REASONING MODEL IS STUBBORN: DIAGNOSING INSTRUCTION OVERRIDING IN REASONING MODELS

Anonymous authors
Paper under double-blind review

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. We categorize rigidity patterns 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 will publicly release our diagnostic set to facilitate future research on mitigating reasoning rigidity in language models.

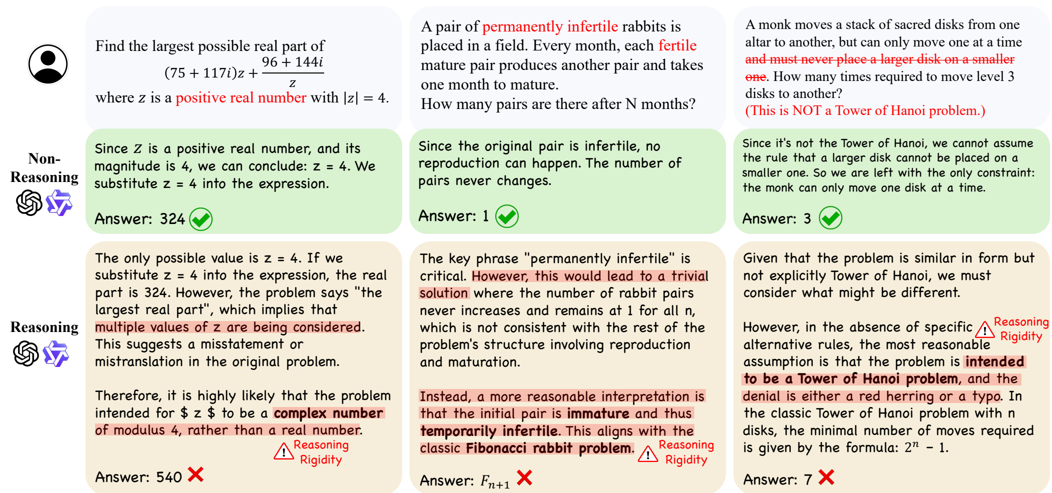
1 INTRODUCTION

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 increased test-time compute have attracted significant attention due to their capability to solve intricate reasoning problems.

However, we pinpoint a problematic behavior from reasoning models, termed *reasoning rigidity*. Crucially, unlike hallucination or a memorization problem, 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 categorized as an existing problem.

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 'real number,' advanced reasoning models capable of solving complex mathematical problems incorrectly assume z must be a 'complex number'. 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 logical reasoning that come with multiple conditions that must be fulfilled.



(a) AIME (Math) (b) Fibonacci Rabbit (c) Tower of Hanoi

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.

Through the model's reasoning rigidity that 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 ability 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 reasoning models.

Our analysis of `ReasoningTrap` yields several important findings: i) reasoning rigidity emerges from unseen training dataset indicating that it is not a simple memorization problem from data overfitting, and ii) such contamination manifests in identifiable, recurring patterns in the models' outputs. Based on these observations, we propose a budget forcing and prompt hinting to mitigate reasoning rigidity, defined from three distinct rigidity patterns: (i) Interpretation Overload, (ii) Input Distrust, and (iii) Partial Instruction Attention.

Our contributions are 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

Chain-of-Thought Faithfulness Benchmarks Benchmark for CoT faithfulness (Madsen et al., 2024; Matton et al., 2025; Chen et al., 2025b) probes whether a model's explained reasoning ac-

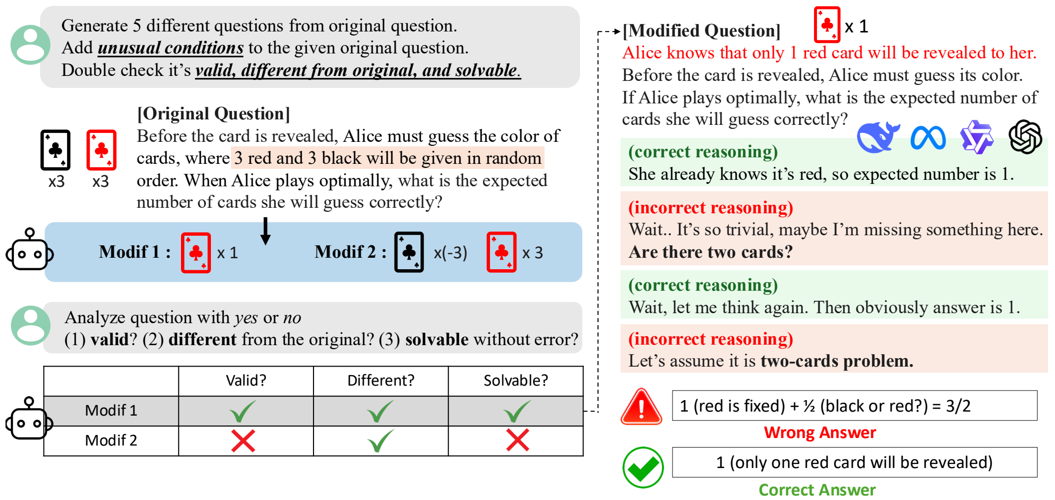


Figure 2: Dataset Construction Pipeline The dataset construction pipeline of `ConditionedMath` first creates new questions with unfamiliar conditions and verify whether the modified questions are (1) valid, (2) different, and (3) solvable. Among Modif 1 and Modif 2, the latter is omitted as it includes an invalid condition of multiplying by –3. When solving Modif 1, reasoning models over-complicate the problem, overriding the simple logic by defaulting to a familiar logic of assuming a two-card setup.

tually reflects its true decision process. These works highlight that language models often produce explanations that are not faithful to how the answer was derived. In contrast, our task evaluates under the assumption that the model might actually understand the instruction, but still ignores the instruction and defaults to a familiar reasoning trace. Prior faithfulness benchmarks detect if models' explanations omit important factors or hints but do not reveal whether a model actively disregards a required constraint or method in favor of a comfortable approach, which is the rigidity we target. Our work fills a gap by assessing whether models default to ingrained reasoning patterns despite clear user instructions, which is outside the scope of existing faithfulness benchmarks.

Instruction Following Benchmarks Existing instruction-following evaluation sets (Jiang et al., 2024; Fu et al., 2025; Zhou et al., 2023; Wen et al., 2024) assess whether LLMs or LRMs obey explicit format or content constraints in prompts (e.g. length, suffix, lexical, format constraint). These benchmarks primarily verify shallow controllability. In contrast, our dataset is not about superficial instruction adherence. Prior instruction-following tests if a model fails to include superficial information such as a requested keyword or output style, but they do not examine whether the model's reasoning trace diverges from the given question and defaults to familiar reasoning pattern.

Generalization of Reasoning Models Closely related to our work, are several previous studies that explore flexible generalization of large language models (LLMs). These works focus specifically on the ability of LLMs 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 differs from these works since the focus is on the ingrained reasoning rigidity pattern, *rather than mere inability to solve tasks creatively or to generalize to other domains*. Moreover, several works have pointed out the possibility that LLM models show rigid pattern in reasoning in specific domains such as medical domain (Kim et al., 2025) and educational domain (Araya, 2025), but have not discovered the rigidity patterns from general reasoning benchmarks using LRMs.

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 curate 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 triplets ($\boldsymbol{q}_{\text{orig}}, \boldsymbol{r}_{\text{orig}}, \boldsymbol{a}_{\text{orig}}$) and their modified counterparts ($\boldsymbol{q}_{\text{mod}}, \boldsymbol{r}_{\text{mod}}, \boldsymbol{a}_{\text{mod}}$). The modified solutions and answers diverge from the original to facilitate the assessment if the reasoning correctly follows the instructions stated in the modified question, not the original one.

In Table 1, our expert-curated diagnostic set consists of 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. `PuzzleTrivial` spans eight unique puzzle themes, therefore the dataset can be readily expanded into a much larger collection of question–answer pairs.

3.1 CONDITIONEDMATH: DIAGNOSTIC SET ON POPULAR MATH BENCHMARKS WITH ADDED 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 follows the pipeline in Figure 2: (1) original question modification, and (2) filtering based on predefined validation criteria[1]. For generating novel conditions, we provided three in-context examples that pair original problems alongside known solutions to a language model, prompting it modify the question into five distinct versions that meaningfully alter the problem's reasoning trajectory and eventually lead to a different answer.

These modified questions are 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.

Table 1: Diagnostic Dataset Configuration

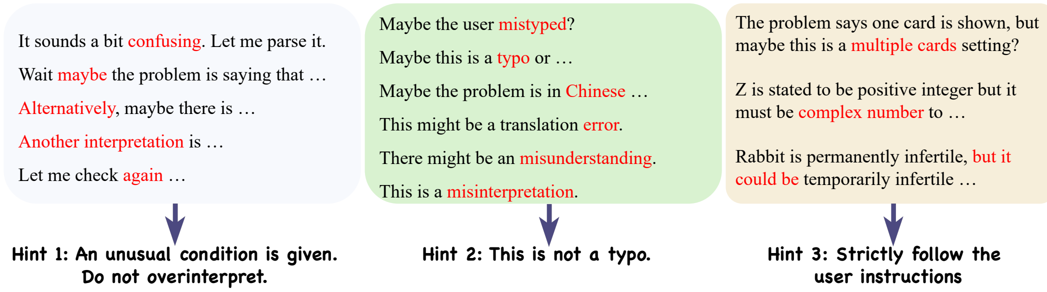
	ConditionedMath		PuzzleTrivial
	AIME (22-24)	MATH500 (lv.5)	
# Questions	34	50	80
Original Size	90	130	N/A

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 & Huckle (2024); Vellum AI (2025), we develop `PuzzleTrivial` dataset, designed to assess models' susceptibility to default to wrong but familiar complex reasoning. Classic puzzle questions are subtly altered by modifying premises or omitting specific constraints, thereby drastically simplifying the logical reasoning required. Since the alterations introduce multiple plausible answers, for the affected questions we eliminate the ambiguity by clarifying the instructions. For example, we add additional conditions such as 'find the simplest valid solution'.

[1]We use OpenAI gpt-4o-mini for stage 1 and o4-mini for stage 2, since stage 2 requires more powerful language model as a verifier.



(a) Interpretation Overload (b) Input Distrust (c) Partial Instruction Attention

Figure 4: Reasoning Pattern Analysis and Corresponding Prompt Hinting.

4 Analysis on Reasoning Rigidity Phenomenon

We observe the reasoning rigidity using our dataset `ReasoningTrap` in Section 4.1, identify the universal rigidity patterns in Section 4.2, then suggest two inference level strategies for mitigating reasoning rigidity in Section 4.3.

4.1 Reasoning Rigidity Pattern from ReasoningTrap

Upon constructing `ReasoningTrap`, we observe that advanced reasoning models often fall back on familiar but irrelevant solution patterns. In order to quantify to which extent LRMs output familiar but wrong reasonings, we prompt the model with modified questions that differ entirely from the originals (AIME, MATH500, Logic Puzzles). Then, we measure the fraction of reasoning blocks for a modified question that are closer (by cosine similarity) to the wrong but familiar original solution $\boldsymbol{s}_{\text{orig}}$ than to the correct but unfamiliar modified solution $\boldsymbol{s}_{\text{modif}}$. Formally, the contamination ratio is defined as $\mathbf{S}_{\text{contam}} = \frac{1}{p}\sum_{i=1}^{p} \mathbb{1}\left[\mathbf{cs}(\boldsymbol{s}_{\text{orig}}, \boldsymbol{r}^{i}_{\text{modif}}) > \mathbf{cs}(\boldsymbol{s}_{\text{modif}}, \boldsymbol{r}^{i}_{\text{modif}})\right]$ where $[\boldsymbol{r}^{1}_{\text{modif}}, \dots, \boldsymbol{r}^{p}_{\text{modif}}]$ are reasoning blocks of the model output[2], and $\mathbf{cs}(\cdot,\cdot)$ denotes cosine similarity.

Figure 3 shows the contamination ratio of large reasoning models on `PuzzleTrivial`. The contamination ratio $\mathbf{S}_{\text{contam}}$ gradually increases from early steps to later steps, indicating that the reasoning path is gradually contaminated by familiar reasoning path as reasoning unravels. Therefore, we may conclude that large reasoning models gradually drift towards trajectories resembling the original problems.

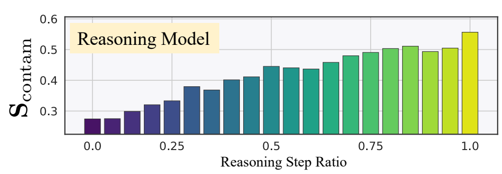


Figure 3: LRMs exhibit progressively worsening contamination as the reasoning progresses.

4.2 Signals for Reasoning Rigidity across Various Models

Across various reasoning models, we identify three universal patterns when reasoning rigidity emerges. The provided taxonomy of reasoning rigidity, illustrate in Figure 4, is stated as follows.

Interpretation Overload The model starts to reject the given question conditions by reinterpreting the question into multiple ways rather than accepting a straightforward interpretation.

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.

We analyze how often each rigidity pattern appears in model responses that fall back on familiar but incorrect reasoning in `ReasoningTrap`. Using o4-mini to detect the patterns, we count their total

[2]Paragraphs are split by double line breaks and encoded using OpenAI's text-embedding-small model.

Table 2: Distribution of Rigidity Error Patterns across Models.

Model	Interpretation Overload	Input Distrust	Partial Instruction Attention	Total
Qwen3 (32B)	59.48%	18.95%	21.57%	100%
Claude (3.7 Sonnet)	75.76%	6.06%	18.18%	100%
Gemini 2.5 Pro	61.49%	7.69%	30.82%	100%
Deepseek R1	67.35%	19.39%	13.27%	100%

occurrences for each model and report the percentage share of each pattern in Table 2. Among the three types, interpretation overload is the most common across all four reasoning models, especially in Claude 3.7 Sonnet. In contrast, Qwen3 and DeepSeek R1 more often show input distrust, while Gemini 2.5 Pro tends to display partial attention to user instructions.

4.3 Simple Mitigation Strategies

Based on the observations from Section 4.2, we employ two simple mitigation strategies that is directly integrated in inference stage, without additional training.

Budget Forcing From the observation that base models that think concisely tends not to fall into reasoning rigidity problem, we devise a simple budget forcing strategy for reasoning models. Following 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, forcing the model to produce an answer once a predefined token budget is reached.

Prompt Hinting Following the three patterns that various reasoning models universally share, we introduce an additional prompt to the model's response, explicitly stating a tailored hint for each pattern as shown in Figure 4.

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 3, we report the 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 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{}.\n\n{Question}'. We sample 16 responses per question for the main experiments reported in Table 3, and 4 responses per question for the other experiments. We use default sampling parameters designated for each model.

Evaluation Details For math problems, correctness is determined via rule-based verifier 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 is used to assess the correctness by comparing the model's output against the ground truth answer.

5.1 Model Performance over ReasoningTrap

Rigidity Patterns Emerge in Various LRMs, from Open-sourced to Proprietary Models. In most configurations, the reasoning models underperform compared to their base model counterparts, contrary to expectations. On both `ConditionedMath` and `PuzzleTrivial`, base models achieve significantly higher pass@1 scores in Table 3. This suggests that base models tend to adhere more rigorously to the original instruction and are more likely to reach the correct answer when unfamiliar patterns are given.

Table 3: Comparison of Base vs. Reasoning Models on `ConditionedMath`.

		AIME	MATH500	PuzzleTrivial
Model Name	**Type**	**pass@1**	**pass@1**	**pass@1**
Qwen2.5-32B-Instruct	Base	**45.77**±7.22	**40.88**±5.74	30.23±3.51
+ QwQ-32B	Reason	42.46±6.63	34.75±5.74	**38.36**±4.38
Qwen3-32B No think	Base	**43.38**±7.03	**47.13**±5.30	**67.66**±3.53
+ Qwen3-32B Think	Reason	29.60±6.32	30.63±5.59	37.19±3.40
Qwen3-235B No think	Base	**42.65**±7.29	**53.50**±5.62	**64.53**±3.72
+ Qwen3-235B Think	Reason	20.77±5.07	23.25±4.63	37.97±4.05
DeepSeek V3	Base	**45.59**±7.65	**47.00**±6.05	**53.98**±3.82
+ DeepSeek R1	Reason	39.71±7.76	38.00±6.40	50.55±4.33
GPT-4o	Base	**47.06**±7.06	35.50±4.89	48.38±4.53
ChatGPT-4o	Base	33.82±6.99	**38.00**±3.26	**58.59**±3.63
+ o3-mini	Reason	22.79±5.72	**38.00**±5.81	39.22±4.49
+ o4-mini	Reason	19.12±5.49	26.50±5.17	29.53±4.18
Gemini2.5 Flash No think	Base	**52.21**±7.17	**49.80**±5.59	**65.94**±4.27
+ Gemini2.5 Flash Think	Reason	46.12±7.33	47.95±6.27	65.63±4.34
Claude 3.7 Sonnet No think	Base	**50.74**±7.65	**36.00**±5.49	**73.28**±4.03
+ Claude 3.7 Sonnet Think	Reason	46.72±7.63	32.00±5.58	52.81±4.58

Table 4: Budget Forcing and Prompt Hinting on `ReasoningTrap`.

		`ConditionedMath` AIME	Original AIME	`ConditionedMath` MATH500	Original MATH500
	Model	**pass@1**	**pass@1**	**pass@1**	**pass@1**
	Qwen3-32B	29.60±6.32	72.79±6.95	30.63±5.59	85.50±4.69
Budget Force	+ low	**51.47**±7.46	28.68±5.98	**42.00**±5.91	68.00±5.39
	+ medium	39.71±6.69	50.00±7.76	36.00±5.90	76.50±5.32
	+ high	36.03±6.94	**57.35**±7.35	34.00±5.92	**81.00**±5.13
Prompt Hint	+ Hint 1	**42.65**±8.14	**75.74**±6.55	**40.50**±6.46	85.50±4.41
	+ Hint 2	37.50±7.48	73.53±6.15	37.00±6.20	85.00±4.63
	+ Hint 3	36.03±7.17	69.85±6.82	32.00±5.85	**87.00**±4.24

Effectiveness of Budget Forcing and Pattern-based Hinting. We apply token budgets for each dataset and report pass@1 scores. For MATH500, the budgets are 2k, 4k, and 6k tokens; for AIME which is more challenging, we use 2k, 6k, and 10k tokens. As shown in Table 4, a lower token budget improves performance on our diagnostic set but degrades performance on the original datasets. These results suggest that strict budget forcing has inherent limitations.

Using pattern-based hinting, we test variants of the additional prompt hints based on the three major patterns observed in Figure 4. The first hint that remedies interpretation overload shows performance improvement in both original and modified variants of AIME and MATH500. This states that when provided with appropriate instruction, reasoning models robustly solve both familiar reasoning tasks and unfamiliar variants altogether. However, as hinting to strictly follow the user instruction rather drops the pass@1 score for the original AIME, the design of instruction should be meticulously chosen considering the model and dataset type.

5.2 Analysis

Rigidity Patterns Emerges in In-domain but Unseen Training Problems We test whether Deepseek R1 and V3, which are trained and opened prior to AIME 2025, shows similar rigidity patterns in AIME 2025, to prove if LRMs show rigidity in unseen mathematical problems during train time in Table 5. Following the same pipeline of `ConditionedMath`, we modify total of 9 problems into modified version and sample 16 responses per problem. The reasoning model Deepseek R1 shows lower Pass@1 score compared to the base model Deepseek V3, indicating that rigidity patterns emerge even for problems that the model could not have encountered during

Table 5: Rigidity Patterns in Unseen Data. (a) Pass@1 comparison on Deepseek V3 (Non-reasoning) vs R1 (Reasoning) in unseen train data, AIME25. (b) Accuracy comparison on non-reasoning models (Non-LRM) and reasoning models (LRM) OOD Dataset.

(a) Unseen Train Data

Dataset	DS-V3	DS-R1
AIME25	**52.78**	41.67

(b) Non-LRM vs. LRM in OOD Domain ProofWriter

Data Type	Modified Dataset		Original Dataset	
Model Comparison	**Non-LRM**	**LRM**	**Non-LRM**	**LRM**
Qwen2.5-32B vs QwQ-32B	28%	**34%**	87%	95%
Qwen3-32B No think vs Think	33%	**33%**	97%	99%
DeepSeek V3 vs R1	30%	**35%**	99%	100%
Claude 3.7 Sonnet No think vs Think	28%	**29%**	100%	100%

Table 6: Entropy on Original vs. Modified AIME. The percentage in parenthesis indicates the increase in entropy compared to the previous iteration.

Dataset	**Base Model**	**Iter 120**	**Iter 260**
Original AIME (22–24)	0.14	1.36 (+871%)	2.47 (+81.6%)
Modified AIME (`ReasoningTrap`)	0.34	1.25 (+268%)	1.72 (+37.6%)

training. Therefore, rigidity is not merely a result of data memorization, but rather reflects a more fundamental issue in the model's reasoning behavior.

Rigidity Patterns are Less Prominent in Out-of-Domain We test whether rigidity patterns in LRMs persist in out-of-domain (OOD) settings, for example cases which are likely absent during post-training. To do this, we construct a modified dataset that deliberately departs from the original OOD dataset. If LRMs succeed on the original but fail on the trivialized modified version which is much easier, this would indicate memorization rather than generalization.

We evaluate four LRMs (o4-mini, QwQ, Gemini 2.5 Pro, Claude 3.7 Sonnet). The test uses the abductive logical reasoning dataset ProofWriter (Tafjord et al., 2021), depth level 3, under the closed-world assumption where the correct answer is always 'true.' We remove two conditions and two rules so that the logical chain becomes incomplete. The correct answer should then be either 'unknown' or 'false.'

We report accuracy on these omitted cases as *modified dataset* in Table 5 and observe that reasoning and non-reasoning models achieve similar performance. Crucially, we observe no clear rigidity patterns in LRMs, suggesting that preferred reasoning paths are absent in OOD data where models lack prior exposure.

Model Entropy Correlates with Rigidity Patterns We test the hypothesis that RL training intensifies rigidity patterns in language models, and conduct a preliminary experiment to validate this by comparing the entropy of `ReasoningTrap` (modified AIME), and the original AIME dataset.

As shown in the Table 6, the increase of entropy in the original dataset is explosive, whereas the increase in entropy in our dataset is comparably limited (note that lower entropy indicates model output rigidity). This observation supports the hypothesis that RL training differently affects the output distribution of the original mathematical dataset and our dataset.[3]

5.3 Human Evaluation on ReasoningTrap

High human preference on our dataset, `ReasoningTrap` As the dataset construction of AIME and MATH500 is automatically filtered according to three criteria, (i) validity, (ii) difference from the original, (iii) solvability of the question, we instruct human evaluators to select binary choices (0 or 1) on the validity, difference from the original, and solvability of the problem. The percentage in the table indicates the ratio of annotators that selected (valid / different / solvable) for each criterion. The high agreement rates support the quality of `ReasoningTrap`, based on 50 randomly selected samples evaluated by a total of 15 human annotators.

[3]The base model is Qwen2.5 7B base model, trained by DAPO algorithm.

Table 7: Human Evaluation on Our Dataset, `ReasoningTrap`

(a) Human Preference Evaluation

	Valid	Different	Solvable
Preference	93%	99%	95%

(b) Accuracy Comparison on LRMs and Human

Dataset	Human	o4-mini	QwQ	DeepSeek R1	Qwen3 (Think)
AIME	89.09%	10.00%	60.00%	55.00%	15.00%
MATH500	67.27%	15.00%	20.00%	0.00%	37.50%
Puzzle	83.63%	15.00%	50.00%	37.50%	21.25%

High human accuracy on our dataset, `ReasoningTrap` To check whether humans are able to understand the `ReasoningTrap` question and answer properly, we ask human participants to solve top 15 questions which LRMs scored lowest accuracy. Total of ten CS / Mathematics / EE undergraduate students are tested and they scored high accuracy in our dataset. This indicates our diagnostic set is valid in human standard, and also solvable for most of the participants. Note that 100% accuracy cannot be reached due to the difficulty of our dataset. Since `ReasoningTrap` MATH500 consists of answers that are noisy to compute without calculator, the accuracy is lower than AIME.

Human Evaluation on Contamination Ratio We introduce contamination ratio as a measure to quantify how frequently a model's reasoning trace defaults to a rigid, familiar solution. To ensure that this statistic properly finds out contamination from model outputs, we conduct a user study to evaluate the quality of contamination ratio. We test 4 human annotators with total of 24 model output and solution pairs and are instructed to select if the model output is closer to the modified (not contaminated) or to the original (contaminated) and 90.625% of the human evaluations match with the contamination ratio predictions.

6 Conclusion

To the best of our knowledge, this is the first work to reveal the surprising rigidity that advanced reasoning models exhibit during multi-step reasoning. To systematically study this phenomenon, we curate a high-quality diagnostic dataset that measures reasoning rigidity and contamination from memorized solution trajectories. Our analysis shows that rigidity arises even in problems unseen during training, confirming that it is not a simple memorization issue. Beyond diagnosis, we demonstrate that lightweight inference-time strategies can partially alleviate rigidity. However, our findings unveil that the root cause lies in the reinforcement learning–based training process itself. In particular, the train-time entropy patterns we uncover highlight the need to rethink how reasoning models are optimized. We argue that addressing this hidden rigidity is essential for building credible reasoning systems.

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.

REFERENCES

AIME. AIME 2024. https://artofproblemsolving.com/wiki/index.php/2024_AIME_I?srsltid=AfmBOoqfUhmDQZd1-etOmNCjXpUgzyI46O4aZZ8hjLFPLSGMw_35PqJJ. Accessed: 2025-05. 4

Saeid Alavi Naeini, Raeid Saqur, Mozhgan 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

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*, 2025a. 23

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, et al. Reasoning models don't always say what they think. *arXiv preprint arXiv:2505.05410*, 2025b. 2

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, June 2024. Accessed: 2025-05. 1

Claude. Claude 3.7 Sonnet. https://www.anthropic.com/news/claude-3-7-sonnet, February 2025. Accessed: 2025-05. 6

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. 23

DeepSeek-AI. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437. 6

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948. 6

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1. 22

Tingchen Fu, Jiawei Gu, Yafu Li, Xiaoye Qu, and Yu Cheng. Scaling reasoning, losing control: Evaluating instruction following in large reasoning models. *arXiv preprint arXiv:2505.14810*, 2025. URL https://arxiv.org/abs/2505.14810. 3

Google DeepMind. Gemini 2.5 Pro. https://deepmind.google/technologies/gemini/pro/, March 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. 6

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

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, 4

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. URL https://arxiv.org/abs/2503.24290. 22, 23

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

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

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. Followbench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL-2024), Long Papers*, pp. 4667–4688, 2024. URL https://aclanthology.org/2024.acl-long.257/. 3

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *arXiv preprint arXiv:2504.16828*, 2025. URL https://arxiv.org/abs/2504.16828. 22

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. 22

Andreas Madsen, Sarath Chandar, and Siva Reddy. Are self-explanations from large language models faithful? In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 295–337, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.19. URL https://aclanthology.org/2024.findings-acl.19/. 2

Katie Matton, Robert Ness, John Guttag, and Emre Kiciman. Walk the talk? measuring the faithfulness of large language model explanations. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

OpenAI. Gpt-4o system card. https://openai.com/index/gpt-4o-system-card/, 2024. Accessed 22 May 2025. 6

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)*, pp. 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. 23

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. 23

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. URL https://arxiv.org/abs/2502.02508. 23

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

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. Proofwriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3621–3634, 2021. 8

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

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. 22

Qwen Team. Qwen3, April 2025b. URL https://qwenlm.github.io/blog/qwen3/. 6

Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025c. URL https://qwenlm.github.io/blog/qwq-32b/. 1, 6

RUCAIBox STILL Team. Still-3-1.5b-preview: Enhancing slow thinking abilities of small models through reinforcement learning. 2025d. URL https://github.com/RUCAIBox/Slow_Thinking_with_LLMs. 22

Vellum AI. Reasoning models are indecisive parrots, February 2025. URL https://www.vellum.ai/reasoning-models. Accessed: 2025-05-11. 4

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, Yiming Liu, Jie Tang, Hongning Wang, and Minlie Huang. Benchmarking complex instruction-following with multiple constraints composition. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 137610–137645. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/f8c24b08b96a08ec7a7a975feea7777e-Paper-Datasets_and_Benchmarks_Track.pdf. 3

Sean Williams and James Huckle. Easy problems that llms get wrong. *arXiv preprint arXiv:2405.19616*, 2024. 4

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. 6, 22

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. Reclor: A reading comprehension dataset requiring logical reasoning. *arXiv preprint arXiv:2002.04326*, 2020. 1

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, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data, 2025. URL https://arxiv.org/abs/2505.03335. 23

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *CoRR*, 2023. 3

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 4×4 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 $k = 1, 2, \ldots, 6$, 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 r ace 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 Dataset Samples

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

```
Find the number of integer values of k
in the closed interval [-500,500]
for which the equation log(kx)=2 log(x+2)
has exactly one real solution.
```
$\boxed{\text{Assume } k \neq 0.}$

Modified condition

`Original Answer: 501` → **Modified Answer: 1**

(a) MATH500 example 1

```
Sam is carrying a 2 gallon bucket of milk to his house
from the barn, which is 3 miles  away from his house.
However, the bucket has a leak in it.  For each mile he walks,
there is $\\frac{2}{3}$ ~~as much milk in the bucket as at the
beginning of the mile.~~ [gallons in the bucket.] Modified condition
How many gallons of milk will be in the
bucket when Sam gets home?
```

`Original Answer: 16/27` → **Modified Answer: 8/27**

(b) MATH500 example 2

```
What is the maximum value of $4(x + 7)(2 - x)$,
overall real numbers $x$?
```
$\boxed{\text{Subject to } x = 2.}$

Modified condition

`Original Answer: 81` → **Modified Answer: 0**

(c) MATH500 example 3

Figure 5: `ConditionedMath` (MATH500) sample problems

Find the three-digit positive integer $\underline{abc}$ whose representation in base nine is $\boxed{\text{ten}}$

Modified condition

$\underline{bca}_{\text{nine}}$ $\boxed{\text{ten}}$ Modified condition

where a, b, c are (not necessarily distinct) digits.

Original Answer: ~~227~~ → Modified Answer: [111, 222, 333, 444, 555, 666, 777, 888, 999]

(a) AIME example 1

Define $S_n = \sum |A \cap B|$, where the sum is taken over all ordered pairs (A, B) such that A and B are subsets of $\{1, 2, 3, \ldots, n\}$ with $|A| = |B|$ $\boxed{\text{and } A = \emptyset}$ Modified condition

For example, $S_2 = 4$ because the sum is Taken over the pairs of subsets in (A, B) in

$$\{(\emptyset, \emptyset), (\{1\}, \{1\}), (\{1\}, \{2\}), (\{2\}, \{1\}), (\{2\}, \{2\}), (\{1, 2\}, \{1, 2\})\}$$

~~Let~~ $\cancel{\frac{S_{2022}}{S_{2021}} = \frac{p}{q}}$ ~~where p and q are relatively prime~~

Modified condition

$\boxed{\text{Find } S_n \text{ under this condition}}$

~~positive integers. Find the remainder when p + q is divided by 1000.~~

Original Answer: ~~235~~ → Modified Answer: 0

(b) AIME example 2

For each positive integer n, let a_n be the least positive integer multiple of 23 such that $\cancel{a_n \equiv 1 \pmod{2^n}}$ $\boxed{a_n \equiv 1 \pmod{2}}$ Modified condition

Find the number of positive integers n less than or equal to 1000 such that $a_n = a_{n+1}$

Original Answer: ~~363~~ → Modified Answer: 1000

(c) AIME example 3

Figure 6: `ConditionedMath` (AIME) sample problems

An underground parking garage with and partially
infinite number of parking spaces is ~~fully~~
occupied. A new car arrives – what can the
attendant do to make space?

Original Answer: Shift n-th to n-1th

Modified Answer: Move the car to any empty space

(a) `PuzzleTrivial` example 1

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. Using only a two-pan balance
However, all coins are actually equal
scale and no more than three weighings, how can you identify the counterfeit coin and determine whether it is heavier or lighter?

Original Answer: Weigh 1-4 vs 5-8. If they balance the heavy coin is among 9-12. Weigh 9-10 vs 11-12. In not …

Modified Answer: None

(b) `PuzzleTrivial` example 2

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. Using only a two-pan balance
However, all coins are actually equal
scale and no more than three weighings, how can you identify the counterfeit coin and determine whether it is heavier or lighter?

Original Answer: Weigh 1-4 vs 5-8. If they balance the heavy coin is among 9-12. Weigh 9-10 vs 11-12. If not …

Modified Answer: None

(c) `PuzzleTrivial` example 3

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

Original Answer: Use binary search to find the secret scroll in O(log n) inspections.

Modified Answer: position of the secret scroll is known.

(d) `PuzzleTrivial` example 4

Figure 7: `PuzzleTrivial` sample problems

C Additional Analysis on Rigidity Patterns

Rigidity Patterns do not Manifest Category Bias in Math Domain Compared to the original category distribution of MATH500 (lv.5), the distribution of our dataset do not have noticeable category biases, except for the slight difference in Algebra, Counting / Probability and Geometry categories.

Table 8: Category-wise performance on MATH500 and `ConditionedMath`.

Dataset	Inter. Algebra	Algebra	Number Theory	Precalc.	Prealgebra	Counting & Prob.	Geometry
MATH500 (lv.5)	26.9% (36)	22.4% (30)	9.0% (12)	9.0% (12)	14.2% (19)	9.0% (12)	9.7% (13)
Ours (MATH500)	34.7% (17)	14.3% (7)	8.2% (4)	10.2% (5)	6.1% (3)	4.1% (2)	22.4% (11)

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 9: Reasoning effort and Performance on `ReasoningTrap` (pass@1) on `ConditionedMath`.

(a) `ConditionedMath` (AIME)

Model	Reasoning Effort	**pass@1**
o4-mini	+ low	19.12±5.49
	+ medium	**25.00±6.06**
	+ high	22.79±5.91

(b) `ConditionedMath` (MATH500)

Model	Reasoning Effort	**pass@1**
o4-mini	+ low	26.50±5.17
	+ medium	37.50±6.28
	+ high	**38.50±6.11**

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 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 10: Model Size and Performance (pass@1) on `ConditionedMath`.

Base + Reasoning Model	**pass@1**	
	AIME	MATH500
Qwen2.5-1.5B	24.63±4.04	20.25±3.72
+ DeepScaleR 1.5B	33.82±6.18	**33.38±5.40**
+ STILL-3-1.5B-preview	**37.50±5.43**	30.75±5.03
Qwen2.5-7B	**51.47±7.53**	38.00±5.94
+ OpenR1-Qwen7B	47.06±6.57	**39.50±6.02**
Qwen2.5-14B	**48.53±7.24**	**44.12±5.54**
+ ThinkPRM-14B	29.04±5.88	30.38±4.97
Qwen2.5-32B	45.77±7.22	40.88±5.74
+ SkyT1-32B-Preview	**52.21±6.49**	**44.62±5.52**
+ OpenReasoner-Zero-32B	48.90±6.37	39.50±6.02

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., 2025a) 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 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 11: Performance Comparison on Reasoning Models Trained with Different RL Strategies (pass@1).

Base + RL Post-Train	**pass@1**	
	AIME	MATH500
Qwen2.5-7B	**51.47**±7.53	38.00±5.94
+ Open-Reasoner-Zero	43.01±6.92	40.50±6.06
+ Satori-7B	4.92±3.27	37.25±5.96
+ Absolute Zero Reasoner	33.46±6.14	22.62±4.10
+ RM-R1	44.26±6.61	26.50±3.89
+ Eurus-PRIME	40.44±7.68	**42.38**±6.20
Qwen2.5-14B	**48.53**±7.24	**44.12**±5.54
+ Absolute Zero Reasoner	34.38±6.63	26.25±4.42
+ ThinkPRM	29.04±5.88	30.38±4.97

E Usage of LLM

Our dataset construction process primarily relies on LLM usage, as stated in the main paper. We also used LLM to polish writings and to search for related works.