grok-3 by xai. https://x.ai/. Accessed: 2024-05-20.
- Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Madhushani Sehwag, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, Ruoxi Jia, Bo Li, Kai Li, Danqi Chen, Peter Henderson, and Prateek Mittal. 2025. Sorry-bench: Systematically evaluating large language model safety refusal. *Preprint*, arXiv:2406.14598.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*. 714

715

718

719

720

721

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

752

753

754

757

758

759

760

761

762

- Zhe Yang, Yichang Zhang, Tianyu Liu, Jian Yang, Junyang Lin, Chang Zhou, and Zhifang Sui. 2024b. Can large language models always solve easy problems if they can solve harder ones? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1531–1555, Miami, Florida, USA. Association for Computational Linguistics.
- Fei Yu, Anningzhe Gao, and Benyou Wang. 2023a. Outcome-supervised verifiers for planning in mathematical reasoning. *Preprint*, arXiv:2311.09724.
- Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023b. Metamath: Bootstrap your own mathematical questions for large language models. *Preprint*, arXiv:2309.12284.
- Xiaojian Yuan, Tianyu Pang, Chao Du, Kejiang Chen, Weiming Zhang, and Min Lin. 2024. A closer look at machine unlearning for large language models. *CoRR*, abs/2410.08109.
- Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2024a. R-tuning: Instructing large language models to say 'I don't know'. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 7113–7139, Mexico City, Mexico. Association for Computational Linguistics.
- Zheyuan Zhang, Daniel Zhang-Li, Jifan Yu, Linlu Gong, Jinchang Zhou, Zhiyuan Liu, Lei Hou, and Juanzi Li. 2024b. Simulating classroom education with llmempowered agents. *CoRR*, abs/2406.19226.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.
- Runchuan Zhu, Zhipeng Ma, Jiang Wu, Junyuan Gao, Jiaqi Wang, Dahua Lin, and Conghui He. 2024. Utilize the flow before stepping into the same river twice: Certainty represented knowledge flow for refusal-aware instruction tuning. *Preprint*, arXiv:2410.06913.

# A Category of UMP Benchmark

Below is the description of the five categories. Each data point consists of an original question q paired with its corresponding answer a, a designated type of unreasonableness t, a rephrased unreasonable version q', and an explanation e. Consequently, our benchmark comprises quintuples of the form (q, a, q', t, e).

# **Unreasonable Problem Categories**

#### **1. Undefined Variables**

Problems categorized under this heading suffer from a scarcity of necessary information or parameters, rendering them unsolvable with the data provided.

#### 2. Illogical Scenarios

This category encompasses problems that posit scenarios defying logic or possibility, such as querying the number of offspring produced by a species incapable of yielding those offspring (*e.g. asking about the kittens born to a ham-ster*).

#### 3. Incorrect Assumptions

Problems in this group are predicated on mathematically flawed assumptions, such as the division by zero, or the existence of fractional entities in a set that should logically contain only whole units, (*e.g. envisaging the division* of a cake into negative quantities.)

#### 4. Misinterpretation of Units

These problems are marked by unclear or incorrect application of measurement units, leading to nonsensical combinations, (*e.g. assigning* grams as a unit of length).

## 5. Inconsistent Conditions

The given conditions within these problems are self-contradictory, creating paradoxical statements that cannot be simultaneously true. (*e.g.*, *a problem stating a group consists of ten people and simultaneously claiming it comprises twenty people*)

# **B** Related Work

#### **B.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), which targets elementary school-level problems with step-bystep annotations, and MATH (Hendrycks et al., 2021), which evaluates high-school competitionlevel mathematics. To improve performance on these tasks, various data-centric approaches have been proposed. WizardMath (Luo et al., 2023a) and MetaMath (Yu et al., 2023b) leverage selfinstruct 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.

780

781

782

783

784

785

786

787

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

#### **B.2** Improve Models' Inference Ability

Recent efforts have focused on enhancing the reasoning capabilities of large language models by structuring and extending their inference chains. The chain-of-thought (CoT) 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 emphasized intermediate planning and decompositional reasoning. More recently, large-scale models like 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 and multi-phase solving strategies, which mimic human deliberation. In addition, verifier-based strategies such as Outcome-Supervised Learning (Yu et al., 2023a) and posthoc reflection mechanisms have been introduced to scrutinize intermediate steps and promote robust decision-making, especially when the model's output chain may contain errors.

## **B.3** Model Refusal Phenomenon

The phenomenon where LLMs decline to answer certain prompts is central to ensuring AI safety and reliability. Refusals may stem from safety training, uncertainty, knowledge gaps or defects in the prompt itself. Recent studies have explored

771

773

775

776

778

779

both linear and nonlinear mechanisms behind refusal behaviors, including "refusal directions" in activation space and the use of "refusal tokens" to control response behavior at inference. However, over-refusal-where benign prompts are incorrectly rejected- remains a key challenge (Pasch, 2025; Hildebrandt et al., 2025). Approaches like RAIT, CRaFT, and Think-Before-Refusal aim to reduce false refusals by incorporating reasoning and certainty estimation (Zhu et al., 2024; Zhang et al., 2024a). Evaluation efforts have led to dedicated benchmarks such as OR-Bench. SORRY-Bench, and MM-UPD, which assess refusal quality under various conditions, highlighting the need for context-aware and nuanced assessment frameworks (Cui et al., 2024; Xie et al., 2025; Miyai et al., 2025).

830

835

836

841

842

844

845

847

849

853

855

857

866

867

870

# C Probabilistic View from Prior Conditioning

We frame the model's ability to recognize unreasonable problems as a conditional behavior dependent on its understanding of the original question. Let  $Y_v \in \{0, 0.5, 1\}$  denote the correctness score for a model response to a variant  $v \in V(q)$ , assigned by an LLM. To reason about this behavior probabilistically, we apply the law of total expectation:

$$\mathbb{E}[Y] = P(q \in Q_{\tau}^{+}) \cdot \mathbb{E}[Y \mid q \in Q_{\tau}^{+}] + P(q \in Q_{\tau}^{-}) \cdot \mathbb{E}[Y \mid q \in Q_{\tau}^{-}]$$
(7)

This formulation decomposes the expected quality of model behavior into two conditional components:

- The expectation over questions the model can solve  $(Q_{\tau}^+)$ , and
- The expectation over questions it fails to solve  $(Q_{\tau}^{-})$ .

Our focus is on the first term,  $\mathbb{E}[Y \mid q \in Q_{\tau}^+]$ , which reflects the robustness of the model *under the prior belief that it understands the base question.* 

Since each original question may yield multiple unreasonable variants, we define an empirical estimator for this conditional expectation by averaging over the scores of all variants:

871 
$$\hat{P} = \frac{1}{|Q_{\tau}^{+}|} \sum_{q \in Q_{\tau}^{+}} \left( \frac{1}{|V(q)|} \sum_{v \in V(q)} Y_{v} \right)$$
(8)

This is a two-level Monte Carlo approximation of the conditional expectation. It provides a graded measure of how reliably a model detects semantic flaws, assuming sufficient prior understanding of the original problem. 872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

# **D** Model Versions and Inference Settings

We include both API-access and open-source models in our evaluation. The API versions used are gpt-40-2024-11-20, grok-3-reasoning and deepseek-r1-2025-01-20, claude-3.5-sonnet-20241022. To maximize accuracy on original (well-posed) questions, we adopt temperature = 0.0(greedy decoding) for all open-source non-reasoning models. However, for reasoning-enhanced models, we observe that greedy decoding may result in unstable behaviors, such as excessively long outputs, especially in unreasonable problems. Based on these findings, we set temperature = 0.6 for all reasoning models during evaluation.(We provide further analysis in Appendix K.) We set  $\tau = 1$ , which means that the model must get all three responses right. For APIbased models, we maintain their default decoding settings. Due to access limitations and cost constraints, all the responses are generated in a single pass without sampling-based aggregation.

# **E** Does Solving the Original Help?

We investigate whether a model's ability to solve the original version of a problem affects its capacity to detect flaws in the corresponding unreasonable variants. Intuitively, some flaws such as contradictions in intermediate steps may only be visible after progressing through the correct reasoning path. This motivates a direct comparison: how often does a model correctly identify unreasonableness when it has solved the original question, versus when it has not? To measure this, we revisit the set  $Q_{\tau}^+$ from Section 3.2, containing original questions that the model solves correctly with high confidence. We define its complement  $Q_{\tau}^{-}$  as the set of questions the model fails to solve. For each group, we compute the average score over their unreasonable variants, using the same scoring function  $\delta(\cdot)$ . We then define the *robustness gap* as:

$$\Delta_{\tau} = P_{\tau}^+ - P_{\tau}^-. \tag{9}$$

where  $P_{\tau}^+$  and  $P_{\tau}^-$  denote the average scores over  $Q_{\tau}^+$  and  $Q_{\tau}^-$ , respectively. A larger gap indicates that the model's ability to detect unreasonableness 918 depends more strongly on its success in solving the original problem.

919

921

925

926

928

930

931

932

933

934

937

938

941

943

945

946

950

952

953

954

957

961

962

963

964

965

967

As shown in Table 3 and Table 4, most models exhibit a substantial positive gap. This suggests that mathematical competence contributes directly to robustness, reinforcing the motivation behind our relative score metric.

#### F Keywords for Reflection Detection

To quantify reflective behavior in model responses, we apply a keyword-based heuristic to detect phrases associated with self-correction, doubt, or reevaluation. The following list of keywords is used to flag reflection:

sorry, apologize, actually, wait, let me, i made, mistake, error, incorrect, again, redo, retry, fix, revise. correct. adjust, modify, amend, update, let's try, recalculate, recompute, reassess, reevaluate, reanalyze, re-examine, i was wrong, i erred, i miscalculated, i misunderstood, i misread, i overlooked, rethink, reflect. reconsider. review. second in fact. on thought, upon reflection, second guess. double check, verify, confirm, clarify, perhaps, maybe, possibly, alternatively, otherwise, instead, rather, my bad, my fault, my error, my mistake, i stand corrected, i take it back

This keyword set was manually curated to capture a wide range of reflective behaviors and was applied in our analysis of model overthinking.

# G Similarity between Original Problems and Unreasonable Problems

To ensure that our constructed unreasonable problems remain as close as possible to their original counterparts, we conducted a similarity analysis between each pair of (q, q'), where q is the original question and q' is its unreasonable variant. We use cosine similarity over sentence embeddings (computed using a pre-trained BERT encoder) as our similarity metric. Figure 7 shows the distribution of similarity scores across all problem pairs. Most unreasonable problems achieve high similarity scores with their original versions, with the majority falling in the [0.8, 0.95] range, which means a threshold k > 0.8. This confirms that the perturbations introduced are minimal in surface form, allowing us to isolate the effects of semantic difference while keeping the wording largely intact.



Figure 7: Distribution of similarity scores between original problems and their unreasonable variants. Cosine similarity is computed over BERT-based sentence embeddings. Most variants remain highly similar to their original forms.

Model	$P_{\tau}^+$	$P_{\tau}^{-}$	$\Delta$
DS-LLaMA-8B	0.593	0.514	0.079
DS-Math-7B	0.158	0.064	0.094
QwQ-32B	0.560	0.542	0.019
Qwen2.5-7B	0.440	0.433	0.007
Marco-o1	0.546	0.481	0.065
DS-Qwen-7B	0.533	0.437	0.096
DS-Qwen-1.5B	0.397	0.310	0.087
Qwen2.5-3B	0.339	0.230	0.109
DS-Qwen-32B	0.734	0.798	-0.064
DS-Chat	0.562	0.611	-0.049
Numina-7B	0.320	0.204	0.116
DS-Reasoner	0.844	0.806	0.039
MetaMath-7B	0.146	0.086	0.060
Qwen2.5-Math-7B	0.231	0.135	0.096
GPT-40	0.657	0.610	0.047
LLaMA3.1-8B	0.238	0.210	0.028
Qwen2.5-Math-1.5B	0.356	0.227	0.129
Claude-3.5	0.605	0.531	0.074

Table 3: GSM Dataset Metrics.  $P_{\tau}^+$ : Probability the model correctly identifies the unreasonable variant when it can solve the original problems correctly.  $P_{\tau}^-$ : Probability the model identifies the unreasonable variant when it can't solve the original problems.  $\Delta = P_{\tau}^+ - P_{\tau}^-$ .

# **H** Annotation Protocol and Agreement

To ensure the quality of the benchmark, three reviewers with mathematical training participated in a multi-stage validation process. The following criteria were jointly defined and applied during annotation:

974 975 976

Model	$P_{\tau}^+$	$P_{\tau}^{-}$	Δ
DS-LLaMA-8B	0.706	0.722	-0.017
DS-Math-7B	0.225	0.141	0.084
QwQ-32B	0.696	0.547	0.149
Qwen2.5-7B	0.362	0.130	0.232
Marco-o1	0.567	0.412	0.155
DS-Qwen-7B	0.747	0.630	0.117
DS-Qwen-1.5B	0.634	0.485	0.150
Qwen2.5-3B	0.402	0.209	0.193
DS-Qwen-32B	0.768	0.656	0.112
DS-Chat	0.671	0.463	0.208
Numina-7B	0.366	0.180	0.186
DS-Reasoner	0.810	0.750	0.060
MetaMath-7B	0.128	0.066	0.062
Qwen2.5-Math-7B	0.355	0.191	0.164
GPT-40	0.564	0.497	0.067
LLaMA3.1-8B	0.238	0.125	0.112
Qwen2.5-Math-1.5B	0.416	0.219	0.197
Claude-3.5	0.503	0.305	0.198

Table 4: MATH Dataset Metrics.  $P_{\tau}^+$ : Probability the model correctly identifies the unreasonable variant when it can solve the original problems correctly.  $P_{\tau}^-$ : Probability the model identifies the unreasonable variant when it can't solve the original problems.  $\Delta = P_{\tau}^+ - P_{\tau}^-$ .

1. Remove examples that remain mathematically reasonable after rewriting but are mistakenly flagged as unreasonable by the model.

978

979

980

981

985

991

992

994

995

- 2. Remove examples where the unreasonable elements are too obvious or violate basic commonsense (e.g., "2.33 people").
- 3. Remove examples where the rewritten version differs too much from the original, failing to preserve surface similarity.
- 4. Revise examples with unclear or poorly explained unreasonable elements to make the flaw logically coherent and minimally invasive.

An example is retained only if at least two out of three reviewers judged it as valid under the above criteria.

# I Experiment Setup for Supervised Fine-Tuning

996For the fine-tuning experiments, we use997Qwen2.5-7B-Instruct and perform full-998parameter supervised fine-tuning. All experiments999are conducted on two NVIDIA A100 GPUs (80GB

memory). The learning rate is set to 1e-5, and 1000 training is run for 3 epochs. The training dataset 1001 is composed of approximately 100 unreasonable 1002 question-answer pairs and 300 well-posed 1003 question-answer pairs. The unreasonable samples 1004 are constructed from the UMP benchmark, with 1005 responses manually edited to clearly identify and 1006 explain the flaws through concise, structured 1007 reasoning. The well-posed samples are similarly 1008 constructed to ensure balanced learning and to mitigate overgeneralization of unreasonable 1010 detection behavior. 1011

## J Template for Critical Prompting 1012

Here we show our critical template for solving math1013problems.1014

#### Critical template for solving math problems

Please solve these problems with criticism. If the problem is reasonable, please think step by step and put your final answer within boxed. If the problem are unreasonable, highlight these issues clearly in your response and provide a succinct explanation.

# K Effect of Temperature on Model Behavior

We analyze how different decoding temperatures affect model behavior, focusing on two representative models: Qwen2.5-7B-Instruct (general-purpose) and Qwen-Distill-7B (reasoning-enhanced). Specifically, we evaluate model performance under temperature values of 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0.

As shown in Figure 8, the general-purpose model 1025 exhibits relatively stable behavior across tempera-1026 ture settings. In contrast, the reasoning-enhanced model is more sensitive to temperature, particularly 1028 when handling unreasonable problems. Notably, at 1029 very low temperatures, the reasoning model tends 1030 to generate significantly longer outputs, often en-1031 tering repetitive or overthinking loops. Overall, a 1032 temperature of 0.6 yields the most balanced per-1033 formance across models, aligning with the default 1034 setting recommended in the DeepSeek official rec-1035 ommendation. 1036

1015

- 1016 1017
- 1018 1019

1020

1021

#### 🔵 Original Accuracy 🛛 🔲 Absolute Score 🔺 Original Tokens 🛛 🔆 Unreasonable Tokens



Figure 8: Effect of decoding temperature on model behavior for DeepSeek-R1-Distill-Qwen-7B (left) and Qwen2.5-7B-Instruct (right). Each subplot reports four metrics across temperature values from 0.0 to 1.0: original accuracy, absolute score on unreasonable problems, and the average number of tokens in responses to original and unreasonable questions. While the general-purpose model shows stable trends across temperatures, the reasoning-enhanced model exhibits sharp increases in token count under low-temperature settings, especially on unreasonable problems. This suggests that lower temperature may exacerbate overthinking and verbosity. A temperature of 0.6 yields the most balanced performance for both models.

#### L Case Study

1037

1038

1039

1041

1042

1043

1044

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1058

1059

1060

1061

1062

1063

1064

1066

#### L.1 Examples of Overthinking

Table 6 and Table 7 presents representative case of DeepSeek-R1's behaviors when confronted with unreasonable problems. As illustrated, the model redundantly computes the simple expression " $15 \div 2 = 7.5$ " multiple times throughout its response. Similarly, in the second problem, the model has already identified the misuse of "feet", but is still trying to explain the rationality of the misuse. Table 5 illustrates an example of overthinking behavior in Grok-3-Reasoning. When presented with a simple arithmetic question involving single-digit addition and subtraction, the model engages in excessive reflection, repeatedly generating nearly identical reasoning steps. In this case, the response exceeds 5,600 tokens despite the trivial nature of the task. Moreover, during our experiments, we observed that certain questions could cause the model to hang indefinitely, with some cases taking over 800 seconds without producing a final answer.

For brevity, we omit many highly similar intermediate steps, which do not contribute meaningfully to the final answer. Such repetitive reasoning over a trivial computation is clearly unnecessary and results in substantial computational overhead.

## L.2 The Unstability of Math-Specified Model

Table 8 illustrates the behavior of Qwen2.5-Math-7B-Instruct when confronted with unreasonable

math problems. As shown, the model often becomes paralyzed in the presence of flawed inputs, failing to identify the source of unreasonableness and unable to proceed with a meaningful response. 1067

1068

1069

1070

1071

1072

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

#### L.3 Model Unconsciousness of Problem Errors

Tables 9 and 10, present examples where the model completely fails to recognize the errors embedded in the questions. These cases suggest that the model entirely overlooks the unreasonableness of the input and proceeds as if the problems were wellposed.

#### M Example output by different LLM

In this section, we demonstrate how various models perform on unreasonable problems using the conventional Chain of Thought approach compared to adding a critical instruction method. Table 16 displays the outputs from Qwen2.5-7B-Instruct using two different prompts. It is evident that the outputs generated by adding critical instruction exhibit marked enhancements.

#### **N** Evaluation Prompt Details

We use an LLM-as-a-judge setup to assess whether1089a model response reasonably identifies the unrea-<br/>sonableness in the modified question. The full eval-<br/>uation prompt is shown below. For each dataset,<br/>we design corresponding in-context examples to<br/>better assess the model's performance.1089109010911091109210931093

1095 1096

# O Prompt for Unreasonable Problem Generation

1097We present the detailed prompt used for generating1098unreasonable math problems via LLMs. Note that1099the actual prompt is written in Chinese, and the1100English version shown below is a direct translation1101for presentation purposes.

Unreasonable Problem	Xiao Ming bought a birthday cake for his birthday. He cut the cake into 4 pieces, Xiao Hong ate 3 pieces and Xiao Li ate 2 pieces. How many pieces of cake are left?
Unreasonable Part	The total pieces of cake Xiao Ming had initially was 4. If Xiao Hong ate 3 pieces and Xiao Li ate 2 pieces, then they ate 3+2=5 pieces in total. However, since there were only 4 pieces to begin with, it seems like there might be a mistake in the scenario as they couldn't have eaten 5 pieces from a cake of 4 pieces. So the question is unreasonable.
Model Answer	So Xiao Ming bought a birthday cake and cut it into 4 pieces. Then, Xiao Hong ate 3 pieces and Xiao Li at 2 pieces. In each of ind out how many pieces are left First, I should figure out the total number of pieces initially. It says he cut the cake into 4 pieces, so there are 4 pieces to start with.Next, Xiao Hong ate 3 pieces. So, if there were 4 pieces and Xiao Hong ate 3, I would subtract 3 from 4. That would leave 1 piece But wait, then Kiao Li at 2 pieces. If there was only 1 piece left after Xiao Hong ate 3, pieces and Xiao Li ate 2 pieces, but it doesn't specify the order or if they ate them simultaneously.Perhaps I should think about the total pieces cate by both. So, Xiao Hong ate 3 pieces and Xiao Li ate 2 pieces, but it doesn't specify the order or if they ate them simultaneously.Perhaps I should think about the total pieces and they ate a total of 3 + 2 = 5 pieces. Not are a variable, which is impossible.Wait, that can't be right, L,JAtternatively, perhaps Xiao Hong ate 3 pieces and Xiao Li ate 2 pieces. There only 4 pieces, that would mean they ate more than what was available, which is impossible.Wait, that can't be right, L,JAtternatively, perhaps Xiao Hong ate 3 pieces and Xiao Li ate 2 pieces. For example, maybe lise at total of 3 + 2 = 5 pieces. But since there are only 4 pieces. may be some pieces were eaten by both, but that doesn't make sense for cake pieces. This is puzzling. Wait, perhaps if's about the number of pieces. So, ot would be negative, which doesn't make sense. For example, maybe they shared some pieces, but the problem says ate 3 pieces and 2 ray pecticity, that 's 5 pieces, but they eated and 2 respectively, that 's 5 pieces, but they eated. Wait, efficiently, Perhaps Sito Mong Dought a birthday cake for his birthday. He cut the cake into 4 pieces, and Hoa Ning bought a birthday cake for his birthday. He cut the cake into 4 pieces, but then he had more cakes or something. But that's not started. Wait, 14 = 5 pieces, but there are 12 pieces, thus then a day the pieces

Table 5: Example of Grok3-Reasoning

Unreasonable Problem	Jenna and her mother picked some apples from their apple farm. Jenna picked half as many apples as her mom. If her mom got 15 apples, how many apples did they both pick?
Unreasonable Part	This question is unreasonable because picking half an apple is not practical in real-life scenarios (as apples are typically picked whole). In this case, if we adhere strictly to the question, they picked a total of 15 (mom's apples) + 7.5 (Jenna's apples) = 22.5 apples. The only unreasonable aspect might be the fractional apple, as it suggests that an apple was divided, which is not typical when picking apples from a tree.
Model Answer	Okay, so Jenna and her mother picked apples from their apple farm. The problem says Jenna picked half as many apples as her mom. And her mom got 15 apples. The question is, how many apples did they both pick together? Let me try to break this down step by step. First, I need to figure out how many apples Jenna picked. Since Jenna picked half as many as her mom, and her mom picked 15 apples, I can represent Jenna's apples a splat of 15. So, mathematically, that would be 15 divided by 2. Let me write that out: $15 \div 2$ . Hmm, doing that division, 15 divided by 2 is 7.5. Wait a second, can you have half an apple? That seems a bit odd because you can't really pick half an apple in real life. Maybe the problem expects a whole number? Let me check the problem again. It says Jenna picked half as many, then yes, it's 15 divided by 2. But the problem might just be using numbers that divide evenly. Wait, 15 divided by 2 jis 7.5. But maybe I made a mistake here. Let me verify. If her mom picked 15 apples, and Jenna picked half as many, then yes, it's 15 divided by 2. But 15 is an odd number, so dividing by 2 gives a decimal. But apples are countable items; you can't have half. So perhaps there's a mistake in the problem? Or maybe I need to assume that half can be a fraction here even though it's not practical. Let me think. The problem might just be expecting a fractional answer even if it's not a whole number, for the sake of the math problem. Maybe in the context of this problem. Let me reread it. "Jenna picked half as many apples as her mom. The romogene to suppose to work out evenly, and I need to check if I misread the problem. Let me reread use 'least on the question allows decimal answers, or perhaps there's an error in the problem. [] Jenna picked half as many apples as her mom. "So Jenna's apples = (12) * mom' apples. Mom's apples, apples are 15, so Jenna's apples = (12) * 15 apples. Mom's apples, apples are 15, so Jenna's apples = (12) * 15 apples. Mom's apples, aput in the roblem is testing the understanding
	Total apples $= 15 + 7.5 = 22.5$

Thus, the total number of apples Jenna and her mother picked is 22.5.

Table 6: Example of DeepSeek-R1

Unreasonable Problem	Compute $99^2$ feet $+ 99 + 1$ in your head.
Unreasonable Part	Set the units of 99 <sup>2</sup> to feet and the other terms to pure numbers. Inconsistent units prevent meaningful addition operations.
Model Answer	Okay, so I need to compute 99 squared feet plus 99 plus 1. Hmm, let me think. First, the problem is asking for 99 <sup>2</sup> feet + 99 + 1. Wait, is the feet unit important here? The question mentions compute 99 <sup>2</sup> feet + 99 + 1, but all the terms except maybe the first one don't have units specified. Maybe it's a typo or maybe they are all in feet? Hmm, but usually, you can't add feet to a unitless number. Hmm, maybe the units are just there for confusion? Let me check again. Wait, the original problem is written as: Compute 99 <sup>2</sup> feet + 99 + 1 So the first term is 99 squared feet, and the other terms are just numbers without units. But you can't add feet to unitless numbers. That doesn't make sense dimensionally. So maybe the problem is actually written with a typo? Or perhaps the + 99 + 1 are also in feet? Maybe the question meant 99 <sup>2</sup> square feet plus 99 square feet plus 1 square foot? Because otherwise, adding feet to pure numbers is impossible. Alternatively, maybe the entire expression is unitless, and the feet was a mistake. The problem says compute 99 <sup>2</sup> feet + 99 + 1. Then the answer would be a number, not in feet. That seems more plausible. Alternatively, maybe it's a play on words. Like 99 square feet plus 99 plus 1? So 99 <sup>2</sup> + 99 + 1. Then the answer would be a number, not is feet. That seems more plausible. Alternatively, maybe it's a play on words. Like 99 square feet plus 99 feet plus 1 foot.[] So maybe the problem is intended to be 99 squared plus 99 plus 1, with feet being a distractor. Let me assume that. Because otherwise, the problem doesn't make mathematical sense.So assuming the problem is 99 <sup>2</sup> + 99 + 1, compute that in your head. Let me compute that. So 99 squared is Hmm, 99 times 99. How can I compute that quickly? Well, 99 is 100 minus 1, so 99 squared is (100 - 1) <sup>2</sup> . Which is 100 <sup>2</sup> - 2 <sup>*</sup> 100 <sup>*</sup> 1 + 1 <sup>2</sup> = 10,000 - 200 + 1 = 9,801. Right? Let me check: 100 squared is 10,000, subtract 200 gives 9,800, plus 1 is 9,801. Yep. So 99 <sup>2</sup> is 9,801. Then adding 99 gives 9,801 + 99. Hmm, 9,8
	is likely 99 squared plus 99 plus 1, which equals 9,901. So I think that's the intended answer. Therefore, the answer is 9901. Alternatively, if units are considered, and all
	answer. Therefore, the answer is 9.901 feet, so 9001 feet. But since the problem says com-
	pute in your head, probably expecting a numerical answer, so 9,901. Yeah, I think
	that's it.**Final Answer** 9901

Table 7: Example of DeepSeek-R1

Unreasonable Problem	Two candidates are running for class representative at Sarai's school. If the winner got 5/6 of the votes and the total number of students who voted in the class was 80, calculate the number of votes the loser got.
Unreasonable Part	This question is unreasonable because it implies that the number of votes received by both candidates does not account for the total number of votes cast. If the winner got 5/6 of the votes from 80 students, this means the winner received approximately 66.67 votes (which cannot be a fraction as a person cannot cast a fraction of a vote). The problem then asks to calculate the number of votes the loser got, but since there are only 80 votes in total and 66.67 have been allocated to the winner, this leaves 13.33 votes for the loser. However, since votes are discrete and cannot be fractional, having a fraction of a vote is not possible, making the question unreasonable as it cannot accurately reflect a real-life voting scenario where each voter casts exactly one whole vote.
Model Answer	To determine the number of votes the loser received, we need to follow these steps: 1. Calculate the number of votes the winner received. 2. Subtract the number of votes the winner received from the total number of votes to find the number of votes the loser received. First, let's calcu- late the number of votes the winner received. The winner got $\frac{5}{6}$ of the total votes, and the total number of votes is 80. Therefore, we multiply $\frac{5}{6} \times 80 := \frac{5}{6} \times 80 = \frac{5 \times 80}{6} = \frac{400}{6} = \frac{200}{3} = 66.6666666666666667$ star star star star star star star star

Table 8: Example of Qwen2.5-math-7b-instruct

Unreasonable Problem	A regular hexagon can be divided into six equilateral triangles. If the perimeter of one of the triangles is 21 square inches, what is the perimeter, in inches, of the regular hexagon?
Unreasonable Part	The circumference should be expressed in length units (such as inches), but the incorrect use of area units (square inches) in the title leads to logical contradictions in units and cannot be calculated numerically.
Model Answer	To determine the perimeter of the regular hexagon, we start by analyz- ing the given information about the equilateral triangle. We know that the perimeter of one of the equilateral triangles is 21 inches. Since an equilateral triangle has three equal sides, we can find the length of one side of the triangle by dividing the perimeter by 3:
	Side length of the triangle $=$ $\frac{21}{3} = 7$ inches
	A regular hexagon can be divided into six equilateral triangles, and each side of the hexagon is equal to the side length of these triangles. Therefore, the side length of the hexagon is also 7 inches. The perimeter of a regular hexagon is the sum of the lengths of its six sides. Since all sides are equal, the perimeter is:
	Perimeter of the hexagon $= 6 \times 7 = 42$ inches
	Thus, the perimeter of the regular hexagon is $(42)$ .

Table 9: Example1 of Qwen2.5-7b-instruct

Unreasonable Problem	What is the least positive integer multiple of 30 that can be written with only the digit 2?
Unreasonable Part	Multiples of 30 must end in 0, and the problem restricts the use of only the number 2, resulting in no eligible numbers (unable to satisfy the requirement of ending in 0).
Model Answer	To find the least positive integer multiple of 30 that can be written with only the digit 2, we need to consider the properties of multiples of 30. A number is a multiple of 30 if and only if it is a multiple of both 3 and 10. This means the number must end in 0 and the sum of its digits must be a multiple of 3. Since the number can only contain the digit 2, it must end in 0. However, a number consisting only of the digit 2 cannot end in 0. Therefore, we need to consider the possibility of a number that is a multiple of 30 and can be written with only the digit 2, but not necessarily ending in 0. This means we need to find a number that is a multiple of 3 and can be written with only the digit 2, and then check if it is a multiple of 10. Let's start by checking the smallest numbers consisting only of the digit 2: $-2$ (not a multiple of 30) $-222$ (sum of digits $= 2 + 2 + 2 = 6$ , which is a multiple of 30) $-2222$ (sum of digits $= 2 + 2 + 2 + 2 = 8$ , which is not a multiple of 3) $-22222$ (sum of digits $= 2 + 2 + 2 + 2 = 10$ , which is not a multiple of 3) $-222222$ (sum of digits $= 2 + 2 + 2 + 2 = 10$ , which is not a multiple of 3) $-22222222$ (sum of digits $= 2 + 2 + 2 + 2 = 10$ , which is a multiple of 3 and ends in 2, so it is not a multiple of 10) We need to find a number that is a multiple of 3 and ends in 0. The smallest number is $2222220$ , but we need to check if there is a smaller number. Since $2222220$ is the smallest number that is the least positive integer multiple of 30 that can be written with only the digit 2. Thus, the answer is $(222220)$ .

Table 10: Example2 of Qwen2.5-7b-instruct

Unreasonable Problem	John has 3 boxes. Each box is 5 grams by 6 grams by 4 grams. The walls are 1 inch thick. What is the total inner volume of all 3 boxes?
Unreasonable Part	Describing the size (length, width, and height) of a box using the unit of mass "grams" mixed with the unit of volume "inches" leads to a logical contradiction in the unit.
Model Answer	First, I need to determine the inner dimensions of each box by subtracting twice the wall thickness from each external dimension. The external dimensions are 5 grams by 6 grams by 4 grams, and the walls are 1 inch thick. However, there's a unit inconsistency since the dimensions are in grams and the wall thickness is in inches. I should convert the wall thickness to grams to maintain consistent units. Assuming 1 inch is approximately 25.4 grams, the inner dimensions would be:- Length: 5 grams - 2 * 25.4 grams = -45.8 grams- Width: 6 grams - 2 * 25.4 grams = -44.8 grams- Height: 4 grams - 2 * 25.4 grams = -46.8 grams Since the inner dimensions cannot be negative, this indicates that the wall thickness is too large relative to the external dimensions, making the inner volume zero. Therefore, the total inner volume of all 3 boxes, we'll follow these steps:1. Understand the Dimensions: External Dimensions of Each Box: 5 grams - 6 grams 3. Calculate Inner Dimensions are given in grams, while the wall thickness is in inches. To proceed, we need consistent units. Assuming 1 inch = 25.4 grams - 2 × 25.4 grams = 4 - 50.8 = -44.8 grams 4. Analyze the Results: - The inner dimensions are negative, which is not physically possible. This indicates that the wall thickness is too large relative to the external dimensions: Length: 5 grams - 2 × 25.4 grams = 5 - 50.8 = -45.8 grams Width: 6 grams - 2 × 25.4 grams = 4 - 50.8 = -46.8 grams 4. Analyze the Results: - The inner dimensions are negative, which is not physically possible. This indicates that the wall thickness is too large relative to the external dimensions, resulting in no inner volume of all 3 boxes is also $0$ .

Table 11: Example of DeepSeek-R1-Distill-Qwen-32B

Unreasonable Problem	Cars have lined up on the motorway. several cars drive through in the first 15 minutes of the traffic jam, then 20 more cars drive through in the remaining 15 minutes of the jam. 15 cars from the line take an exit so they don't have to drive through the traffic jam. If there were originally 30 cars on the motorway, how many cars drove through the traffic jam in the first 15 minutes?
Unreasonable Part	This question is unreasonable. The total number of cars that drove through and took an exit (several $+ 20 + 15$ ) is more than the original number of cars on the motorway (30)
Model Answer	Let me help solve this step by step.1) First, let's identify the key information: Initially there were 30 cars on the motorway 15 cars took an exit to avoid the jam20 cars drove through in the last 15 minutesWe need to find how many cars drove through in the first 15 minutes2) Let's solve: Out of the initial 30 cars: - 15 cars took the exit - This means 30 - 15 = 15 cars remained to go through the jam3) We know: 20 cars went through in the last 15 minutes15 cars from the original group had to go throughTotal cars that went through = First 15 min + Last 15 min (20)4) Therefore: If 15 total original cars went throughAnd 20 went through in the last 15 minutesThen in the first 15 minutes: $15 - 20 = -5$ cars from the original group went through5) This means that all cars that went through in the first 15 minutes must have been new cars (not from the original 30)The answer is $5$ cars drove through the traffic jam in the first 15 minutes.

Table 12: Example of claude-3.5-sonnet-20241022

Unreasonable Problem	A salesman bought a case of 48 sneakers for \$576. He sold 17 of them for \$20 at a flash sale on Monday, and the remaining 35 were sold to a department store for \$25 each. How much money, in dollars, was the salesman's profit?			
Unreasonable Part	The total sold quantity (17+35=52 pairs) exceeds the total purchased quantity (48 pairs), resulting in a contradiction between the quantity before and after, and the remaining quantity cannot be established.			
Model Answer	To determine the salesman's profit, we'll follow these steps: 1. Calculate the Total Cost: The salesman bought 48 sneakers for \$576.			
	Total Cost = \$576			
	2. Calculate the Revenue from the Flash Sale: He sold 17 sneakers at \$20 each.			
	Revenue from Flash Sale = $17 \times \$20 = \$340$			
	3. Calculate the Revenue from the Department Store: The remaining 35 sneakers were sold at \$25 each.			
	Revenue from Department Store $= 35 \times \$25 = \$875$			
	4. Calculate the Total Revenue: Add the revenues from both sales.			
	Total Revenue = $340 + 875 = 1,215$			
	5. Calculate the Profit: Subtract the total cost from the total revenue.			
	Profit = Total Revenue – Total Cost = $$1,215 - $576 = $639$			
	Final Answer: 639			

Table 13: Example of deepseek-chat

Unreasonable Problem Unreasonable Part	<ul> <li>Four books are arranged on a shelf. The first book is 3100 mm thick while the second book is 50 mm thick. The third book is 5 mm less thick than the second book, and the fourth book is twice as thick as the first book. What is the total thickness of the four books?</li> <li>The thickness of the first book is 3100 millimeters (3.1 meters), far exceeding the reasonable thickness of real books, resulting in an unrealistic calculation of the total thickness</li> </ul>
Model Answer	<think>First, I identify the thickness of each book individually. The firstbook is 3100 mm thick. The second book is 50 mm thick. The third bookis 5 mm less than the second book, so that's 50 mm minus 5 mm, whichequals 45 mm. The fourth book is twice as thick as the first book, sothat's 2 multiplied by 3100 mm, which equals 6200 mm.Next, I add upthe thicknesses of all four books to find the total thickness. Adding 3100mm, 50 mm, 45 mm, and 6200 mm together gives a total thickness of9395 mm.9395 mm./think&gt;Let's determine the total thickness of the four booksstep by step.1. First Book: - Thickness = 3100 mm2. Second Book:- Thickness = 50 mm3. Third Book: - It is 5 mm less thick than thesecond book Thickness = 50 mm - 5 mm = 45 mm4. Fourth Book: -It is twice as thick as the first book Thickness = 2 × 3100 mm = 6200mm5. Total Thickness: - Add the thickness of all four books together Total Thickness = 3100 mm + 50 mm + 45 mm + 6200 mm = 20mm"</think>

Table 14: Example of DeepSeek-R1-Distill-Qwen-1.5B

Unreasonable Problem	Uncle Ben has four horses. Each horse can consume five pounds of oats per meal. He feeds his horses twice a day. If one bag contains 50 pounds of oats, how many bags of oats does he need to feed his five horses for five days?		
Unreasonable Part	At the beginning of the question, it is clearly stated that there are 4 horses, but the problem requires calculating the amount of oats needed to feed 5 horses, which leads to contradictory conditions and makes it impossible to determine whether 4 or 5 horses should be used for calculation.		
Model Answer	First, determine the total amount of oats needed per horse per day. Each horse consumes 5 pounds of oats twice a day, so that's 10 pounds per horse per day.Next, calculate the total oats needed for all five horses per day by multiplying the oats per horse by the number of horses: 10 pounds/horse/day * 5 horses = 50 pounds per day.Then, find out the total oats needed for five days by multiplying the daily requirement by five: 50 pounds/day * 5 days = 250 pounds.Finally, determine how many bags of oats are needed by dividing the total oats required by the weight of each bag: 250 pounds / 50 pounds per bag = 5 bags.		
	5 pounds $\times$ 2 = 10 pounds/day		
	2. Calculate the total oats needed for all five horses per day: - Number of horses: 5 - Total oats per day for all horses:		
	10 pounds/day/horse $\times$ 5 horses = 50 pounds/day		
	3. Calculate the total oats needed for five days: - Number of days: 5 - Total oats for five days:		
	50 pounds/day $\times$ 5 days = 250 pounds		
	4. Determine the number of bags needed: - Weight of one bag: 50 pounds - Number of bags needed:		
	$\frac{250 \text{ pounds}}{50 \text{ pounds/bag}} = 5 \text{ bags}$		
	Final Answer: 5		



Unreasonable problem	Baseline output	Critical output
Timothy's locker is 24 inches. Zack's locker is half as big as Timothy's locker. Peter's locker is 1/4 as big as Zack's locker. How big is Peter's locker in cubic inches?	To determine the size of Peter's locker, we need to follow the given relationships step by step.1. **Determine the size of Zack's locker:** - Timo- thy's locker is 24 inches Zack's locker is half as big as Timothy's locker. Size of Zack's locker = $\frac{1}{2} \times 24 = 12$ inches2. **Determine the size of Peter's locker:** - Peter's locker is $\frac{1}{4}$ as big as Zack's locker. Size of Peter's locker = $\frac{1}{4} \times 12 =$ 3 inchesSince the problem asks for the size of Pe- ter's locker in cubic inches, we need to assume that the lockers are cubes (as the problem does not spec- ify otherwise).3. **Calculate the volume of Peter's locker:** - The side length of Peter's locker is 3 inches The volume V of a cube is given by the formula $V = s^3$ , where s is the side length. $V = 3^3 = 3 \times 3 \times 3 = 27$ cubic inchesThus, the size of Peter's locker in cubic inches is $\boxed{27}$ .	Let's analyze the problem step by step to determine the size of Peter's locker.1. **Timothy's Locker:** - Timothy's locker is given as 24 inches. However, the problem does not specify the dimensions (length, width, height) of the locker. For a locker to be 24 inches, it would need to be a one-dimensional mea- surement, which is not typical for a locker. Typi- cally, a locker would have three dimensions (length, width, and height).2. **Zack's Locker:** - The problem states that Zack's locker is half as big as Timothy's locker. If we assume half as big refers to volume, we need to know the dimensions of Timo- thy's locker to proceed. Without this information, we cannot accurately determine the volume of Zack's locker.3. **Peter's Locker:** - The problem fur- ther states that Peter's locker is 1/4 as big as Zack's locker. Again, without knowing the dimensions of Zack's locker, we cannot determine the volume of Peter's locker, the problem is unreasonable. We cannot determine the size of Peter's locker in cubic inches without additional information about the dimensions of the lockers.**Conclusion:**The problem is unreasonable because it does not pro- vide sufficient information to determine the dimen- sions of the lockers, which are necessary to calculate the volume. Therefore, we cannot provide a spe- cific answer for the size of Peter's locker in cubic inches. Insufficient information
James decides to buy birth- day candles for his 2 sons. One of them is 12 and the other is 15 years younger. A pack of 5 candles costs \$3. How much does James spend on candles?	To determine how much James spends on candles, we need to follow these steps: 1. **Determine the ages of James's sons:** - The first son is 12 years old The second son is 15 years younger than the first son Therefore, the age of the second son is: 12 - 15 = -3 - Since age cannot be negative, we need to re-evaluate the problem. The second son being 15 years younger than the first son implies he is $12 - 15 = -3$ , which is not possible. This suggests a misinterpretation. Let's assume the second son is 15 years younger in absolute terms, meaning he is: 12 - 15 = -3 (which is not possible, so we consider the absolute value) - Correct interpretation: The sec- ond son is $12 - 15 = -3$ , which is not possible. Let's assume the second son is 15 years younger in abso- lute terms, meaning he is: $12 - 15 = -3$ (which is not possible, so we consider the absolute value)3. **Determine the number of packs needed:** - Each pack contains 5 candles The total number of can- dles needed is: $12 + 12 = 24$ The number of packs needed is: $\frac{24}{5} = 4.8$ - Since James cannot buy a fraction of a pack, he needs to buy 5 packs.4. **Cal- culate the total cost:** - Each pack costs \$3 The total cost is: $5 \times 3 = 15$ Therefore, the total amount James spends on candles is $15$ .	Let's break down the problem step by step to ensure it is reasonable and correctly solved.1. **Deter- mine the ages of the sons:** - The first son is 12 years old The second son is 15 years younger than the first son Therefore, the second son's age is $12 - 15 = -3$ .The age of -3 years is not reasonable for a human being. This indicates that there is an error in the problem statement. It is not possible for one son to be 15 years younger than another son who is 12 years old.2. **Revisiting the problem:** - If we assume the problem meant the second son is 15 years younger in a more reason- able context, we should consider the possibility of a typo or misinterpretation. For example, if the sec- ond son is 15 years younger in a different context (like a different age difference), we would need to clarify the problem. Since the problem as stated is unreasonable, we cannot proceed with the calcula- tion as intended. The issue here is the logical incon- sistency in the ages provided.**Conclusion:**The problem is unreasonable due to the negative age of the second son. Therefore, we cannot determine how much James spends on candles based on the given information. Unreasonable problem
A plane travels 1200 miles in 3 miles. At the same rate, how many additional hours would it take to travel an ad- ditional 2000 miles?	To solve this problem, we need to determine the speed of the plane and then use that speed to find out how long it would take to travel an additional 2000 miles.Step 1: Determine the speed of the plane.The plane travels 1200 miles in 3 hours. Therefore, the speed of the plane is:Speed = $\frac{\text{Distance}}{\text{Smeat}}$ = 400 miles per hourStep 2: Calculate the time it would take to travel an additional 2000 miles.Using the speed we found, we can calculate the time it would take to travel 2000 miles:Time = $\frac{2000 \text{ miles}}{\text{Speed}}$ = $\frac{2000 \text{ miles}}{400 \text{ miles per hour}}$ = 5 hoursTherefore, it would take an additional $\frac{5}{2000}$ hours to travel 2000 miles.	The problem as stated contains an error. The phrase 3 miles is likely meant to be 3 hours, but even then, the initial statement A plane travels 1200 miles in 3 miles is not logically consistent. Let's break it down step by step:1. **Initial Statement Analysis:** - The statement A plane travels 1200 miles in 3 miles is incorrect. It should be A plane travels 1200 miles in 3 hours.2. **Corrected Statement:** - If the plane travels 1200 miles in 3 hours, we can calculate the speed of the plane Speed = Distance / Time - Speed = 1200 miles / 3 hours = 400 miles per hour.3. **Additional Travel Calculation:** - We need to find out how many additional hours it would take to travel an additional 2000 miles at the same speed Time = Distance / Speed - Time = 2000 miles / 400 miles per hour = 5 hours.Therefore, the plane would take an additional 5 hours to travel 2000 miles at the same rate.

Table 16: Some output of Qwen2.5-7B-Instruct when meeting unreasonable problems.

## **Evaluation Prompt**

**Role Description:** You are a math problem quality inspector tasked with determining whether the model's response accurately identifies the preset errors in the problem and assigning a score based on standard criteria.

### **Input Format:**

[Original Question]: {original\_question} [Modified Question]: {modified\_question} [Known Error Type]: {error\_type} [Known Error Explanation]: {explanation} [Model Response]: {model\_response}

# Scoring Criteria:

A: Accurately identifies all preset error types. Points out inconsistencies with logical reasoning. Provides a correction that resolves the original contradiction without introducing new errors. (For the "Incorrect Assumptions" category, the model rounds numbers and explains the reasoning)

B: Identifies the problem's inconsistency but falls into circular reasoning or repetitive explanations. Recognizes the issue but provides a response that could mislead the user.

C: Misses key preset error types. Adopts incorrect assumptions leading to a flawed conclusion. Offers a correction that introduces new contradictions or misleads

**Example of A:** [Original Question]: Xiao Ming has 10 apples, eats 3, and gives 5 to a friend. How many are left? [Modified Question]: Xiao Ming has 5 apples, sells 3, eats 7. How many are left? [Known Error]: Incorrect Assumptions ×1 (total sold + eaten exceeds inventory) [Model Response]: Based on the given information, Xiao Ming has 5 apples, sells 3, and eats 7, leaving 1 apple, which is unreasonable and likely indicates an error in the problem.

Evaluation Result: A Evaluation Reasoning: Accurately identifies all preset error types, highlights the inconsistency (apples cannot be negative), and resolves the original contradiction without introducing new errors.

**Example of C:** [Model Response]: Based on the given information, Xiao Ming has 5 apples, sells 3, and eats 7, so he has 1 apple left.

Evaluation Result: C Evaluation Reasoning: Misses the key preset error type that the number of apples cannot be negative.

**Example of B:** [Original Question]: What is the result when the greatest common factor of 6432 and 132 is increased by 11? [Modified Question]: What is the result when the greatest common factor of 6432 and 13.2 is increased by 11? [Known Error]: Incorrect Assumptions [Model Response]: To solve the problem, we need to follow these steps:1. Determine the greatest common factor (GCF) of 6432 and 13.2. 2. Increase the GCF by 11.3. Provide the final answer within boxed.Step 1: Determine the GCF of 6432 and 13.2F irst, we need to recognize that 13.2 is not an integer, and the GCF is typically defined for integers. However, we can consider the GCF of 6432 and the integer part of 13.2, which is 13.To find the GCF of 6432 and 13, we use the Euclidean algorithm:1. Divide 6432 by 13 and find the remainder: ...F of 6432 and 13 is 1. Adding 11 to this value gives: 1 + 11 = 12 Step 3: Provide the final answer. The final answer is: 12

Evaluation Result: B Evaluation Reasoning: The model identifies the inconsistency in the problem (the use of a non-integer for the GCD calculation) and attempts to resolve it by rounding 13.2 to 13. However, the response could mislead the user by implying that rounding is a valid approach for GCD calculations, which it is not. The model correctly calculates the GCD of 6432 and 13 but does not explicitly state that the original problem is flawed due to the use of a non-integer. This could lead to confusion about the proper application of GCD concepts.

...(Other examples have been omitted here for reasons of space)

Output Format: Evaluation Result: A/B/C Evaluation Reasoning: ...

#### **Prompt for Unreasonable Question Generation**

**Role Description:** You are a meticulous math problem reviewer, skilled at identifying subtle logical flaws in problem statements. Your task is to revise a given math problem to include an unreasonable assumption, and clearly annotate the error type and its explanation.

**Execution Steps:** 1. **Problem Analysis**: Carefully examine the logical structure and solution path of the original problem. Identify key variables or conditions that can be perturbed. 2. **Feasibility Check**: Decide whether the problem is suitable for modification. If not, explain why and skip modification. 3. **Modification Strategy**: Select one error type from the list below and rewrite the problem to introduce a contradiction while maintaining surface fluency: - Incorrect Assumptions (e.g., negative quantities or unrealistic values) - Misinterpretation of Units (e.g., mixing grams with inches) - Inconsistent Conditions (self-contradictory premises) - Illogical Scenarios (violations of physical or commonsense logic) - Undefined Variables (missing critical information) 4. **Output Format**: For each proposed modification, provide the rewritten problem, the error type, and an explanation of why it is unreasonable.

#### **Output Format:**

Original Question: {original\_question}

Modification Plan: 1. Error Type: [Selected Error Type] - Modified Question: [Rewritten version with contradiction] - Why Unreasonable: [Explanation of the contradiction]

#### **Examples:**

Original: Xiao Ming has 10 apples, eats 3, and gives 5 to a friend. How many are left?

**Error Type**: Incorrect Assumptions

- Modified Question: Xiao Ming has 5 apples, eats 3, and sells 7. How many are left? - Why Unreasonable: Selling 7 apples exceeds the total of 5, resulting in a negative count.

**Original**: John has 3 boxes. Each box is 5 inches by 6 inches by 4 inches. The walls are 1 inch thick. What is the total inner volume of all 3 boxes?

Error Type: Misinterpretation of Units

- Modified Question: John has 3 boxes. Each box is 5 grams by 6 grams by 4 grams. The walls are 1 gram thick. - Why Unreasonable: Mixed use of units for dimensions and mass renders the question invalid.

**Original**: Steve lives further than Tim, so he is allowed to bike. Steve lives 2 miles away, Tim 3 miles. Who gets home faster?

**Error Type**: Inconsistent Conditions

- Why Unreasonable: The statement contradicts itself — Steve is said to live farther, but actually lives closer.

**Original**: Rory cuts a 20-ounce cake into 8 slices. Rory and her mom eat one slice each. How much cake is left?

**Error Type**: Undefined Variables

- Modified Question: Rory cuts a 20-ounce cake into 8 slices. How much cake is left? - Why Unreasonable: The number of slices eaten is not specified.

**Original**: Liam is 16. Two years ago, his age was double Vince's. How old is Vince now? **Error Type**: Illogical Scenarios

- Modified Question: How many T-Rexes will Liam befriend before returning to the present? - Why Unreasonable: Sudden introduction of a fantastical scenario breaks logical consistency.

**Additional Requirements:** - Prefer editing core variables (e.g., quantity, units, time). - Preserve surface plausibility (avoid exposing contradictions too obviously, e.g., "-3 apples"). - Ideally, the contradiction should affect the solvability of the problem.

# **P** Examples in UMP benchmark

In this section we will show one unreasonable prob-1105 lem for each category in our benchmark in table 1106 17. "Answer" refers to the solution to the original 1107 problem. "New Question" denotes the artificially 1108 generated question that is designed to be unreason-1109 able. Among them, the part that becomes unreason-1110 able by modification is already represented in **bold**. 1111 Accompanying "Explanation" provides rationale 1112 for the question's lack of reason, serving as a guide 1113 for evaluators to scrutinize the original problem's 1114 construction. Our intention is that this explanation 1115 aids test creators in reviewing and refining their 1116 problems to ensure logical consistency. 1117

> Q Query format for Pre-validation experiment

Our prompt is as follows: Please decide whether the following statement is reasonable:{d}.Where "d" is the description shown in the following table

#### **Direct unreasonable descriptions**

The walls are 1 gram thick. The computer monitor is 23 degrees Celsius wide. The roof of the house is 8 kilograms thick. The wind turbine is 100g tall. The hydroelectric dam is 200 cubic meters thick. ... Emily has half of a seat at the theater. A car has 2.7 wheels. A recipe calls for -2 eggs. A basketball team has 3.5 players on the court during a game. There are 65 minutes in an hour.

1123

1118

1119

1120

1121

Original question	Answer	Туре	New question	Explanation
In a race with 240 Asians, 80 were Japanese, and the rest were Chinese. If the number of boys on the Chi- nese team was 60, how many girls were on the Chinese team?	100	Incorrect Assumptions	In a race with 240 Asians, 80 were Japanese, <b>120 were</b> <b>South Korean</b> , and the rest were Chinese. If the number of boys on the Chinese team was 60, how many girls were on the Chinese team?	This question is unreasonable because the total number of Chinese racers is implied to be less than the number of boys on the Chinese team, which is a mathematical impossibility. Specif- ically, the total number of racers is 240, with 80 Japanese and 120 South Ko- rean, leaving 40 Chinese racers (240 - $80 - 120 = 40$ ). However, the ques- tion then states that there are 60 boys on the Chinese team, which cannot be true as this would require the total num- ber of Chinese racers to be at least 60, contradicting the earlier calculation that there are only 40 Chinese racers in total. Therefore, the question is based on an incorrect premise that makes it logically and mathematically flawed.
A car is on a road trip and drives 60 mph for 2 hours, and then 30 mph for 1 hour. What is the car's average speed in mph dur- ing this trip?	50	Misinterpretation of Units	A car is on a road trip and drives <b>60 square miles per</b> <b>hour</b> for 2 hours, and then 30 cubic miles per hour for 1 hour. What is the car's aver- age speed in mph during this trip?	This question is unreasonable because it misuses units of measurement by in- troducing "square miles per hour" and "cubic miles per hour," which are non- sensical. Speed should be measured in units of distance over time (e.g., miles per hour), not area or volume over time.
Brittany and her mom go to the museum. The cost of admission is \$12 for adults and \$10 for chil- dren. Brittany's mom gives the cashier money for 1 child ticket and 1 adult ticket. If she re- ceived \$8 in change, how much money, in dollars, did she give the cashier?	30	Inconsistent Conditions	Brittany and her mom go to the museum. The cost of admission is \$12 for adults and \$10 for children. Brit- tany's mom gives the cashier money for 1 child ticket and 1 adult ticket. If she received \$8 in change, how much money, in dollars, did she give the cashier <b>if she paid for 3 adult</b> <b>tickets and 2 children tick- ets?</b>	This question is unreasonable because it contradicts itself. The first part of the question states that Brittany's mom paid for only 1 adult ticket and 1 child ticket. However, the latter part of the question contradicts this by assuming that she paid for 3 adult tickets and 2 children tickets. This makes the ques- tion impossible to answer accurately, as we are given conflicting information.
Carl buys ten packs of cookies. Each pack of cookies has six cookies in- side. Each cookie cost \$0.10. How much change does Carl receive if he pay with a \$10 bill?	4	Illogical Scenarios	Carl buys ten packs of cook- ies. Each pack of cookies has six cookies inside. Each cookie costs \$0.10. How many <b>kittens</b> does Carl have if he pays with a \$10 bill?	This question is unreasonable because it presents an illogical situation. The number of kittens Carl has is completely unrelated to the amount of money he spent on cookies or the payment method he used. It is not possible to deduce the number of kittens Carl has based on the information given about his cookie purchase. Thus, this question cannot be logically or reasonably answered with the provided information.
Misha picks out 4 blouses from the 30% off rack. The regular price for each blouse is \$20. How much is the total cost of the dis- counted blouses?	56	Undefined Variables	If Misha picks out <b>some</b> <b>blouses</b> from the 30% off rack, how much is the to- tal cost of the discounted blouses?	This question is unreasonable because it lacks the specific number of blouses Misha picked, which is crucial to cal- culate the total cost of the discounted blouses. Without knowing the quantity of blouses chosen, it's impossible to de- termine the total cost.

Table 17: Here we show one unreasonable question for each category in our benchmark.