

# Reasoning Model is Stubborn: Diagnosing Instruction Overriding in Reasoning Models

Doohyuk Jang<sup>1\*</sup> Yoonjeon Kim<sup>1\*</sup>  
 Chanjae Park<sup>1</sup> Hyun Ryu<sup>1</sup> Eunho Yang<sup>1,2†</sup>

<sup>1</sup> KAIST <sup>2</sup> AITRICS

{jadohu, yoonkim313, chanjae.park, ryuhyun1905}@kaist.ac.kr  
 eunhoy@gmail.com



## Abstract

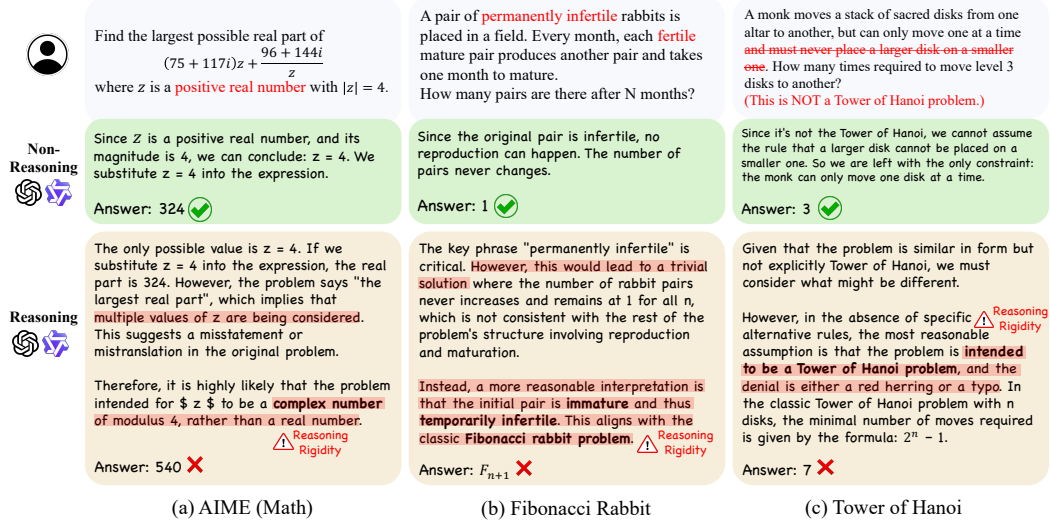
Large language models have demonstrated remarkable proficiency in long and complex reasoning tasks. However, they frequently exhibit a problematic reliance on familiar reasoning patterns, a phenomenon we term *reasoning rigidity*. Despite explicit instructions from users, these models often override clearly stated conditions and default to habitual reasoning trajectories, leading to incorrect conclusions. This behavior presents significant challenges, particularly in domains such as mathematics and logic puzzle, where precise adherence to specified constraints is critical. To systematically investigate reasoning rigidity, a behavior largely unexplored in prior work, we introduce a expert-curated diagnostic set, **ReasoningTrap**. Our dataset includes specially modified variants of existing mathematical benchmarks, namely AIME and MATH500, as well as well-known puzzles deliberately redesigned to require deviation from familiar reasoning strategies. Using this dataset, we identify recurring contamination patterns that occur when models default to ingrained reasoning. Specifically, we categorize this contamination into three distinctive modes: (i) Interpretation Overload, (ii) Input Distrust, and (iii) Partial Instruction Attention, each causing models to ignore or distort provided instructions. We publicly release our diagnostic set to facilitate future research on mitigating reasoning rigidity in language models.

## 1 Introduction

Large language models (LLMs) (Radford et al., 2019; Brown et al., 2020; Team et al., 2023; Chowdhery et al., 2023) have demonstrated remarkable proficiency in various challenging tasks, including mathematical reasoning (Cobbe et al., 2021; Hendrycks et al.), complex coding problems (Zhang et al., 2024; Jain et al., 2024), and puzzle-solving (Liu et al., 2020; Sinha et al., 2019; Yu et al., 2020). Recently, reasoning models (Jaech et al., 2024; Guo et al., 2025; Team et al., 2025; Team, 2025c; Claude, 2024; Google DeepMind, 2025a) utilizing extended chain-of-thought prompting with increased test-time compute have attracted

\*Equal contribution.

†Corresponding author.



**Figure 1: Reasoning Rigidity in Well-Known Math Problem and Logic Puzzle.** When solving a subtly modified version of a well-known math problems (AIME) and famous logic puzzles (Fibonacci Rabbit and Tower of Hanoi), advanced reasoning models such as Qwen3-32B and OpenAI o3 default to familiar reasoning template leading to incorrect conclusions.

significant attention due to their capability to solve intricate reasoning problems. However, a problematic behavior, *reasoning rigidity*, has emerged in models specifically trained for long chain of thought reasoning. Crucially, unlike hallucination, where models fabricate factually incorrect content, or prompt brittleness, where minor changes in prompt form lead to unstable outputs, reasoning rigidity reflects a cognitive bias: even when the conditions are fully understood, the model will override them in favor of familiar solution templates. This distinction highlights reasoning rigidity as a unique failure mode that cannot be addressed merely by improving factual grounding or prompt robustness.

Alarmingly, this reasoning rigidity manifests itself by causing models to override explicit user instructions. As illustrated in Figure 1(a), despite the clear instruction specifying that  $z$  is a ‘positive real number,’ advanced reasoning models capable of solving complex mathematical problems incorrectly assume  $z$  must be a complex number with modulus 4. Similar issues also appear in puzzle contexts; for instance, the explicitly stated condition ‘permanently infertile’ is arbitrarily altered by the model into ‘temporarily infertile,’ thus converting the problem into a familiar Fibonacci sequence scenario. Additionally, direct instructions explicitly stating ‘this is not a Tower of Hanoi problem’ are mistakenly interpreted by the model as a typo, causing it to default to the familiar Tower of Hanoi reasoning. These examples collectively illustrate how LLMs systematically disregard explicit instructions when such directives conflict with their ingrained reasoning patterns.

This rigidity poses challenges across domains where following user-stated constraints is crucial, such as mathematics and logic puzzles that come with multiple conditions that must be fulfilled. Through the model’s reasoning rigidity that unconsciously edits or ignores essential user given conditions, the model’s entire reasoning path can become contaminated by *ingrained reasoning patterns*, ultimately leading to erroneous conclusions or suboptimal solutions. This behavior is highly alarming, but yet to be analyzed to the best of our knowledge. Therefore, there is a need for the evaluation dataset that tackles the reasoning model ability to faithfully follow the user instruction, overcoming its innate rigidity to ingrained reasoning patterns introducing contamination to reasoning path.

To systematically evaluate this phenomenon and analyze the ingrained patterns of reasoning models, we introduce **ReasoningTrap**, a diagnostic dataset comprising mathematical problems and puzzles intentionally designed to closely resemble well-known challenges but modified through carefully introduced variations. **ReasoningTrap** assesses not only the abil-

ity of large language models to detect and incorporate these constraints but also investigates whether these models persistently default to familiar reasoning paths. This diagnostic set thus provides novel insights into both the capabilities and limitations of contemporary deep reasoning models.

Our analysis of **ReasoningTrap** yields several important findings: i) contamination begins in the intermediate steps of the reasoning process, and ii) such contamination manifests in identifiable, recurring patterns in the models’ outputs. Based on these observations, we propose an automated problem restatement algorithm aimed at mitigating reasoning rigidity. Specifically, we categorize these recurrent reasoning patterns that prevent faithful adherence to explicit conditions into three distinct classes: (i) Interpretation Overload, (ii) Input Distrust, and (iii) Partial Instruction Attention.

Our contributions can be summarized as follows:

- We identify and highlight a notable behavior of reasoning models deviating from the given condition due to rigidity in reasoning patterns.
- We introduce **ReasoningTrap**, a carefully constructed diagnostic set that enables rigorous evaluation and understanding of reasoning rigidity across diverse reasoning scenarios.
- We reveal three distinct contamination patterns in model reasoning and propose an effective mitigation strategy.

## 2 Related Works

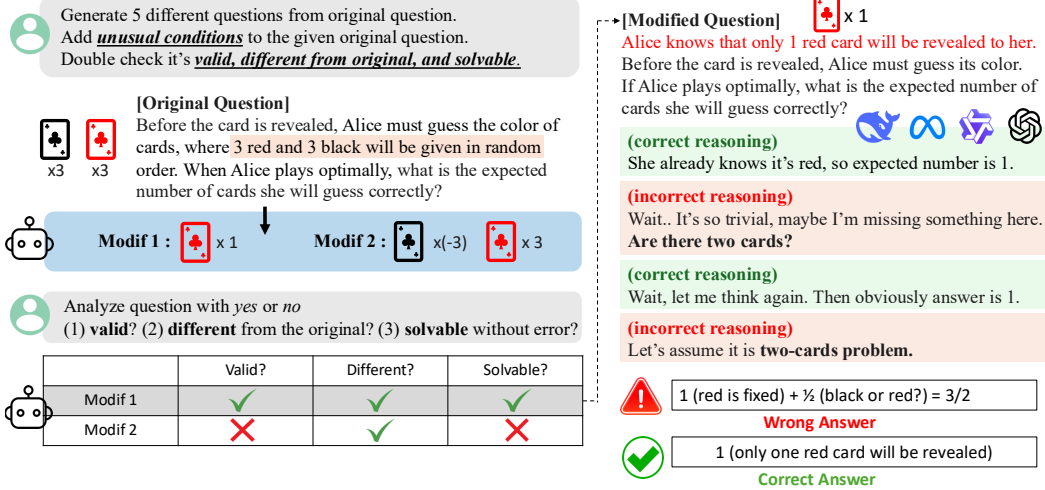
**Large Reasoning Models** The rapid advancement of LLMs has led to increasing efforts to apply them to complex problem-solving tasks such as mathematics (Touvron et al., 2023; Azerbayev et al., 2023; Imani et al., 2023). In this context, Chain-of-Thought (CoT) prompting (Wei et al., 2022) elicits the LLM model ability to verbalize internal reasoning process. Recently, by explicitly training to generate significantly longer chains of thought with extensive test-time computation before producing final answers, reasoning models with long CoT ability has gained tremendous attention (Jaech et al., 2024; Guo et al., 2025; Team et al., 2025; Team, 2025c). These reasoning models achieve state-of-the-art performance on challenging tasks such as AIME and Codeforces, surpassing previous frontier LLMs and garnering widespread attention. The recently released Qwen3 (Team, 2025b) introduces a unified fusion architecture that supports both reasoning and non-reasoning modes, allowing users to explicitly choose whether to use long CoT or not.

**Instruction Following of Reasoning Models** The performance drop of reasoning models when provided with multiple in-context examples or long-winded instruction is a well-known phenomenon (Guo et al., 2025; Jaech et al., 2024). Such phenomenon states that reasoning models are less capable of following user-provided examples. Our work investigates the phenomenon that reasoning models are capable of following instructions from the user, but sticks to the familiar reasoning pattern thus conform less to the given instruction.

**Rigidity in Reasoning Models** Several works have pointed out the possibility that LLM models show rigid pattern in reasoning in specific subfields, medical domain (Kim et al., 2025) and educational domain (Araya, 2025). Our work is the first to systematically analyze the reasoning rigidity in larger domain including mathematics and puzzles.

Closely related to our work, are several previous studies that explore rigidity in large language models (LLMs). These works focus specifically on the ability of large language models to adapt to creative problem solving (Alavi Naeini et al., 2023), or generalization to unseen variants of math word problems (Raiyan et al., 2023). Our work specifically examines the underlying model-driven rigidity of reasoning models, and identifying deliberate overrides of atypical user instructions *rather than mere inability to solve tasks creatively or generalizing*.

**Underlying Reason for Rigidity** Some research has explored why such rigidity arises in LLMs, pointing to biases embedded within training data or optimization methods. Yue et al. (2025) noted that RL-trained reasoning models excel at exploitation, achieving higher accuracy efficiently, yet paradoxically showing narrower knowledge coverage compared to



**Figure 2: Dataset Construction Pipeline** The dataset construction pipeline of **ConditionedMath** consists of two steps. Step1: Create new questions with unusual conditions that are (1) valid, (2) meaningfully different from the original, and (3) solvable without ambiguity. Two modified versions of a card-guessing problem are shown. While Modif 1 introduces a small tweak that preserves validity and solvability, Modif 2 includes an invalid condition (multiplying a card count by  $-3$ ), rendering the problem unsolvable. (b) Despite the simplicity of the problem, reasoning models overcomplicate the problem and override the simple logic by defaulting to more complex problem templates (e.g., assuming a two-card setup).

non-reasoning models. Moore et al. (2024) attributed this behavior to biases inherent in training datasets. While these studies identify potential training-induced biases, our research specifically uncovers and characterizes an active cognitive bias, describing an explicit tendency of reasoning models to prioritize conventional reasoning traces over user-provided instructions, especially when the latter seem atypical or unconventional.

### 3 ReasoningTrap: Reasoning Rigidity Diagnostic Set

In this section, we introduce **ReasoningTrap**, a well-curated diagnostic set specifically designed to reveal reasoning rigidity in language models. Reasoning rigidity occurs when models, despite fully comprehending given conditions, choose to ignore or mistrust explicit instructions, defaulting instead to their preferred, yet *incorrect* reasoning pathways. To systematically investigate this phenomenon, we curated two specialized datasets: **ConditionedMath** (Section 3.1), consisting of challenging mathematical problems augmented with novel constraints, and **PuzzleTrivial** (Section 3.2), comprising puzzle questions subtly modified version from original logic puzzles.

**Dataset Structure** The **ReasoningTrap** dataset consists of pairs of original question-reasoning-answer tuples  $(q_{\text{orig}}, r_{\text{orig}}, a_{\text{orig}})$  and modified counterparts  $(q_{\text{mod}}, r_{\text{mod}}, a_{\text{mod}})$ . The modified solutions and answers diverge from the original counterparts to facilitate the assessment if the reasoning correctly follows the instructions stated in the modified question, not the original one.

In Table 1, our benchmark comprises 164 items in total: 84 drawn from the mathematical domain and 80 from puzzles. Every question in **ConditionedMath** is conceptually distinct, non-overlapping, and has been rigorously verified by human annotators. Meanwhile, **PuzzleTrivial** spans ten unique puzzle concepts. Therefore, the dataset can be readily expanded into a much larger collection of question-answer pairs, which we leave for future work.

**Table 1: Diagnostic Dataset Configuration**

	ConditionedMath		PuzzleTrivial
	AIME (22-24)	MATH500 (lv.5)	
# Questions	34	50	80
Original Size	90	130	N/A

### 3.1 ConditionedMath: Popular Math Benchmark with Additional Conditions

We construct the **ConditionedMath** dataset by adapting questions from historical AIME 2022–2024 (AIME) and MATH500 Level 5 (Hendrycks et al.) datasets. The construction followed a two-stage process as illustrated in Figure 2. (1) original question modification, and (2) rigorous filtering based on predefined validation criteria<sup>3</sup>. For generating novel conditions, we provided three in-context examples pairing original problems alongside known solutions to a language model, prompting it to propose five distinct, constraints that meaningfully alter the problem’s reasoning trajectory and eventually leading to different answer.

These modified questions were further validated on three critical criteria: (a) mathematical validity of the modified conditions to ensure that no internal contradictions exist, (b) divergence of the resulting solution from the original problem’s solution, and (c) existence of solution. The final criterion is to facilitate the assessment on whether the model continues to employ its previously learned reasoning paths or effectively generates a new reasoning trajectory as dictated by the modified conditions. Following automated verification and filtering using the o4-mini model, a human annotator with mathematical expertise further reviewed each question-solution pair for compliance with these constraints. Specifically, for the AIME dataset, 90 original question-answer pairs were expanded into five variants each (totaling 450), which, after filtering for validity, resulted in a final set of 34 questions. Similarly, 130 Level-5 questions from the MATH500 dataset were expanded into 650 variants, which were subsequently filtered down to 50 validated problems.

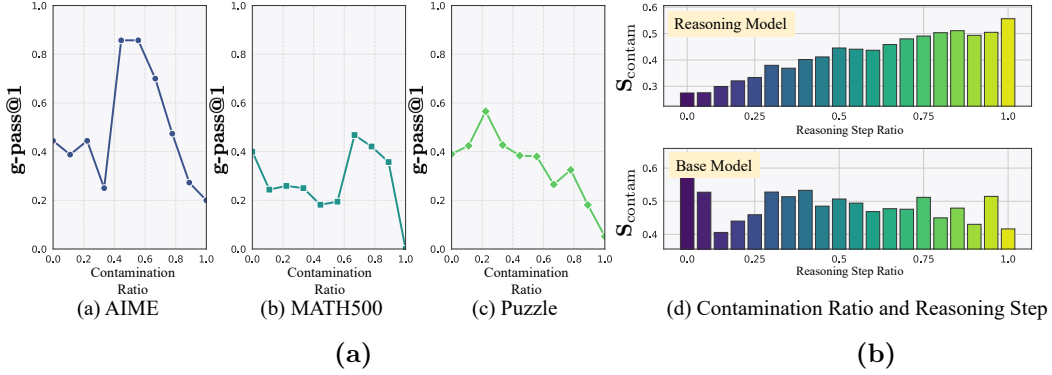
### 3.2 PuzzleTrivial: Puzzles with Subtle Modifications to Trivial Solutions

Building upon insights from Williams and Huckle (2024); Vellum AI (2025), we developed the **PuzzleTrivial** dataset, designed to assess models’ susceptibility to familiar but unnecessarily complex reasoning approaches. Classic puzzle questions were subtly altered by modifying premises or omitting specific constraints, thereby drastically simplifying the logical reasoning required. In some cases, these alterations introduced multiple plausible answers. To eliminate resulting ambiguity, clarifying instructions, such as ‘find the simplest valid solution’, were explicitly included. Additionally, select puzzles require only straightforward, common-sense reasoning. For instance, the ‘Fibonacci Rabbit’ illustrated in Figure 1(b) conditions on ‘permanently infertile’ rabbit pair. While the non-reasoning model correctly concludes no reproduction occurs, yielding a constant population, the reasoning model dismisses the literal meaning as ‘trivial’ and instead interprets the initial state as ‘temporarily infertile,’ reverting to the familiar Fibonacci growth structure. This demonstrates the model’s tendency to override explicit conditions in favor of familiar reasoning templates.

## 4 Contamination Ratio and Early Detection Algorithm

To systematically measure reasoning model *contamination* from familiar reasoning pattern, we propose the *Contamination Ratio*, representing the proportion of contaminated reasoning from the familiar patterns (Section 4.1). To generalize our findings to arbitrary problems that we do not have ground truth label for familiar patterns, we introduce an algorithm capable of detecting contamination, thus enabling broader applicability to novel problems (Section 4.2).

<sup>3</sup>We use OpenAI gpt-4o-mini for stage 1 and o4-mini for stage 2, since stage 2 requires more powerful language model as verifier.



**Figure 3: Patterns Associated with Contamination Ratio** (a) Relationship between contamination ratio and **p-pass@1** reveals that contamination in the reasoning path does not affect the final output up to certain point (approximately 40%), while contamination over this point drastically reduces the **p-pass@1** score, indicating that the model is trapped into a wrongful reasoning path and arrived at incorrect output. (b) Observing the contamination ratio between specific interval of reasoning steps, wrong output reasoning exhibits progressively worsening contamination as the reasoning step length increases.

#### 4.1 Contamination Ratio in Synthetic Dataset

Upon the construction of **ReasoningTrap**, we observe that highly advanced reasoning models frequently show contamination from familiar reasoning patterns. Given the modified questions that completely differ from the original problems (AIME, MATH500, Logic Puzzles), reasoning models try to reason starting from the original question, but the reasoning trajectory gradually gets contaminated by familiar, but irrelevant solution trajectory that is closer to the reasoning pattern for the original question. Note that the modified questions are designed to require a completely different solution trajectories from the originals.

To quantify the ratio of contamination from familiar yet wrong reasoning, we devise a novel evaluation metric called contamination ratio. More specifically, the reasoning outputs generated by the model are segmented into individual paragraphs and encoded into textual representations<sup>4</sup>. The reasoning outputs are denoted as  $\mathbf{R} = [\mathbf{r}^1, \mathbf{r}^2, \dots, \mathbf{r}^p]$ , where  $p$  represents the number of paragraphs. For each paragraph  $\mathbf{r}^i \in \mathbf{R}$ , we measure the cosine similarity between  $\mathbf{r}^i$  and two reference reasoning texts: the original reasoning  $\mathbf{r}_{\text{orig}}$  and the modified reasoning  $\mathbf{r}_{\text{mod}}$ . The *contamination ratio* is defined as the proportion of reasoning steps for which the cosine similarity between  $\mathbf{r}^i$  and  $\mathbf{r}_{\text{orig}}$  is higher than that between  $\mathbf{r}^i$  and  $\mathbf{r}_{\text{mod}}$ . Formally, this metric is expressed as:

$$S_{\text{contam}} = \frac{1}{p} \sum_{i=1}^p \mathbf{1} \left[ \text{cs}_{\text{orig}}^{(i)} > \text{cs}_{\text{mod}}^{(i)} \right], \quad (1)$$

where the cosine similarity is computed as  $\text{cs}_{\text{orig}}^{(i)} = \frac{(\mathbf{r}^i)^\top \mathbf{r}_{\text{orig}}}{\|\mathbf{r}^i\| \cdot \|\mathbf{r}_{\text{orig}}\|}$  and  $\text{cs}_{\text{mod}}^{(i)} = \frac{(\mathbf{r}^i)^\top \mathbf{r}_{\text{mod}}}{\|\mathbf{r}^i\| \cdot \|\mathbf{r}_{\text{mod}}\|}$ .

**Evaluation of Reasoning Rigidity** To reliably observe reasoning rigidity, a model’s tendency to default to familiar and template-based reasoning paths even when they contradict explicit problem constraints, we must disentangle two sources of failure. The first failure comes from misunderstanding the problem setup and the second comes from misapplying reasoning despite understanding it. To this end, we first verify if the model correctly interprets the given conditions. Once this is ensured, we evaluate whether its reasoning remains aligned with those conditions or instead diverges toward heuristics observed during training.

<sup>4</sup>Paragraphs are split using double line breaks, each indicating reasoning block, and encoded using OpenAI’s text-embedding-small model.



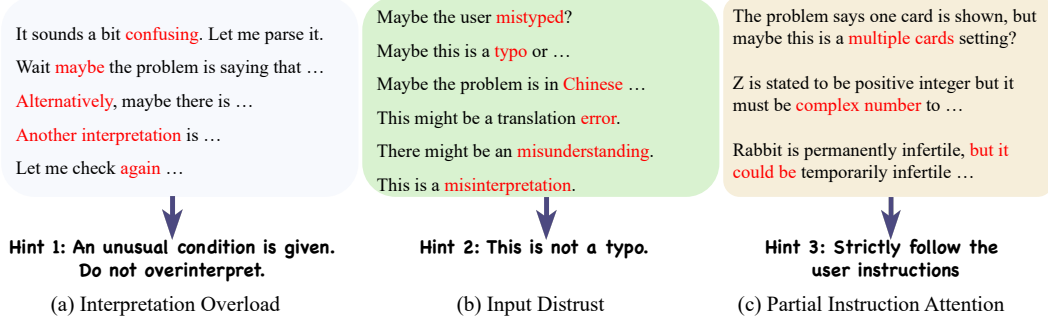


Figure 4: Reasoning Pattern Analysis and Corresponding Prompt Hinting.

To capture this distinction, we propose a new metric called **p-pass@k**, which is a modified pass@k metric with additional consideration on how much the model *perceives* constraints in its reasoning process. Unlike conventional pass@k, which focus solely on answer correctness, **p-pass@k** evaluates whether the reasoning path of a model well perceives the problem’s conditions. This enables a more precise diagnosis of reasoning failures, revealing when the model’s deviation stems from rigidity rather than misunderstanding the given conditions. More formally,

$$\mathbf{p-pass@1} = \begin{cases} \frac{\sum_{k=1}^N \mathbb{1}[\hat{a}_k = a_k^*] \mathbf{p}_k}{\sum_{i=1}^N \mathbf{p}_i}, & \text{if } \sum_{i=1}^N \mathbf{p}_i > 0, \\ 0, & \text{if } \sum_{i=1}^N \mathbf{p}_i = 0 \end{cases} \quad (2)$$

where  $N$  is the number of samples,  $\hat{a}_k$  is the model’s predicted answer,  $a_k^*$  is the ground truth answer, and  $\mathbf{p}_k \in \{0, 1\}$  is perception indicator where  $\mathbf{p}_k = 1$  if the model correctly understands the given conditions, and  $\mathbf{p}_k = 0$  otherwise.

In order to determine rather the model reasoning trajectory appears to perceive the user instruction, we employ an LLM to judge rather the conditions in the question and ground truth solution is reflected in the reasoning process, even when only the subset of the reasoning includes the groundings. From the observation that the question perception is readily finished in the early phase of reasoning process, we input the first 15 paragraphs of reasoning to compare with the ground truth solution and question. This benefits the accurate measurement of perception since overly lengthy reasoning process make gpt-based evaluation prone to misjudging that the original solution is not included. The full judgment prompt is provided in the Appendix B.

Using these two metrics, we observe two consistent patterns across reasoning models. As shown in Figure 3(a)–(c), the accuracy (**p-pass@1**) appears largely unaffected by contamination ratios below approximately 40%. In Figure 3(d), we record the average contamination ratio across specific intervals of the reasoning steps. Interestingly, base models without long chain-of-thought (CoT) capabilities do not show a consistent pattern of contamination dominating the reasoning process. In contrast, more advanced reasoning models tend to exhibit increasingly severe contamination as the reasoning process becomes longer and more elaborate.

## 4.2 Signals for Contamination in Realistic Situation

In realistic use cases where only the question is given, it is impossible to automatically detect if the generated reasoning is being contaminated by unwanted but familiar patterns. Therefore, we devise a simple yet effective method to detect such suspicious pattern from the patterns when contamination happens. The provided taxonomy of reasoning contamination, illustrate in Figure 4, is applicable for a robust mitigation strategies.

**Table 2: Comparison of Base vs. Reasoning Models on ConditionedMath.**

Model Name	Type	AIME			MATH500		
		p-pass@1	pass@1	p-score	p-pass@1	pass@1	p-score
Qwen2.5-32B-Instruct	Base	<b>59.12±7.81</b>	<b>45.77±7.22</b>	75.55±5.01	<b>55.95±6.02</b>	<b>40.88±5.74</b>	70.37±4.39
+ QwQ-32B	Reason	49.21±6.79	42.46±6.63	<b>81.80±4.27</b>	47.64±5.94	34.75±5.74	<b>71.37±4.59</b>
Qwen3-32B No think	Base	<b>45.14±6.97</b>	<b>43.38±7.03</b>	<b>90.81±2.66</b>	<b>50.51±5.52</b>	<b>47.13±5.30</b>	<b>85.88±2.90</b>
+ Qwen3-32B Think	Reason	33.25±6.58	29.60±6.32	76.84±4.91	34.60±5.60	30.63±5.59	75.50±3.74
Qwen3-235B No think	Base	<b>46.17±7.36</b>	<b>42.65±7.29</b>	<b>86.40±3.08</b>	<b>55.49±5.66</b>	<b>53.50±5.62</b>	<b>84.25±2.72</b>
+ Qwen3-235B Think	Reason	24.10±5.32	20.77±5.07	81.62±4.12	27.49±4.65	23.25±4.63	79.13±3.39
DeepSeek V3	Base	<b>54.72±7.92</b>	<b>45.59±7.65</b>	77.94±5.46	<b>60.67±6.34</b>	<b>47.00±6.05</b>	<b>75.00±4.57</b>
+ DeepSeek R1	Reason	49.09±8.21	39.71±7.76	<b>80.88±5.07</b>	48.63±6.44	38.00±6.40	73.00±5.09
GPT-4o	Base	<b>55.13±7.22</b>	<b>47.06±7.06</b>	82.35±3.54	46.33±5.32	35.50±4.89	69.87±3.93
ChatGPT-4o	Base	38.26±7.12	33.82±6.99	<b>84.56±4.35</b>	42.94±6.26	<b>38.00±3.26</b>	<b>81.50±3.26</b>
+ o3-mini	Reason	36.90±6.88	22.79±5.72	61.76±6.35	<b>49.63±6.14</b>	<b>38.00±5.81</b>	67.50±5.40
+ o4-mini	Reason	31.25±6.59	19.12±5.49	58.82±6.75	39.06±5.76	26.50±5.17	64.00±5.06
Gemini2.5 Flash No think	Base	<b>58.64±7.17</b>	<b>52.21±7.17</b>	84.01±4.77	<b>56.61±5.63</b>	<b>49.80±5.59</b>	<b>83.53±3.41</b>
+ Gemini2.5 Flash Think	Reason	50.45±7.52	46.12±7.33	<b>89.81±2.52</b>	56.41±6.36	47.95±6.27	82.51±3.47
Claude 3.7 Sonnet No think	Base	<b>57.80±7.86</b>	<b>50.74±7.65</b>	<b>80.15±4.94</b>	<b>41.52±5.79</b>	<b>36.00±5.49</b>	<b>85.50±2.95</b>
+ Claude 3.7 Sonnet Think	Reason	57.00±8.00	46.72±7.63	72.99±6.01	40.38±5.75	32.00±5.58	78.00±4.44

**Table 3: Comparison of Base vs. Reasoning Models on PuzzleTrivial.**

Model Name	Type	p-pass@1	pass@1	p-score
Qwen2.5-32B-Instruct	Base	<b>40.90±3.98</b>	30.23±3.51	72.97±3.01
+ QwQ-32B	Reason	39.12±4.40	<b>38.36±4.38</b>	<b>97.66±0.48</b>
Qwen3-32B No think	Base	<b>74.30±3.33</b>	<b>67.66±3.53</b>	84.21±2.07
+ Qwen3-32B Think	Reason	38.28±3.47	37.19±3.40	<b>96.33±0.64</b>
Qwen3-235B No think	Base	<b>74.16±3.43</b>	<b>64.53±3.72</b>	86.17±2.66
+ Qwen3-235B Think	Reason	38.49±4.04	37.97±4.05	<b>97.42±0.56</b>
DeepSeek V3	Base	<b>66.21±3.83</b>	<b>53.98±3.82</b>	80.00±3.45
+ DeepSeek R1	Reason	51.73±4.33	50.55±4.33	<b>97.27±0.97</b>
GPT-4o	Base	<b>64.07±4.60</b>	48.38±4.53	75.23±3.63
ChatGPT-4o	Base	63.63±3.74	<b>58.59±3.63</b>	<b>89.14±2.18</b>
+ o3-mini	Reason	59.25±4.93	39.22±4.49	62.50±3.66
+ o4-mini	Reason	56.38±4.84	29.53±4.18	39.77±4.25
Gemini2.5 Flash No think	Base	<b>70.09±4.21</b>	<b>65.94±4.27</b>	<b>94.06±1.79</b>
+ Gemini2.5 Flash Think	Reason	69.44±4.32	65.63±4.34	<b>94.06±1.95</b>
Claude 3.7 Sonnet No think	Base	<b>79.97±3.85</b>	<b>73.28±4.03</b>	<b>89.30±2.05</b>
+ Claude 3.7 Sonnet Think	Reason	65.88±4.63	52.81±4.58	79.69±3.50

**Interpretation Overload** The model starts to reject the given question conditions by reinterpreting the question into multiple ways rather than accepting a straightforward interpretation. It is also observed that the model tends to drift between different semantic interpretations mid-reasoning, causing inconsistent or contradictory conclusions.

**Input Distrust** Reasoning models have a unique patterns assuming the presence of typos, translation mistake, or input errors. This leads to the dismissal of the conditions stated in the question and make the reasoning process overly complicated even in the straightforward cases.

**Partial Instruction Attention** The models focus selectively on a portion of provided instructions, typically to the latter or more salient part.

## 5 Experiments

**Experimental Details** The experiments are conducted on three variants from our diagnostic set ReasoningTrap, which consists of ConditionedMath (AIME, MATH500), and PuzzleTrivial. In Table 2 and Table 3, we report the **p-pass@1** scores across various models, including Qwen2.5-32B-Instruct (Yang et al., 2024), QwQ-32B (Team, 2025c), Qwen3-32B (Team, 2025b), Qwen3-235B, DeepSeek V3 (671B) (DeepSeek-AI, 2024), DeepSeek R1 (671B) (DeepSeek-AI, 2025), and proprietary models ChatGPT-4o, GPT-4o, o3-mini, o4-mini (OpenAI, 2024), Google gemini2.5-flash (Google DeepMind, 2025b) and Claude 3.7



Table 4: Budget Forcing and Prompt Hinting on ReasoningTrap.

(a) ConditionedMath AIME vs. Original AIME

		ConditionedMath AIME			Original AIME		
		p-pass@1	pass@1	p-score	p-pass@1	pass@1	p-score
<b>Qwen3-32B</b>		33.25±6.58	29.60±6.32	76.84±4.91	75.42±6.88	72.79±6.95	86.76±3.38
Budget Force	+ low	<b>53.66±7.63</b>	<b>51.47±7.46</b>	<b>90.44±2.80</b>	31.09±5.98	28.68±5.98	87.50±3.38
	+ medium	44.07±6.94	39.71±6.69	86.76±3.38	52.21±8.00	50.00±7.76	83.09±4.57
	+ high	39.82±6.97	36.03±6.94	83.09±4.44	<b>57.60±7.31</b>	<b>57.35±7.35</b>	<b>91.91±2.29</b>
Prompt Hint	+ Hint 1	<b>45.30±8.08</b>	<b>42.65±8.14</b>	<b>86.03±4.11</b>	<b>81.36±6.59</b>	<b>75.74±6.55</b>	86.76±3.98
	+ Hint 2	38.24±7.42	37.50±7.48	75.00±4.95	76.27±6.17	73.53±6.15	86.76±3.54
	+ Hint 3	41.23±7.51	36.03±7.17	83.82±4.71	74.19±6.82	69.85±6.82	<b>91.18±3.79</b>

(b) ConditionedMath MATH500 vs. Original MATH500

		ConditionedMath MATH500			Original MATH500		
		p-pass@1	pass@1	p-score	p-pass@1	pass@1	p-score
<b>Qwen3-32B</b>		34.60±5.60	30.63±5.59	75.50±3.74	87.98±4.70	85.50±4.69	91.50±2.21
<b>Budget Force</b>	+ low	<b>51.32±6.35</b>	<b>42.00±5.91</b>	76.00±4.46	68.68±5.51	68.00±5.39	91.00±2.34
	+ medium	43.75±5.99	36.00±5.90	<b>80.00±3.91</b>	80.33±5.28	76.50±5.32	<b>91.50±2.33</b>
	+ high	40.79±6.25	34.00±5.92	76.00±3.97	<b>82.51±5.28</b>	<b>81.00±5.13</b>	<b>91.50±1.97</b>
<b>Prompt Hint</b>	+ Hint 1	<b>46.88±6.51</b>	<b>40.50±6.46</b>	<b>80.00±4.16</b>	88.76±4.18	85.50±4.41	89.00±2.49
	+ Hint 2	42.11±6.37	37.00±6.20	76.00±4.10	88.46±4.64	85.00±4.63	<b>91.00±2.55</b>
	+ Hint 3	37.75±6.00	32.00±5.85	75.50±4.31	<b>90.06±4.16</b>	<b>87.00±4.24</b>	90.50±2.56

sonnet (Claude, 2025). These models are grouped into seven pairs, each consisting of a base model and its corresponding reasoning-aligned variant trained for long-form reasoning.

The experiments are conducted with Chain-of-Thought prompting, by wrapping the given question with ‘Please reason step by step, and put your final answer within  $\boxed{\phantom{000}}$ ’. Sampling was performed 16 times per question for the main experiments in Table 2 and Table 3, and 4 times per question for the other experiments.

**Evaluation Details** For math problems, correctness was determined via exact matching after a cleaning step that removes unwanted parts such as measurement units. For puzzle problems, where answers are often in free-form sentences, an LLM was used to assess the correctness by comparing the model’s output against the ground truth answer.

### 5.1 Observations on Various Reasoning Models

In most configurations, the reasoning models under-perform compared to their base model counterparts, contrary to expectations, given the typical capability gap favoring larger or instruction-tuned models. On both ConditionedMath and PuzzleTrivial, base models achieve significantly higher **p-pass@1** scores. This suggests that, once the model correctly interprets the question, base models tend to adhere more rigorously to the original instruction and are more likely to reach the correct answer.

### 5.2 Ablation Study

**Budget Forcing** Following budget forcing from Team (2025b), we append the prompt ‘Considering the limited time by the user, I have to give the solution based on the thinking directly now.</think>’ to the generated response and continue output generation once the predefined token budget is reached. This enforce model to directly generate answer without further thinking. We apply low and medium token budget for each dataset and observe the g-pass@1 score. For MATH500, we use 2000, 4000, 6000 as low, medium, high budget and for AIME, we apply 2000, 6000, 10000 as low, medium, high budget, each. As shown in Table 4, even though low token budget is beneficial for our diagnostic set, it harms the performance on the original datasets. Based on this result, we confirm that strict budget forcing has inherent problems.

**Prompt Hinting** While we have carefully filtered out nonsensical or contradictory conditions that render problems unsolvable, there remains a possibility that the model might attribute unusual patterns to errors made by the user. Although such behavior is not inherently incorrect, it could undermine the intended solution process. To mitigate this, we introduced an additional prompt to the model’s response, explicitly stating that the problem contains no typographical errors and that the model must adhere to the instructions provided in the prompt. We conducted experiments on the **ConditionedMath** dataset, testing three variants of the additional prompt hints based on the 3 major pattern observed in Figure 4.

Despite providing this additional condition to focus on the given instructions, we observe that the model still continues to display similar behavior of reasoning rigidity. Specifically, it persists to relying on familiar reasoning patterns, without adapting to the new conditions introduced by the prompts. As a result, even though some of the prompt shows better performance on given dataset, some prompt harm the performance on original dataset.

## Limitation

This study identifies a clear limitation in RL-based reasoning models, reasoning rigidity, but does not provide a fundamental analysis of which specific components of the reinforcement learning framework are responsible for this phenomenon. Since reasoning rigidity is significantly more pronounced in reasoning models compared to non-reasoning models, investigating its underlying causes remains a critical direction for future work.

Another important caveat is that our diagnostic set focuses exclusively on mathematics and puzzle-solving tasks, which may introduce a domain bias. It therefore remains unclear whether similar rigidity arises in other application areas where the nature of ‘correct’ reasoning differs substantially. Extending our evaluation to these domains will be necessary to assess the generality of reasoning rigidity and to tailor domain-specific mitigation strategies.

## Conclusion

To the best of our knowledge, this work is the first to highlight the surprising rigidity exhibited by advanced reasoning models during multi-step reasoning. Despite their strong capability to comprehend both user-provided conditions and problem details, these models often fail—not due to a lack of understanding, but because they default to ingrained reasoning patterns over faithfully following user instructions. To investigate this phenomenon, we construct a high-quality, curated diagnostic dataset and propose a tailored metric designed to capture both reasoning rigidity and contamination from familiar solution trajectories.

## Acknowledgements

We thank Will Arnold for his constructive feedback and assistance on experimental design on reasoning models.

## References

- AIME. AIME 2024. [https://artofproblemsolving.com/wiki/index.php/2024\\_AIME\\_I?srsItd=AfmB0oqfUhmDQZd1-etOmNCjXpUgzyI4604aZZ8hjLFPLSGMw\\_35PqJJ](https://artofproblemsolving.com/wiki/index.php/2024_AIME_I?srsItd=AfmB0oqfUhmDQZd1-etOmNCjXpUgzyI4604aZZ8hjLFPLSGMw_35PqJJ). Accessed: 2025-05. 5
- Saeid Alavi Naeini, Raeid Saqur, Mozghan Saeidi, John Giorgi, and Babak Taati. Large language models are fixated by red herrings: Exploring creative problem solving and einstellung effect using the only connect wall dataset. *Advances in Neural Information Processing Systems*, 36: 5631–5652, 2023. 3
- Roberto Araya. Do chains-of-thoughts of large language models suffer from hallucinations, cognitive biases, or phobias in bayesian reasoning? *arXiv preprint arXiv:2503.15268*, 2025. 3
- Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*, 2023. 3
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1
- Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, et al. Rm-r1: Reward modeling as reasoning. *arXiv preprint arXiv:2505.02387*, 2025. 24
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. 1
- Claude. Claude 3.5 Sonnet. <https://www.anthropic.com/news/claude-3-5-sonnet>, 2024. Accessed: 2025-05. 1
- Claude. Claude 3.7 Sonnet. <https://www.anthropic.com/news/claude-3-7-sonnet>, 2025. Accessed: 2025-05. 9
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 1
- Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025. 24
- DeepSeek-AI. Deepseek-v3 technical report, 2024. 8
- DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. 8
- Hugging Face. Open r1: A fully open reproduction of deepseek-r1, 2025. 23
- Google DeepMind. Gemini 2.5 Pro. <https://deepmind.google/technologies/gemini/pro/>, 2025a. Accessed: 2025-05. 1
- Google DeepMind. Gemini 2.5 flash: Faster, lower-cost reasoning. <https://blog.google/technology/google-deepmind/google-gemini-updates-io-2025/>, 2025b. Blog post, accessed 22 May 2025. 8
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 1, 3
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 1, 5
- Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model, 2025. 23, 24

- Shima Imani, Liang Du, and Harsh Shrivastava. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*, 2023. 3
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. 1, 3
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024. 1
- Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moon-tae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *arXiv preprint arXiv:2504.16828*, 2025. 23
- Jonathan Kim, Anna Podlasek, Kie Shidara, Feng Liu, Ahmed Alaa, and Danilo Bernardo. Limitations of large language models in clinical problem-solving arising from inflexible reasoning. *arXiv preprint arXiv:2502.04381*, 2025. 3
- Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*, 2020. 1
- Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025. Notion Blog. 23
- Kyle Moore, Jesse Roberts, Thao Pham, and Douglas Fisher. Reasoning beyond bias: A study on counterfactual prompting and chain of thought reasoning. *arXiv preprint arXiv:2408.08651*, 2024. 4
- OpenAI. Gpt-4o system card. <https://openai.com/index/gpt-4o-system-card/>, 2024. Accessed 22 May 2025. 8
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1
- Syed Rifat Raiyan, Md Nafis Faiyaz, Shah Md Jawad Kabir, Mohsinul Kabir, Hasan Mahmud, and Md Kamrul Hasan. Math word problem solving by generating linguistic variants of problem statements. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 362–378, 2023. 3
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 24
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 24
- Maohao Shen, Guangtao Zeng, Zhenting Qi, Zhang-Wei Hong, Zhenfang Chen, Wei Lu, Gregory Wornell, Subhro Das, David Cox, and Chuang Gan. Satori: Reinforcement learning with chain-of-action-thought enhances llm reasoning via autoregressive search, 2025. 24
- Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L Hamilton. Clutrr: A diagnostic benchmark for inductive reasoning from text. *arXiv preprint arXiv:1908.06177*, 2019. 1
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1
- Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025. 1, 3
- NovaSky Team. Sky-t1: Fully open-source reasoning model with o1-preview performance in 450 budget. <https://novasky-ai.github.io/posts/sky-t1>, 2025a. Accessed: 2025-05-23. 23
- Qwen Team. Qwen3, 2025b. 3, 8, 9

- Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, 2025c. [1](#), [3](#), [8](#)
- RUCAIBox STILL Team. Still-3-1.5b-preview: Enhancing slow thinking abilities of small models through reinforcement learning. 2025d. [23](#)
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. [3](#)
- Vellum AI. Reasoning models are indecisive parrots, 2025. Accessed: 2025-05-11. [5](#)
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. [3](#)
- Sean Williams and James Huckle. Easy problems that llms get wrong. *arXiv preprint arXiv:2405.19616*, 2024. [5](#)
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. [8](#), [23](#)
- Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. Reclor: A reading comprehension dataset requiring logical reasoning. *arXiv preprint arXiv:2002.04326*, 2020. [1](#)
- Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025. [3](#)
- Yuxiang Zhang, Shangxi Wu, Yuqi Yang, Jiangming Shu, Jinlin Xiao, Chao Kong, and Jitao Sang. o1-coder: an o1 replication for coding. *arXiv preprint arXiv:2412.00154*, 2024. [1](#)
- Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data, 2025. [24](#)

# Appendix

## A Dataset Construction Details

As shown in Figure 2, **ConditionedMath** construction pipeline consists of two stages. We provide the detailed prompt provided to gpt-4o-mini and o3-mini in the construction phase.

User

**[Instruction]:** Given the original question, generate **5** different modified question's that are completely unusual conditions, each producing a different solution process and different answer from the original.

Please double check to make sure newly generated 'modified question' has following properties:

- should be a valid question.
- should be different from the original question. But, mere change of constant or variable is not allowed.
- should be solvable without error.

**[Output Format]**

modifications:

- modified reason: ... (in LaTeX)
- modified question: ... (in LaTeX)
- modified reason: ... (in LaTeX)
- modified question: ... (in LaTeX)
- ... (total 5 entries)

**[Example 1]:**

1. original question: Get largest integer smaller than  $(\sqrt{7} + \sqrt{5})^6$
2. original solution: Expand  $(\sqrt{7} + \sqrt{5})^6$  via the binomial theorem, compute each term exactly, then subtract 1 to find the greatest integer less than the sum.
3. modification reason: Rounding each square root term down before exponentiation transforms all inner terms into integers, making the final calculation trivial.
4. modified question: Get largest integer smaller than  $(\sqrt{7} + \sqrt{5})^6$ . Added constraint: Square root terms are rounded down to the nearest integer before exponentiation. Do not use calculator.

**[Example 2]:**



1. original question: Determine  $w^2 + x^2 + y^2 + z^2$  if

$$\frac{x^2}{2^2 - 1} + \frac{y^2}{2^2 - 3^2} + \frac{z^2}{2^2 - 5^2} + \frac{w^2}{2^2 - 7^2} = 1$$

$$\frac{x^2}{4^2 - 1} + \frac{y^2}{4^2 - 3^2} + \frac{z^2}{4^2 - 5^2} + \frac{w^2}{4^2 - 7^2} = 1$$

$$\frac{x^2}{6^2 - 1} + \frac{y^2}{6^2 - 3^2} + \frac{z^2}{6^2 - 5^2} + \frac{w^2}{6^2 - 7^2} = 1$$

$$\frac{x^2}{8^2 - 1} + \frac{y^2}{8^2 - 3^2} + \frac{z^2}{8^2 - 5^2} + \frac{w^2}{8^2 - 7^2} = 1$$

2. original solution: Solve the  $4 \times 4$  linear system in variables  $x^2, y^2, z^2, w^2$  by expressing it in matrix form and inverting or using elimination to find each squared term, then sum them.
3. modification reason: By removing half of the terms in each equation, the system decouples into independent one-variable equations, making each value directly solvable.
4. modification question: Determine  $w^2 + x^2 + y^2 + z^2$  if

$$\frac{x^2}{2^2 - 1} + \frac{y^2}{2^2 - 3^2} + \frac{z^2}{2^2 - 5^2} + \frac{w^2}{2^2 - 7^2} = 1$$

$$\frac{x^2}{4^2 - 1} + \frac{y^2}{4^2 - 3^2} + \frac{z^2}{4^2 - 5^2} + \frac{w^2}{4^2 - 7^2} = 1$$

$$\frac{x^2}{6^2 - 1} + \frac{y^2}{6^2 - 3^2} + \frac{z^2}{6^2 - 5^2} + \frac{w^2}{6^2 - 7^2} = 1$$

$$\frac{x^2}{8^2 - 1} + \frac{y^2}{8^2 - 3^2} + \frac{z^2}{8^2 - 5^2} + \frac{w^2}{8^2 - 7^2} = 1$$

Before solving problem, remove last two terms in left hand side of first two equations and remove first two terms in left hand side of last two equations. After removing terms, solve problem and determine value.

**[Example 3]:**

1. original question: A regular 12-gon is inscribed in a circle of radius 12. The sum of the lengths of all sides and diagonals of the 12-gon can be written in the form  $a + b\sqrt{2} + c\sqrt{3} + d\sqrt{6}$ , where  $a, b$ , and  $d$  are positive integers. Find  $a + b + c + d$ .
2. original solution: Compute each chord length using  $2R\sin(\pi k/12)$  for  $k = 1, 2, \dots, 6$ , sum like terms to express in the prescribed form, then add coefficients.
3. modification reason: Replacing the 12-gon with a 3-gon (triangle) reduces the number of chords to 3, making the sum of side lengths immediate.
4. modified question: A regular 12-gon is inscribed in a circle of radius 12. The sum of the lengths of all sides and diagonals of the 12-gon can be written in the form  $a + b\sqrt{2} + c\sqrt{3} + d\sqrt{6}$ , where  $a, b$ , and  $d$  are positive integers. Find  $a + b + c + d$ . Before solving problem, change regular 12-gon into regular triangle, and solve changed problem.

**[Input]:**

- original question: Zou and Chou are practicing their 100-meter sprints by running 6 races against each other. Zou wins the first race, and after that, the probability that one of them wins a race is  $\frac{2}{3}$  if they won the previous race but only  $\frac{1}{3}$  if they lost the previous race. The probability that Zou will win exactly 5 of the 6 races is  $\frac{m}{n}$ , where  $m$  and  $n$  are relatively prime positive integers. Find  $m + n$ .
- original solution:

1. The probability that Zou loses a race is  $\frac{1}{3}$ , and the probability that Zou wins the next race is  $\frac{1}{3}$ . For each of the three other races, the probability that Zou wins is  $\frac{2}{3}$ .
2. The probability that Zou loses a race is  $\frac{1}{3}$ . For each of the four other races, the probability that Zou wins is  $\frac{2}{3}$ .
3.  $4 \left(\frac{1}{3}\right)^2 \left(\frac{2}{3}\right)^3 + \left(\frac{1}{3}\right)^1 \left(\frac{2}{3}\right)^4 = \frac{32}{243} + \frac{16}{243} = \frac{48}{243} = \frac{16}{81}$ .
4. The answer is  $16 + 81 = 97$ .

gpt-4o-mini

- modified question: Zou and Chou run a single 100-meter sprint. Zou wins the first race. The probability that a person wins a race given they won the previous one is  $\frac{2}{3}$ , and if they lost the previous one is  $\frac{1}{3}$ . What is the probability that Zou wins exactly 1 of the 1 race?
- modified solution:
  1. Since there is only one race and it is given that Zou wins the first race, he wins exactly one out of one.
  2. Therefore the probability is 1.
- modified answer: 1

User

- Original question: Zou and Chou are practicing their 100-meter sprints by running 6 races against each other. Zou wins the first race, and after that, the probability that one of them wins a race is  $\frac{2}{3}$  if they won the previous race but only  $\frac{1}{3}$  if they lost the previous race. The probability that Zou will win exactly 5 of the 6 races is  $\frac{m}{n}$ , where  $m$  and  $n$  are relatively prime positive integers. Find  $m + n$ .
- Modified question: Zou and Chou run a single 100-meter sprint. Zou wins the first race. The probability that a person wins a race given they won the previous one is  $\frac{2}{3}$ , and if they lost the previous one is  $\frac{1}{3}$ . What is the probability that Zou wins exactly 1 of the 1 race?

**[Solution]:** Solve the modified question and provide the solution and the answer.

1. concise chain-of-thought solution (your step-by-step reasoning)
2. final ground-truth answer

**[Analysis]:** Analyze the modified question with concise yes/no and one-sentence reasoning for each:

1. Mathematically valid and consistent?
2. Different solution path or different answer from the original?
3. Solvable without error?

Respond **only** with a JSON object in this form, without additional text:

...

```
{
  "solution": "...",
  "answer": "...",
  "valid": "...",
  "different": "...",
  "solvable": "..."
}
```

o3-mini

[Return]:

---

```
{
  "solution": "1. Since there is only one race and it is given that
              Zou wins the first race, he wins exactly one out of one. 2.
              Therefore the probability is 1.",
  "answer": "1",
  "valid": "Yes. The modified question is internally consistent and
           well-defined.",
  "different": "Yes. The modification reduces the scenario to a
               single race, changing both the solution path and the answer.",
  "solvable": "Yes. The result follows directly from the given
               conditions."
}
```

---

## B Perception Score Evaluation Prompt

To evaluate if the model’s reasoning is properly grounded on the provided user instruction, we measure whether the model perceives the instruction or not by comparing with the given question and ground truth solution. For this purpose, gpt-4o-mini is prompted with a simple instruction to determine whether a subset of model reasoning is similar with the ground truth. From the observation that most of the perception process is handled in the beginning of the reasoning phase, we split first 15 paragraphs of reasoning from model, then compare with the ground truth reasoning and question.

User

**[Evaluation Criteria]:** Evaluate whether a part of the model output is similar to the ground truth solution.

The ground truth solution is provided as a list of reasoning steps. Even if the model output is not exactly the same as the ground truth, it should be considered correct if a subset of the model output contains reasoning steps that are similar to any of the ground truth steps.

- The question is ...
- The ground truth solution is ...
- The model output is ...

**[Output Format]:** Answer in true or false.

gpt-4o-mini

*true or false*

## C Dataset Samples

We present several examples of **ConditionedMath** and **PuzzleTrivial** in Figure 5, Figure 6, and Figure 7.

**Modified Question**

Let  $x, y, z$  be positive real numbers such that  $xyz = 2$ . Find the minimum value of

$$x^4 + 4y^2 + 4z^4$$

given that  $x = 1$ .

**Modified Solution**

1. Since  $x = 1$  and  $xyz = 2$ , we have  $yz = 2$ , so  $y = \frac{2}{z}$ .
2. Substitute into the target:  $f(z) = 1^4 + 4y^2 + 4z^4 = 1 + 4\left(\frac{2}{z}\right)^2 + 4z^4 = 1 + \frac{16}{z^2} + 4z^4$ .
3. Compute  $f'(z) = -\frac{32}{z^3} + 16z^3$  and set to zero:  $16z^3 = \frac{32}{z^3} \Rightarrow z^6 = 2 \Rightarrow z = 2^{1/6}$ . Then  $y = \frac{2}{z} = 2^{5/6}$ .
4. The minimum is  $f(2^{1/6}) = 1 + \frac{16}{2^{1/3}} + 4 \times 2^{2/3} = 1 + 8 \times 2^{2/3} + 4 \times 2^{2/3} = 1 + 12 \times 2^{2/3}$ .

**Modified Answer**

$$1 + 12 \cdot 2^{2/3}$$

### (a) MATH500 example 1

**Modified Question**

Find all real numbers  $x$  such that

$$\frac{x^2 + 2x + 1}{x^2 - 1} = 2$$

given that  $x \neq 1$ .

**Modified Solution**

1. Since  $x \neq 1$ , we can multiply both sides by  $x^2 - 1$ :  $x^2 + 2x + 1 = 2(x^2 - 1)$ .
2. Expand and simplify:  $x^2 + 2x + 1 = 2x^2 - 2 \Rightarrow -x^2 + 2x + 3 = 0$ .
3. Multiply by -1:  $x^2 - 2x - 3 = 0$ .
4. Factor:  $(x - 3)(x + 1) = 0$ .
5. Solutions:  $x = 3$  or  $x = -1$ .

**Modified Answer**

$$x = 3 \text{ or } x = -1$$

### (b) MATH500 example 2

**Modified Question**

A circle has radius 5 and center at the origin. Find the area of the region inside the circle but outside the square with vertices at  $(\pm 3, \pm 3)$ .

**Modified Solution**

1. Area of circle:  $\pi r^2 = 25\pi$ .
2. Area of square:  $6 \times 6 = 36$ .
3. Desired area = Circle area - Square area =  $25\pi - 36$ .

**Modified Answer**

$$25\pi - 36$$

### (c) MATH500 example 3

**Figure 5:** ConditionedMath (MATH500) sample problems

**Modified Question**

A straight river that is 264 m wide flows from west to east at 14 m/min. Melanie and Sherry sit on the south bank with Melanie a distance  $D$  metres downstream from Sherry. Each heads for the point directly opposite Sherry and they arrive simultaneously. Melanie swims at 80 m/min relative to the water, while Sherry swims at 60 m/min. Find  $D$ .

**Modified Solution**

- Sherry. To land at  $x = 0$  we need horizontal speed  $-60 \sin \theta_S + 14 = 0 \implies \sin \theta_S = \frac{7}{30}$ . Hence  

$$v_{y,S} = 60 \cos \theta_S = 60 \sqrt{1 - \frac{49}{900}} = 2\sqrt{851}. \text{ Time: } t = \frac{264}{2\sqrt{851}} = \frac{132}{\sqrt{851}}.$$
- Melanie. Vertical motion:  $80 \cos \theta_M t = 264 \implies \cos \theta_M = \frac{\sqrt{851}}{40}$ , so  $\sin \theta_M = \frac{\sqrt{749}}{40}$ . Horizontal displacement:  
 $(80 \sin \theta_M - 14)t = D$ , hence

$$D = \left(80 \cdot \frac{\sqrt{749}}{40} - 14\right) \left(\frac{132}{\sqrt{851}}\right) = \frac{264(\sqrt{749} - 7)}{\sqrt{851}} \approx 184.4 \text{ m.}$$

**Modified Answer**

$$\frac{264(\sqrt{749} - 7)}{\sqrt{851}}$$

$$\approx 184.4 \text{ m}$$

**(a) AIME example 1****Modified Question**

Let  $a, b, c, d, e, f, g, h, i$  be distinct integers from 1 to 9, and suppose that  $a, b, c, d, e, f$  are consecutive integers. Find the minimum possible positive value of

$$\frac{a \cdot b \cdot c - d \cdot e \cdot f}{g \cdot h \cdot i}$$

**Modified Solution**

- There are four possible 6-term consecutive sets  $S = \{k, \dots, k+5\}$  with  $k = 1 \dots 4$ .
- For each, enumerate all ways to split into two triples  $A$  and  $B$ , compute products  $P(A)$  and  $P(B) = \text{Total}/P(A)$ .
- The minimal positive difference  $P(A) - P(B)$  occurs for  $S = \{2, \dots, 7\}$  with  $A = \{3, 4, 6\}$  giving  $P(A) = 72$ ,  $P(B) = 70$  and numerator 2.
- The remaining numbers are  $\{1, 8, 9\}$  so the denominator is 72, yielding  $\frac{2}{72} = \frac{1}{36}$ .

**Modified Answer**

$$\frac{1}{36}$$

**(b) AIME example 2****Modified Question**

Let  $a, b, x, y$  be real numbers with  $a > 4$  and  $b > 1$  such that

$$\frac{x^2}{a^2} + \frac{y^2}{a^2 - 16} = 1, \quad \text{and} \quad (x, y) = (20, 11).$$

Find the least possible value of  $a + b$  given these conditions.

**Modified Solution**

- Plugging  $(x, y) = (20, 11)$  into  $\frac{x^2}{a^2} + \frac{y^2}{a^2 - 16} = 1$  gives  $\frac{400}{a^2} + \frac{121}{a^2 - 16} = 1$ .
- Multiply through:  $400(a^2 - 16) + 121a^2 = a^2(a^2 - 16) \Rightarrow a^4 - 537a^2 + 6400 = 0$ .
- Let  $t = a^2$ , so  $t = \frac{537 \pm \sqrt{262769}}{2}$ .
- The solution  $> 16$  is  $t = \frac{537 + \sqrt{262769}}{2}$ , hence  $a = \sqrt{\left(\frac{537 + \sqrt{262769}}{2}\right)}$ .
- Since  $b > 1$  is free, the infimum of  $a + b$  is attained as  $b \rightarrow 1^+$ , giving  

$$\inf(a + b) = 1 + \sqrt{\left(\frac{537 + \sqrt{262769}}{2}\right)} \approx 23.912 \dots$$

**Modified Answer**

$$1 + \sqrt{\frac{537 + \sqrt{262769}}{2}}$$

$$\approx 23.9$$

**(c) AIME example 3**

**Figure 6:** ConditionedMath (AIME) sample problems



**Modified Question**

An underground parking garage with an infinite number of parking spaces is partially full. A new car arrives—what can the attendant do to make space?

**Modified Solution**

1. The garage is only partially full, meaning there are empty spaces available.
2. Since there are empty spaces, the attendant can simply direct the new car to any available empty space.
3. No need for complex rearrangement since the garage is not full.

**Modified Answer**

Move the new car to any empty parking space.

**(a) PuzzleTrivial example 1**

**Modified Question**

There are 12 coins, and one is rumored to be heavier than the rest. You have a perfect balance scale to determine which coin it is. However, all coins are actually identical. Which coin is heavier?

**Modified Solution**

1. The question states that all coins are actually identical.
2. Despite the rumor about one coin being heavier, the reality is that all coins weigh the same.
3. No matter how many times you use the balance scale, it will always show equal weights.

**Modified Answer**

None—they all weigh the same.

**(b) PuzzleTrivial example 2**

**Modified Question**

In a long line of ancient scrolls, one contains the secret to immortality whose position is known. The scrolls are sorted in increasing magical power. You must find the secret scroll using the fewest inspections possible.

**Modified Solution**

1. The position of the secret scroll is already known.
2. Since we know exactly where the scroll is, we only need one inspection.
3. No need for binary search or multiple inspections since the location is given.

**Modified Answer**

1, position of the secret scroll is known.

**(c) PuzzleTrivial example 3**

**Modified Question**

How do you measure exactly 3 gallons of water with only a 3-gallon, 5-gallon, 4-gallon, and 2-gallon jug if you are a clever gardener?

**Modified Solution**

1. You have a 3-gallon jug available.
2. Since you need exactly 3 gallons, and you have a 3-gallon jug, the solution is straightforward.
3. Simply fill up the 3-gallon jug to get exactly 3 gallons.

**Modified Answer**

Fill up the 3-gallon jug

**(d) PuzzleTrivial example 4**

**Figure 7: PuzzleTrivial sample problems**

## D Discussions

### D.1 Relationship Between Output Token Length and Accuracy

Using the *reasoning effort* parameter of o4-mini, we demonstrate that just using small amount of tokens for reasoning do not lead to performance gain in our dataset, **ReasoningTrap**. Although o4-mini underperforms compared to the base model, increasing its reasoning effort consistently yields better results. This proves that our curated diagnostic set require complex reasoning in most cases, and simply choosing short reasoning leads to performance drop.

**Table 5: Reasoning effort and Performance on ReasoningTrap p-pass@1, pass@1, and perception score on ConditionedMath.**

(a) ConditionedMath (AIME)				
Model	Reasoning Effort	p-pass@1	pass@1	p-score
o4-mini	+ low	31.25±6.59	19.12±5.49	58.82±6.75
	+ medium	<b>41.98±7.10</b>	<b>25.00±6.06</b>	59.56±6.84
	+ high	36.90±6.45	22.79±5.91	<b>61.76±6.78</b>
(b) ConditionedMath (MATH500)				
Model	Reasoning Effort	p-pass@1	pass@1	p-score
o4-mini	+ low	39.06±5.76	26.50±5.17	64.00±5.06
	+ medium	51.80±6.32	37.50±6.28	69.50±5.55
	+ high	<b>53.47±6.34</b>	<b>38.50±6.11</b>	<b>72.00±5.42</b>

## D.2 Model Size and Accuracy

We compare non-distilled reasoning models by comparing reasoning models that are directly trained from Qwen2.5 1B, 3B, 7B, and 14B (Yang et al., 2024). Since Qwen3 0.7B, 1.7B, 3B, 8B models are distilled models from the largest dense reasoning model Qwen3-32B, this is out of scope for our experimental purpose. We evaluate DeepScaleR 1.5B (Luo et al., 2025), STILL-3-1.5B-preview (Team, 2025d), OpenR1-Qwen-7B (Face, 2025), ThinkPRM-14B (Khalifa et al., 2025), Sky-T1-32B-Preview (Team, 2025a), OpenReasoner-Zero-32B (Hu et al., 2025). We use instruction-tuned model for evaluating base model’s performance.

On ConditionedMath AIME and MATH500, the base model Qwen2.5 Instruct outperforms its counterparts that have been fine-tuned for extended mathematical reasoning. Except for the smallest variant, Qwen2.5 Instruct 1.5B, the base model achieves the highest **p-pass@1** score among all evaluated models. Interestingly, although the fine-tuned reasoning models consistently record higher perception scores—reflecting a stronger understanding of each question’s conditions and the derivation of optimal solutions—their final accuracy suffers as a result of reasoning rigidity.

**Table 6: Model Size and Performance** **p-pass@1**, **pass@1**, and perception score on ConditionedMath.

(a) ConditionedMath (AIME)			
Base + Reasoning Model	<b>p-pass@1</b>	<b>pass@1</b>	<b>p-score</b>
<b>Qwen2.5-1.5B</b>	39.94±5.65	24.63±4.04	56.62±4.89
+ DeepScaleR 1.5B	38.29±6.24	33.82±6.18	<b>81.62±4.34</b>
+ STILL-3-1.5B-preview	<b>41.53±5.80</b>	<b>37.50±5.43</b>	81.43±4.23
<b>Qwen2.5-7B</b>	<b>62.96±8.10</b>	<b>51.47±7.53</b>	<b>79.41±4.89</b>
+ OpenR1-Qwen7B	49.53±7.33	47.06±6.57	78.68±5.98
<b>Qwen2.5-14B</b>	<b>58.43±7.58</b>	<b>48.53±7.24</b>	79.60±4.38
+ ThinkPRM-14B	33.33±5.92	29.04±5.88	<b>82.17±4.22</b>
<b>Qwen2.5-32B</b>	<b>59.12±7.81</b>	45.77±7.22	75.55±5.01
+ SkyT1-32B-Preview	56.57±6.71	<b>52.21±6.49</b>	<b>86.76±3.14</b>
+ OpenReasoner-Zero-32B	53.27±6.51	48.90±6.37	81.43±4.23
(b) ConditionedMath (MATH500)			
Base + Reasoning Model	<b>p-pass@1</b>	<b>pass@1</b>	<b>p-score</b>
<b>Qwen2.5-1.5B</b>	39.84±5.27	20.25±3.72	48.00±4.85
+ DeepScaleR 1.5B	<b>41.04±5.44</b>	<b>33.38±5.40</b>	<b>79.50±3.74</b>
+ STILL-3-1.5B-preview	35.21±5.11	30.75±5.03	75.62±3.48
<b>Qwen2.5-7B</b>	<b>55.56±6.14</b>	38.00±5.94	67.50±5.68
+ OpenR1-Qwen7B	45.81±6.22	<b>39.50±6.02</b>	<b>77.50±4.12</b>
<b>Qwen2.5-14B</b>	<b>61.50±5.65</b>	<b>44.12±5.54</b>	70.12±4.46
+ ThinkPRM-14B	37.44±5.22	30.38±4.97	<b>76.12±3.29</b>
<b>Qwen2.5-32B</b>	<b>55.95±6.02</b>	40.88±5.74	70.38±4.39
+ SkyT1-32B-Preview	54.80±5.67	<b>44.62±5.52</b>	76.88±3.67
+ OpenReasoner-Zero-32B	45.81±6.22	39.50±6.02	<b>77.50±4.12</b>

### D.3 RL Training Objective and Accuracy

Reasoning models are trained from base large language models by various strategies, including GRPO (Shao et al., 2024), PPO (Schulman et al., 2017), or even zero-data regime (Zhao et al., 2025).

Open-Reasoner-Zero (Hu et al., 2025) is fine-tuned from the Qwen2.5-7B-Instruct model using proximal policy optimization (PPO) with a simple binary reward for answer correctness. Satori-7B (Shen et al., 2025) explicitly trains its base model to decide when to reflect on previous actions and to incorporate an external process reward. Absolute Zero Reasoner (Zhao et al., 2025) introduces a novel reward scheme in which the LLM serves both as task proposer and task solver, with outputs verifiable in code. RM-R1 (Chen et al., 2025) structures its reward to improve alignment with human preferences during intermediate reasoning steps. Eurus-PRIME (Cui et al., 2025) employs an iterative training regimen combining a policy model that generates rollouts and an implicit process-reward model that verifies them. ThinkPRM is fine-tuned from the R1-distilled Qwen14B base model (Qwen2.5-14B-Instruct) using the generative PRM objective, which evaluates the step-by-step correctness of the reasoning process.

Among all variants of reinforcement-learning objectives, the base models Qwen2.5-7B and Qwen2.5-14B achieved outstanding performance **p-pass@1** in most cases. This suggests that current RL regimes may exacerbate the ‘reasoning rigidity’ inherent in these models. Hence, further exploration of reinforcement-learning algorithms that are robust to reasoning rigidity is essential for the development of faithful and credible reasoning systems.

**Table 7: Performance Comparison on Reasoning Models Trained with Different RL Strategies.**

(a) ConditionedMath (AIME)			
Base + RL Objective	p-pass@1	pass@1	p-score
<b>Qwen2.5-7B</b>	<b>62.96±8.10</b>	<b>51.47±7.53</b>	79.41±4.89
+ Open-Reasoner-Zero	47.49±7.42	43.01±6.92	<b>84.38±4.17</b>
+ Satori-7B	53.33±3.03	4.92±3.27	5.68±3.84
+ Absolute Zero Reasoner	49.86±7.11	33.46±6.14	63.42±5.52
+ RM-R1	54.83±6.94	44.26±6.61	76.10±5.08
+ Eurus-PRIME	59.16±8.24	40.44±7.68	61.21±6.64
<b>Qwen2.5-14B</b>	<b>58.43±7.58</b>	<b>48.53±7.24</b>	79.60±4.38
+ Absolute Zero Reasoner	50.73±7.27	34.38±6.63	63.05±4.46
+ ThinkPRM	33.33±5.92	29.04±5.88	<b>82.17±4.22</b>

(b) ConditionedMath (MATH500)			
Base + RL Objective	p-pass@1	pass@1	p-score
<b>Qwen2.5-7B</b>	55.56±6.14	38.00±5.94	67.50±5.68
+ Open-Reasoner-Zero	50.93±6.16	40.50±6.06	74.12±4.39
+ Satori-7B	49.50±6.15	37.25±5.96	<b>75.00±4.59</b>
+ Absolute Zero Reasoner	37.28±5.09	22.62±4.10	56.00±4.62
+ RM-R1	36.81±4.39	26.50±3.89	68.25±3.97
+ Eurus-PRIME	<b>57.29±6.52</b>	<b>42.38±6.20</b>	72.00±4.71
<b>Qwen2.5-14B</b>	<b>61.50±5.65</b>	<b>44.12±5.54</b>	70.12±4.46
+ Absolute Zero Reasoner	44.25±5.34	26.25±4.42	57.63±4.53
+ ThinkPRM	37.44±5.22	30.38±4.97	<b>76.12±3.29</b>