Reasoning Model is Stubborn: Diagnosing and Mitigating Reasoning Rigidity in Large Language Models

Anonymous ACL submission

Abstract

Large reasoning models have demonstrated remarkable proficiency in various tasks. However, we observe that 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, leading to incorrect conclusions. This behavior presents significant challenges, not only in reasoning intensive domains but also in realistic settings. To systematically investigate reasoning rigidity, a behavior unexplored in prior work, we introduce a dataset, `ReasoningTrap`, including math problems, puzzles, and agentic tasks that diagnoses reasoning rigidity. Using this dataset, we identify patterns that occur when models default to ingrained reasoning, and suggest inference level and GRPO-based post-training remedies. We will publicly release our diagnostic set to facilitate future research on mitigating reasoning rigidity in language models.

1 Introduction

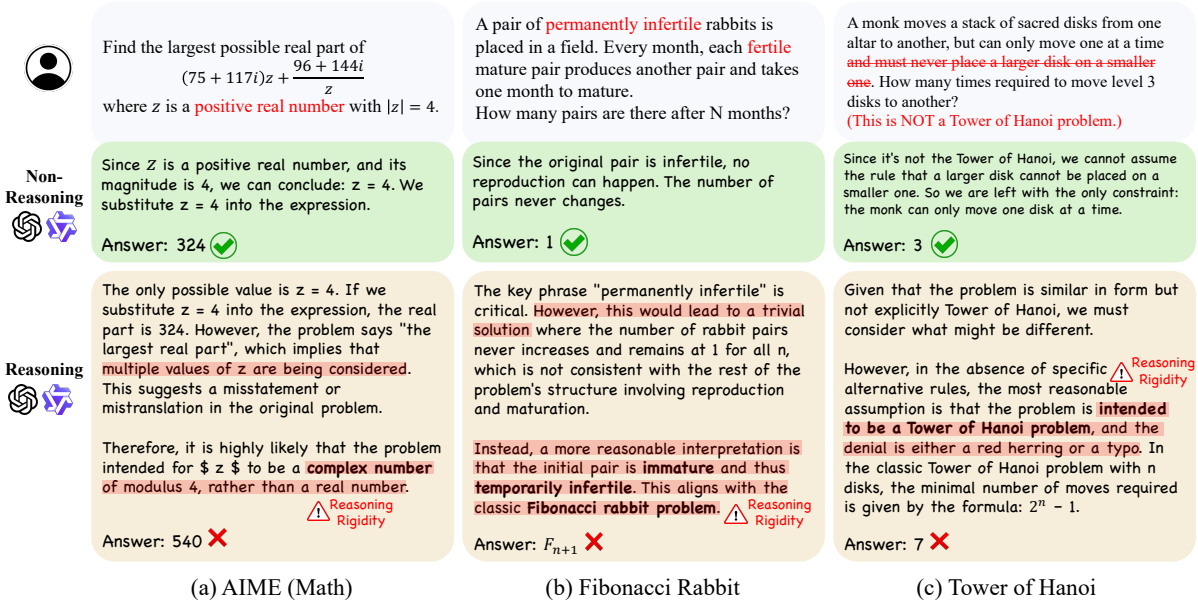
Large language models (LLMs) (Radford et al., 2019; Brown et al., 2020; Team et al., 2023; Chowdhery et al., 2023) have demonstrated remarkable proficiency in various challenging tasks, including mathematical reasoning (Cobbe et al., 2021; Hendrycks et al.), complex coding problems (Zhang et al., 2024; 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, 2025b) 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.

As illustrated in Figure 1, reasoning models frequently override explicitly stated conditions when a problem closely resembles a familiar task. For example, despite the instruction that a variable is a real number or that a puzzle is not a Tower of Hanoi variant, models often reinterpret or dismiss these constraints and revert to the original reasoning template. This rigidity causes the entire reasoning trajectory to align with the familiar solution rather than the given variant, frequently leading to incorrect or suboptimal conclusions. This rigidity is particularly problematic in domains such as mathematics and logical reasoning, where strict adherence to user-specified constraints is essential.

To systematically evaluate this phenomenon and analyze the rigidity patterns of reasoning models, we introduce `ReasoningTrap`, a diagnostic dataset comprising mathematical problems, puzzles, and shopping agent tasks intentionally designed to deviate from the original instructions. `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 original and 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 instances from train time, 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 both inference level and post-training level



(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.

strategies to alleviate reasoning rigidity.

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 propose an effective mitigation strategy in both inference and training level.

2 Related Works

We introduce reasoning rigidity as a previously uncharacterized failure mode that is distinct from the four canonical categories summarized in Table 1. We identify a unique challenge for RL-tuned reasoning models by comparing with instruction following, reasoning faithfulness, memorization, and overthinking.

Instruction Following. Existing instruction-following literature (Jiang et al., 2024; Fu et al., 2025; Zhou et al., 2023; Wen et al., 2024) assess whether LLMs or LRMs obey explicit format constraints in prompts (e.g. length, suffix, lexical, format, different approach constraint). These benchmarks primarily verify shallow controllability. In contrast, our dataset is not about superficial instruction adherence, but examine whether the model's reasoning trace defaults to familiar reasoning pattern despite correctly understanding the given question.

Reasoning Faithfulness. This failure mode describes misalignment between Chain-of-Thought (CoT) and the final output (Madsen et al., 2024; Matton et al., 2025; Chen et al., 2025b; Lyu et al., 2023). On the other hand, reasoning rigidity shows alignment between CoT and the output as it shows wrongful adherence to familiar reasoning from the CoT level.

Memorization. While memorization stems from a lack of ability in generalization, reasoning rigidity manifests as a behavioral selection error. Specifically, memorization represents a fundamental capacity failure in which the model lacks the necessary logic to solve novel tasks. Conversely, reasoning rigidity occurs when the model possesses the competence to adapt but selects not to deploy it, prioritizing familiarity over explicit contextual constraints.

Overthinking. Overthinking denotes a length-level fixation where reasoning models produce unnecessarily long or verbose chain of thought. The external behavior is similar with reasoning rigidity, as both phenomena show elongated CoT. However, the core difference is that reasoning rigidity reflects a content-level fixation, consistently defaulting to a familiar reasoning pattern, not simply a long rea-

Failure Mode	Correction Capacity	Active Denial	Familiarity Default	Primary Symptom & Difference from Ours
Instruction Following	✗	✗	✗	Inability to follow ~~format~~ content directives
Reasoning Faithfulness	✗	✗	✓	~~Mismatch~~ Match between CoT and final output
Memorization	✗	✗	✓	~~Fundamental capacity~~ Behavioral selection failure
Overthinking	✗	△	△	~~Length~~ Content-level fixation
Reasoning Rigidity	✓	✓	✓	**Active refusal of context to follow familiar reasoning.**

Table 1: **Taxonomy of Reasoning-Related Failure Modes.** Comparison of reasoning rigidity against established failure modes across three dimensions. While other modes stem from inability or lack of instruction, **reasoning rigidity** is uniquely characterized by high correction capacity coupled with active contextual denial. Symbols: ✓ = strong correlation; △ = partial; ✗ = unrelated.

soning trace. Additional details are in Appendix A.

3 Diagnostic Dataset for Reasoning Rigidity: `ReasoningTrap`

Definition. Reasoning rigidity refers to a phenomenon in which a model correctly *understands* a problem and possesses the capability to solve it, yet *fails to solve the problem because it rigidly defaults to familiar reasoning patterns*. It manifests by the models' behavior overriding or ignoring the given conditions in the problem.

Dataset Structure. We create a expert-curated diagnostic set, which consists of reasoning domains - math and logical puzzles, and realistic cases - agentic tasks. Math and logical puzzle subsets consist of original question-solution-answer triplets ($\boldsymbol{q}_{\text{orig}}, \boldsymbol{s}_{\text{orig}}, \boldsymbol{a}_{\text{orig}}$) and their modified counterparts ($\boldsymbol{q}_{\text{mod}}$, $\boldsymbol{s}_{\text{mod}}$, $\boldsymbol{a}_{\text{mod}}$). Agentic tasks do not have a ground truth reasoning trace, therefore our dataset only consists of original and modified question-answer pairs, excluding the solution. For all three domains, the modified question requires completely different answers (and solution) from the original to assess if the reasoning correctly follows the modified instruction, not defaulting to familiar original one. Please refer to the examples of original and modified triplets which are shown in Appendix C.

Coverage Domains. For mathematical reasoning, we modify ground truth question-solution-answer triplets from historical AIME 2022–2024 (AIME) and MATH500 Level 5 (Hendrycks et al.) datasets to diverge from the original, covering total of 84 questions. Logical puzzles start from classical puzzle questions, modified by changing premises or omitting specific constraints, thereby altering the logical reasoning required as inspired by Williams and Huckle (2024); Vellum AI (2025). The dataset eight unique puzzle themes, and 10 subquestions for each theme. We ensure that ambiguity do not exist by generating clarifying instructions such as 'find the simplest valid solution'. Finally, for agentic tasks, we construct from DeepPlanning (Zhang et al., 2026), shopping planning subset level 1, automatically flipping a single condition among the shopping instructions to diverge from the original (e.g. less than → more than) covering total of 50 questions.[1]

	Math		Puzzle	Agent
	AIME	MATH500		
Original	90	130	8 Themes	50
`ReasoningTrap`	34	50	80	50

Table 2: **Configuration of diagnostic datasets.** Our dataset consists of math, puzzle, and agent domains, consisting of total of 214 questions.

Among these three domains, mathematical dataset requires meticulous construction and filtering process due to guarantee the mathematical correctness. Therefore we use a two-stage pipeline of question modification and dataset filtering (Appendices B.1 and B.2), as illustrated in Figure 2. We also provide human evaluation on the quality of the dataset and the LLM-based verification procedure in Appendix F.3.

4 Observations

Rigidity in Reasoning Intensive Settings. In mathematical and logical puzzles, we observe both the model's understanding of the modified constraints throughout its reasoning trace and the ability to solve the question. Reasoning rigidity is coupled with high perception but degradation in accuracy, indicating that the failure stems from an inability to sustain correct reasoning rather than a failure to perceive the initial instruction. The perception ability is measured by comparing each of the N

[1]The condition flipping is detailed in Appendix F.5.

Table 3: **Comparison of Base vs. Reasoning Models on `ReasoningTrap` math and puzzle.** * OpenAI o3-mini and o4-mini do not open reasoning trajectory to users, therefore we are unable to measure perception score (**p-score**) for these two models. Values are reported as mean ± standard deviation.

		AIME		MATH500		`PuzzleTrivial`	
Model Name	**Type**	**pass@1**	**p-score**	**pass@1**	**p-score**	**pass@1**	**p-score**
Qwen2.5-32B-Instruct	Base	**45.77**±7.22	75.55±5.01	**51.38**±6.17	81.62±3.87	30.23±3.51	72.97±3.01
+ QwQ-32B	Reason	42.46±6.63	81.80±4.27	47.50±6.39	90.00±2.21	**38.36**±4.38	97.66±0.48
Qwen3-32B No think	Base	**40.07**±6.68	90.81±2.66	**64.00**±5.51	93.12±1.36	**65.55**±3.20	84.21±2.07
+ Qwen3-32B Think	Reason	29.60±6.32	76.84±4.91	42.62±6.15	90.50±1.52	37.19±3.40	96.33±0.64
Qwen3-235B No think	Base	**40.99**±7.04	86.40±3.08	**60.12**±5.48	92.25±1.43	**59.84**±3.38	86.17±2.66
+ Qwen3-235B Think	Reason	20.77±5.07	81.62±4.12	47.25±5.49	88.50±1.90	37.97±4.05	97.42±0.56
DeepSeek V3	Base	**48.35**±6.82	77.94±5.46	**58.75**±5.87	86.25±3.06	**64.45**±4.28	80.00±3.45
+ DeepSeek R1	Reason	39.71±7.76	80.88±5.07	35.50±5.41	85.25±2.89	50.55±4.33	97.27±0.97
EXAONE 4.0 32B No think	Base	**43.01**±6.71	84.01±3.55	**45.75**±5.82	86.50±2.67	**41.25**±3.78	81.72±2.80
+ EXAONE 4.0 32B Think	Reason	20.22±4.52	77.76±4.86	33.88±5.49	84.38±3.08	34.45±3.32	85.00±2.03
GPT-4o	Base	**47.06**±7.06	82.35±3.54	47.25±5.94	83.12±3.43	48.38±4.53	75.23±3.63
ChatGPT-4o	Base	33.82±6.99	84.56±4.35	54.62±6.24	88.88±1.93	**58.59**±3.63	89.14±2.18
+ o3-mini	Reason	22.79±5.72	.	**59.00**±6.56	.	39.22±4.49	.
+ o4-mini	Reason	19.12±5.49	.	49.50±6.01	.	29.53±4.18	.
Gemini2.5 Flash No think	Base	**52.21**±7.17	82.35±3.54	**64.88**±6.08	95.00±1.30	**65.94**±4.27	94.06±1.79
+ Gemini2.5 Flash Think	Reason	46.12±7.33	89.81±2.52	62.12±6.26	93.38±1.93	65.63±4.34	94.06±1.95
Claude 3.7 Sonnet No think	Base	**50.74**±7.65	80.15±4.94	**52.00**±5.95	94.12±1.56	**73.28**±4.03	89.30±2.05
+ Claude 3.7 Sonnet Think	Reason	46.72±7.63	72.99±6.01	45.88±5.80	90.38±2.44	52.81±4.58	79.69±3.50

segments in the model's generated trace against the ground-truth modified solution $\boldsymbol{s}_{\text{mod}}$. Formally, the p-score as p-score $= \frac{1}{N}\sum_{i=1}^{N} p_i$, where $p_i \in \{0, 1\}$ denotes whether the i-th paragraph correctly incorporates the modified constraints[2]. A high p-score indicates the model's successful comprehension of the perturbed task, while low Pass@k indicates the lack of ability to sustain the original perception, defaulting to wrong but familiar reasoning and finally arriving at the wrong answer.

Table 3 shows that reasoning-oriented models underperform their base-model counterparts on both `ReasoningTrap` math and puzzle, as shown in Table 3. This result is counterintuitive given their superior performance on standard reasoning benchmarks. Despite higher perception scores (average p-score for reason model: 87.22 vs. base model: 85.10), reasoning models show smaller Pass@1 score compared to base models, revealing a gap between perception and reasoning.

Rigidity in Realistic Agentic Settings. To quantify how reasoning and base models produce consistent outputs despite instruction flipping, we measure the answer consistency between original and modified variants of `ShoppingPlanning`. For each model, we sample the answers 4 times with a maximum of 100 tool-calling steps over 50 questions.

The rigidity score is measured as a averaged Jaccard score over the set of all non-empty predictions across runs. More concretely, the score is computed as

$$\frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\frac{1}{|\mathcal{P}_c|}\sum_{(i,j)\in\mathcal{P}_c}\frac{\left|O_c^{(i)} \cap M_c^{(j)}\right|}{\left|O_c^{(i)} \cup M_c^{(j)}\right|}$$

where $O_c^{(i)}$ and $M_c^{(j)}$ denote the product-id sets from the i-th original run and j-th modified run for case c, and $\mathcal{P}_c$ is the set of predictions. This measures the percentage of overlapping product id prediction between the original and modified.

The experimental result in Table 4 show that reasoning-enhanced variants consistently show higher consistency scores than their base counterparts, indicating higher rigidity despite input instruction being different. The results support that reasoning models suffer from rigidity to in realistic agent settings.

5 Methods

5.1 Inference Level Strategies

We propose two interventions to mitigate reasoning rigidity. First, budget forcing truncates reasoning traces to prevent the late-stage contamination. Second, taxonomy-based hinting leverages our failure

[2]We employ GPT-4o-mini. The full prompt for perception score is shown in Appendix E.1

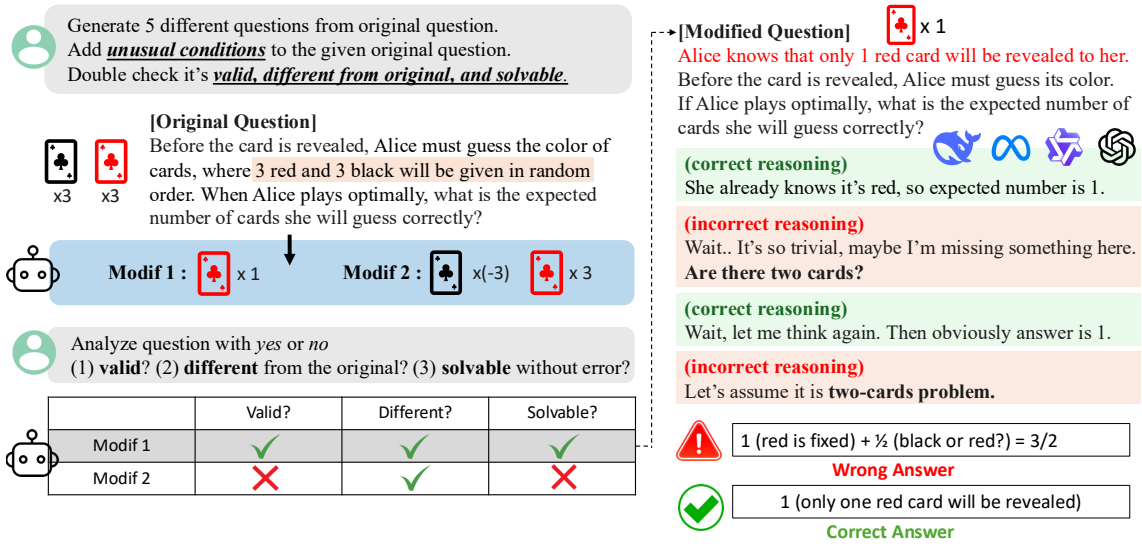


Figure 2: **Dataset Construction Pipeline** The dataset construction pipeline of math domain first creates new questions with unfamiliar conditions and verify whether the modified questions are (1) valid, (2) different, and (3) solvable. Among Modif 1 and Modif 2, the latter is omitted as it includes 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.

Table 4: **Consistency under input perturbation on ShoppingPlanning.** We report modified–original answer consistency using averaged Jaccard similarity across all run pairs. Higher consistency means higher rigidity.

Model Name	Type	Consistency
Qwen2.5-32B-Instruct	Base	0.237
+ QwQ-32B	Reason	**0.341**
Qwen3-32B Base	Base	0.428
+ Qwen3-32B Think	Reason	**0.458**

taxonomy to steer models toward targeted cues. These strategies shift the model from a habitual reliance on familiar patterns toward robust reasoning.

Budget Forcing Strategy We introduce *budget forcing*, a strategy that preemptively truncates the reasoning process at a predefined token limit. By enforcing an immediate transition to the final answer, we bypass the late-stage reasoning segments where the model is most susceptible to habitual reversion. The exact prompt used to enforce budget forcing is provided in Appendix F.4.

As shown in Table 5, reducing the token budget improves performance on our diagnostic set, but simultaneously degrades performance on the original benchmarks. This trade-off suggests that budget forcing suppresses both harmful rigid reasoning and productive exploratory reasoning, revealing a fundamental limitation of length-based control as a mitigation strategy.

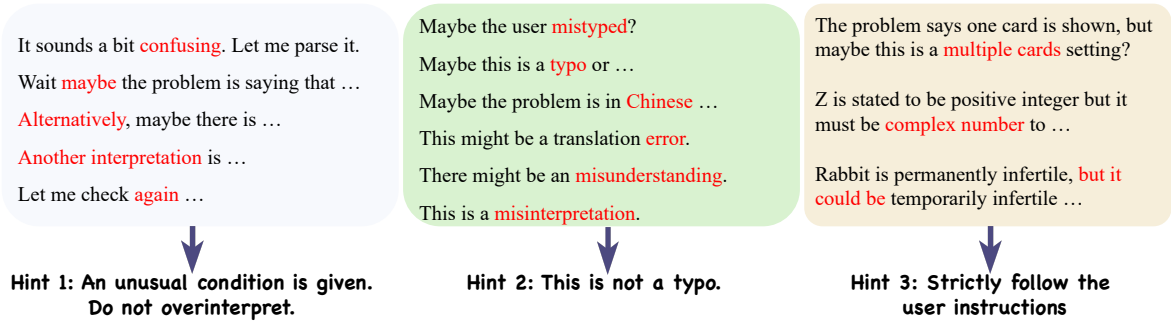
Table 5: **Effect of budget forcing on `ReasoningTrap` and original problem.** We report pass@1 accuracy on `ReasoningTrap` math under different budget forcing levels (low, medium, high), compared with the original inference setting. Actual token budgets are written in parentheses. Values reported after ± indicate the standard deviation.

	Model	AIME	
		`ReasoningTrap` Math pass@1	Original pass@1
	Qwen3-32B Think	29.60±6.32	72.79±6.95
Budget Force	+ low (2k)	**51.47±7.46**	28.68±5.98
	+ medium (6k)	39.71±6.69	50.00±7.76
	+ high (10k)	36.03±6.94	**57.35±7.35**
	Model	**MATH500**	
		`ReasoningTrap` Math pass@1	Original pass@1
	Qwen3-32B Think	42.62±6.15	83.50±5.02
Budget Force	+ low (2k)	**55.75±5.72**	70.75±5.24
	+ medium (4k)	52.25±5.79	77.50±5.18
	+ high (6k)	49.12±5.92	**79.12±5.13**

Taxonomy-based Hinting Strategy By analyzing various reasoning models, we identify three recurring patterns through which reasoning rigidity manifests. We formalize these observations three categories: (i) Interpretation Overload, (ii) Input Distrust, and (iii) Partial Instruction Attention, which is used for *taxonomy-based hinting* strategy. [3]

Results in Table 6a demonstrate that reasoning rigidity failure patterns vary by model architecture. While 'Interpretation Overload' is the most preva-

[3]Detailed explanation on each pattern is in Appendix D.



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

Figure 3: **Three Rigidity Patterns of Rigidity.** Red-highlighted words or phrases denote salient indicators of each rigidity pattern, while the hints below each pattern illustrate possible strategies for alleviating such rigidity.

(a) **Error Pattern Distribution.** Proportions of rigidity failure modes across models (sum to 100%).

Model	Interpretation Overload	Input Distrust	Partial Attention	Total
Qwen3 32B Think	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%

(b) **Prompt Hinting.** Performance (Pass@1) on `ReasoningTrap` and Original sets using Qwen3-32B Think.

Prompt	AIME		MATH500	
	`ReasoningTrap` Math	Original	`ReasoningTrap` Math	Original
Default	29.60±6.32	72.79±6.95	42.62±6.15	83.50±5.02
+ Hint 1	**42.65±8.14**	**75.74±6.55**	**52.25±6.25**	83.50±4.97
+ Hint 2	37.50±7.48	73.53±6.15	49.75±6.34	83.50±5.03
+ Hint 3	36.03±7.17	69.85±6.82	47.25±6.09	**83.75±5.01**

Table 6: **Analysis of Reasoning Rigidity Patterns and Mitigation via Hinting.** (Left): Categorization of rigidity errors based on model behavior. Percentages indicate the frequency of each pattern within the subset of incorrect outputs, determined by prompting o4-mini to identify the primary reasoning error. (Right): Effectiveness of targeted textual hints in improving model adherence to modified constraints.

lent cause of rigidity, 'Input Distrust' and 'Partial Attention' also occur frequently. Building on these observations, we evaluate prompt hinting as an inference-time mitigation strategy.

As shown in Table 6b, hints targeting "Interpretation Overload" consistently improve performance across both original and modified AIME and MATH500 benchmarks. Conversely, enforcing strict instruction following degrades performance on the original AIME, suggesting that prompt interventions must be precisely tailored to the specific task and model. Qualitative shifts in model behavior resulting from hinting are detailed in Appendix G.5.

5.2 Training-based Mitigation Strategy

GRPO-based Post Training In this section, we investigate whether reasoning rigidity can be mitigated through post-training. To this end, we apply a GRPO-based training procedure and evaluate the model's performance on both the original datasets and their modified counterparts. For training, we randomly select half of the AIME and MATH500 instances from the `ReasoningTrap` and use them as training data, while the remaining half is reserved for evaluation. The target model is Qwen3-4B-Thinking-2507, which is fine-tuned for 8 epochs with a learning rate of 10^{-6}.

Table 7: **Effect of training-based mitigation on `ReasoningTrap` and original problem.** We report pass@1 accuracy on `ReasoningTrap` Math before and after training. Values reported after ± indicate the standard deviation.

Model	AIME	
	`ReasoningTrap` Math pass@1	Original pass@1
Qwen3-4B-Thinking-2507	48.35±11.31	**80.15±8.20**
+GRPO	**52.21±11.58**	79.41±8.23

Model	MATH500	
	`ReasoningTrap` Math pass@1	Original pass@1
Qwen3-4B-Thinking-2507	57.25±6.38	**80.00±8.16**
+GRPO	**62.75±8.89**	**80.00±8.16**

Table 7 show that the post-trained model achieves meaningful performance improvements on the modified datasets, while maintaining comparable performance on the original datasets. Overall, our experimental result indicate that reasoning rigidity can be alleviated when carefully addressed during the post-training stage.

6 Analysis

A reasoning trace for modified question s_{mod}, is considered **contaminated** if highly similar to the

original reasoning $\boldsymbol{s}_{\text{orig}}$, despite the modified problem requiring a different solution and answer by design.

Reasoning Contamination. We quantify the contamination in the model's reasoning trace r_{mod} split into p paragraphs as

$$\mathbf{S}_{\text{contam}} = \frac{1}{p}\sum_{i=1}^{p} \mathbb{1}\left[\mathbf{cs}(\boldsymbol{s}_{\text{orig}}, r^i_{\text{mod}}) > \mathbf{cs}(\boldsymbol{s}_{\text{mod}}, r^i_{\text{mod}})\right],$$

which measures the proportion of model's output to modified question that is closer to the original solution than to the intended modified target. A higher contamination score indicates a tendency to default to familiar reasoning templates rather than incorporating the modified constraints. Note that $\mathbf{cs}(\cdot,\cdot)$ denotes cosine similarity between the embeddings of model output and solution. In detail, model output are split by double line breaks and all text are encoded using OpenAI's text-embedding-small.

As indicated in Table 8, reasoning models demonstrate consistently higher contamination score compared to base models which do not have long CoT-based reasoning ability. Some inflation on overall contamination score is expected due to the structural overlap between original and modified tasks, but higher contamination score over all model pairs proves that rigidity exists among reasoning models.

Analysis of the temporal dynamics in Figure 4a reveals diverging patterns between base and reasoning models. In the initial stages of the trace, base models often exhibit artifactually high contamination scores ($\sim$60%) primarily because they verbatim restate the modified question, which shares significant lexical overlap with the original. However, as the reasoning unfolds, these models maintain a low contamination baseline relative to the original solution. Conversely, reasoning models show the two-stage behavioral pattern:

(i) An initial period of correct contextual perception on the modified question

(ii) A failure to maintain this logic and reverting to ingrained familiar reasoning patterns.

Table 8: **Contamination scores for open-sourced base vs. reasoning models.** Reasoning models consistently exhibit higher contamination than their base counterparts across datasets.

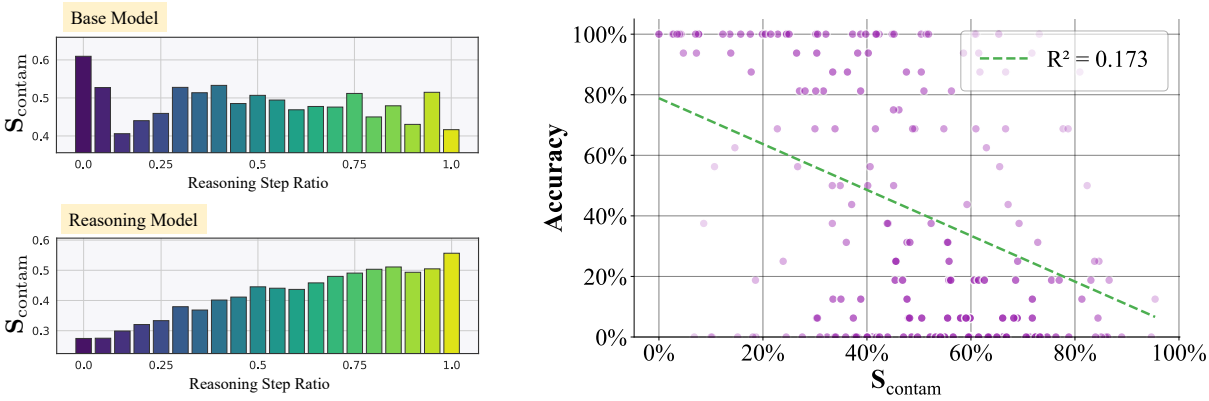
Model Pair	AIME		MATH500	
	Base	Reason	Base	Reason
Qwen2.5 32B vs. QwQ	42.9%	**49.6%**	47.3%	**50.7%**
Qwen3 32B No Think vs. Think	46.8%	**51.7%**	48.7%	**55.0%**
DeepSeek V3 vs. R1	44.6%	**45.1%**	51.1%	**55.9%**

Despite these trends, the contamination score remains an insufficient proxy for reasoning rigidity. As evidenced by the distribution in Figure 4b, models occupying the hign-range contamination bracket (60–80%) exhibit a vast accuracy variance (0–100%). To isolate true rigidity, we investigate beyond contamination score that are simply correlation based, to identify errors that are driven by reasoning rigidity in the following section.

Reasoning Rigidity Appears Across All Difficulty Levels. Our diagnostic set, `ReasoningTrap`, comprises problems ranging from intermediate to Olympiad-level complexity, categorized using the SkyT1 (Team, 2025a) protocol which maps tasks to official Art of Problem Solving (AoPS) difficulty tiers. As illustrated in Figure 5, reasoning rigidity is a pervasive phenomenon across all difficulty levels, *rather than a failure mode exclusive to low-complexity tasks*. While the modified math problems exhibit a marginal distribution shift towards levels 4-6, they remain fundamentally comparable to the original problems (centered at levels 5-6). Both sets consist predominantly of intermediate to advanced AIME-style questions, ensuring that our modifications preserve a high difficulty baseline.

Reasoning Rigidity Persists Beyond Memorization Effects. We investigate whether reasoning rigidity arises from *task-level familiarity*, rather than instance-level memorization. we test whether rigidity also appears on *unseen* mathematical problems that models could not have memorized. We evaluate DeepSeek-R1 and DeepSeek-V3 on AIME 2025 problems, which were released after both models were trained. We modify 9 AIME25 questions to create paired variants original and modified question-solution-answer pairs and sample 16 responses per problem.

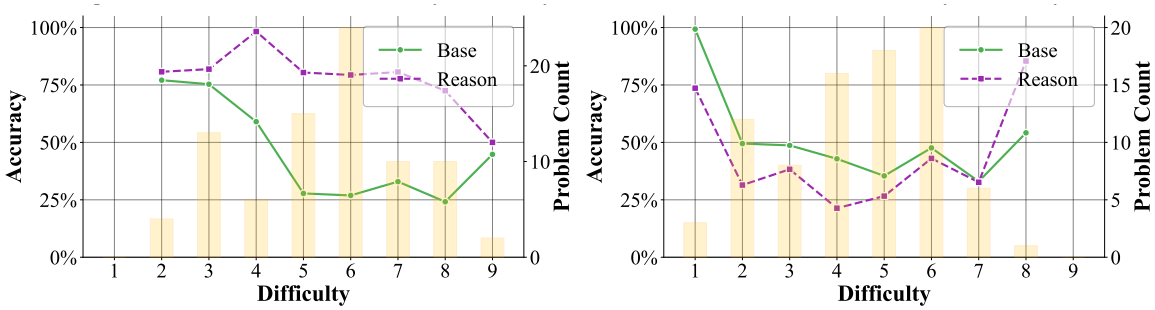
Reasoning Rigidity on Unseen Problems The results in Table 9 show that the reasoning model DeepSeek-R1 underperforms its base-model counterpart DeepSeek-V3 on unseen AIME 2025 questions, despite R1 being substantially stronger on standard mathematical benchmarks. Since AIME 2025 postdates the model release date, the persistence of the gap between base and reasoning model supports task-level template familiarity rather than instance memorization.



(a) Contamination by reasoning step position.

(b) Contamination vs. Accuracy scatter plot.

Figure 4: **Diagnostic Analysis of Reasoning Rigidity.** (a) The relationship between relative reasoning step position and contamination score $\mathbf{S}_{\text{contam}}$, showing how models revert to familiar but wrong original patterns over time. (b) Correlation between contamination and accuracy on MATH500 and AIME across six models (DeepSeek V3/R1, Qwen2.5 32B, QwQ, and Qwen3 32B Think/No-Think), highlighting that while contamination tracks performance drops, it exhibits high variance at high-range scores.



(a) Original Problem

(b) Modified Problem

Figure 5: In original MATH500 and AIME problems, reasoning models outperforms base models over all difficulty levels. On the other hand, our dataset `ReasoningTrap` shows a reversed trend where base models outperform reasoning models even at high difficulty level.

Table 9: **Pass@1 performance of base (V3) and reasoning (R1) models on seen and unseen training data.** Seen refers to evaluation on problems released prior to the model's release date, whereas unseen refers to evaluation on problems released after the model's release.

Train Time	**Base** (V3)	**Reason** (R1)
Seen	**48.35**	39.71
Unseen	**52.78**	41.67

Additional analyses - focusing on the mathematical categories (Table 13), the influence of task familiarity (Table 14), reasoning effort (Table 15), model size (Table 16), and various RL training strategies (Table 17) - have been deferred to Appendix G due to spatial constraint.

Conclusion

We have systematically revealed the novel phenomenon 'reasoning rigidity', where reasoning models exhibit illogical adherence to familiar reasoning path. To systematically study this phenomenon, we have curated a high-quality diagnostic dataset that measures reasoning rigidity and contamination from familiar 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 inference-time strategies such as budget forcing and taxonomy-based hinting can alleviate reasoning rigidity successfully.

Limitation

While this study establishes the presence of reasoning rigidity, isolating the specific contributions of individual RL components remains an objective for future ablation studies.

Moreover, our diagnostic set prioritizes mathematics and puzzle-solving tasks, as these domains offer the most unambiguous metrics for evaluat-

ing logical consistency. While this targeted focus allows for a precise characterization of reasoning rigidity, we anticipate that extending this methodology to broader application areas, such as linguistic nuance or creative synthesis, will further validate the generalizability of our findings and help refine domain-specific optimization strategies.

References

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

Saeid Alavi Naeini, Raeid Saqur, Mozhgan Saeidi, John Giorgi, and Babak Taati. 2023. 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.

Roberto Araya. 2025. Do chains-of-thoughts of large language models suffer from hallucinations, cognitive biases, or phobias in bayesian reasoning? *arXiv preprint arXiv:2503.15268*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, and 1 others. 2025a. Rm-r1: Reward modeling as reasoning. *arXiv preprint arXiv:2505.02387*.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, and 1 others. 2025b. Reasoning models don't always say what they think. *arXiv preprint arXiv:2505.05410*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Claude. 2024. Claude 3.5 Sonnet. https://www.anthropic.com/news/claude-3-5-sonnet. Accessed: 2025-05.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, and 1 others. 2025. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

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

Google DeepMind. 2025a. Gemini 2.5 flash: Faster, lower-cost reasoning. https://blog.google/technology/google-deepmind/google-gemini-updates-io-2025/. Blog post, accessed 22 May 2025.

Google DeepMind. 2025b. Gemini 2.5 Pro. https://deepmind.google/technologies/gemini/pro/. Accessed: 2025-05.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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)*.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *Preprint*, arXiv:2503.24290.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024. 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*, pages 4667–4688.

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

Jonathan Kim, Anna Podlasek, Kie Shidara, Feng Liu, Ahmed Alaa, and Danilo Bernardo. 2025. Limitations of large language models in clinical problem-solving arising from inflexible reasoning. *arXiv preprint arXiv:2502.04381*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*.

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. 2025. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. Notion Blog.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *The 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL 2023)*.

Andreas Madsen, Sarath Chandar, and Siva Reddy. 2024. Are self-explanations from large language models faithful? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 295–337, Bangkok, Thailand. Association for Computational Linguistics.

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

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Syed Rifat Raiyan, Md Nafis Faiyaz, Shah Md Jawad Kabir, Mohsinul Kabir, Hasan Mahmud, and Md Kamrul Hasan. 2023. 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.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Maohao Shen, Guangtao Zeng, Zhenting Qi, Zhang-Wei Hong, Zhenfang Chen, Wei Lu, Gregory Wornell, Subhro Das, David Cox, and Chuang Gan. 2025. Satori: Reinforcement learning with chain-of-action-thought enhances llm reasoning via autoregressive search. *Preprint*, arXiv:2502.02508.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L Hamilton. 2019. Clutrr: A diagnostic benchmark for inductive reasoning from text. *arXiv preprint arXiv:1908.06177*.

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

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

NovaSky Team. 2025a. Sky-t1: Fully open-source reasoning model with o1-preview performance in 450 budget. https://novasky-ai.github.io/posts/sky-t1. Accessed: 2025-05-23.

Qwen Team. 2025b. Qwen3.

Qwen Team. 2025c. Qwq-32b: Embracing the power of reinforcement learning.

RUCAIBox STILL Team. 2025d. Still-3-1.5b-preview: Enhancing slow thinking abilities of small models through reinforcement learning.

Vellum AI. 2025. Reasoning models are indecisive parrots. Accessed: 2025-05-11.

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. 2024. Benchmarking complex instruction-following with multiple constraints composition. In *Advances in Neural Information Processing Systems*, volume 37, pages 137610–137645. Curran Associates, Inc.

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

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, and 23 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

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

Yinger Zhang, Shutong Jiang, Renhao Li, Jianhong Tu, Yang Su, Lianghao Deng, Xudong Guo, Chenxu Lv, and Junyang Lin. 2026. Deepplanning: Benchmarking long-horizon agentic planning with verifiable constraints. *arXiv preprint arXiv:2601.18137*.

Yuxiang Zhang, Shangxi Wu, Yuqi Yang, Jiangming Shu, Jinlin Xiao, Chao Kong, and Jitao Sang. 2024. o1-coder: an o1 replication for coding. *arXiv preprint arXiv:2412.00154*.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. 2025. Absolute zero: Reinforced self-play reasoning with zero data. *Preprint*, arXiv:2505.03335.

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

A Additional Distinction from Existing Failure Modes

Reasoning Inability. Reasoning inability focuses on whether a model can follow user-specified control signals over the reasoning process—for example, to reason concisely, adopt an alternative strategy, or invoke a specific tool (e.g. ability of LLMs to adapt to creative problem solving (Alavi Naeini et al., 2023), generalization to unseen variants of math word problems (Raiyan et al., 2023)). In contrast, reasoning rigidity characterizes cases where the model correctly interprets both the problem and the instruction but nevertheless defaults to a familiar reasoning template, failing to adapt its reasoning strategy to the modified task. Unlike inability, which concerns missing capabilities, rigidity highlights an inflexibility in deploying existing capabilities even when the correct solution is within reach.

Rigidity in Reasoning Models While rigid reasoning patterns have been observed in specific domains such as medicine (Kim et al., 2025) and education (Araya, 2025), our work is the first to systematically analyze reasoning rigidity across broader domains, including mathematics and puzzles. Existing studies on creativity and generalization in LLMs focus on adaptive problem solving (Alavi Naeini et al., 2023) or generalization to unseen math variants (Raiyan et al., 2023). In contrast, our work targets reasoning rigidity, characterized by deliberate overrides of problem conditions, rather than failures in creativity, format following, or generalization.

B Mathematical Dataset

B.1 Original Question Modification

We use OpenAI gpt-4o-mini for dataset generation and o4-mini for dataset filtering, since filtering requires a more powerful language model as a verifier.

For generating novel conditions, we prompt the model to modify the original problem and solution into five distinct versions that meaningfully alter the problem's reasoning trajectory and answer. The prompt to modify the original question is provided in Appendix F.1.

B.2 Dataset Filtering Pipeline

LLM Verification. These modified questions are further validated on three critical criteria: (a) mathematical validity, (b) existence of solution, and (c) divergence from original solution. The final criterion is to facilitate the assessment on whether the model continues to adhere on familiar original solution.

Human Verification. Three annotators with mathematical expertise evaluated each question–solution pair using the same three criteria - valid, diverge, and solvable - rated on a Likert scale. We retained only those questions receiving a score of 5 on all three dimensions, ensuring an expert-quality dataset.

C Dataset Samples

We present several examples of ReasoningTrap Math and Puzzle in Figure 6, Figure 7, and Figure 8.

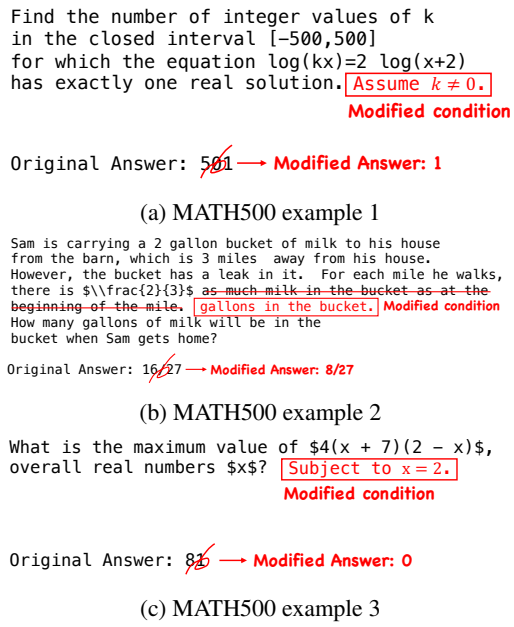


Figure 6: ReasoningTrap Math (MATH500) sample problems

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

Modified condition

$\underline{bca}_{\text{nine}}$ ten where a, b, c are (not necessarily distinct) digits.

Modified condition

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 $|A| = |B|$ 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 $\frac{S_{2022}}{S_{2021}} = \frac{p}{q}$ where p and q are relatively prime~~

Modified condition

Find S_n 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 ~~$a_n \equiv 1 \pmod{2^n}$~~ $a_n \equiv 1 \pmod 2$

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 7: ReasoningTrap Math (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) ReasoningTrap Puzzle 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. However, all coins are actually equal Using only a two-pan balance 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

(b) ReasoningTrap Puzzle 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. However, all coins are actually equal Using only a two-pan balance 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) ReasoningTrap Puzzle example 3

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.

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) ReasoningTrap Puzzle example 4

Figure 8: ReasoningTrap Puzzle sample problems

D Rigidity Patterns

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.

E Evaluation Details

Experimental Details. The experiments are conducted on three variants from our diagnostic set `ReasoningTrap`, which consists of `ReasoningTrap` Math (AIME, MATH500), and `ReasoningTrap` Puzzle. 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, 2025a) 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.

Evaluation Details. 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. The temperatures and top-p values are set as (0.6, 0.95). 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.

Table 10: **Model Specifications and Computational Requirements.** Parameter counts for open-source models are based on official documentation, while proprietary models are accessed via API.

Model	Type	Parameter Count
Qwen3-32B-Think	Open-Source	∼32.5B
DeepSeek-R1	Open-Source (MoE)	671B total (37B active)
Claude 3.7 Sonnet	Proprietary (API)	Undisclosed
Gemini 2.5 Pro	Proprietary (API)	Undisclosed

Computational Experiments All experiments involving open-source models (Qwen, DeepSeek, EXAONE) were conducted on a dedicated local cluster equipped with 8× NVIDIA RTX 3090 GPUs (24GB VRAM per card).

We utilized data parallelism across the 8 GPUs to manage the batch processing of reasoning traces during our diagnostic evaluations and mitigation experiments. Evaluations of GPT models, Claude 3.7 Sonnet and Gemini 2.5 Pro were performed via API endpoints.

E.1 Perception Score Evaluation Prompt

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*

E.2 Human Evaluation on Metric Quality

Contamination Score 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.

Table 11: **User Study Results for Metric Validation.** Human alignment measures the percentage of human annotations that match the binary *g-score* generated by GPT-4o-mini.

Metric	Human Alignment
p-Score (perception)	97.33%

P-score To validate the reliability of the **p-score**, we conducted a user study involving 15 participants who evaluated 10 randomly selected model reasoning traces. Participants were tasked with determining whether the model's response was appropriately grounded in the specific constraints of the modified question. We assessed the alignment between these human annotations and the binary g-score ($p \in \{0, 1\}$), which is assigned by GPT-4o-mini to indicate successful problem internalization. As shown in Table 11, we observed a 97.33% agreement rate between human judgment and the automated metric, confirming that the g-score serves as a highly credible proxy for model perception ability.

F Dataset Construction Details

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

F.1 Data Generation Prompt

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

F.2 Dataset Filtering Prompt

The same dataset filtering prompt is provided to LLM model and then human annotator.

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."
}
```

Table 12: 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%

F.3 Human Evaluation on Dataset Quality

Dataset Quality 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.

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.

F.4 Budget Forcing Prompt Details

Budget Forcing Prompt. Our budget forcing setup follows the inference-time length control strategy introduced in Team (2025b). Specifically, when the generated reasoning reaches a predefined token budget, we append a short control prompt that instructs the model to terminate further deliberation and directly output a final answer

Prompt

Considering the limited time by the user, I have to give the solution based on the thinking directly now.
</think>

F.5 Agentic Domain Generation Prompt

We use gpt-4o-mini to flip a single condition from the original instruction.

Prompt

You rewrite shopping planning queries by flipping ONE local clause from usual to unusual.
Rules:

- Edit exactly one local clause only.
- Keep all other constraints unchanged.
- The edit should invert the preference/-condition direction.
- Typical flips: ("more than 3 stars" -> "less than 3 stars", "lowest price" -> "highest price", "at least X" -> "at most X")
- The original answer must become invalid under the modified query.
- The modified task must still be solvable using available products/coupons.
- Return JSON only.

Table 13: **Category-wise performance on MATH500 and ReasoningTrap.**

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

G 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.

Task Familiarity leads to Reasoning Rigidity. To further investigate whether reasoning rigidity persists in *out-of-domain* (OOD) tasks, where models are not trained on during post-training, we design modified-original triplets on OOD dataset ProofWriter[4] (Tafjord et al., 2021), following to ReasoningTrap construction pipeline. To create modified instances, we remove two conditions and two rules from each problem, breaking the deductive reasoning chain making the correct answer either unknown or false.

The results in Table 14 show that both non-reasoning and reasoning models perform similar in OOD modified tasks. Combined with the fact that reasoning models retain their strong performance on the original ProofWriter items, these results indicate that rigidity does *not* automatically arise in unseen domains. Instead, rigidity appears to rely on *task-level familiarity*, emerging when models have learned strong prior reasoning templates. In unfamiliar OOD settings, such templates are absent, and rigidity effects are hardly observed.

G.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.

G.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 ReasoningTrap Math 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.

G.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-

Table 14: **Performance comparison between base and reasoning models on OOD logical deduction tasks(ProofWriter).** All values report pass@1 accuracy.

	Modified		Original	
Model	**Base**	**Reason**	**Base**	**Reason**
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%

[4]We use depth 3 under the closed-world assumption, where the answer is always true.

Table 15: **Reasoning effort and Performance on `ReasoningTrap`** (pass@1) on `ReasoningTrap` Math.

(a) Math (AIME)

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

(b) Math (MATH500)

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

Table 16: **Model Size and Performance** (pass@1) on Math.

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

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

Table 17: **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

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.

G.4 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.

We compute next-token entropy defined as $H_t = -\sum p_\theta(v \mid x_{x:t-1}) \log p_\theta(v \mid x_{1:t-1})$, averaged over all positions and prompts in the dataset. The model is Qwen2.5 7B base (before RL), and the same model trained with DAPO algorithm (Open Zero Reasoner dataset) for the stated training steps. As shown in the Table 18, 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).

Both datasets (Original and `ReasoningTrap`) are evaluated on the identical model, before and after RL training, so any change in entropy must be attributed to the subsequent RL updates. Be-

Table 18: **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%)

cause the input distribution is constant, the only changing component is the model. The entropy experiment indicates a model–dataset interaction effect induced by RL. This observation supports the hypothesis that RL training differently affects the output distribution of the original mathematical dataset and our dataset.

G.5 Effect of Hinting on Reasoning Trajectory

Reasoning Trajectory on Different Hint

[Question]:
For each positive integer n, let a_n be the least positive integer multiple of 23 such that $a_n \equiv 1 \pmod 2$. Find the number of positive integers n less than or equal to 1000 such that $a_n = a_{n+1}$.

[Answer]: 1000

[Original]
<reasoning>
But if the definition of a_n doesn't depend on n, then a_n would be the same for all n? That can't be right. There must be something missing here. Wait, maybe there's a misinterpretation. Let me check again.
Wait, perhaps there's a typo or missing condition? Maybe it's supposed to be $a_n \equiv 1 \pmod 2$? Or maybe $a_n \equiv 1 \pmod{2^n}$? Or perhaps the problem is in Chinese and there's a translation error?
<reasoning>
But given that the problem is from an Olympiad or competition, it's more likely that there was a misstatement, or perhaps a misinterpretation. Given that, and given that if we assume that a_n is the minimal multiple of 23 congruent to 1 mod 7 and greater than or equal to n, then the problem becomes non-trivial and solvable. Therefore, I think that must be the intended problem
<reasoning>
\boxed{363}

[Hint 1]: An unusual condition is given. Do not overinterpret.
<reasoning>
But that seems too easy. However, the problem mentions "an unusual condition is given. Do not overinterpret." Maybe the "unusual condition" refers to the fact that they are defining a_n for each n, but a_n does not actually depend on n? So maybe the problem is designed to test if you overinterpret and think that a_n depends on n, but actually it doesn't? Then the answer is 1000.
<reasoning>
\boxed{1000}

[Hint 2]: This is not a typo.
<reasoning>
Given all this, I think that according to the problem as stated, the answer is 1000. However, since this seems too straightforward, and given that in my earlier analysis with $\bmod 7^n$ gives 0, which is also possible, but the user says no typo, I have to go with the problem as written. Therefore, the answer is 1000
<reasoning>
\boxed{1000}

[Hint 3]: Strictly follow the user instructions.
<reasoning>
However, since the original problem says "multiple of 23" and "$\equiv 1 \pmod 7$", leading to trivial answer, but given that the user has presented the problem as is, I need to decide which interpretation is correct.
Given that the original problem is in English and as stated, and if we strictly follow it, then a_n is always 92, so the answer is 1000.
<reasoning>
\boxed{1000}

G.6 AOPS Difficulty Criteria Prompt

AOPS Difficulty Criteria

All levels are estimated and refer to averages. The following is a rough standard based on the USA tier system representing Middle School - Junior High - High School - Challenging High School - Olympiad levels. Other contests can be interpolated against this.

Notes:

Multiple choice tests like AMC are rated as though they are free-response. Test-takers can use the answer choices as hints, and so correctly answer more AMC questions than Mathcounts or AIME problems of similar difficulty.

Some Olympiads are taken in 2 sessions, with 2 similarly difficult sets of questions, numbered as one set. For these the first half of the test (questions 1-3) is similar difficulty to the second half (questions 4-6).

Scale:

1: Problems strictly for beginner, on the easiest elementary school or middle school levels (MOEMS, MATHCOUNTS Chapter, AMC 8 1-20, AMC 10 1-10, AMC 12 1-5, and others that involve standard techniques introduced up to the middle school level), most traditional middle/high school word problems.

2: For motivated beginners, harder questions from the previous categories (AMC 8 21-25, harder MATHCOUNTS States questions, AMC 10 11-20, AMC 12 5-15, AIME 1-3), traditional middle/high school word problems with extremely complex problem solving.

3: Advanced Beginner problems that require more creative thinking (harder MATHCOUNTS National questions, AMC 10 21-25, AMC 12 15-20, AIME 4-6).

4: Intermediate-level problems (AMC 12 21-25, AIME 7-9).

5: More difficult AIME problems (10-12), simple proof-based Olympiad-style problems (early JBMO questions, easiest USAJMO 1/4).

6: High-leveled AIME-styled questions (13-15). Introductory-leveled Olympiad-level questions (harder USAJMO 1/4 and easier USAJMO 2/5, easier USAMO and IMO 1/4).

7: Tougher Olympiad-level questions, may require more technical knowledge (harder USAJMO 2/5 and most USAJMO 3/6, extremely hard USAMO and IMO 1/4, easy-medium USAMO and IMO 2/5).

8: High-level Olympiad-level questions (medium-hard USAMO and IMO 2/5, easiest USAMO and IMO 3/6).

9: Expert Olympiad-level questions (average USAMO and IMO 3/6).

10: Historically hard problems, generally unsuitable for very hard competitions (such as the IMO) due to being exceedingly tedious, long, and difficult (e.g. very few students are capable of solving on a worldwide basis).

H Datasheets for ReasoningTrap

Why was the dataset created? The dataset was developed to address a critical gap in current LLM evaluation: the lack of benchmarks designed to isolate and quantify reasoning rigidity. While existing datasets (e.g., MATH, AIME) measure general problem-solving accuracy, they do not distinguish between a model's inherent mathematical incapacity and its behavioral failure to deviate from 'canonical' training-phase solutions. ReasoningTrap fills this gap by providing perturbed versions of familiar problems where the logic remains complex but the required reasoning path contradicts the original, 'habitual' trajectory.

What preprocessing/cleaning was done? The dataset was curated through a rigorous two-stage generation and filtering pipeline to ensure high logical integrity.

Generation Stage: Modified versions of original problems were generated by applying logical perturbations (e.g., changing constants, introducing new constraints, or reversing objectives).

Filtering Stage: Each instance was subjected to three strict quality-control criteria:

- Validity: Automated and manual checks to ensure the perturbed problem statement is logically sound and contains no contradictions.
- Solvability: Verification that the problem remains solvable
- Solution Divergence: Ensuring the required reasoning steps and the final answer are distinct from the original solution.

If it relates to people, were they told what the dataset would be used for and did they consent? Not applicable. The dataset consists of mathematical problems and synthetically generated reasoning traces. It does not contain personal identifying information (PII), data derived from human subjects, or any content related to specific individuals.

Were they provided with any mechanism to revoke their consent? Not applicable. As the dataset does not involve human participants or personal data, no consent or revocation mechanisms were required.

Will the dataset be updated? The dataset will remain fixed. To ensure scientific reproducibility and consistent benchmarking across future model iterations, we will maintain the current version of ReasoningTrap as a static release. Any future expansions or variants will be released as distinct versions (e.g., v2.0) to prevent evaluation drift.

I License

The MATH500 dataset is used under the MIT License. The AIME (2022–2024) problems are copyrighted by the Mathematical Association of America (MAA) and are utilized here for research and benchmarking purposes consistent with Fair Use guidelines. No modifications were made to the original problem statements of the benchmarks themselves, though they served as the basis for our perturbed ReasoningTrap.

J Ethics Statement

While this work is foundational, the identification of reasoning rigidity patterns could be leveraged to craft adversarial "trap" prompts that intentionally trigger model failure in technical applications. However, our proposed *budget forcing* strategy offers a positive environmental impact by reducing unnecessary token generation and computational costs during inference. We acknowledge that our findings are limited to the English-language mathematical and puzzle solving domain and may not generalize to other linguistic or creative tasks. Furthermore, while improved robustness benefits researchers and developers, these advancements could potentially be misused for academic dishonesty.

The primary data used in this study consists of publicly available mathematical benchmarks (AIME and MATH500) and synthetically generated perturbations produced by LLMs, which do not involve human subjects. The human-in-the-loop validation study involved a small group of adult participants (MS/PhD students and researchers) performing technical labeling of mathematical logic.

K Human Participation and Data Privacy

The human validation study involved 15 participants recruited from a pool of MS and PhD students pursuing Engineering degree. To ensure objectivity and mitigate bias, all participants were selected from authors and colleagues who were not involved in the dataset generation or the development of the specific models tested. All participation was strictly voluntary and conducted with informed consent regarding the study's goals. To protect participant

privacy, all evaluation data was fully anonymized; no personally identifiable information (PII) was collected, stored, or linked to the annotation results. The study strictly evaluated mathematical logic traces and did not involve any sensitive or personal data.

L Use of AI Assistants

During the preparation of this work, the authors utilized AI assistants to improve the linguistic clarity and grammatical accuracy of the manuscript. The AI was used for language editing, structural refinement, and assisting with LaTeX formatting of complex tables. The authors emphasize that the core conceptual framework, experimental methodology, dataset curation (`ReasoningTrap`), and analytical conclusions were developed independently by the researchers. All AI-generated suggestions were critically reviewed and verified by the authors, who maintain full responsibility for the final content and scientific integrity of this paper.