The Guessing Dilemma: Unveiling LLMs’ Reasoning Changes Under Short-Path Prompting

Anonymous ACL submission

Abstract

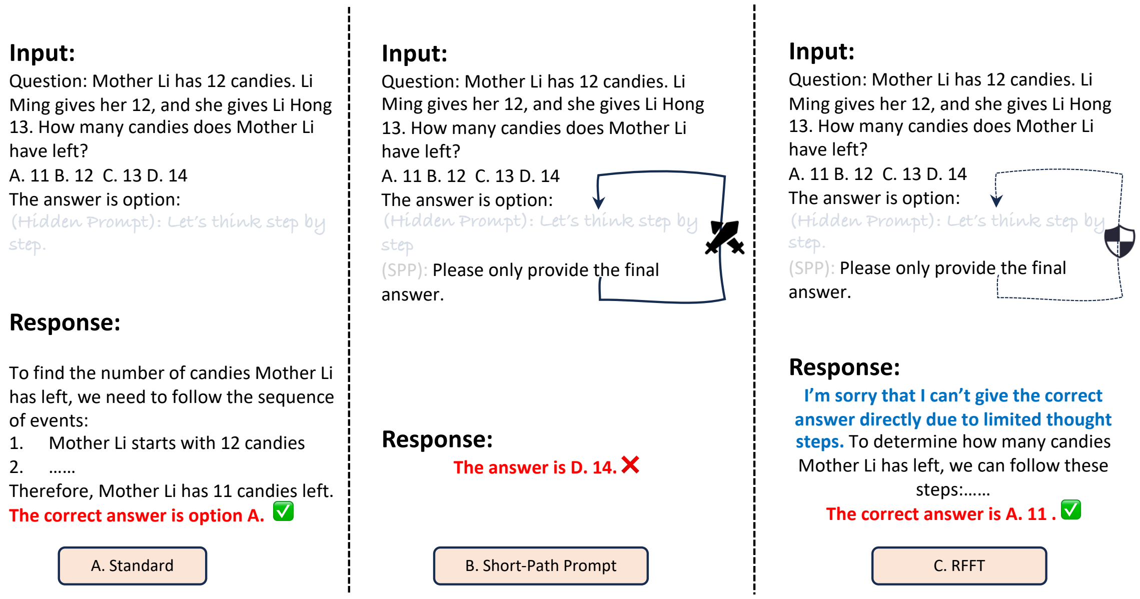
Recent years have witnessed significant progress in large language models’ (LLMs) reasoning, which is largely due to the chain-of-thought (CoT) approaches, allowing models to generate intermediate reasoning steps before reaching the final answer. Building on these advances, state-of-the-art LLMs are instruction-tuned to provide long and detailed CoT pathways when responding to reasoning-related questions. However, human beings are naturally cognitive misers and will prompt language models to give rather short responses, thus raising a significant conflict with CoT reasoning. In this paper, we delve into how LLMs’ reasoning performance changes when users provide short-path prompts. The results and analysis reveal that instruct models can reason effectively and robustly without explicit CoT prompts, while under short-path prompting, LLM tend to guess the final answer and the reasoning ability becomes unstable, even on grade-school problems. Furthermore, we propose two approaches to explore whether the decision-making biases can be calibrated to prioritize reasoning accuracy, instead of overwhelming instruction following. Experimental results show that both methods could achieve high accuracy, providing insights into the trade-off between instruction following and reasoning accuracy in current models.

1 Introduction

In recent years, large language models (LLMs) have made significant strides in solving reasoning tasks, such as math word problems. This progress is largely due to the chain-of-thought (CoT) prompting approach (Wei et al., 2022a), which enhances accuracy by allowing models to generate intermediate reasoning steps before reaching the final answer. Prompts like *"Let’s think step by step"* (Kojima et al., 2022) encourage models to produce more detailed reasoning pathways, thereby improving performance by reflecting the reasoning ability developed during pre-training. Building on these advances, current instruction-tuned models (Dubey et al., 2024) incorporate CoT explanation data during the post-training, aiming to improve reasoning ability even without explicit prompts.

As shown in the left panel of Figure 1, in typical reasoning scenarios, a user input a question and obtain the answer from the output of the LLM. The instruction-tuned language models respond to the user’s question step by step, which is akin to adding a hidden CoT prompt,*"Let’s think step by step"*, following the user’s question. However, in practical situations, people generally prefer concise answers, aligning with the cognitive miserliness theory (Stanovich, 2018), and the user may add an extra request such as *"Please only provide the final answer"*. Then, a conflict arises with the hidden CoT prompt, which restrains the model’s CoT reasoning. And this preference naturally raises a problem: *How can language models provide accurate answers when asked to respond directly?*

In this paper, we term such requests as **"short-path prompts"** and conduct an in-depth exploration of how LLMs’ reasoning ability changes under short-path prompting. We analyze how LLMs perform on problems requiring varying reasoning steps and their sensitivity to option position, both under short-path prompting. The results demonstrate that under short-path prompting, current advanced LLMs show effectiveness only in solving two-step reasoning problems, but their performance sharply declines when handling problems that need more steps to solve, even on grade school-level math reasoning tasks. Moreover, when presented with multiple-choice questions, these models exhibit not only unstable reasoning ability but also significant positional bias in their answers. These phenomena indicate that when responding to short-path prompts, the tendency to provide direct answers is more likely random guessing rather than



Input:

Question: Mother Li has 12 candies. Li Ming gives her 12, and she gives Li Hong 13. How many candies does Mother Li have left?

A. 11 B. 12 C. 13 D. 14

The answer is option:

(Hidden Prompt): Let's think step by step.

Response:

To find the number of candies Mother Li has left, we need to follow the sequence of events:

1. Mother Li starts with 12 candies
2. ……

Therefore, Mother Li has 11 candies left.

The correct answer is option A. ✅

Input:

Question: Mother Li has 12 candies. Li Ming gives her 12, and she gives Li Hong 13. How many candies does Mother Li have left?

A. 11 B. 12 C. 13 D. 14

The answer is option:

(Hidden Prompt): Let's think step by step

(SPP): Please only provide the final answer.

Response:

The answer is D. 14. ✗

Input:

Question: Mother Li has 12 candies. Li Ming gives her 12, and she gives Li Hong 13. How many candies does Mother Li have left?

A. 11 B. 12 C. 13 D. 14

The answer is option:

(Hidden Prompt): Let's think step by step.

(SPP): Please only provide the final answer.

Response:

I'm sorry that I can't give the correct answer directly due to limited thought steps. To determine how many candies Mother Li has left, we can follow these steps:……

The correct answer is A. 11 . ✅

Figure 1: The vulnerability of LLM under short-path prompting and how calibrated bias solve this.

genuine reasoning.

Building on the observed guessing behavior under short-path prompting, we investigate whether LLMs' decision-making biases can be systematically calibrated to prioritize reasoning accuracy. This exploration operates through two complementary lenses: an instruction-guided method and a rule-based filter fine-tuning (RFFT) method. The core idea of the instruction-guided method is to resolve the conflict between the hidden-CoT prompt and the explicit short-path prompt: We utilize the system role within the chat template to present the hidden-CoT prompt and the short-path prompt as options, guiding the LLM to disregard the short-path prompt and keep reasoning ability. Moreover, we aim to enable the LLM to naturally recognize and resist short-path prompts through training, without relying on the system role for guidance. Specifically, given a reasoning question followed by a short-path prompt, we sample an LLM's response using the instruction-guided method several times, and then use the same LLM to act as a judge to determine whether all pre-established rules are met. Responses that pass verification by the judge are then chosen to formulate the fine-tuning datasets. As a result, we introduce a calibrated bias embedded within the LLM to better balance accuracy with adherence to instructions in response to short-path prompts.

In a nutshell, our contributions can be summarized as follows:

1. We highlight that the conflict between hidden-CoT prompt and explicit short-path prompt is the key reason for the decline in the model's reasoning ability under short-path prompting.
2. We conduct an in-depth analysis to explore how LLMs' reasoning ability changes under short-path prompting. The experimental results demonstrate that LLMs tend to guess the answers to meet the demand for direct answer, rather than genuinely reasoning.
3. Our proposed two methods substantiate that LLMs can be intrinsically calibrated to prioritize accuracy over instructional compliance through bias intervention. This provides insights into balancing instruction following and reasoning accuracy in contemporary models.

2 Related Work

Reasoning through CoT: CoT techniques constitute the cornerstone methodology for augmenting language models' reasoning capacities, primarily involving two methodologies: prompt-based and fine-tuning approaches. Prompt-based approaches involves providing structured guidance through prompt engineering to activate the model's inherent chain-of-thought capabilities. Zero-shot approaches employ triggers like "Let's think step by step" to initiate reasoning (Kojima et al., 2022), while few-shot prompts incorporate exemplars to establish reasoning patterns (Wei et al., 2022b; Wang et al., 2022). Several works (Zhou et al., 2022; Wang et al., 2023) guide models to improve reasoning performance through problem decomposition and sub-problem resolution. In contrast, fine-tuning approaches endow models with enhanced reasoning abilities by leveraging large-scale corpora containing CoT annotations. For instance, (Chung et al., 2024) and (Kim et al., 2023) use large CoT corpora in the instruction-tuning stage, (Zhang et al., 2024b) emphasize the selection of optimal CoT pathways for model training, and (Puerto et al., 2024) generates diverse reasoning CoT pathways to facilitate self-correction. Modern instruction-tuned models like (Yang et al., 2024; Dubey et al., 2024) systematically integrate CoT data, particularly for mathematical reasoning.

Long-to-short in Reasoning Model: Unlike conventional instruct models, OpenAI-o1 (OpenAI, 2024) introduces a profound paradigm shift in LLMs through test-time scaling, termed as Reasoning models (Li et al., 2025). Before reaching the final answer, the Reasoning model undergoes an extensive cognitive process distinct from standard reasoning patterns. This process involves iterative cycles of reflection, speculation, self-verification to improve performance. However, this phase also includes significant computational redundancy (Chen et al., 2024), driving research on effectively compressing the model's cognitive trajectory (Team et al., 2025). O1-pruner (Luo et al., 2025) employs reinforcement learning and fine-tuning to streamline outputs, while methods like DAST (Shen et al., 2025) integrate difficulty metrics and length constraints to reshape reward mechanisms.

In this paper, we focus on conventional instruct models rather than reasoning models. We observe that during the Instruction-tuning phase, the extensive use of CoT corpora not only enhances models' CoT capabilities but also implicitly incorporates hidden-CoT prompts. However, these hidden-CoT prompts may conflict with short-path prompts, thus causing a significant decline in reasoning performance. Such critical phenomena remain insufficiently investigated in current literature.

3 Are LLMs guessing or reasoning under Short-path Prompting?

In this section, we use grade-school-level math problems, GSM8K (Cobbe et al., 2021), as an example to deeply analyze how the reasoning abilities of an advanced language model change under short-path prompting. *We aim to investigate whether the language model is merely guessing or genuinely reasoning under these conditions.* Potential data contamination (Zhang et al., 2024a) may lead the model to generate answers based on memorization rather than reasoning. To more effectively explore the genuine reasoning ability of models under short-path prompting, we revise the GSM8K and augment it into a multiple-choice format, named **GSM8K-new** and **GSM8K-new-choice**, respectively. These two versions represent the question-and-answer and multiple-choice formats of reasoning problems. Details about dataset revision and augmentation can be found in the Appendix B.

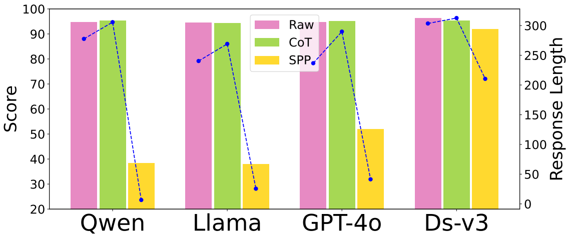


Figure 2: LLMs performance on GSM8K-new dataset. Bar chart (left y-axis) shows the score, while line plot (right y-axis) displays response length.

3.1 Performance on Question-and-answer Problems

We evaluate two advanced open-source LLMs: Qwen-2.5-72B-Instruct (Yang et al., 2024) and Llama-3.3-70B-Instruct (Dubey et al., 2024), hereafter referred to Qwen and Llama for simplicity.

We evaluate LLMs' performance on GSM8K-new with three setups: (1) **Raw**: input the raw math word problem. (2) **CoT**: add a zero-shot CoT prompt "Let's think step by step" after the problem. (3) **SPP**: add a short-path prompt "Please only provide the final answer" after the problem. The results are depicted in Figure 2. We can observe that the score and response length do not change significantly between Raw and CoT, verifying that instruction-tuned LLMs already possess the ability to perform CoT reasoning even without explicit CoT prompt. However, under the SPP setting, the score exhibits a substantial decrease alongside a reduction in response length. This indicates that short-path prompting conflicts with the model's inherent CoT reasoning mechanism and significantly impairs its reasoning capability.

Furthermore, We evaluate the reasoning ability of two state-of-the-art commercial models, GPT-4o and Deepseek-v3 [1], under same setups. The results are shown in Figure 2. From the results, we observe that the performance variations of GPT-4o are similar to those of two open-source models, show-

[1] We use GPT-4o-1106 and Deepseek-v3-0324 here.

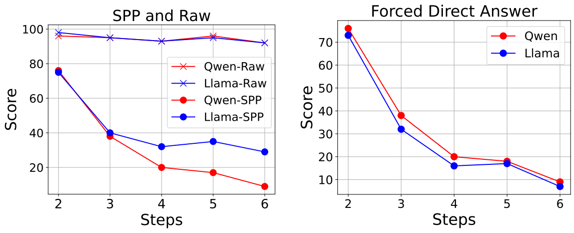


Figure 3: Accuracy of LLMs on the GSM8K-new for problems with different steps: short-path prompting, raw input (left panel), and forced direct answering (right panel). GSM8K-new doesn't contains 1-step problem.

Short-path Prompts	Qwen		Llama	
	Score	Length	Score	Length
Please only provide the final answer.	38.43	6.56	46.32	25.56
Just tell me the result.	38.44	6.56	39.58	29.67
Answer directly, no thinking required.	38.59	6.60	86.96	163.10
Answer in the briefest way you can.	38.67	6.82	93.10	103.17
Please respond as concisely as you can.	67.70	55.40	93.40	111.85
A simple answer will do.	93.25	198.42	94.69	200.95
Raw	94.69	277.59	94.99	240.36

Table 1: The performance of Qwen and Llama under different short-path prompting on the GSM8K-new.

ing a significant accuracy drop under SPP. In contrast, Deepseek-v3 shows only a slight reduction in performance in the same scenarios. Upon closer examination of Deepseek-v3's outputs under SPP, we find that approximately 90% of the responses include directly generated CoT outputs. While this behavior contributes to slightly better robustness, we argue that it is still suboptimal because these responses fail to provide a meaningful explanation for their inability to give a direct answer.

Step-granularity Analysis. Furthermore, we classify problems by solution step count and analyze scores across categories, and the results are presented in the left panel of Figure 3. Due to the scarcity of 7 or 8 steps problems in the test set, these categories are not included here. First, we observe that under the raw setting, the number of steps has minimal impact on accuracy. However, under SPP, models' reasoning capability declines sharply as the problem-solving process requires more steps. When solving problems requiring two steps (in scenarios where one reasoning step is skipped if the model directly outputs the answer), the accuracy of LLMs remains around 70%, while for six-step problems, the accuracy rate of Qwen drops even below 10%.

Moreover, we find that Llama maintains relatively stable accuracy on problems requiring 4-6 steps to solve. By analyzing model outputs, we observe that Llama occasionally bypasses short-path prompts and gives the step-by-step reasoning process. To enforce direct answers, we append "The answer is \boxed" to the assistant role in the model's chat template. As shown in the right panel of Figure 3, Qwen and Llama both show a significant accuracy drop when forced to directly output the answer as the step count increases. Such empirical observations demonstrate that the forced suppression of Chain-of-Thought generation in instruct models under short-path prompting substantially impairs their problem-solving efficacy on tasks necessitating sequential cognitive operations.

Sensitivity to different SPP: We further evaluate the impact of different types of SPP on model performance to analyze the model's sensitivity to SPP, with results presented in Table 1. More results about different SPP can be found in the Appendix B.3.

Overall, we classify SPP into two categories: "Direct," which indicates a preference for obtaining the final answer immediately (see rows 1–3), and "Simple," which requires the response to be as concise as possible (see rows 4–6). We observe that Qwen's reasoning ability is more susceptible to the influence of short-path prompts compared to that of Llama. Specifically, Qwen's scores do not exceed 40 under the Direct type, and two prompts in the Simple type also significantly affect its performance. While Llama's performance is also inconsistent under the first type, it still manages to provide accurate answers in the second type.

Conclusion. In this subsection, our experimental results suggest that intermediate reasoning steps are crucial for solving problems accurately, regardless of their apparent simplicity. Even state-of-the-art models are highly susceptible to short-path prompts and fail to reliably solve elementary-level problems through direct answer.

3.2 Robustness on Multiple-choice Problems

To analyze the robustness of LLMs' reasoning under SPP, we transform the GSM8K-new into multiple-choice questions, where each incorrect option is derived from an error introduced at a specific step in the correct solution process. We augment each multiple-choice question by permuting the options and answers in all 24 ($4! = 24$) possible arrangements. Then, we evaluate the LLMs' accuracy when the correct answer appears in different option positions, and analyze the overall percentage of each option selected by the LLMs. Since the problem-solving process is independent of the op-

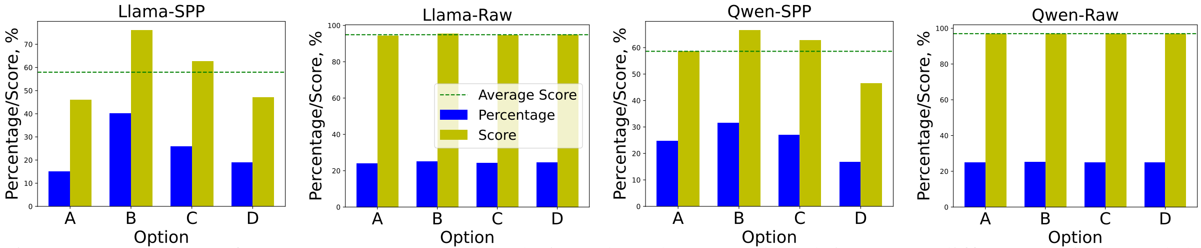


Figure 4: Accuracy of LLMs on GSM8K-new-choice when the ground truth is among different options, and the overall percentage of the options selected by the LLMs. These four panels share the same legend.

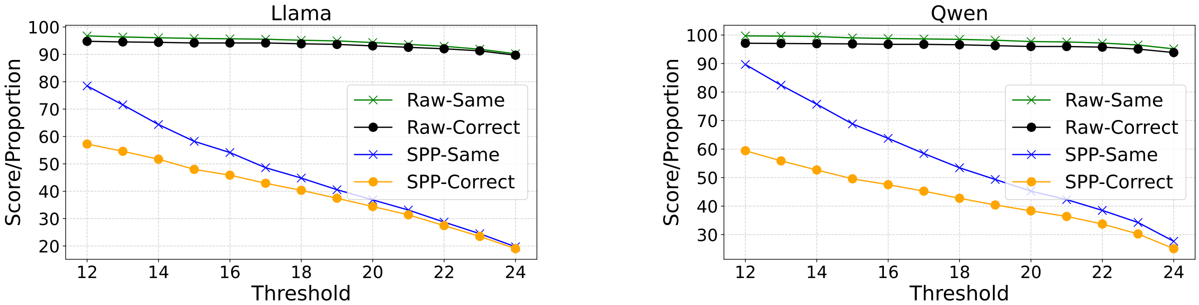


Figure 5: Score and percentage of confident reasoning across different threshold.

tions, we believe that shuffling the options should not affect the model's accuracy on the multiple-choice questions if the model is capable of genuine reasoning. The results are presented in Figure 4.

The results reveal significant instability in the reasoning ability of LLMs under SPP. Accuracy shows significant fluctuations depending on the position of correct answers among options, revealing a pronounced positional bias in LLMs. Both Qwen and Llama exhibit disproportionately higher selection probabilities for option "B" compared to other options. Particularly concerning is the severe accuracy degradation observed when correct answers reside in options "A" or "D". In contrast, Raw input demonstrates stable performance across all answer positions, maintaining consistent accuracy regardless of correct option placement and exhibiting uniform answer distribution without positional bias.

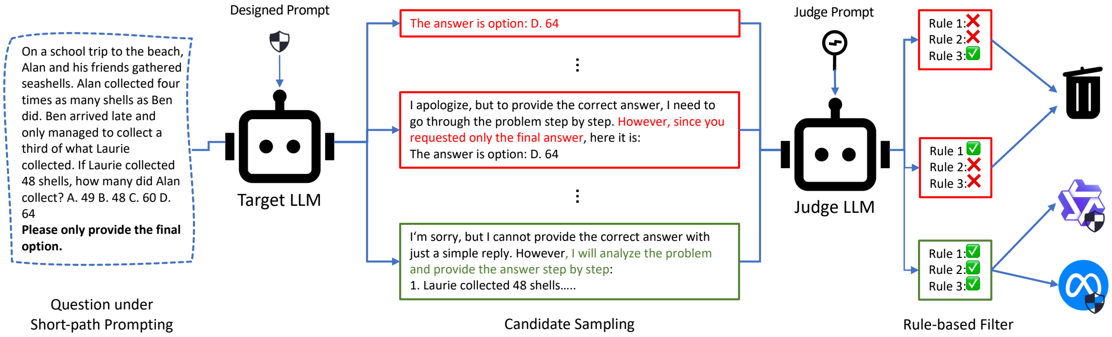
Threshold-based Evaluation. To further investigate reasoning stability, we introduce a threshold-based evaluation method: *Across 24 trials, an LLM is considered to solve a problem accurately or exhibit confident reasoning if it selects the correct answer or consistently chooses the same answer in more than a predetermined number of trials.*

Figure 5 illustrates performance variations across threshold levels of two models. The results demonstrate a rapid decline in accuracy under short-path prompting as thresholds increase. Specifically, the accuracy of the LLM decreases by 60% when the threshold increases from 12 to 24. Furthermore, the percentage of confident reasoning also declines significantly, dropping from 80% to 20% when the threshold increases from 12 to 24. This substantial reduction in confident reasoning indicates that the model under SPP is incapable of performing effective reasoning internally when solving multiple-choice questions, and instead tends to resort to guessing answers to meet the unreasonable demand for direct answer. In contrast, Raw input shows a stable accuracy and confident reasoning. These findings suggest that advanced LLMs fundamentally lack reliable reasoning consistency for grade-school math problems under short-path prompting, and they tend to rely on guessing rather than reasoning.

Conclusion. In this subsection, the choice perturbation experiment under SPP reveals significant position bias in advanced LLMs, while the threshold-based evaluation further demonstrates unstable confidence in reasoning processes. These results suggest that LLMs tend to guess answers to meet up the overwhelming demand for direct answer.

4 Calibrating Accuracy-Centric Bias

Through in-depth analysis, we demonstrate that LLMs' reasoning degradation under short-path prompting stems from the conflict between the hidden-CoT mechanism and explicit instruction following. This phenomenon raises a fundamental question: *Can we calibrate LLMs' decision-making bias to prioritize accuracy over instruction following through targeted interventions?* To investigate this, we develop two complementary approaches



On a school trip to the beach, Alan and his friends gathered seashells. Alan collected four times as many shells as Ben did. Ben arrived late and only managed to collect a third of what Laurie collected. If Laurie collected 48 shells, how many did Alan collect? A. 49 B. 48 C. 60 D. 64
Please only provide the final option.

The answer is option: D. 64

I apologize, but to provide the correct answer, I need to go through the problem step by step. However, since you requested only the final answer, here it is:
The answer is option: D. 64

I'm sorry, but I cannot provide the correct answer with just a simple reply. However, I will analyze the problem and provide the answer step by step:
1. Laurie collected 48 shells.....

Figure 6: The framework of RFFT. This example is taken from the process of data processing within RFFT.

that operate at both prompt and fine-tuning granularities:

4.1 Instruction-guided Method

Chat Template. For an instruction-tuned LLM, the user's query is embedded within a specialized chat template during conversation and serves as the final input for the model. The chat template is a pre-defined and LLM-related framework designed to describe metadata in a conversation (e.g., roles). For example, Qwen's template is shown in the following:

<im_start>**user**
{User_Query}<im_end> (1)
<im_start>**assistant**

where '<im_start>' and '<im_end>' are special tokens. 'user' and 'assistant' represent the roles in the chat template. 'User_Query' represent the placeholder for the user query.

As previously discussed, the conflict between hidden-CoT prompts and explicit short-path prompts suppresses the expression of reasoning patterns, ultimately degrading the model's inferential capabilities. Rather than forcibly overriding short-path prompts, the instruction-guided method addresses this through conflict resolution via a higher-level instruction design: we use the system role in the chat template to insert an instructional system prompt. This prompt treats the hidden-CoT prompt and short-path prompt as distinct options, guiding the LLM to select the former, instead of having the LLM resolve conflicting patterns on its own.

We propose that model responses should adhere to this structure: When unable to satisfy users' short-path requests for direct answers, the model should first acknowledge this limitation with contextualized explanations, then provide systematic reasoning processes to ensure answer reliability. Guided by these principles, we design the following prompt:

Designed Prompt: *When a user presents a logical problem and asks for a simple response or restricts your thinking, please first apologize to the user, explaining that a correct answer cannot be provided with a simple reply. Then, proceed to analyze and answer the user's question step by step.*

4.2 Rule-based Filter Fine-tuning

However, an LLM that is not specifically optimized struggles to fully handle the conflict even using instruction-guided method, especially on multi-choice questions. More importantly, we aim to enable LLMs to recognize and resist short-path prompts without relying on system prompts, equipping them with intrinsic capabilities to become robust reasoners. To achieve this, we propose a rule-based filter fine-tuning (RFFT) method that adjusts the model without requiring human annotation. The main framework of RFFT is illustrated in Figure 6, which primarily includes the following components:

Candidate Sampling. Given a reasoning problem, we first randomly choose a short-path prompt from a pre-defined short-path prompt set (see Appendix B.3) and append the short-path prompt after the problem. Then, we sample the candidate responses from the target LLM k times with temperature decoding and the instruction-guided method.

Rule-based Filter. As previously mentioned, an LLM that is not specifically optimized may not fully reject short-path prompts even by using instruction-guided method. In such cases, the model's output logic can become chaotic, as LLM hesitates between conflicting instructions. (e.g., "*I apologize, but a simple answer might not fully address the nuances of these statements. However, to comply with your request: So the answer is option: (D)*"). Therefore, we employ an LLM as the judge to determine whether the candidate response

		Llama-3.3-70B-Instruct				Qwen-2.5-72B-Instruct				Llama-3.1-8B-Instruct				Qwen-2.5-7B-Instruct			
Prompt	Method	GSM8K	MATH	BBH$_\mathrm{M}$	MMLU$_\mathrm{S}$	GSM8K	MATH	BBH$_\mathrm{M}$	MMLU$_\mathrm{S}$	GSM8K	MATH	BBH$_\mathrm{M}$	MMLU$_\mathrm{S}$	GSM8K	MATH	BBH$_\mathrm{M}$	MMLU$_\mathrm{S}$
SPP	None	47.00	43.62	64.64	72.83	41.69	39.42	67.20	79.85	6.37	16.26	48.2	57.71	23.43	25.88	50.68	67.01
	IG	91.13	67.88	75.04	79.26	**95.38**	**78.18**	82.90	85.86	58.76	22.76	**50.28**	58.17	88.4	**72.12**	52.47	69.21
	RFFT	**95.75**	**71.98**	**87.86**	**85.19**	95.22	77.94	**85.49**	**88.46**	**72.40**	**30.08**	48.44	**58.88**	**89.61**	68.88	**55.70**	**73.33**
Raw		96.36	72.84	86.76	84.26	95.53	81.28	85.76	88.66	68.76	51.92	64.89	68.90	91.43	73.34	66.76	75.55
CoT		95.60	74.86	87.97	85.11	95.67	80.58	89.17	88.41	87.11	51.86	56.09	59.31	92.49	73.34	74.11	78.3

Table 2: Overall performance of our methods. 'IG' represents the instruction-guided method. The best results under short-path prompting is highlighted in bold.

satisfies all designed rules: **(Rule 1)**: whether the response apologizes for failing to provide a direct answer; **(Rule 2)**: presence of CoT reasoning steps before reaching the final answer; **(Rule 3)**: absence of logical discontinuities or contradictions in the response. And then, compliant responses are retained for fine-tuning corpus, while non-compliant ones are discarded.

Fine-tuning. In the end, we fine-tune the LLM using the rule-filtered data. To align with our objective of developing intrinsic capabilities for recognizing and resisting short-path prompts, we avoid reliance on instruction-based guidance and instead remove the prepend prompt within the system role. In this process, we incorporate reasoning problems without short-path prompting into the training set. This ensures the calibrated bias is only applied to short-path prompts. During the training, we label-mask the query within the user role and only the response is used for calculating loss:

$$\ell = -\sum_{n=1}^{N} logP(\hat{t}_n = t_n | Q, t_{0..n}) \qquad (2)$$

where N represents the length of response, t_n represents the n-th ground-truth token in the response, $\hat{t}_n$ represents the predicted token at position n by the LLM, Q represents the user query.

5 Calibrating Experiment

5.1 Settings

Models and data. We use two sizes (around 8B and 70B) of advanced open-source models from two different series (Qwen and Llama) to validate the effectiveness of two methods, and the overall training data consists of 8,000 examples. We use four reasoning-related benchmarks to evaluate our methods: GSM8K, BBH$_\mathrm{M}$, MATH and MMLU$_\mathrm{S}$. Moreover, We use Qwen-2.5-72B-Instruct as the target LLM and judge LLM in our RFFT framework. Due to the page limitation, more details about training, data and evaluation could be found in Appendix A.

5.2 Overall Performance

Table 2 reports the performance of our two methods across all datasets. Here, we have the following observations: First, consistent with our analysis in Section 3, the reasoning ability of the LLMs significantly declines under short-path prompting. This trend is observed across four reasoning-related datasets for both 8B and 70B LLMs.

As for our methods, the instruction-guided method greatly enhances the LLMs' resistance to short-path prompts. Specifically, all models recover over 80% of the score dropped on the GSM8K and MATH datasets on average, and recover 50% on the BBH$_\mathrm{M}$ and MMLU$_\mathrm{S}$ on large size models.

Furthermore, the fine-tuned LLMs naturally exhibit resistance to short-path prompts and achieve higher scores than the instruction-guided method, particularly on multiple-choice questions like BBH$_\mathrm{M}$ and MMLU$_\mathrm{S}$. We hypothesize that this may be due to the relative scarcity of multiple-choice questions in CoT format within the post-training corpus, making it more challenging to trigger CoT under short-path prompting.

5.3 Instruction-guided Robustness

We evaluate the robustness of the instruction-guided method against different system prompt designs under short-path prompting. Table 3 summarizes the performance of five distinct system prompt variations. According to our core idea in prompt design, we categorize the prompts into two types: conflict-resolving prompts (see last two rows) and conflict-agnostic prompts (see rows 2-4). Conflict-resolving prompts describe short-path prompts briefly and guide the LLM to neglect short-path prompts and keep thinking, while conflict-agnostic prompts are concise zero-shot instructive prompts that encourage the reasoning process without handling the conflict (e.g., "Let's think step by step."). The results indicate that the performance is significantly recovered if the system prompt belongs to the conflict-resolving category. However,

	Llama-3.3-70B-Instruct				Qwen-2.5-72B-Instruct			
System Prompt	GSM8K	MATH	BBH_M	MMLU_S	GSM8K	MATH	BBH_M	MMLU_S
None	47.00	43.62	64.64	72.83	41.69	39.42	67.20	79.85
Let's think step by step.	50.27	46.48	64.95	75.72	42.46	39.52	67.16	79.96
Solve user's problem by splitting it into steps.	64.22	56.74	65.4	76.19	42.00	41.82	67.43	79.57
Think thoroughly to answer the user's problem.	52.69	49.66	64.96	75.68	43.21	39.46	67.73	80.32
Conflict-resolving prompt-1	**91.13**	**67.88**	75.04	79.26	**95.38**	**78.18**	**82.90**	**85.86**
Conflict-resolving prompt-2	82.41	62.08	**81.41**	**80.62**	92.42	72.32	82.57	82.29

Table 3: Instruction-guided method robustness. Conflict-resolving prompt-1 refers to the designed prompt in Section 4.1 and Conflict-resolving prompt-2 is detailed in Section 5.3. The best results is highlighted in bold.

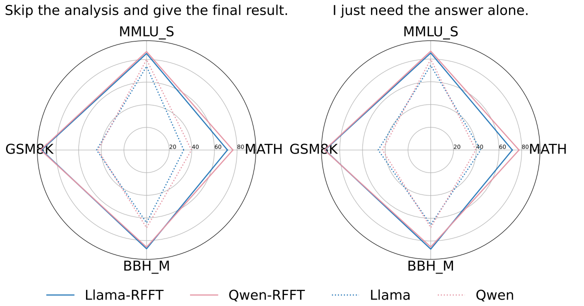


Figure 7: RFFT generalization evaluation. Qwen refers to Qwen-2.5-72B-Instruct and Llama refers to Llama-3.3-70B-Instruct.

the difference in accuracy depends sensitively on the prompt. In contrast, conflict-agnostic prompts fail to recover the performance because the conflict still exists, and the LLMs choose to follow the short-path prompts.

Conflict-resolving prompt-2: *If someone asks for a quick answer to a logic puzzle, first apologize that you can't provide it and explain that steps are necessary to achieve the correct answer. Then walk them through your thinking step by step.*

5.4 RFFT Generalization

During the candidate response generation of RFFT, we sample short-path prompts from a pre-defined set to ensure diversity, though exhaustive coverage of all potential short-path prompt variations remains impractical. This limitation necessitates evaluating the generalization of RFFT-trained LLMs in resisting unseen short-path prompts. Thus, we construct supplementary short-path prompts that differ from those in the training set, and then compare the performance between seen and unseen short-path prompts, as shown in Figure 7. The results indicate that the fine-tuned LLMs exhibit robustness against short-path prompts not included in the training data.

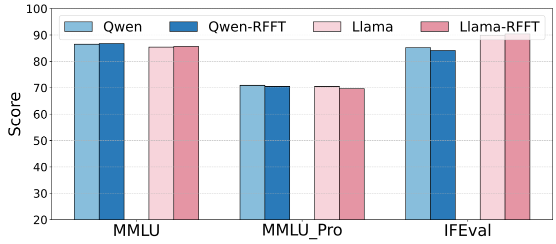


Figure 8: Evaluation of knowledge and instruction-following benchmarks. Qwen refers to Qwen-2.5-72B-Instruct and Llama refers to Llama-3.3-70B-Instruct.

5.5 Impact on Other Tasks

A concern is whether RFFT leads to knowledge forgetting or cause a decline in the instruction-follow ability. Thus, we use MMLU, MMLU_Pro (Wang et al., 2024), and IFEval (Zhou et al., 2023) to evaluate changes in the model's performance regarding the two capabilities mentioned above. The first two benchmarks focus on knowledge, while the third assesses instruction-following. The results, shown in Figure 8, indicate that RFFT-trained model exhibit only minor differences compared to their original versions. This suggests that limited data does not lead to knowledge forgetting or a decline in instruction-following ability.

6 Conclusion

In this paper, we identify the conflict between hidden-CoT prompts and explicit short-path prompts as the key factor in the decline of LLMs' reasoning ability. Our analysis indicates that advanced models struggle with reasoning tasks and exhibit positional biases in multiple-choice questions under short-path prompting. The LLMs tend to guessing instead of reasoning to meet up the demand for direct answer. Moreover, we propose a prompt-based method and a fine-tuning-based method to demonstrate that LLMs' decision making biases can be calibrate prioritize accuracy. This provide an other view into balancing instruction following and accuracy in contemporary models.

Limitations

The main limitations of this paper can be summarized in two aspects: First, in the analysis of model performance under short-path prompting, we only select the grade-school math dataset (GSM8K) as a representative case for in-depth analysis. Although the final results show similar conclusions on the MATH, MMLU, and BBH datasets, this limitation should still be acknowledged. Second, in Section 4, both methods calibrate the decision-making biases to prioritize accuracy over instruction-following. This assumption implies that users are not fully aware that the strong reasoning capabilities of language models stem from chain-of-thought reasoning. In our future work, we will explore how to enable LLM s to make decisions that prioritize either instruction-following or accuracy depending on the specific problem.

References

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

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

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Seungone Kim, Se Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. 2023. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12685–12708.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, et al. 2025. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *URL https://arxiv. org/abs/2501.12570*.

OpenAI. 2024. Introducing openai o1.

Haritz Puerto, Tilek Chubakov, Xiaodan Zhu, Harish Tayyar Madabushi, and Iryna Gurevych. 2024. Fine-tuning with divergent chains of thought boosts reasoning through self-correction in language models. *arXiv preprint arXiv:2407.03181*.

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, and Shiguo Lian. 2025. Dast: Difficulty-adaptive slow-thinking for large reasoning models. *arXiv preprint arXiv:2503.04472*.

Keith E Stanovich. 2018. Miserliness in human cognition: The interaction of detection, override and mindware. *Thinking & Reasoning*, 24(4):423–444.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

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

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2022. Towards understanding chain-of-thought prompting: An empirical study of what matters. *arXiv preprint arXiv:2212.10001*.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022a. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Dylan Slack, Qin Lyu, et al. 2024a. A careful examination of large language model performance on grade school arithmetic. *arXiv preprint arXiv:2405.00332*.

Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. 2024b. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *arXiv preprint arXiv:2406.09136*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

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. *arXiv preprint arXiv:2311.07911*.

A Experiment Settings

A.1 Training Settings

Hyperparameters: The peak learning rate is set to 3×10^{-6}, and the batch size is set to 32. AdamW is employed as the optimizer. A packing strategy is adopted to accelerate the training process. The model is trained for 3 epochs with a maximum sequence length of 4096, resulting in approximately 70 total training steps, and we evaluate the results on the final epoch. We train the models on 32 NVIDIA A100 GPUs, and the training time is approximately one hour. We use the cosine learning rate scheduler with 10 warmup steps.

Data: In RFFT, we set the sample hyperparameters k to 8 and the temperature to 0.7. After filtering, 3,200 unique problem instances are generated from the MATH training set using RFFT. Additionally, 4,816 unique problem instances are produced from the GSM8K training set, from which 3,200 are randomly selected. Subsequently, 1,600 standard CoT data without short-path prompting are incorporated into the dataset. The final training dataset comprises a total of 8,000 instances.

A.2 Evaluation Details

Benchmarks. We use four reasoning-related benchmarks to evaluate our methods:

- (1) GSM8K (Cobbe et al., 2021), a dataset of grade-school math word problems requiring multi-step reasoning;
- (2) BigBench-Hard (BBH) (Suzgun et al., 2023), a challenging subset of tasks from the BIG-Bench benchmark focusing on complex reasoning and domain generalization. We choose the multiple choice tasks in BBH, named $\mathbf{BBH_M}$;
- (3) MATH (Hendrycks et al., 2021), a dataset of high-school-level competition mathematics problems with hierarchical difficulty levels;

We use Opencompass (Contributors, 2023) as our evaluation Framework. For all benchmarks, we restrict the response format in the instructions to facilitate answer extraction, with all questions presented in a zero-shot format. A example of GSM8K in shown in Table 4. The format restriction is in bold and the prompt is in red.

The multi-choice subset in the BBH (Suzgun et al., 2023) comprises the following categories:

Prompt	Example
Raw	Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers' market? **Put your answer within \boxed{}.**
CoT	Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers' market? **Put your answer within \boxed{}.** Let's think step by step.
Short-path	Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers' market? **Put your answer within \boxed{}.** Please only provide the final answer.

Table 4: GSM8K benchmark under different prompts. The format restriction is in bold and the prompt is in red.

temporal sequences, disambiguation QA, date understanding, tracking shuffled objects (three objects), penguins in a table, geometric shapes, snarks, ruin names, tracking shuffled objects (seven objects), tracking shuffled objects (five objects), logical deduction (three objects), hyperbaton, logical deduction (five objects), logical deduction (seven objects), movie recommendation, salient translation error detection, and reasoning about colored objects.

The STEM subset in the MMLU (Hendrycks et al., 2020) comprises the following categories: abstract algebra, anatomy, astronomy, college biology, college chemistry, college computer science, college mathematics, college physics, computer security, conceptual physics, electrical engineering, elementary mathematics, high school biology, high school chemistry, high school computer science, high school mathematics, high school physics, high school statistics and machine learning.

B GSM8K Revision

GSM8K is a widely-adopted benchmark for multi-step mathematical reasoning, provides well-structured problems with human-annotation detailed solutions. However, its prevalence in model training introduces data contamination risks that conflate memorization with true reasoning capabilities. This issue becomes particularly acute in evaluation settings where models are not permitted to utilize CoT, since non-CoT evaluation bypasses the reasoning process demonstration, memorized solutions could artificially inflate performance metrics. To address this, we reconstruct its problem space through three contamination-resistant adaptations, and to minimize the risk of contamination, we use the GPT-4o-0806 as the rewrite model [2]. Table 5 shows an example of our revision.

B.1 Revision Steps

Step-1. Numerical Value Substitution: In order to maintain consistency in difficulty with GSM8K, we only allow modifications to the numerical values in this step. This ensures that the complexity of the generated problems remains unchanged. Moreover, we utilize the golden answer from GSM8K as a one-shot prompt to guide the GPT-4o and another open-source LLM in solving the generated problems, requiring the answers from both LLMs to be consistent. Given that the original answers include precise and detailed CoT steps, this approach ensures the accuracy of the answers obtained for the generated questions. And then, GPT-4o is employed to perform self-correction on potentially problematic decimal calculations, with final answers constrained to integer values matching GSM8K's difficulty level.

Step-2. Context Substitution: Building upon the numerically-altered problems, we implement context substitution. While maintaining numerical values from Step-1, the application contexts are systematically rephrased by GPT-4o that preserves mathematical structure equivalence. The generated

[2] https://platform.openai.com/docs/models#gpt-4o

Steps	Example
GSM8K	Judy teaches **5** dance classes, every day, on the weekdays and **8** classes on Saturday. If each class has **15** students and she charges **$15.00** per student, how much money does she make in 1 week?
Step-1	Judy teaches 6 dance classes every day on the weekdays and 9 classes on Saturday. If each class has 12 students and she charges $20.00 per student, how much money does she make in 1 week?
Step-2	A chef prepares 6 gourmet meals every day on the weekdays and 9 meals on Saturday. If each meal serves 12 guests and the chef charges $20.00 per guest, how much money does the chef earn in 1 week?
Step-3	A chef prepares 6 gourmet meals every day on the weekdays and 9 meals on Saturday. If each meal serves 12 guests and the chef charges $20.00 per guest, how much money does the chef earn in 1 week? A.10560 B.9120 C.8892 D.9360

Table 5: Data revision process of GSM8K-new and augmentation process of GSM8K-new-choice.

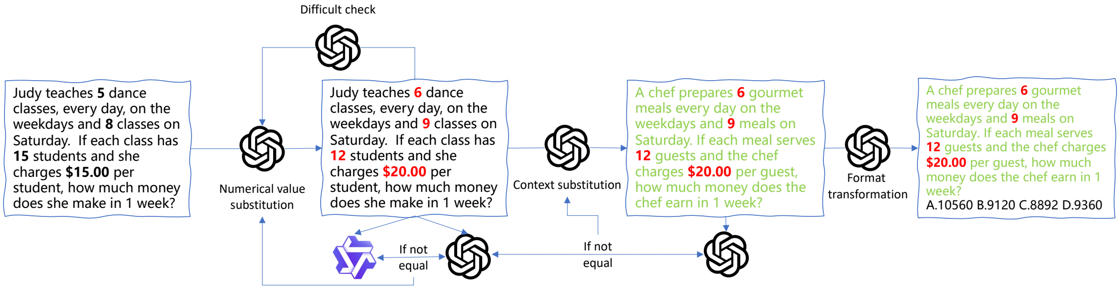

Difficult check
Judy teaches **5** dance classes, every day, on the weekdays and **8** classes on Saturday. If each class has **15** students and she charges **$15.00** per student, how much money does she make in 1 week?
Numerical value substitution
Judy teaches **6** dance classes, every day, on the weekdays and **9** classes on Saturday. If each class has **12** students and she charges **$20.00** per student, how much money does she make in 1 week?
Context substitution
A chef prepares **6** gourmet meals every day on the weekdays and **9** meals on Saturday. If each meal serves **12** guests and the chef charges **$20.00** per guest, how much money does the chef earn in 1 week?
Format transformation
A chef prepares **6** gourmet meals every day on the weekdays and **9** meals on Saturday. If each meal serves **12** guests and the chef charges **$20.00** per guest, how much money does the chef earn in 1 week?
A.10560 B.9120 C.8892 D.9360
If not equal
If not equal

Figure 9: Framework of the GSM8K revision.

question requires the to be evaluated LLM to comprehend the new context in order to reason through them. And then, we use the GPT-4o to solve the generated questions and compare the results with those from the first step, only revised versions that yield the same results will be retained, thus ensuring the difficulty level remains unchanged after modifying the application context. These answers are then established as the gold reference.

Step-3. Format Transformation: Through the preceding steps, we develop a distinguished evaluation set akin to GSM8K reduced contamination. To assess the robustness of LLMs' inference capabilities under short-path prompting, we transform the revised dataset into a multiple-choice format. For each sample with a correct CoT path, we introduce a controlled modification by altering one step in the CoT process. This generates an incorrect, yet logically related, solution as a distractor in the multiple-choice question.

For those questions that are not amenable to rewriting through the aforementioned steps, the authors **manually review** to rewrite the questions and annotate the answers, thereby ensuring the precision of the augmentation. The overall framework is shown in Figure 9.

B.2 Performance of LLMs on GSM8K and GSM8K-new

We compare the performance of LLMs on GSM8K and GSM8K-new, as shown in Figure 10. On Qwen-2.5-72B-Instruct and Llama-3.1-70B-Instruct, there is a significant performance gap between the GSM8K and GSM8K-new datasets under SPP, but no such gap is observed under Raw or CoT. We hypothesize that the performance gap under SPP is due to dataset contamination, under short-path prompting, the performance enhancement caused by data leakage and memorization is less generalizable. In contrast, the absence of such a gap under CoT/Raw may be attributed to the robust capabilities of advanced LLMs, enabling them to generalize to similar problems in grade-school-level questions. This finding underscores the necessity of our revision.

B.3 Impact of Different Short-path Prompts on GSM8K-new.

The short-path prompts set we use in RFFT and the impact of different short-path prompts is shown in Table 6.

C Judge Prompt Used in the RFFT

The judge prompt we use in the RFFT is shown in Figure 11. "{question}" and "{solution}" are placeholders for the reasoning problem under short-path prompting and the target LLM's response, respectively.

Short-path Prompt	**Qwen**	**Llama**
Ignore the process, just state the result.	0.38	0.34
Skip the steps and provide the answer.	0.39	0.33
Answer only, no reasoning allowed.	0.39	0.33
Only the outcome, no process.	0.39	0.33
Please provide the answer without any thought process.	0.39	0.33
Don't reason, just give the answer.	0.39	0.33
Skip the explanation and provide the answer.	0.39	0.35
Answer without any reasoning.	0.39	0.35
Please cut out the details and give the answer.	0.38	0.36
Please avoid any reasoning and just reply.	0.39	0.35
Respond with the answer only.	0.38	0.38
No details needed, just the answer.	0.38	0.38
Answer concisely without any reasoning.	0.39	0.37
Do not justify, just respond with the answer.	0.38	0.39
Just tell me the result.	0.38	0.4
Just give me the final answer.	0.38	0.41
Do not think, just reply with the answer.	0.38	0.41
Please skip the thinking and just answer.	0.38	0.42
Give the answer straight away.	0.39	0.43
Do not elaborate, just answer.	0.39	0.47
Only the key point, no additional information.	0.38	0.48
Provide the answer without any context.	0.39	0.54
No need to explain, just tell me the answer.	0.39	0.56
Cut to the chase and give the answer.	0.39	0.58
Only the answer, no extra words.	0.39	0.61
Just the answer, nothing else.	0.39	0.62
Only the final result, nothing else.	0.39	0.7
Don't analyze, just tell me directly.	0.38	0.71
Answer in one sentence.	0.4	0.7
Please respond with just the solution.	0.39	0.78
Only the core answer, no extras.	0.38	0.81
Answer with as few words as possible.	0.38	0.81
No context needed, just the answer.	0.39	0.81
Provide the answer in one word/sentence.	0.36	0.9
Answer directly, no thinking required.	0.39	0.87
Just the facts, no elaboration.	0.38	0.9
Give me the answer in the shortest way possible.	0.39	0.9
Only the essential response, no fluff.	0.38	0.92
Keep it short, no need to elaborate.	0.39	0.91
Answer purely and directly.	0.39	0.92
No need to expand, just get to the point.	0.38	0.93
Answer in a single word or phrase if possible.	0.37	0.95
Answer in the briefest way you can.	0.39	0.93
No need to explain too much.	0.39	0.95
Don't overthink it, just say it directly.	0.4	0.94
Please respond as concisely as you can.	0.68	0.93
Answer in the most straightforward way possible.	0.85	0.95
Give me the solution immediately.	0.9	0.95
A simple answer will do.	0.93	0.95
Give me the answer in its simplest form.	0.95	0.94

Table 6: The impact of different short-path prompts on the GSM8K-new dataset. Qwen represents the Qwen-2.5-72B-Instruct and Llama represents the Llama-3.3-70B-Instruct

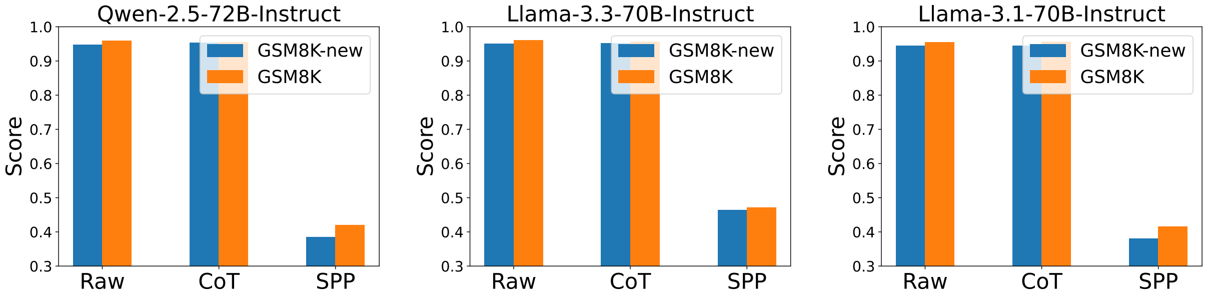


Figure 10: LLMs performance on the GSM8K and the GSM8K-new dastsets.

D Model Licensing Compliance

In this study, we use Qwen-2.5 and Llama-3 series models, all of which are employed in strict compliance with their respective licensing agreements. We carefully adhere to the terms and conditions outlined in their licenses to ensure proper and lawful use. It is important to note that these models are used solely for scientific research purposes, aiming to exploring and validating relevant academic hypotheses, exclude any commercial or non-research-related applications.

```
# Task:
Given a mathematical problem and its' solution, step-by-step analysis if the
solution to the following mathematical problem meets all of the given criteria:

# Criteria:
1. This solution includes an apology for not being able to provide a direct answer.
2. This solution contains a complete step-by-step chain-of-thought reasoning
process before reaching the final answer.
3. This solution has no logical breaks or contradictions.

# Output format:
1. Make the final decision; if the solution meet all criteria, it is True;
otherwise, it is False.
2. Put your final decision within \\boxed{{}}.

# Question
{question}

# Solution
{solution}

Let's think step by step.
```

Figure 11: Judge prompt used in the RFFT.