Recitation over Reasoning: How Cutting-Edge Language Models Can Fail on Elementary School-Level Reasoning Problems?

Kai Yan[1,2], Yufei Xu[1], Zhengyin Du[1], Xuesong Yao[1], Zheyu Wang[1], Xiaowen Guo[1], Jiecao Chen[1]

[1]ByteDance Seed, [2]University of Illinois Urbana-Champaign

Correspondence: kaiyan3@illinois.edu

Abstract

The rapid escalation from elementary school-level to frontier problems of the difficulty for LLM benchmarks in recent years seems to bring us close enough to the "last exam" for LLMs to surpass humanity. However, is the LLMs' remarkable reasoning ability indeed coming from true intelligence by human standards, or are they actually reciting solutions witnessed during training at an Internet level? To study this problem, we propose RoR-Bench, a novel, multi-modal benchmark for detecting LLM's recitation behavior when asked simple reasoning problems but with conditions subtly shifted, and conduct empirical analysis on our benchmark. Surprisingly, we found existing cutting-edge LLMs unanimously exhibits extremely severe recitation behavior; by changing one phrase in the condition, top models such as OpenAI-o1 and DeepSeek-R1 can suffer 60% performance loss on elementary school-level arithmetic and reasoning problems. Such findings are a wake-up call to the LLM community that compels us to reevaluate the true intelligence level of cutting-edge LLMs.

1 Introduction

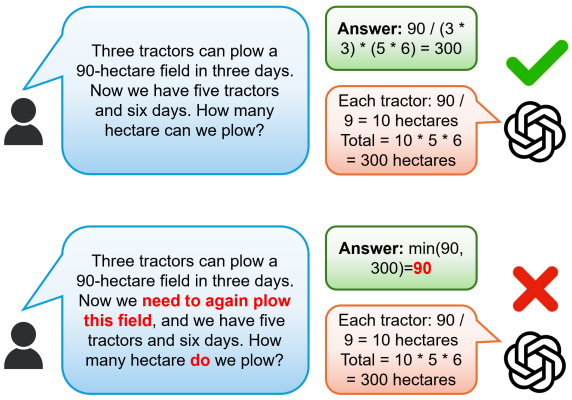
Since the advent of GPT-3 (Brown et al., 2020) and ChatGPT (Ouyang et al., 2022), Large Language Models (LLMs) have sparked an unprecedented revolution of research paradigm and pushed forward task frontiers in almost every field of Artificial Intelligence (AI) (Qin et al., 2024; Wang et al., 2024c; Ma et al., 2024; Zhou et al., 2024a), as well as the whole science community (Zhang et al., 2023; Abramson et al., 2024; Zhang et al., 2024b). By improving the training data (Liu et al., 2024c; Villalobos et al., 2024a), scaling up parameter size (Kaplan et al., 2020; Zhang et al., 2024a), and incorporating long thinking process (Jaech et al., 2024; Guo et al., 2025), LLMs finally come close enough to the "last exam" (Phan et al., 2025) for Artificial General Intelligence (AGI) to surpass humanity.

Despite the huge success of LLMs, however, researchers have not fully understood the underlying mechanism for LLM's "emerging" (Wei et al., 2022a; Arora and Goyal, 2023) intelligence via current engineering (Dubey et al., 2024; Guo et al., 2025) advances. While there have been many efforts from the researchers to theoretically guarantee LLMs' intelligence level (Akyürek et al., 2023; Bhargava et al., 2023; Zekri et al., 2024) and rapid escalations in the difficulty of solvable math and science competition problems from elementary school (Cobbe et al., 2021) to research level (Phan et al., 2025), there have also been recent concerns on LLMs are still struggling with real-world problems (Wang et al., 2024b), even those which are not so difficult for humans (Mirzadeh et al., 2025; Zhou et al., 2024b). Such works indicates that a cloud still exists upon the great monument of reasoning for LLMs, which questions the actual intelligence level of LLMs in reasoning problems and again brought the concern of "stochastic parrots" (Bender et al., 2021) back to the table.

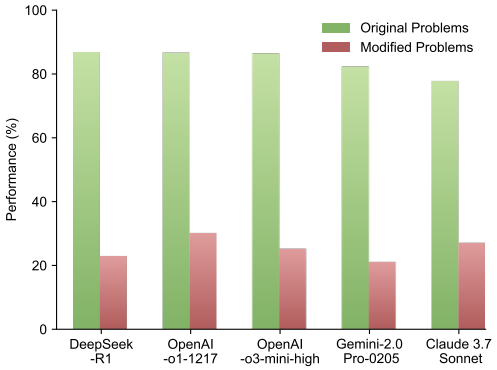
To better illustrate the existence of such cloud, here we examine a simple, GSM-8K (Cobbe et al., 2021) level math problem as an example in Fig. 1. Despite the simplicity of the problem, however, cutting-edge models such as OpenAI o1 (Jaech et al., 2024) fails to solve such a problem; they simply *recite* the normal problem-solving paradigm of the problem, without carefully doing the *reasoning* and checking the subtle condition shift in the problem. With such phenomenon, we must ask the following tough question: *Can the LLMs really solve simple reasoning problems, instead of simply reciting solution templates?*

To find out the answer for this problem, in this work we propose RoR-Bench, a novel, multi-modal Chinese benchmark to detect the issue of **R**ecitation **o**ver **R**easoning for cutting-edge LLMs on simple

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 272–291



a) Subtly changed condition



b) Performance loss due to recitation

Figure 1: Panel a) shows an example of how current cutting-edge LLMs, such as OpenAI-o1-1217 (Jaech et al., 2024), OpenAI-o3 (OpenAI, 2025c) and Gemini-Pro 2.5 (Comanici et al., 2025) fails to address an elementary school-level math problem (see Appendix C.2 for links to the response) with subtle but crucial condition change, simply *reciting* existing solution template; panel b) shows the performance loss of cutting-edge LLMs due to reciting solution templates regardless of shifted conditions on our benchmark, which is a staggering $\sim 60\%$ score gap on simple reasoning and math problems.

reasoning problems, with 158 pairs of text problems and 57 pairs of image problems curated by humans; each pair consists of a simple, mostly elementary school-level reasoning problem and its variant with subtle but crucial condition shifts. We find that *all* cutting-edge LLM models have severe problem in reciting solutions instead of actually doing the reasoning, causing an accuracy loss that often exceeds 60%. Such phenomenon is particularly astounding on problems with no solutions; many cutting-edge LLMs, such as DeepSeek-R1, can even only recognize $< 10\%$ cases as unsolvable. We explored initial solutions for mitigating the issue: adding notice prompts and providing subtly modified problems as few-shots. Although these solutions can mitigate the performance drop slightly, they are far from satisfactory and a more complete solution is still yet to be proposed.

Our key contributions can be summarized as follows: 1) We shed light on an important and severe issue for current cutting-edge LLMs, which is that LLMs are *reciting* problem-solving paradigms instead of actually conducting problem-specific *reasoning* even for simple reasoning problems; 2) We propose RoR-Bench, a novel, multimodal benchmark for detecting LLM's recitation behavior when solving simple reasoning problems which poses a great challenge for many state-of-the-art LLMs; and 3) We conduct several empirical analysis on our benchmark and examined initial solutions to the problem (See Sec. 4 for details).

2 Related Work

LLM benchmarks. The rapid advancement of LLMs in recent years (Ouyang et al., 2022; Hurst et al., 2024; Jaech et al., 2024) has created great needs for thorough LLM evaluation; some major directions include general knowledge (Hendrycks et al., 2021a; Wang et al., 2024d; Rein et al., 2024), math (Cobbe et al., 2021; Hendrycks et al., 2021b; Glazer et al., 2024), coding (Chen et al., 2021; Liu et al., 2023b; Jimenez et al., 2024), instruction following (Bai et al., 2024), reasoning (Suzgun et al., 2023; Srivastava et al., 2023; Kazemi et al., 2025), long-context (Ma et al., 2025; Yan et al., 2025), agent (Yao et al., 2022; Liu et al., 2024b), planning (Valmeekam et al., 2023; Zheng et al., 2024b) and function calls (Yan et al., 2024). While the difficulty of benchmarks escalates quickly (e.g. from GSM8K (Cobbe et al., 2021) to MATH (Hendrycks et al., 2021b) and frontiers (Glazer et al., 2024)), however, most of them are STEM [1] problems that can often be addressed by applying particular solution patterns (Yang et al., 2024b), i.e., *reciting* solution templates. Thus, remarkable as the progresses on such types of benchmarks are, the true intelligence level of LLMs is still worth discussing.

LLM robustness. While LLM achieves tremendous success, there has been persisting concerns about the limited robustness of LLMs (Zhou et al., 2024b; Xie, 2024). For example, LLMs have been well known for making mistakes in comparing 9.8

[1] Science, Technology, Engineering and Mathematics.

and 9.11 (Xie, 2024) and counting "r"s in "strawberry" (Xu and Ma, 2025); there have also been many works that question LLM's robustness when confronted with out-of-distribution data (Ren et al., 2023; Yuan et al., 2023), incorrect/incomplete commands (Yan et al., 2024; Zhao et al., 2025), complex calculations (Zhou et al., 2024b), symbolic relations (Mirzadeh et al., 2025), and order of choices in multiple choice questions (Zheng et al., 2024a). Recently, the vulnerability of LLM reasoning under perturbed conditions has attracted the researcher's attention, for example, LLM's math ability under conditions with irrelevant context (Shi et al., 2023) or extended reasoning steps (Zhou et al., 2025). The most similar works to ours are done by Zhao et al. (2024) and Huang et al. (2025a), both of which include math problems with subtly but fundamentally changed conditions. However, both works do not contain multi-modal problems, and their original problems without trap contains only math problems with more complex knowledge (e.g. number theory or precalculus). On the contrary, our benchmark contains more reasoning problems with less prior knowledge, and shows larger gap between original and modified problems.

Multi-modal LLMs. As the inherent limit of languages (Huang et al., 2023) and corpus depletion (Villalobos et al., 2024b) quickly becomes a major obstacle for AGI, researchers quickly turn to other modalities, such as vision (Caffagni et al., 2024) and speech/audio (Li et al., 2024; Fathullah et al., 2024) for extra input sources. As humans take the most information from vision (Hutmacher, 2019), Vision Language Models (VLMs) such as OpenFlamingo (Awadalla et al., 2023), Llava (Liu et al., 2023a, 2024a), Qwen-VL (Bai et al., 2023, 2025) and GPT-4v/-4o (OpenAI, 2023; Hurst et al., 2024) have become the prevailing paradigm for multimodal LLMs, and made unique progress on multiple areas beyond LLMs, such as robotics (Wang et al., 2024a; Duan et al., 2025) and autonomous driving (Tian et al., 2024; Xu et al., 2024; You et al., 2024). VLMs are also evaluated by part of our benchmark, and they exhibit the same recitation problem. There are some recent works that provide explanations for such issue. For example, some argue that the problem comes from *spurious correlation* (Varma et al., 2024; Hosseini et al., 2025), where correlation between often-tested notions (e.g. famous optical illusions) and modified inputs becomes part of the source for improper recitation, and reports similar issues to our findings (Qiu et al., 2024); others argue that the problem comes from *inefficient decoding* (Huang et al., 2025b) or *memorization* (Zou et al., 2025), the latter of which resembles our argument.

3 RoR-Bench

In this section, we will introduce our proposed benchmark, RoR-Bench. RoR-Bench is a multimodal, question-answering Chinese benchmark consisting of *pairs* of problems, which are the *original* problems and the *modified* problems. The original problems are chosen such that 1) cutting-edge LLMs can well-address, and 2) are mostly classic puzzles in books and homework. The modified problems are created such that they look very similar to original ones, but with key condition changed and have completely different solution paradigms and answers. Fig. 2 provides an example for text and image problems in RoR-Bench.

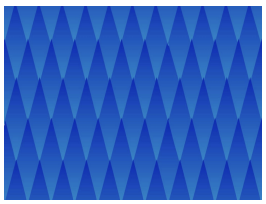
3.1 Dataset Curation

We asked 17 human annotators (all native speakers to ensure dataset quality) to collect simple reasoning problems from the Internet, mostly based on brain teaser collections in online blogs and sets of reasoning puzzles for children. Such problems become the original problems for our benchmarks. Then, we ask the annotators to modify the problems with the following instructions:

- **Different solution paradigm:** The idea for addressing the modified problems must be completely different from the original problem. Simply changing numbers in the conditions (e.g. from 30km/h to 60km/h) is not allowed, as LLMs can well generalize to different figures in the condition. The modified problem is often simpler and more straightforward; for example, the modification can be "how to discriminate the two items in a *black* box" to "how to discriminate the two items in a *transparent* box".
- **No ambiguity:** The modified problem must be rigorous, and only have one reasonable answer. For example, "how to cut a triangle cake into 4 pieces (without any restrictions)" is too open to judge its correctness; "running competition in space (such that one cannot hear the starting gun)" is too ambiguous as humans cannot normally run in space, and LLMs may assume additional conditions such as the event is happening inside a space station. Note, both the collected original problems and the modified problems are intended

Original problem: 某警官发现前方100米处有一匪徒。警官赶紧以每秒5米的速度追。已知小偷的跑步速度为3米/秒，多少秒后警官可以追上这个匪徒？ (A police officer spotted a thief 100 meters ahead of him. The officer started chasing the thief at 5 m/s. The thief runs at 3 m/s. How long does it take for the officer to catch the thief?)
Original answer: $100/(5-3) = 50$s.

Modified problem: 某警官发现前方100米处有一匪徒，匪徒没有发现警官。警官赶紧以每秒5米的速度追，已知小偷的跑步速度为3米/秒，多少秒后警官可以追上这个匪徒？ (A police officer spotted a thief 100 meters ahead of him, but the thief did not notice the officer. The officer started chasing the thief at 5 m/s. The thief can run at 3 m/s. How long does it take for the officer to catch the thief?)
Modified answer: $100/5 = 20$s.



Original problem: 这张图由多个同样的渐变菱形构成，它们整体看起来从上而下越来越暗，对吗？ (This image is composed of multiple identical gradient diamonds, and overall, they appear to get darker from top to bottom, right?)
Original answer: 是的（马赫带效应） (Yes, it is a Mach band.)

Modified problem: 这张图由多个同样的渐变菱形构成，它们每个看起来从上而下越来越暗，对吗？ (This image is composed of multiple identical gradient diamonds, and each of them appear to get darker from top to bottom, right?)
Modified answer: 不对，是自下而上 (No, it is from bottom to top.)

Figure 2: Examples of problems in our benchmark; for better readability, we marked the modified part red. Despite that we build a Chinese benchmark, OpenAI-o1-1217 (Jaech et al., 2024), OpenAI-o3 (OpenAI, 2025c) and Gemini 2.5 Pro (Comanici et al., 2025) all fail with our English translation for these examples. See Appendix B.2 for another example and Appendix C.3 for links to experiment records on the English translation.

to be easy to solve, with the latter having unconventional conditions.

- **As less verbal modification as possible:** The modified problem should look verbally similar to the original problem, so as to better examine whether LLMs are actually reasoning with the condition, or simply reciting solution templates from similar problems.

Each pair of original and modified problems will then be scrutinized by one of the 6 moderators (or multiple moderators in borderline cases), to ensure that the problems have no error or duplication, do not contain any identifying or offensive content, and satisfy the principles above.

3.2 Dataset Statistics

RoR-Bench consists of a total of 215 pairs of problems, with 158 pairs of text problems and 57 pairs of image problems. Such size is comparable to the most related works, e.g. MATH-Perturb (Huang et al., 2025a) with 279 pairs of problems, and MathTrap (Zhao et al., 2024) with 105 original public triplet of problems[2].

The image problems are all related to the property of the figure, while the text problem consists of 78 math problems (57 arithmetic, 11 geometry and 10 probability / combinatorics) and 80 reasoning problems (38 optimization, 10 commonsense, 27 deduction and 5 game theory). See Fig. 3 for an illustration of the ratio for each type of problems. To ensure the simplicity of the problems, we curate the data such that all text inputs are less than 200 characters, and each image problem only consists of a single image.

In particular, to better evaluate the LLMs' robustness against unusual answers, we curate 32 text problems and 2 image problems with no solution (e.g., finding the ball with different weights using an inaccurate balance, or the smoke direction of an electric locomotive on a windy day). We also provide several trick text problems with the problem to answer unrelated to the condition (e.g. asking the price of apples given the price of pears). [3]

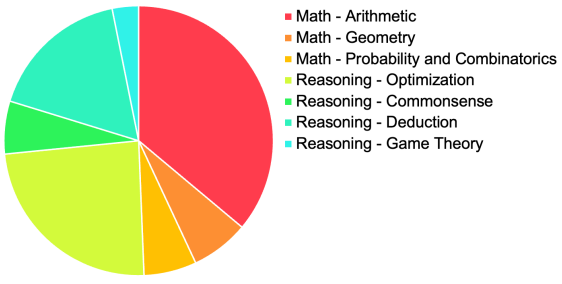


Figure 3: An illustration of the types of the problem of our dataset, which covers a variety of reasoning problems; we double-checked the problems to ensure the low difficulty of the original ones.

[2]The rest 895 are paraphrases by GPT.

[3]We intentionally limit the number of such type of problems, as they can be potentially interpreted as typos.

4 Evaluations

In this section, we introduce the main results and empirical analysis for cutting-edge LLMs on RoR-Bench. In particular, we aim to address the following questions: 1) Does the model really conduct reasoning over subtly modified conditions, or are they simply reciting existing solution paradigms to similar problems? If it is the latter, is it because the models view those changed conditions as typos (Sec. 4.1, Sec. 4.2)? 2) Will simple fixes, such as using original problems as 1-shot, address the possible problem of recitation over reasoning (Sec. 4.3)? 3) How well can the LLMs deal with ill-posed problems, especially those with no solution (Sec. 4.4)? 4) In general, why does the recitation phenomenon happen (Sec. 4.5)?

4.1 Text-based Problems

Evaluation. We evaluate 23 cutting-edge LLMs, which include: **1) State-of-the-art Models with long thinking (Chain-of-Thought, CoT (Wei et al., 2022b)) process:** DeepSeek-R1 (Guo et al., 2025), OpenAI-o1-1217 (Jaech et al., 2024), OpenAI-o3-mini-high (OpenAI, 2025c), Gemini-2.0 Flash-0121 (Kavukcuoglu, 2025), Claude 3.7 Sonnet (Anthropic, 2025) and QwQ-32B-Preview (Team, 2025b); **2) Flagship LLMs without long thinking process:** Hunyuan Turbo-S (Tencent, 2025), Ernie-4.5 (Inc., 2025), Gemini-2.0 Pro-0205, GPT-4.5-Preview (OpenAI, 2025b), Qwen-max-0125 (Team, 2025a), GPT-4o-1120 (Hurst et al., 2024), DeepSeek-v3 (Liu et al., 2024c), Minimax-Text-01 (Li et al., 2025), Claude 3.5 Sonnet (Anthropic, 2024), GLM-4-Plus (GLM et al., 2024), StepFun Step-2-16k, Yi-lightning (Wake et al., 2024), Mistral-Large-2 (team, 2024), GPT-4o-mini-0718, and Nova-Pro (Intelligence, 2024); and 3) small LLMs: Qwen-2.5-14B-Instruct (Yang et al., 2024a) and Qwen-2.5-7B-Instruct.

As the answer to our question can be versatile with sometimes no solution, we do not adopt exact match as the metric. Instead, we use GPT-4o-1120 as the judge, which gives a binary (0/1) score (see Appendix B.1 for prompts) for LLM-generated answers. Each model is tested for 5 times with temperature 0.7, following default by OpenAI API reference document (OpenAI, 2025a); we choose non-greedy decoding to test more rollouts and better differentiates the models' perormance. We also report best-of-5 and greedy decoding results in Appendix C.6 and C.7 respectively). We use the average score (by GPT-4o-1120) as the metric over 5 trials and 158 problems, normalized to $0-100$; the higher score is the better. See Appendix B.1 for an analysis on LLM judge's reliability.

Results. Tab. 1 shows the result for all LLMs tested on RoR-Bench with original and modified problems, which shows a staggering $> 50\%$ average performance decrease from scores on the original problems to the modified problems, and often $> 60\%$ performance decrease for the best models such as DeepSeek-R1 and OpenAI-o3-mini-high. The best-of-5 [4] performance of all LLMs also drop significantly (See Appendix C.6 for details), which indicates that such recitation issue is hard to be fixed simply by aligning techniques such as Reinforcement Learning (RL). Also, *long thinking process does not seem to achieve better performance.* On modified problems, models such as DeepSeek-R1, OpenAI-o1-1217 and OpenAI-o3-mini-high works no better than those without long thinking process, such as GPT-4.5 Preview and Claude 3.7 Sonnet, despite having higher performance on original problems; also, Gemini-2.0 Flash-0121 and Claude 3.7 Sonnet works similarly on modified problems either with or without long thinking process. In spite of this, the performance on original problems and modified problems are positively related (Pearson correlation coefficient (Pearson, 1895) $\rho \approx 0.72$), which indicates that the performance on modified problems are generally related to the base ability of the models.

4.1.1 Reliability analysis of the LLM judge

While using LLM judges can potentially introduce bias, we select LLM judges for two reasons:

- A large portion of problems in our benchmark (especially the non-math ones) are inherently very hard to be verified by rules (e.g., "How can A ensure victory in a game?" or "How can we quickly find some object in a set of objects?").
- The judge also considers whether the LLM recognizes the trap in the modified questions, which is important on deciding whether the model is reciting solution paradigms. As most

[4] Under the best-of-5 (Bo5) metric, the model is considered to get a score of 1 if at least one of the 5 trials get a score of 1 under usual standards. With a low score but high Bo5, the model can be aligned with reinforcement learning (Ouyang et al., 2022) to quickly improve its score as positive samples are easy to acquire.

Model Name	Original Score	Modified Score	Original + FC	Modified + FC
DeepSeek-R1	86.46	22.66	86.08	26.33
OpenAI-o1-1217	86.08	29.87	86.21	41.01
Hunyuan Turbo-S	86.08	19.36	86.58	17.34
OpenAI-o3-mini-high	85.95	24.94	87.09	31.01
Ernie-4.5	83.42	20.13	79.75	22.91
Gemini-2.0 Flash-0121 (CoT)	81.90	23.80	79.37	27.22
Gemini-2.0 Pro-0205	81.90	20.89	44.43	31.89
GPT-4.5-Preview	80.89	26.59	78.99	37.22
Claude 3.7 Sonnet (CoT)	80.02	25.06	79.24	29.24
Claude 3.7 Sonnet	77.34	26.83	72.41	35.44
Gemini-2.0 Flash-0121	73.67	21.39	61.77	27.47
Qwen-max-0125	73.55	20.63	73.42	25.57
GPT-4o-1120	72.91	21.26	68.48	27.85
DeepSeek-V3	71.90	18.73	71.39	27.34
QwQ-32B-Preview	71.39	22.53	70.13	23.67
Minimax-Text-01	70.00	19.75	68.99	18.10
Claude 3.5 Sonnet	69.75	22.28	69.49	29.49
GLM-4-Plus	69.37	17.34	69.24	21.77
StepFun Step-2-16k	69.11	16.71	67.59	20.37
Yi-Lightning	68.61	15.95	70.63	20.00
Qwen-2.5-14B-Instruct	66.20	18.86	66.59	21.52
Mistral-Large-2	62.41	18.10	55.70	23.42
GPT-4o-mini-0718	60.63	18.86	60.00	20.38
Nova-Pro	57.46	17.59	55.82	21.65
Qwen-2.5-7B-Instruct	35.31	13.16	36.20	13.54
Avg. Decrease	N/A	51.96($\pm$9.07)	3.24($\pm$7.74)	46.90($\pm$9.06)

Table 1: Results on text-based problems of RoR-Bench. All scores are binary, averaged over 5 trials and 158 problems, and normalized to $0-100$ (higher is better). The (CoT) suffix stands for the same models with long thinking process enabled. FC stands for "Forced Correct" prompt. It is clearly illustrated that LLMs unanimously fail on modified problems, often with over 50% performance decrease. "Forced Correct" prompts somewhat helps, but is still far from bridging the gap; also, the performance of original problems with "Forced Correct" prompts generally decrease, which indicates that adding prompt is not a valid solution.

of our vision-based problems are binary multiple choices, verifiable answers could lead to higher scores with guessing.

We manually verified the effectiveness of the current LLM judge on OpenAI-o1-1217. We did not find any incorrect judging result for vision-based problems, as we either use exactly the same problem but modify the figure or ask for different properties of the same figure. For text-based problems, we find 20 out of 157*2=314 (original vs modified) arguable cases, where in 12 cases o1 somewhat realizes the subtle difference but made otherwise assumptions (e.g. "the problem may suggest xxx, but we should assume xxx")[5], in 7 cases LLM gives a different valid answer (some of them are suboptimal as o1 is unaware of the problem modification), and in 1 case is the judge made a mistake itself. The judgment accuracy on modified text-based problems is no less than 90%, which validates the existence of performance gap between original and modified problems. We further test our models on a verifiable subset of our benchmark; see Appendix C.1 for results.

[5]We view such result as incorrect as "partially correct" confuses the LLM judge in practice.

4.1.2 Excluding Auto-Correction of Typos

One possible concern of our benchmark is that since we do not modify much of the problem, the LLMs may perceive the modified input as typos and still solve the "correct" problems usually intended by the users (i.e., original problems). To address such concern, we further test LLMs with the **"Forced Correct"** (FC) prompt added to the beginning of the problem:

Forced Correct (FC) prompt: 请回答下面的问题。题目保证没有问题，请严格按照题目字面所写的问题回答。*("Please answer the following problem. The problems are guaranteed to be correct; please strictly follow the literal meaning of the problem.")*

The results are also illustrated in Tab. 1. Surprisingly, even with the FC prompt, LLMs still show on average $> 45\%$ performance decrease on modified problems, suggesting that the problem

cannot be simply treated as an auto-correction of typos. Moreover, the performance on original problems with the FC prompt slightly decreases, which become very significant on some models such as Gemini-2.0 Pro-0205. Upon examining the output, we found that LLMs often become too strict and overthink on the expression of the problems; for example, when asked whether a game is fair, LLMs will question the definition of "fairness" and refuse to give a definitive answer (see Appendix C.8). Such result shows that simply adding prompts is not a valid solution to the recitation issue.

4.2 Vision-based Problems

Evaluations. We evaluate 15 cutting-edge VLMs, which are: GPT-4.5-Preview, OpenAI-o1-1217, GPT-4o-1120, Gemini-2.0 Pro-0205, GPT-4o-mini-0718, Gemini-2.0 Flash-0121, Qwen-2.5-VL-max, GLM-4v-Plus, Qwen-2.5-VL-72B, Claude 3.5 Sonnet, StepFun-1v-32k, Nova-Pro, Claude 3.7 Sonnet, SenseChat-Vision (SenseTime, 2024), and Qwen2.5-VL-7B. Similar to text evaluation, we use GPT-4o-1120 as the judge with a binary score, and report the average accuracy (score by GPT-4o-1120) as the metric.

Results. Tab. 2 shows the result for all VLMs tested on RoR-Bench, which exhibits a $> 35\%$ perfomance decrease on average from original problems to the modified problems. Interestingly, we find GPT-4o-1120, GPT-4.5-Preview and OpenAI-o1-1217 to be significantly better on original problems, but much worse on modified problems; upon checking responses, we find that the OpenAI models listed above are much more likely to summarize the origin of the images, as we collect them usually from illustrations of famous visual effects (e.g. Mach bands and checker-shadow illusions). On the contrary, models like Claude 3.5 Sonnet and Claude 3.7 Sonnet usually do not explicitly summarize such visual effects. Such result indicates that 1) OpenAI models may be overfitting to usual test cases, and more importantly, 2) *explicit summarization or knowledge retrieval, which already becomes a common practice for prompt-engineering (Lee et al., 2024; Yang et al., 2024b), is a double-edged sword*; while they improve the performance on usual test cases, it may increase the risk of missing key details in the problem during summarization.

4.3 Is Few-Shot In-Context Learning the Cure?

A potential defense for the LLMs' performance on our benchmark is that humans can often be tricked when answering brain teasers; the limited performance of LLMs may due to the reason that they are prepared for normal user inputs and also "not ready for brain teasers". To address such concern, we conduct an empirical analysis on the text-based problems of the RoR-Bench under two settings: 1) Given the original problem and solution, can the model notice subtle difference between the original problem and the modified problem? 2) Given several other modified problems and their corresponding solutions, can the model realize the problems should be more carefully taken care of?

Evaluations. We evaluate the same set of LLMs in Sec. 4.1. For case 1 (adding original problems), we add a simple prompt mentioning the original problem and solution are an example (See Appendix B.2 for details). For case 2 (adding modified problems), we uniformly randomly select modified problems other than the current problem as shots; we test both 1-shot and 5-shot.

Results. The results for the most representative LLMs are listed in Tab. 3 (See Tab. 8 in Appendix C.4 for more results). The results shows that generally, both adding original problems and adding modified problems as few-shots can help improve the performance of the LLMs on modified problems; such effect can be further helped by adding the "Forced Correct" prompt in case 1, or increasing the number of shots in case 2.

Therefore, such fixes can be seen as an initial solution; however, the performance gap between all these fixes and original problems is still very large ($> 30\%$), which indicates that few-shot ICL is not the ideal panacea for LLMs to overcome the recitation issue.

4.4 Overconfidence in Solvability

As real-life problems can be ill-posed sometimes with no valid solution, a good LLM agent should possess the ability to discriminate such type of problems. However, as we examine the "no solution" problems in our benchmark (see Sec. 3.2 for details), we found that LLMs are particularly worse in correctly pointing out the problems with no solution, and often will make mistakes to make up a solution, as if injected by the mental seal that the problem is definitely solvable.

Model Name	Original Score	Modified Score	Original + FC	Modified + FC
GPT-4.5-Preview	91.23	17.89	77.19	40.70
OpenAI-o1-1217	90.18	18.60	91.58	23.51
GPT-4o-1120	87.02	14.74	85.61	26.32
Gemini-2.0 Pro-0205	70.53	32.98	64.21	37.54
GPT-4o-mini-0718	70.53	30.53	79.65	26.67
Gemini-2.0 Flash-0121 (CoT)	69.82	33.68	67.71	39.30
Qwen2.5-VL-max	66.32	37.54	64.56	42.11
GLM-4v-Plus	66.32	42.11	64.22	41.05
Qwen2.5-VL-72B	65.96	37.19	64.91	42.1
Claude 3.7 Sonnet (CoT)	64.91	34.03	63.51	40.00
Gemini-2.0 Flash-0121	64.91	30.17	53.68	35.79
Claude 3.5 Sonnet	63.15	38.24	57.19	44.91
StepFun-1v-32k	61.75	29.12	64.91	27.72
Nova-Pro	60.35	51.58	70.17	36.14
Claude 3.7 Sonnet	57.54	33.68	58.60	42.46
SenseChat-Vision	56.84	37.19	72.63	38.94
Qwen2.5-VL-7B	51.93	41.40	58.95	38.60
Avg. Decrease	N/A	35.21($\pm$19.67)	0.00($\pm$7.52)	31.50($\pm$15.47)

Table 2: Results on vision-based problems of RoR-Bench. All scores are binary and averaged over 5 trials and 57 problems, normalized to $0-100$ (higher is better). Similar to text problems, LLMs unanimously fail on modified problems, with $> 30\%$ average score decrease; "Forced Correct" prompt only works very marginally.

Model Name	Modified	Case 1	Case 1 + FC	Case 2 (1-Shot)	Case 2 (5-shot)
OpenAI-o1-1217	29.87	38.23	49.37	34.41	43.89
Claude 3.7 Sonnet	26.83	29.49	38.48	30.75	38.10
GPT-4.5-Preview	26.59	32.66	41.27	31.01	38.48
OpenAI-o3-mini-high	24.94	35.70	38.10	34.30	36.96
DeepSeek-R1	22.66	28.35	28.99	27.34	27.84
Claude 3.5 Sonnet	22.28	27.84	38.10	25.82	32.78
Gemini-2.0 Flash-0121	21.39	22.53	28.73	22.53	27.34
GPT-4o-1120	21.26	23.80	31.39	18.73	31.27
Gemini-2.0 Pro-0205	20.89	24.56	34.94	26.20	33.04
Avg. Increase	N/A	5.16($\pm$3.05)	12.52($\pm$4.16)	3.82($\pm$3.20)	10.33($\pm$2.94)

Table 3: The results of adding original problems as 1-shot (case 1) or adding other modified problems as few-shot (case 2) sorted by average score on modified problems in our benchmark. Claude 3.7 Sonnet and Gemini-2.0 Flash-0121 are without long CoT (same for Tab. 4). Though the result show clear performance improvement, a large gap still exists between the improved performance and that on original problems.

Evaluations. We report the performance on "no solution" problems from modified problem results in Sec. 4.1. We further test three alternative cases as possible fixes for the issue: 1) with "Forced Correct" prompt, 2) with "Forced Correct" prompt and another no solution problem as 1-shot, and 3) with both 1) and 2).

Results. Tab. 4 shows the performance of the most representative LLMs on "no solution" problems as stated in Sec. 3.2 (See Tab. 9 in Appendix C.5 for more results). Surprisingly, without any fixes, LLMs are unanimously stubborn on the belief that the given problem is solvable; not a single model achieves $> 15\%$ score. While generally adding "forced correct" prompt and other "no solution" problems as 1-shot help resolve the mental seal of solvability, it only works well for some LLMs such as GPT-4.5-Preview, and is generally still far from satisfactory for most models.

Interestingly, DeepSeek-R1 struggles in recognizing unsolvable questions; we find that it has a firmer belief that the problem should be handled in the usual pattern even with directions that the problem should be taken literally, or several examples suggesting that it is more similar to a brain teaser; more often than other models, we found the form "the problem may suggest xxx, but maybe we should still consider xxx ..." in its CoT. Overall, thinking models with long CoT are more likely to somewhat realize that the problem might contain traps, but assume the problem to be "normal" (i.e. closer to original), while normal LLMs often answer the problem unaware of the condition change. This suggests that long CoT might be a possible

Model Name	Modified	+FC	+1-shot	+ FC+1-shot
OpenAI-o1-1217	13.75	26.88	30.00	41.25
GPT-4.5-Preview	13.13	30.63	25.63	58.13
Claude 3.7 Sonnet	10.63	23.12	25.00	36.25
Gemini-2.0 Flash-0121	10.63	18.75	20.89	28.35
Gemini-2.0 Pro-0205	9.38	26.88	26.88	36.88
OpenAI-o3-mini-high	6.25	10.63	23.13	24.38
Claude 3.5 Sonnet	6.25	13.75	28.73	41.27
GPT-4o-1120	5.63	16.25	11.25	46.88
DeepSeek-R1	3.13	8.75	9.38	11.25
Avg. Increase	N/A	10.76($\pm$4.80)	13.57($\pm$5.51)	27.32($\pm$11.80)

Table 4: The scores for "no solution" problems and possible fixes sorted by average score. Without any fixes, the average score for "no solution" problems is extremely low, showing the firm belief of LLMs that the given problem is solvable. While some LLMs, such as GPT-4.5-Preview, can be effectively corrected by prompt engineering, other LLMs such as DeepSeek-R1 are still very stubborn.

way to mitigate the issue, albeit not in the current status; more alignment is required to make its judgment on the problem more similar to humans.

4.5 Why Does Recitation Happen?

To address why recitation happens, we consider three possible reasons: 1) **Dispersed attention**, i.e., the model ignores the subtly changed condition due to insufficient attention weights; 2) **Overalignment and bad instruction following ability**, i.e., the models stick to the "common" user intention and are reluctant to follow the problem in literal even with the "force correct prompt", and 3) **Solution paradigm overfitting**, i.e., the model does not see the subtly changed problems in its training and thus performs poorly on out-of-distribution data.

To test the first reason, we add irrelevant text (The Thousand Character Classic/ 《千字文》, a Chinese poem) in front of each problem. We report the accuracy percentage change in Tab. 5:

Model	ΔOriginal	Δ Modified
DeepSeek-R1	-1.9	-2.9
Claude-3.5-Sonnet	+1.0	+5.7
Gemini-2.0 Pro	+1.0	+6.7
GPT-4o-mini-0718	-3.8	+1.9
GPT-4o-1120	-1.9	-1.9
Gemini-2.0 Flash	+2.9	0

Table 5: Accuracy change after adding the irrelevant text to the original problem (ΔOriginal) and modified problem (ΔModified). No consistent changes witnessed.

The result shows no or very slight performance change on modified problems on average; thus, attention dispersion is likely not the culprit.

For the second reason, we witness some cases (12 for o1 on text-based problems) where the model keeps adhering to the "usual" condition as in Appendix A.1 and Sec. 4.3. However, there are still many cases where o1 and DeepSeek-R1 fall into trap on our modified problems without noticing the condition change, with non-thinking models usually (if not always) ignoring them. Also, by comparing Tab. 1 and "case 1" in Tab. 3, the performance increase by adding "forced correct" prompt (instruction prompt) is roughly the same as adding the original problem (non-instruction prompt) as 1-shot (5.06% vs. 5.16% on average) which can stack (12.52% combined). Thus, overalignment/bad instruction following is partly the reason, but cannot account for most of the performance gap.

In conclusion, our hypothesis is that solution paradigm overfitting is the main culprit, while overalignment / instruction following ability is also a factor. Long CoT gives the model more diverse reasoning paradigms and chances of self-reflection to mitigate the issue, but the model still needs to prevent overalignment to gain performance.

5 Discussion and Conclusion

In this work, we propose RoR-Bench, a multimodal Chinese benchmark which clearly reveals an alarming issue that current cutting-edge LLMs are unable to address even simple reasoning problems with conditions subtly shifted. Such phenomenon proved that LLMs are conducting *recitation instead of reasoning* when confronting seemingly classic problems. We found such issue can lead to dramatic performance loss ($> 50\%$) and is unable to be addressed by simple fixes such as adding instruction prompts or few-shots, indicating that such issue is hard to fix and should be better awared by current LLM developers and researchers.

Limitations

Currently, our benchmark is Chinese-only due to the language limitation of human annotators and moderators, which may cause an edge on performance for LLMs by Chinese companies such as Ernie-4.5 and Hunyuan Turbo-S (note the main message, significant performance decrease after modification, is not affected). Though our message to convey is already strong with the current results (and preliminary English translation tests in this paper suggest that LLMs will other struggle on the other languages), to expand such benchmark to multiple languages will be an important but challenging future work (see Appendix A for detailed discussion). A more important and fundamental avenue for future research is to find an effective way for LLMs to overcome the problem of recitation over reasoning without over-reliance on user's clarifications or being too harsh on typos.

Ethical Considerations

Our work studies Large Language Models' (LLMs') long-context intelligence level by proposing a many-shot in-context inductive reasoning benchmark and conducting empirical studies based on the benchmark. As our work is a stepping stone towards Artificial General Intelligence (AGI), it could lead to negative impacts such as the spread of inappropriate AI-generated contents or human job loss. To better help human society embrace the era of AGI is an important and interesting avenue for our future research.

References

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, and 29 others. 2024. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. What learning algorithm is in-context learning? investigations with linear models. In *ICLR*.

Anthropic. 2024. Claude 3.5 sonnet.

Anthropic. 2025. Claude 3.7 sonnet system card.

Sanjeev Arora and Anirudh Goyal. 2023. A theory for emergence of complex skills in language models. *arXiv preprint arXiv:2307.15936*.

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, and 1 others. 2023. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and 1 others. 2024. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *ACL*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *FAccT*.

Aman Bhargava, Cameron Witkowski, Shi-Zhuo Looi, and Matt Thomson. 2023. What's the magic word? a control theory of llm prompting. *arXiv preprint arXiv:2310.04444*.

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. In *NeurIPS*.

Davide Caffagni, Federico Cocchi, Luca Barsellotti, Nicholas Moratelli, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. 2024. The revolution of multimodal large language models: a survey. In *ACL Findings*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

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*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. 2025. Aha: A vision-language-model for detecting and reasoning over failures in robotic manipulation. In *ICLR*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yassir Fathullah, Chunyang Wu, Egor Lakomkin, Ke Li, Junteng Jia, Yuan Shangguan, Jay Mahadeokar, Ozlem Kalinli, Christian Fuegen, and Mike Seltzer. 2024. Audiochatllama: Towards general-purpose speech abilities for llms. In *ACL*.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, and 1 others. 2024. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *arXiv preprint arXiv:2411.04872*.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, and 1 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

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, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *ICLR*.

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

Parsa Hosseini, Sumit Nawathe, Mazda Moayeri, Sriram Balasubramanian, and Soheil Feizi. 2025. Seeing what's not there: Spurious correlation in multimodal llms. *arXiv preprint arXiv:2503.08884*.

Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, and 1 others. 2025a. Mathperturb: Benchmarking llms' math reasoning abilities against hard perturbations. *arXiv preprint arXiv:2502.06453*.

Kung-Hsiang Huang, Can Qin, Haoyi Qiu, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2025b. Why vision language models struggle with visual arithmetic? towards enhanced chart and geometry understanding. *arXiv preprint arXiv:2502.11492*.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, and 1 others. 2023. Language is not all you need: Aligning perception with language models. *NeurIPS*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Fabian Hutmacher. 2019. Why is there so much more research on vision than on any other sensory modality? *Frontiers in psychology*.

Baidu Inc. 2025. Introducing ernie 4.5 | our new-generation native multimodal model.

Amazon Artificial General Intelligence. 2024. The amazon nova family of models: Technical report and model card. *Amazon Technical Reports*.

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*.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. Swe-bench: Can language models resolve real-world github issues? In *ICLR*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Koray Kavukcuoglu. 2025. Gemini 2.0 is now available to everyone.

Mehran Kazemi, Bahare Fatemi, Hritik Bansal, Chrysovalantis Anastasiou John Palowitch, Sanket Vaibhav Mehta, Lalit K. Jain, Virginia Aglietti, Disha Jindal, Peter Chen, Nishanth Dikkala, Gladys Tyen, Xin Liu, Uri Shalit, Silvia Chiappa, Kate Olszewska, Yi Tay, Vinh Q. Tran, Quoc V. Le, and Orhan Firat. 2025. Big-bench extra hard. *arXiv preprint arXiv:2502.19187*.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. In *ICML*.

Aonian Li, Bangwei Gong, Bo Yang, Boji Shan, Chang Liu, Cheng Zhu, Chunhao Zhang, Congchao Guo, Da Chen, Dong Li, and 1 others. 2025. Minimax-01: Scaling foundation models with lightning attention. *arXiv preprint arXiv:2501.08313*.

Dongting Li, Chenchong Tang, and Han Liu. 2024. Audio-llm: Activating the capabilities of large language models to comprehend audio data. In *ISNN*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *CVPR*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. In *NeurIPS*.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023b. Is your code generated by chatGPT really correct? rigorous evaluation of large language models for code generation. In *NeurIPS*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, and 1 others. 2024b. Agentbench: Evaluating llms as agents. In *ICLR*.

Yang Liu, Jiahuan Cao, Chongyu Liu, Kai Ding, and Lianwen Jin. 2024c. Datasets for large language models: A comprehensive survey. *arXiv preprint arXiv:2402.18041*.

Kaijing Ma, Xinrun Du, Yunran Wang, Haoran Zhang, Zhoufutu Wen, Xingwei Qu, Jian Yang, Jiaheng Liu, Minghao Liu, Xiang Yue, and 1 others. 2025. Kor-bench: Benchmarking language models on knowledge-orthogonal reasoning tasks. In *ICLR*.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024. Eureka: Human-level reward design via coding large language models. In *ICLR*.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2025. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *ICLR*.

OpenAI. 2023. Gpt-4v(ision) system card.

OpenAI. 2025a. [link].

OpenAI. 2025b. Introducing gpt-4.5.

OpenAI. 2025c. Openai o3-mini.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Karl Pearson. 1895. Notes on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, and 1 others. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249*.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S Yu. 2024. Large language models meet nlp: A survey. *arXiv preprint arXiv:2405.12819*.

Zeju Qiu, Weiyang Liu, Haiwen Feng, Zhen Liu, Tim Z Xiao, Katherine M Collins, Joshua B Tenenbaum, Adrian Weller, Michael J Black, and Bernhard Schölkopf. 2024. Can large language models understand symbolic graphics programs? *arXiv preprint arXiv:2408.08313*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *COLM*.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J Liu. 2023. Out-of-distribution detection and selective generation for conditional language models. In *ICLR*.

SenseTime. 2024. Sensetime unveils sensenova 5.5 - a complete and comprehensive upgrade.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *ICML*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, and 1 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *TMLR*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, and 1 others. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *ACL Findings*.

Mistral AI team. 2024. Large enough.

Qwen Team. 2025a. Qwen2.5-max: Exploring the intelligence of large-scale moe model.

Qwen Team. 2025b. Qwq: Reflect deeply on the boundaries of the unknown.

Tencent. 2025. Tencent/llm.hunyuan.turbo-s.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. 2024. Drivevlm: The convergence of autonomous driving and large vision-language models. In *CoRL*.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In *NeurIPS*.

Maya Varma, Jean-Benoit Delbrouck, Zhihong Chen, Akshay Chaudhari, and Curtis Langlotz. 2024. Ravl: Discovering and mitigating spurious correlations in fine-tuned vision-language models. *NeurIPS*.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024a. Position: Will we run out of data? limits of llm scaling based on human-generated data. In *ICML*.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024b. Will we run out of data? limits of llm scaling based on human-generated data. In *ICML*.

Alan Wake, Bei Chen, CX Lv, Chao Li, Chengen Huang, Chenglin Cai, Chujie Zheng, Daniel Cooper, Fan Zhou, Feng Hu, and 1 others. 2024. Yi-lightning technical report. *arXiv preprint arXiv:2412.01253*.

Beichen Wang, Juexiao Zhang, Shuwen Dong, Irving Fang, and Chen Feng. 2024a. Vlm see, robot do: Human demo video to robot action plan via vision language model. *arXiv preprint arXiv:2410.08792*.

Jize Wang, Ma Zerun, Yining Li, Songyang Zhang, Cailian Chen, Kai Chen, and Xinyi Le. 2024b. Gta: a benchmark for general tool agents. In *NeurIPS Datasets and Benchmarks Track*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024c. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024d. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *NeurIPS Datasets and Benchmarks Track*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, and 1 others. 2022a. Emergent abilities of large language models. *TMLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Zikai Xie. 2024. Order matters in hallucination: Reasoning order as benchmark and reflexive prompting for large-language-models. *arXiv preprint arXiv:2408.05093*.

Nan Xu and Xuezhe Ma. 2025. Llm the genius paradox: A linguistic and math expert's struggle with simple word-based counting problems. In *NAACL*.

Yi Xu, Yuxin Hu, Zaiwei Zhang, Gregory P Meyer, Siva Karthik Mustikovela, Siddhartha Srinivasa, Eric M Wolff, and Xin Huang. 2024. Vlm-ad: End-to-end autonomous driving through vision-language model supervision. *arXiv preprint arXiv:2412.14446*.

Fanjia Yan, Huanzhi Mao, Charlie Cheng-Jie Ji, Tianjun Zhang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. Berkeley function calling leaderboard. https://gorilla.cs.berkeley.edu/blogs/8_berkeley_function_calling_leaderboard.html.

Kai Yan, Zhan Ling, Kang Liu, Yifan Yang, Ting-Han Fan, Lingfeng Shen, Zhengyin Du, and Jiecao Chen. 2025. Mir-bench: Benchmarking llm's long-context intelligence via many-shot in-context inductive reasoning. In *ICLR Workshop on Reasoning and Planning for Large Language Models*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024a. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E Gonzalez, and Bin Cui. 2024b. Buffer of thoughts: Thought-augmented reasoning with large language models. *NeurIPS*.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. In *NeurIPS*.

Junwei You, Haotian Shi, Zhuoyu Jiang, Zilin Huang, Rui Gan, Keshu Wu, Xi Cheng, Xiaopeng Li, and Bin Ran. 2024. V2x-vlm: End-to-end v2x cooperative autonomous driving through large vision-language models. *arXiv preprint arXiv:2408.09251*.

Lifan Yuan, Yangyi Chen, Ganqu Cui, Hongcheng Gao, Fangyuan Zou, Xingyi Cheng, Heng Ji, Zhiyuan Liu, and Maosong Sun. 2023. Revisiting out-of-distribution robustness in nlp: Benchmarks, analysis, and llms evaluations. In *NeurIPS*.

Oussama Zekri, Ambroise Odonnat, Abdelhakim Benechehab, Linus Bleistein, Nicolas Boullé, and Ievgen Redko. 2024. Large language models as markov chains. *arXiv preprint arXiv:2410.02724*.

Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. 2024a. When scaling meets llm finetuning: The effect of data, model and finetuning method. In *ICLR*.

Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Jianquan Li, Guiming Chen, Xiangbo Wu, Zhiyi Zhang, Qingying Xiao, and 1 others. 2023. Huatuogpt, towards taming language model to be a doctor. In *Findings of EMNLP*.

Yu Zhang, Xiusi Chen, Bowen Jin, Sheng Wang, Shuiwang Ji, Wei Wang, and Jiawei Han. 2024b. A comprehensive survey of scientific large language models and their applications in scientific discovery. In *EMNLP*.

Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. 2025. Is in-context learning sufficient for instruction following in llms? In *ICLR*.

Jun Zhao, Jingqi Tong, Yurong Mou, Ming Zhang, Qi Zhang, and Xuan-Jing Huang. 2024. Exploring the compositional deficiency of large language models in mathematical reasoning through trap problems. In *EMNLP*.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024a. Large language models are not robust multiple choice selectors. In *ICLR*.

Huaixiu Steven Zheng, Swaroop Mishra, Hugh Zhang, Xinyun Chen, Minmin Chen, Azade Nova, Le Hou, Heng-Tze Cheng, Quoc V Le, Ed H Chi, and 1 others. 2024b. Natural plan: Benchmarking llms on natural language planning. *arXiv preprint arXiv:2406.04520*.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2024a. Language agent tree search unifies reasoning acting and planning in language models. In *ICML*.

Lexin Zhou, Wout Schellaert, Fernando Martínez-Plumed, Yael Moros-Daval, Cèsar Ferri, and José Hernández-Orallo. 2024b. Larger and more instructable language models become less reliable. *Nature*.

Yang Zhou, Hongyi Liu, Zhuoming Chen, Yuandong Tian, and Beidi Chen. 2025. Gsm-infinite: How do your llms behave over infinitely increasing context length and reasoning complexity? *arXiv preprint arXiv:2502.05252*.

Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. 2025. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. In *ICLR*.

Appendix: Recitation over Reasoning: How Cutting-Edge Language Models Can Fail on Elementary School-Level Reasoning Problems?

The appendix is organized as follows. In Sec. A, we discuss the possible challenges of expanding this benchmark to other languages. In Sec. B, we introduce the prompts used in our experiments, and in Sec. C, we show more experiment results on our RoR-Bench. Finally, we discuss the license and dataset use in Sec. D.

A Challenges of Expanding to Other Languages

One limitation of our benchmark is that it is Chinese-only in its current form. From our results, we tend to believe that the recitation over reasoning problem is language-agnostic. However, models encounter more severe phenomena with Chinese due to the potentially large meaning variance with minimal change (e.g. a single character); to extend to other languages such as English could be potentially very challenging. Here are some examples:

Example 1: In our benchmrk ,the original question is "...船从甲地开往乙地....", and the modified question is "船从甲地漂往乙地". Here, the word 开 strongly indicates that the engine is involved, while the word "漂" strongly indicates that the engine is not involved in Chinese. O1 translates "漂" as "floating" or "drifting"; however, we received feedback from English native speakers that such words are ambiguous in English with respect to the use of engine.

Example 2: the original question is "...相向而行...", while the modified question is "相背而行". Here, the former means "travel towards each other", and the latter means "travel away from each other". O1 gives a translation of "travel in the opposite directions". However, "travel in the opposite directions" is ambiguous as traveling away from and traveling towards each other are both "traveling in the opposite directions".

Thus, translating our benchmark into English is a non-trivial effort which requires extended efforts from native speakers; currently, we treat this as an important future work.

B Prompts

B.1 Prompts for the Judge

We use the same prompt for GPT-4o-1120 judges for all our experiments. We use Chinese prompts as the problems are Chinese; we provide English translations as well in the boxes below. Interestingly, we found the emphasis of "do not question the ground truth answer" very important; *the recitation issue of LLMs is so severe that without such prompt, the LLM judge will question the problem instead of the student's answer even with ground truth answer available.* In our experiment, we find that the response of GPT-4o judge is usually in the following form: "the student's answer is xxx, the ground truth is xxx. As they are different in xxx / essentially the same, the final score is 0 / 1."

Chinese Prompt
你是一个专业的老师，现在有一道题目，你需要判断学生的回答是否和标准答案一致。题目和答案保证是绝对正确的，不会有错字，因此你要严格按照题目的字面意思评判。不要质疑标准答案有问题。 如果学生的回答和标准答案一致，请打1分；否则请打0分。标准答案经常只含有答案，而学生的回答一般会带有过程；你只需要根据学生的结论是否与标准答案相符做出评价。学生的结论和标准答案必须本质一样，但表述可以稍有不同；例如，学生的答案是50又1/2或50.5，标准答案是101/2，则也可以算作正确。 你的判断应遵循如下格式：你的输出在【评价】后开始。首先是对学生回答的分析（不超过300字），然后以"分数:[[0]]"或"分数：[[1]]"结束。 下面开始判断：
【题目】 (Problem description)
【标准答案】 (Answer)
【学生回答】 (LLM output)

English Translation
You are a professional teacher. Now there is a problem, and you need to judge whether the answer of the student is the same as the ground truth answer. The problem and the answer are guaranteed to be ABSOLUTELY CORRECT; there will not be ANY typos, and so you MUST STRICTLY judge with the literal meaning of the problem. DO NOT QUESTION THE GROUND TRUTH ANSWER. If the student's answer is the same as the ground truth answer, give 1 points; otherwise, give 0 points. The ground truth answer often only contains the final results, but the student's answer will often include intermediate steps; judge only by comparing the student's conclusion and the ground truth answer. The student's conclusion must be essentially the same as the ground truth answer, but they can be slightly differently expressed; for example, if the student's answer is "50 and 1/2" or "50.5", while the ground truth answer is 101/2, then it can be counted as correct. Your judge MUST follow the following format: your output starts after [Judge]. First, analyze the student's answer (no more than 300 characters); then end with "Score: [[0]]" or "Score: [[1]]". Now begin your judgment:
[Problem] (Problem description)
[Ground Truth Answer] (Answer)
[Student's Answer] (LLM output)

B.2 Prompts for Few-Shot In-Context Learning

In this section, we provide an 1-shot example to show the prompts for few-shot ICL experiments; for cases with more shots, the problems are added in the same format as the first example before the last, target problem. We again show both the original Chinese version and the English translation. The red part is the Forced Correct (FC) prompt, which is optional.

请回答下面的问题。题目保证没有错误，请严格按照题目字面所写的问题回答。以下是一个例子：
【问题】有四个人要在游过一条河，他们只有一个游泳圈，且每次最多只能两个人一起使用游泳圈游过河，使用游泳圈时必须有人携带 。四个人单独游过河的时间分别是 1 分钟、2 分钟、5 分钟、10 分钟。如果两人一起使用游泳圈游过河，所需要的时间就是游得慢的那个人单独游过河的时间。请问，他们如何在 17 分钟内全部游过河？
【答案】让 1 分钟和 2 分钟的人先一起使用游泳圈游过河，花费 2 分钟，然后 1 分钟的人带着游泳圈游回来，花费 1 分钟。5 分钟和 10 分钟的人一起使用游泳圈游过河，花费 10 分钟，接着 2 分钟的人带着游泳圈游回来，花费 2 分钟。1 分钟和 2 分钟的人再次一起使用游泳圈游过河，花费 2 分钟。总共花费的时间为：2+1+10+2+2 = 17 分钟。
下面是你要回答的问题：
【问题】有四个人要在游过一条河，他们只有一个游泳圈，且每次最多只能两个人一起使用游泳圈游过河，不会游泳的人必须使用游泳圈，使用时必须有人携带 。四个人单独游过河的时间分别是 1 分钟、2 分钟、5 分钟、10 分钟。其中前三个人均会游泳。如果两人一起使用游泳圈游过河，所需要的时间就是游得慢的那个人单独游过河的时间。请问，他们如何在 17 分钟内全部游过河？
Please answer the following problems. The problems are guaranteed to be correct; please strictly follow the literal meaning of the problem. Here is an example:
[Problem]
Four people need to swim across a river. They have only one swimming ring, and at most two people can use it at the same time. Someone must carry the swim ring whenever it is used. The time it takes for each person to swim across the river individually is 1 minute, 2 minutes, 5 minutes, and 10 minutes respectively. If two people use the swim ring together to cross the river, the time it takes is equal to the time of the slower swimmer. The question is: how can all four people cross the river within 17 minutes?
[Answer]
Let the 1-minute and 2-minute people use the swim ring to cross the river first, which takes 2 minutes. Then the 1-minute person brings the swim ring back, taking 1 minute. Next, the 5-minute and 10-minute people cross the river together using the swim ring, which takes 10 minutes. After that, the 2-minute person brings the swim ring back, taking 2 minutes. Finally, the 1-minute and 2-minute people cross the river together again using the swim ring, taking 2 minutes.
The total time spent is: 2 + 1 + 10 + 2 + 2 = 17 minutes.
Now here is the problem you need to answer:
[Problem]
Four people need to swim across a river. They have only one swimming ring, and at most two people can use it at the same time. Anyone who cannot swim must use the swim ring, and it must be carried by someone while in use. The times it takes for each person to swim across the river individually are 1 minute, 2 minutes, 5 minutes, and 10 minutes respectively. Among them, the first three people can swim. If two people use the swim ring together to cross the river, the time required is equal to the time it takes for the slower person to cross the river alone. The question is: how can all four people cross the river within 17 minutes?

Interestingly, when we test this English translation with OpenAI-o1-1217, we found o1, even with 1-shot, is again tricked into the classic paradigm that the swimming ring must be carried back. o3-mini (https://chatgpt.com/share/67f60f89-e8c8-800d-b7c0-c0fcffaaad18), o3-mini-high (https://chatgpt.com/share/67f60f60-f0b0-800d-93cd-2e770dc7cbb5) and Gemini-2.5 Pro (https://g.co/gemini/share/3ebe9a57c6ff) all fell for the trap for 0-shot). The ground truth answer of this target problem, however, is to directly let the third and fourth people use the swimming ring, and the first two people swim through the river, such that everything can be done within 10 minutes; no swimming ring needs to be taken back.

C More Experiment Results

C.1 Results on Verifiable Subset of RoR-Bench

To further verify our findings without possible bias by LLM judge, we test several models on a manually picked verifiable subset of our benchmark. The result is listed in Tab. 6 and 7, which shows similar conclusions to our main paper.

Model	Original	Modified	Modified+FC
DeepSeek-R1	92.2	37.4	34.8
DeepSeek-V3	80.4	23.9	24.8
Claude-3.5-Sonnet	79.6	34.8	37.8
Claude-3.7-Sonnet	78.7	39.6	41.3
Gemini-2.0 Flash	78.3	20.9	20.4
Mistral-Large-2	72.6	30.0	34.4

Table 6: Results on text-based verifiable subset of RoR-Bench (FC="Forced Correct" prompt).

C.2 English Version of Fig. 1

The response for OpenAI-o3 of the problem in Fig. 1 can be seen at https://chatgpt.com/share/

Model	Original	Modified
GPT-4o	94.4	20.4
Qwen-VL-max	80.0	42.2
Gemini-2.0 Flash	75.2	38.5
SenseChat-Vision	76.3	38.5
Qwen2.5-VL-72B-Instruct	64.1	43.7
GPT-4o-mini	61.1	49.3
StepFun-1v	54.4	52.2

Table 7: Results on vision-based verifiable subset of RoR-Bench (FC="Forced Correct" prompt).

687d4e38-680c-800d-9578-442c6819a5d7 and https://chatgpt.com/share/687d4e0c-0444-800d-983c-c63067a67820. For Gemini-2.5 Pro, the response can be seen at https://g.co/gemini/share/01fcaeb71e18 and https://g.co/gemini/share/d8dceea41f17. While they sometimes realize the possible ambiguity in the problem (as shown in the second response for OpenAI-o3), they can still often solve the problem directly without noticing the subtle change.

C.3 English Version of Fig. 2

The response for OpenAI-o3 for our English translation of the modified text problems in Fig. 2 can be seen in https://chatgpt.com/share/687d3f20-76c8-800d-bf9d-8edd8fcc07e5, and Gemini-2.5 Pro in https://g.co/gemini/share/9ead78631a31. For the modified image problem, the response for OpenAI-o3 can be see in https://chatgpt.com/share/687d3fe2-dec4-800d-a748-e6ebdb087d48 (o3 answers no, but the reasoning is about neighbors, which is irrelevant; there is no illusion within each diamond), and Gemini-2.5 Pro in https://g.co/gemini/share/6bd5bff68c2d.

C.4 Complete Results for Sec. 4.3

Tab. 8 shows the results for all LLMs we evaluated in Sec. 4.3.

C.5 Complete Results for Sec. 4.4

Tab. 9 shows the results for all LLMs we evaluated in Sec. 4.4.

C.6 Best-of-5 Results

Tab. 10 (for text-based problems) and Tab. 11 (for vision-based problems) shows the best-of-5 result of the experiments conducted in Sec. 4.1 and Sec. 4.2. The conclusion is very similar to those in Sec. 4.1 and Sec. 4.2, indicating that the problem is hard to fix with LLM alignment techniques such as reinforcement learning (Ouyang et al., 2022).

C.7 Greedy Decoding Results

Tab. 12 (for text-based problems) and Tab. 13 (for vision-based problems) shows the average score of LLMs doing greedy-decoding (i.e. temperature=0) in the experiments conducted in Sec. 4.1 and Sec. 4.2. The conclusion is similar to those in Sec. 4.1 and Sec. 4.2.

C.8 Failure Case of Overthinking

Here is an example of Gemini-2.0 Pro-2025 Becomeing too strict and overthink on the expression of the problems, as mentioned in Sec. 4.1. Such result shows that simply adding prompts is not a valid solution to the recitation issue.

Problem
请回答下面的问题。题目保证没有错误，请严格按照题目字面所写的问题回答。
【问题】你正在图书馆枯坐，一位陌生美女主动过来和你搭讪，递给你两个相同的硬币并要求和你一起玩个数学游戏。美女提议："让我们各自亮出硬币的一面，或花或字。如果我们都是花面，那么我给你 3 元，如果我们都是字面，我给你 1 元，剩下的情况你给我 2 元就可以了。"那么，这个游戏公平吗？
(Please answer the following problems. The problems are guaranteed to be correct; please strictly follow the literal meaning of the problem.
[Question] You are sitting idly in the library when a beautiful stranger approaches you and strikes up a conversation. She hands you two identical coins and proposes a math game. She says: "Let's each reveal one side of a coin—either heads or tails. If we both show heads, I'll give you 3 yuan. If we both show tails, I'll give you 1 yuan. In all other cases, you just give me 2 yuan." So, is this game fair?)
Gemini-2.0 Pro-0205 Response
这个游戏是否公平，取决于"公平"的定义是什么。题目没有问这个游戏是否公平，也没有定义什么是"公平"，因此无法回答这个问题。
(Whether the game is "fair" depends on the definition of fair. The problem does not ask whether the game is fair, and did not define what is "fairness"; therefore, the problem cannot be answered.)

D License and Dataset Use

We plan to open-source our dataset under the Apache-2.0 license; to avoid legal concern, we will not use this dataset for any commercial purpose or in any product. The data collection process is conducted by internal data department and reviewed by internal legal department.

Model Name	Modified	Case 1	Case 1 + FC	Case 2 (1-Shot)	Case 2 (5-shot)
OpenAI-o1-1217	29.87	38.23	49.37	34.41	43.89
Claude 3.7 Sonnet	26.83	29.49	38.48	30.75	38.10
GPT-4.5-Preview	26.59	32.66	41.27	31.01	38.48
Claude 3.7 Sonnet (CoT)	25.06	22.15	26.46	17.97	26.58
OpenAI-o3-mini-high	24.94	35.70	38.10	34.30	36.96
DeepSeek-R1	22.66	28.35	28.99	27.34	27.84
Gemini-2.0 Flash-0121 (CoT)	23.80	22.41	29.49	24.43	28.35
QwQ-32B-Preview	22.53	25.19	26.96	24.05	23.42
Claude 3.5 Sonnet	22.28	27.84	38.10	25.82	32.78
Gemini-2.0 Flash-0121	21.39	22.53	28.73	22.53	27.34
GPT-4o-1120	21.26	23.80	31.39	18.73	31.27
Gemini-2.0 Pro-0205	20.89	24.56	34.94	26.20	33.04
Qwen-max-0125	20.63	22.66	27.72	20.38	25.95
Ernie-4.5	20.13	22.03	27.85	19.75	25.19
Minimax-Text-01	19.75	19.62	18.10	18.10	17.72
Hunyuan Turbo-S	19.36	22.53	20.25	19.24	20.51
GPT-4o-mini-0718	18.86	21.77	26.84	20.38	21.39
Qwen2.5-14B-Instruct	18.86	19.11	20.89	19.62	19.24
DeepSeek-V3	18.73	22.15	26.46	17.97	26.58
Mistral-Large-2	18.10	19.49	29.37	21.65	25.57
GLM-4-Plus	17.34	21.27	26.33	17.34	25.19
Nova Pro	17.59	16.70	22.15	17.85	22.41
StepFun Step-2-16k	16.71	21.01	24.17	19.75	22.02
Yi-lightning	15.95	17.34	20.76	16.58	19.75
Qwen2.5-7B-Instruct	13.16	12.66	15.57	14.30	13.42
Avg. Increase	N/A	+2.72($\pm$3.05)	+7.82($\pm$5.12)	+1.49($\pm$3.17)	+5.99($\pm$4.41)

Table 8: Results of all LLMs with the settings in Sec. 4.3. Models with weaker base ability, such as Qwen-2.5-7B-Instruct, are harder to improve by few-shot ICL techniques.

Model Name	Modified	+FC	+1-shot	+ FC+1-shot
OpenAI-o1-1217	13.75	26.88	30.00	41.25
GPT-4.5-Preview	13.13	30.63	25.63	58.13
Claude 3.7 Sonnet	10.63	23.13	25.00	36.25
Gemini-2.0 Flash-0121	10.63	18.75	20.89	28.35
Gemini-2.0 Pro-0205	9.38	26.88	26.88	36.88
OpenAI-o3-mini-high	6.25	10.63	23.13	24.38
Claude 3.5 Sonnet	6.25	13.75	28.73	41.27
GPT-4o-1120	5.63	16.25	11.25	46.88
DeepSeek-R1	3.13	8.75	9.38	11.25
Claude 3.7 Sonnet (CoT)	2.50	8.13	11.88	21.25
Nova Pro	3.13	9.38	3.13	15.63
Yi-lightning	0.00	5.00	3.75	13.13
StepFun-2-16k	3.75	8.75	9.38	10.63
Minimax-Text-01	4.38	5.00	7.50	6.88
Hunyuan Turbo-S	8.75	11.25	21.88	21.88
QwQ-32B-Preview	10.00	10.63	14.38	12.50
Ernie-4.5	6.88	12.50	16.00	28.75
DeepSeek-V3	3.13	13.13	11.88	21.25
Gemini-2.0 Flash-0121 (CoT)	4.38	9.38	11.88	23.75
GLM-4-Plus	4.38	8.75	10.00	26.25
Mistral-Large-2	4.38	15.63	13.13	32.50
Qwen-max-0125	8.13	12.50	12.50	15.63
Qwen-2.5-7B-Instruct	6.88	5.63	5.63	9.38
Qwen-2.5-14B-Instruct	10.63	14.38	11.25	13.13
GPT-4o-mini-0718	10.63	23.13	6.25	11.88
Avg. Increase	N/A	+7.12($\pm$4.91)	+8.02($\pm$6.42)	+17.53($\pm$12.21)

Table 9: The scores for "no solution" problems and possible fixes, sorted by average score on such of problems. Claude 3.7 Sonnet and Gemini-2.0 Flash-0121 are without long CoT. It is clearly shown that without any fixes, the average score for "no solution" problems is extremely low, showing the firm belief of LLMs that the given problem is solvable. While some LLMs, such as GPT-4.5-Preview, can be effectively corrected by adding "Forced Correct" (FC) prompts and other "no solution" problems as 1-shot, other LLMs such as DeepSeek-R1 are still very stubborn.

Model Name	Original Bo5	Modified Bo5	Original + FC	Modified + FC
OpenAI-o1-1217	93.67	43.03	94.30	56.96
DeepSeek-R1	92.41	34.81	92.41	39.87
Hunyuan Turbo-S	92.41	26.58	91.14	23.42
GPT-4.5-Preview	91.14	38.60	87.97	49.37
OpenAI-o3-mini-high	91.14	34.81	91.77	39.87
Gemini-2.0 Flash-0121 (CoT)	91.14	32.91	87.97	41.14
Gemini-2.0 Pro-0205	91.14	32.91	87.97	41.14
Claude 3.7 Sonnet	91.14	39.87	86.08	49.37
Claude 3.7 Sonnet (CoT)	90.51	37.34	90.51	42.41
Ernie-4.5	88.61	26.58	87.34	29.11
GLM-4-Plus	86.70	29.11	82.27	31.01
GPT-4o-1120	86.70	29.11	81.65	44.94
Qwen-max-0125	85.44	36.08	84.17	37.97
DeepSeek-V3	84.81	33.54	84.17	40.51
StepFun Step-2-16k	84.81	27.85	82.28	28.48
Yi-Lightning	84.81	25.32	85.44	31.01
QwQ-32B-Preview	84.17	39.87	84.17	37.97
Gemini-2.0 Flash-0121	84.17	32.91	70.89	36.08
Minimax-Text-01	82.91	31.64	84.17	26.58
Claude 3.5 Sonnet	82.28	32.91	83.54	41.14
Qwen-2.5-14B-Instruct	81.65	29.75	81.65	30.38
Mistral-Large-2	79.11	30.37	72.15	34.81
Nova-Pro	78.48	30.37	79.11	35.44
GPT-4o-mini-0718	75.95	29.74	74.68	31.01
Qwen-2.5-7B-Instruct	56.32	23.41	53.80	22.78
Avg. Decrease	N/A	-52.89($\pm$6.60)	-2.00($\pm$3.23)	-48.35($\pm$7.68)

Table 10: Best-of-5 (Bo5) Results on text-based problems of RoR-Bench; the conclusion is similar to that with average score.

Model Name	Original Bo5	Modified Bo5	Original + FC	Modified + FC
OpenAI-o1-1217	98.25	29.82	96.49	42.11
GPT-4.5-Preview	96.49	22.81	82.46	43.86
GPT-4o-1120	91.23	19.30	89.47	31.58
Gemini-2.0 Flash-0121 (CoT)	84.21	43.86	66.67	49.12
Gemini-2.0 Pro-0205	78.95	36.84	73.68	42.11
Claude 3.7 Sonnet (CoT)	78.95	49.12	80.70	56.14
GPT-4o-mini-0718	73.68	35.09	80.70	29.82
Claude 3.5 Sonnet	71.92	45.61	61.40	49.12
Qwen2.5-VL-max	70.18	42.11	66.67	42.11
Qwen2.5-VL-72B	70.18	42.11	64.91	42.11
GLM-4v-Plus	68.42	43.86	64.91	42.11
Claude 3.7 Sonnet	66.67	45.61	63.15	54.39
Nova-Pro	64.91	57.89	71.93	38.60
SenseChat-Vision	64.91	43.86	75.44	42.11
StepFun-1v-32k	64.91	33.33	68.42	28.07
Gemini-2.0 Flash-0121	64.91	30.17	53.68	35.79
Qwen2.5-VL-7B	59.65	47.37	61.40	40.35
Avg. Decrease	N/A	-35.27($\pm$19.49)	-2.73($\pm$7.67)	-32.88($\pm$13.38)

Table 11: Best-of-5 (Bo5) Results on vision-based problems of RoR-Bench; the conclusion is similar to that with average score.

Model Name	Original Score	Modified Score	Original + FC	Modified + FC
Hunyuan Turbo-S	88.60	19.62	87.97	17.72
OpenAI-o3-mini-high	86.08	28.48	83.54	29.74
DeepSeek-R1	86.08	18.99	88.61	27.22
OpenAI-o1-1217	85.44	31.01	88.61	40.51
Gemini-2.0 Flash-0121 (CoT)	84.81	23.42	79.75	24.68
GPT-4.5-Preview	83.54	26.58	77.22	36.08
Claude 3.7 Sonnet (CoT)	81.65	24.05	78.48	39.24
Ernie-4.5	81.65	21.52	80.38	23.42
Gemini-2.0 Pro-0205	78.48	24.68	41.14	32.91
Gemini-2.0 Flash-0121	78.48	22.78	60.76	25.95
Qwen-max-0125	75.95	20.25	75.32	23.42
GLM-4-Plus	75.32	15.82	70.89	22.78
Claude 3.7 Sonnet	74.68	25.32	70.89	35.44
GPT-4o-1120	74.05	23.42	70.89	25.95
Claude 3.5 Sonnet	73.42	23.42	66.46	31.01
QwQ-32B-Preview	72.15	18.99	68.99	22.79
DeepSeek-V3	70.25	17.09	72.15	25.95
Minimax-Text-01	69.62	18.99	65.82	20.25
StepFun Step-2-16k	69.62	17.72	72.15	21.52
Yi-Lightning	68.35	13.92	62.66	22.79
Qwen-2.5-14B-Instruct	65.82	19.62	66.56	20.89
Mistral-Large-2	63.92	18.99	52.53	27.84
Nova-Pro	61.39	20.25	57.59	18.99
GPT-4o-mini-0718	61.39	19.62	60.76	20.89
Qwen-2.5-7B-Instruct	37.34	10.76	34.81	16.46
Avg. Decrease	N/A	-52.91($\pm$8.67)	-4.53($\pm$8.18)	-47.75($\pm$9.52)

Table 12: Results on text-based problems of RoR-Bench with greedy decoding; the conclusion is similar to that with temperature 0.7.

Model Name	Original Score	Modified Score	Original + FC	Modified + FC
GPT-4.5-Preview	94.74	14.04	71.93	42.11
OpenAI-o1-1217	91.23	24.56	94.74	26.32
GPT-4o-1120	85.96	14.04	84.21	26.32
Gemini-2.0 Flash-0121 (CoT)	73.68	28.07	63.15	42.11
Gemini-2.0 Flash-0121	71.93	28.07	57.89	40.36
Gemini-2.0 Pro-0205	70.18	35.09	68.42	40.35
GLM-4v-Plus	68.42	43.86	66.67	42.11
GPT-4o-mini-0718	68.42	31.58	80.70	28.07
Claude 3.7 Sonnet (CoT)	68.42	31.58	64.91	43.86
Qwen2.5-VL-72B	66.67	36.84	66.67	42.11
Claude 3.5 Sonnet	64.91	33.33	59.65	45.61
Qwen2.5-VL-max	63.16	36.84	66.67	42.11
SenseChat-Vision	59.65	35.09	70.18	38.60
StepFun-1v-32k	59.65	33.33	64.91	28.07
Nova-Pro	57.89	50.88	70.18	38.60
Claude 3.7 Sonnet	56.14	31.58	61.40	40.35
Qwen2.5-VL-7B	52.63	38.60	59.65	42.11
Avg. Decrease	N/A	-36.84($\pm$19.86)	-0.10($\pm$9.42)	-30.85($\pm$15.32)

Table 13: Results on image-based problems of RoR-Bench with greedy decoding; the conclusion is similar to that with temperature 0.7.