Rethinking LLM-as-a-Judge: Biases and Limitations in Evaluating Chain-of-Thought Outputs

Elias Mouaheb[1], Ali Ismail Fawaz[1], Christophe Cudel[1], Jonathan Weber[1], and Germain Forestier[1,2]

[1] IRIMAS, Université de Haute-Alsace, France
elias.mouaheb,ali-el-hadi.ismail-fawaz,christophe.cudel,
jonathan.weber,germain.forestier@uha.fr
[2] FIT, Monash University, Australia

Abstract. The widespread adoption of Large Language Models (LLMs) has created a growing demand for scalable and reliable evaluation methodologies. While human evaluation remains the gold standard, it is costly, time-consuming, and often requires domain expertise. Consequently, LLM-as-a-Judge approaches have emerged as a popular alternative and are increasingly used to assess model outputs across a wide range of tasks. The recent development of o1-like language models, capable of generating long chain-of-thought traces before producing a final answer, has introduced new challenges for evaluation. Assessing the quality of a reasoning process requires more than verifying the correctness of the final output, raising the question of whether LLM judges can reliably evaluate complex reasoning traces. In this work, we conduct a controlled study of LLM-as-a-Judge for evaluating mathematical reasoning, focusing on how changes in the presentation of reasoning traces affect model judgments. Our experiments investigate whether enabling additional reasoning makes LLM judges more robust to changes in the presentation of reasoning traces. Using PRM800K, we construct controlled pairs consisting of the same solution in an uncorrupted form and a corrupted form containing a controlled number of steps with negative human annotations, and compare judgments with and without reasoning. We then perturb the presentation of these solution pairs by changing their order and their surface formulation, allowing us to measure whether judgments remain consistent when the underlying reasoning quality is unchanged. These findings challenge the reliability of existing LLM-as-a-Judge methodologies for reasoning evaluation and highlight the need for more rigorous and robust evaluation frameworks for reasoning-capable language models.

Keywords: LLM-as-a-Judge; Large Language Models; Chain-of-Thought; Reasoning Traces; Evaluation

1 Introduction

Large Language Models (LLMs) have become increasingly capable general-purpose systems, leading to their widespread use across a broad range of tasks. As their capabilities expand, reliably evaluating their outputs has become an important challenge. Human evaluation remains a key reference, but is costly, time-consuming, and may require substantial domain expertise. This has motivated the use of LLMs themselves as evaluators, commonly referred to as LLM-as-a-Judge. However, previous work has shown that LLM-based judgments can be affected by systematic biases, raising concerns about their reliability.

Recent reasoning-oriented models introduce an additional dimension to this problem. By allocating additional computation to inference before producing a final answer, these models have achieved substantial improvements on challenging reasoning tasks. This raises the question of whether additional reasoning also makes judges more robust to factors that are unrelated to the underlying quality of the evaluated reasoning. In particular, a reliable judge should evaluate the quality of a reasoning process rather than relying on properties of how that process is presented.

Evaluating this distinction requires controlled evaluation settings. For reasoning tasks, this can be studied in a controlled setting by comparing the same solution in an uncorrupted form and a corrupted form, while independently manipulating properties that should be irrelevant to their relative quality, such as their position or surface formulation. If such changes systematically affect the judge's preference, they reveal a discrepancy between the quality of the evaluated reasoning and the factors driving the judgment.

In this work, we investigate the robustness of LLM-as-a-Judge for mathematical reasoning using PRM800K, a dataset containing step-level human correctness annotations for solutions to MATH problems [1]. We construct pairs from the same PRM800K solution, using an uncorrupted version containing only positively annotated reasoning steps and a corrupted version containing a controlled number of negatively annotated steps, and compare the judgments of reasoning and non-reasoning models under controlled perturbations of their presentation. This allows us to study how different sources of bias affect the evaluation of reasoning and whether enabling additional reasoning changes the robustness of LLM judges.

2 Related Work

LLMs and reasoning models. Large language models [2, 3] have achieved strong performance across a wide range of tasks. Chain-of-thought prompting [4] showed that eliciting intermediate reasoning can substantially improve performance on complex reasoning tasks. More recently, large reasoning models (LRMs), such as OpenAI's o1 [5], DeepSeek-R1 [6], and QwQ [7], have extended this paradigm by allocating additional computation to inference, achieving strong performance on challenging mathematical and programming problems.

LLM-as-a-Judge and judging biases. LLM-as-a-Judge has emerged as a scalable alternative to human evaluation [8]. However, LLM judges have been shown to exhibit systematic biases, including position, verbosity, and self-enhancement biases [8], as well as biases related to order, authority, salience, and distraction [9, 10]. These findings suggest that LLM judgments can be influenced by properties of the evaluated content that are not necessarily related to its underlying quality.

Judging biases in reasoning models. Recent work has begun to investigate whether reasoning capabilities affect these biases. Wang et al. [11] systematically compared LLMs and LRMs across several judging biases, including bandwagon, authority, position, and distraction biases, and found that LRMs remain susceptible to such effects while also exhibiting a superficial reflection bias. Our work considers a complementary setting: rather than evaluating general preference or factual judgments, we study the evaluation of mathematical reasoning processes using PRM800K, which provides human step-level correctness annotations [1]. This controlled setting allows us to derive pairs from the same reasoning trace with a controlled difference in the number of negatively annotated

steps, and to test whether changes in their presentation affect model judgments while comparing models with and without reasoning.

3 Experimental Setup

3.1 Models

For the purpose of these experiments, we use 2 different LLMs. The first one is *Qwen3.8-27B* [12], specifically the FP8 checkpoint released in August 2026. The model exhibits strong capabilities for its size, making it a prime contender among open-weights models. For the remainder of this paper, it will be referred to as **Qwen**. The second model used is *Gemma-4-26B-A4B-it* [13], the instruct version of the open-weight model released in the first half of 2026. We will refer to it as **Gemma**. Unlike Qwen, this model does not provide multiple reasoning levels, but only a binary option to enable or disable reasoning. To facilitate comparison with Qwen, we will refer to the enabled setting as "High reasoning".

Finally, all experiments are run on a H100 96GB setup, equipped with two Intel Xeon Platinum 8452Y processors (72 physical cores, 144 hardware threads in total) running Ubuntu 24.04 LTS. The software environment was based on Python 3.12. Large language model inference is performed using vLLM [14] (Docker image vllm/vllm-openai:v0.11.2). We use publicly available datasets and models released by their original creators.

We do not redistribute the datasets or model weights. Our released code, which will be released upon acceptance, only contains scripts to reproduce our experiments and references the original sources.

3.2 Datasets

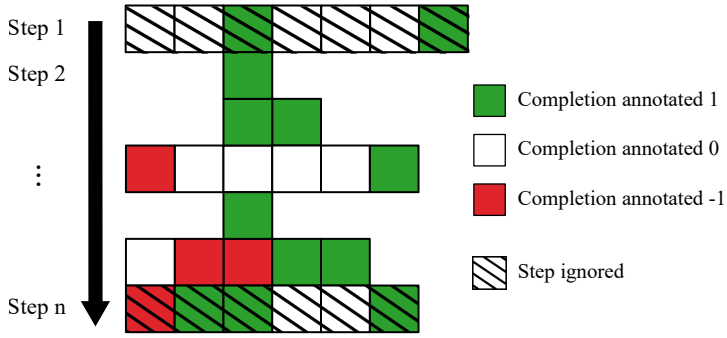
PRM800K [1] is a dataset designed for step-level reasoning evaluation. Each solution consists of multiple reasoning steps, with alternative completions for individual steps annotated by human labelers. In PRM800K, a *step* represents a single reasoning increment in the solution process, while a *completion* is a candidate continuation text proposed for that step, potentially among several alternatives. Each completion is annotated with a label of 1, −1, or 0. We use the positive and negative annotations to distinguish completions that support or hinder the reasoning process, while ignoring ambiguous annotations labeled 0.

In our experiments, we use only the *phase 1* files, namely `phase1_train` and `phase1_test`; the *phase 2* files are not considered, as they serve different purposes. We first retain only solutions whose `finish_reason` is `solution`, indicating that the generation terminated after reaching a final answer. We then verify that the final answer extracted from each reasoning trajectory matches the ground-truth answer provided by the dataset, using the grader module from the official PRM800K repository [3]. This second filtering step removes 18 solutions whose final answer does not match the ground truth, leaving 790 valid solutions for our experiments. For each set of parameters, we perform 5 runs to mitigate the effect of potential outliers.

We introduce the notion of *corruption level* to quantify the amount of negatively annotated reasoning introduced into a solution. We define the corruption level as the number of distinct steps in a solution for which the selected completion is annotated "−1", meaning an irrelevant completion. The first and last steps are excluded from this

[3] https://github.com/openai/prm800k

Fig. 1. Schema of a solution with a corruption level of 2. First and last steps are not counted as part of the corruption level. A bloc is a completion, while a row is a step.



computation (following the C²-Faith [15] method), as they are not considered part of the corruptible reasoning process.

Fig. 1 illustrates a PRM800K solution with a corruption level of 2, where only the fourth and sixth steps contain an irrelevant completion.

When comparing solutions with a given corruption level, we only consider samples belonging to that specific category, rather than using all available solutions from lower corruption levels. For instance, when evaluating the impact of a corruption level of 3 compared to fully correct solutions, we only use the 374 solutions belonging to corruption level 3, even though 601 solutions are available with corruption level 1, i.e. at least one step with a completion annotated -1.

4 Experiments

4.1 Position bias

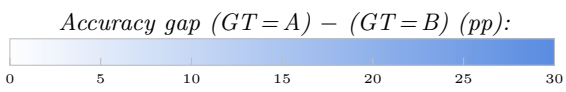
The first presentation factor we consider is the position of the two candidate solutions. For each selected PRM800K solution, we compare its uncorrupted version with a corrupted version at a given corruption level, thereby keeping the underlying problem and solution fixed while controlling the amount of negatively annotated reasoning. We pass each of those tuple in the prompt of our LLM, with the rest of the prompt being comprised of the final answer to the two solutions, the question leading to the answer, and instructions for the LLM, telling it to judge which of the two solutions is better. The criteria is purposely abstract, as to not steer the model too much. Said model can vote "A" if the first solution in the prompt is better, "B" for the second, "N" if both are good, and finally "X" if neither are.

We then vary only the order in which the two versions are presented to the judge, allowing us to measure the effect of position while keeping the evaluated reasoning unchanged.

- **No swap**: The ground truth solution appears first in the prompt.
- **Always swap**: The corrupted solution appears first in the prompt.
- **Alternate**: The order switches every prompt.
- **Random**: The order is randomly decided every prompt.

Table 1. Rate at which *Qwen3.8-27B* correctly predicts the GT class, broken down by shuffle order, corruption level (CL), and reasoning mode. GT = A (resp. GT = B) denotes samples whose ground-truth is the first (resp. second) candidate. *No swap* and *Always swap* are merged for readability; GT = A values stem from *No swap* and GT = B values from *Always swap*. Row background indicates the accuracy gap (GT = A) − (GT = B) in percentage points (see legend below).

Shuffle order	CL	Reasoning mode	GT = A (%)	GT = B (%)
No swap / Always swap	1	No reasoning	3.7	0.5
		High reasoning	70.6	66.7
	2	No reasoning	13.7	2.4
		High reasoning	92.6	84.3
	3	No reasoning	19.1	2.9
		High reasoning	95.5	89.3
Alternate	1	No reasoning	4.2	0.0
		High reasoning	73.4	62.4
	2	No reasoning	12.8	2.0
		High reasoning	92.8	80.6
	3	No reasoning	20.4	2.4
		High reasoning	98.4	86.0
Random	1	No reasoning	3.6	0.8
		High reasoning	70.3	64.2
	2	No reasoning	12.6	1.6
		High reasoning	91.4	86.9
	3	No reasoning	23.7	2.2
		High reasoning	95.9	87.4



The Table 1 shows the obtained results using the Qwen LLM. Regardless of the reasoning level used, accuracy increases with the corruption level. This is consistent with the intuition that larger corruption levels create a larger difference between the two solutions, potentially making them easier to distinguish.

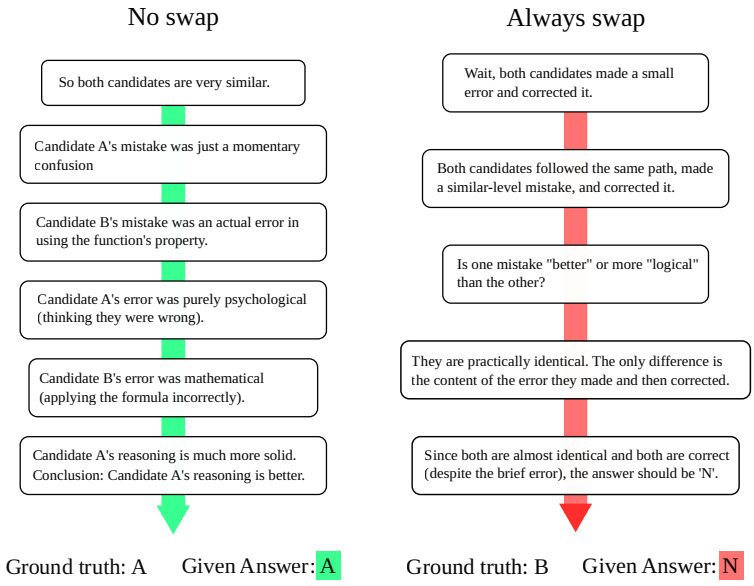
Another noticeable observation is how low the accuracy is overall when no reasoning is used. The model struggles to pick a side and tends to overly rely on the "N" vote [4]. This behavior diminishes greatly with reasoning enabled.

Finally, we observe a sizable position bias when reasoning is disabled, regardless of the model or corruption level used, with the accuracy when the ground truth is first in the prompt ("No swap") being always superior than the accuracy with the same parameters but the ground truth appearing second in the prompt ("Always swap"). For Gemma, the accuracy drops by ∼50 percentage points when putting the ground truth second in the prompt. Qwen sees its accuracy drop by ∼10 points.

These results raise the central question of whether the additional reasoning computation also reduces sensitivity to presentation order. The results indicate that it only partially does. While the accuracy, as stated previously, increases, the gap between "No swap" and "Always swap" stays in the ∼6 points range for Qwen, although it shrinks to ∼11 points for the Gemma model. The corruption level also plays a part

[4] See the corresponding confusion matrices in the accompanying Figshare repository: Mouaheb, Elias (2026), Confusion matrices: https://doi.org/10.6084/m9.figshare.33995610.v2.

Fig. 2. Excerpts of reasoning traces steps from *Gemma* on the same PRM800K solution tuple. The excerpts on the left are taken from a run where the ground truth appears first in the prompt, while right has the ground truth appear second in the prompt given to the judge (here Gemma).



in this, as the accuracy gap for "High reasoning" at CL = 1 tends to be bigger than for "No reasoning" (Table 1), whereas at higher corruption level, the opposite occurs, with "No reasoning" having in most cases a bigger accuracy gap between the ground truth being A and the ground truth being B. This gap especially stands out in Gemma's results (Table 2). The confusion matrices with the full results are available in the accompanying Figshare repository [4].

Looking at the reasoning traces does not help identify the cause of the bias. We take as example a solution with a corruption level of 2 where the 5 runs in "Always swap" saw Gemma answer with "N" instead of "B", whereas the 5 runs in "No swap" saw the model correctly answer "A". Steps of the reasoning traces are shown in Fig. 2.

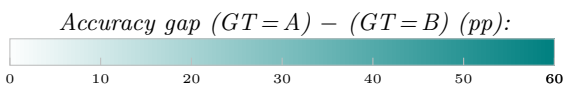
4.2 Formulation bias

Position is not the only presentation factor that can vary without changing the underlying reasoning. We therefore consider a second perturbation: the surface formulation of the solutions.

To isolate the effect of surface formulation, we use *Qwen3.8-27B* to reformulate each solution while preserving its underlying mathematical content, including any existing errors. We provide the model with the complete original solution and prompt it to rewrite it in a casual, everyday style without altering its meaning or mathematical expressions. In particular, the model is instructed to preserve the original reasoning, even when it contains errors, and to leave all mathematical expressions unchanged. The reformulation is therefore deliberately conservative, with the aim of minimizing the risk of modifying either the mathematical content or the overall meaning of the solution.

Table 2. Rate at which *Gemma4-27B* correctly predicts the GT class, broken down by shuffle order, corruption level (CL), and reasoning mode. GT = A (resp. GT = B) denotes samples whose ground-truth is the first (resp. second) candidate. *No swap* and *Always swap* are merged for readability; GT = A values stem from *No swap* and GT = B values from *Always swap*. Row background indicates the accuracy gap (GT = A) − (GT = B) in percentage points (see legend below).

Shuffle order	CL	Reasoning mode	GT = A (%)	GT = B (%)
No swap / Always swap	1	No reasoning	45.5	11.4
		High reasoning	84.9	74.0
	2	No reasoning	76.3	20.9
		High reasoning	98.0	86.3
	3	No reasoning	85.0	35.1
		High reasoning	97.9	86.0
Alternate	1	No reasoning	49.3	10.7
		High reasoning	88.0	70.9
	2	No reasoning	79.0	22.3
		High reasoning	97.8	83.9
	3	No reasoning	88.8	34.2
		High reasoning	99.6	82.4
Random	1	No reasoning	41.4	12.1
		High reasoning	84.6	72.0
	2	No reasoning	78.9	19.6
		High reasoning	98.3	86.0
	3	No reasoning	87.1	36.8
		High reasoning	99.0	86.8



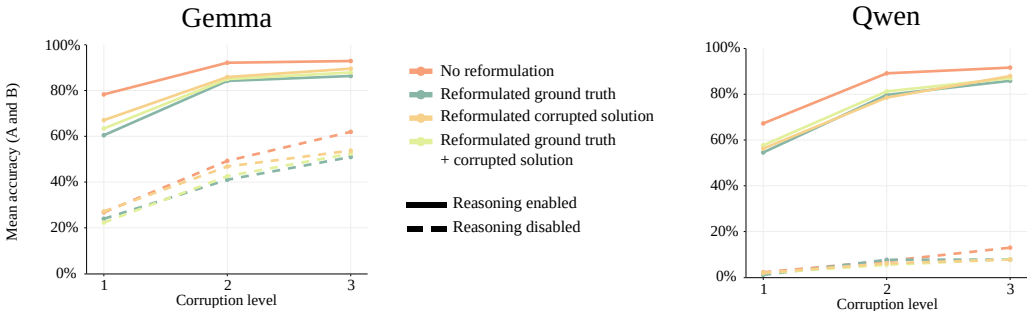
We compare in Fig. 3 the experiments obtained by reformulating different parts of the prompt. As in the position experiment, reasoning-enabled judges achieve higher overall accuracy. We now examine whether this improvement also translates into greater robustness to changes in surface formulation.

We use as our metric the mean of the accuracies obtained for "A" and "B". Regardless of the model or the type of reformulation, this mean accuracy increases with the corruption level. This behavior is expected, as increasing the corruption level makes the distinction between a fully correct solution and a corrupted one more pronounced.

For Gemma, the different reformulation types follow the same ordering for both the reasoning-enabled and non-reasoning models, with the "No reformulation" condition consistently achieving the highest accuracy in both settings. For Qwen, this ordering is less consistent across reasoning settings. Nevertheless, "No reformulation" still achieves the highest accuracy for both the reasoning-enabled and non-reasoning models. The other reformulation types yield relatively similar results for Qwen, indicating a measurable difference between using no reformulation and applying any of the considered reformulations. This pattern is observed for both Qwen and Gemma.

Interestingly, reformulating the ground truth alone leads to the largest decrease in accuracy among the considered reformulation settings, even compared to reformulating both the ground truth and the corrupted solution. This observation holds across corruption levels and models, with the exceptions of Gemma at corruption level 1 and Qwen at corruption level 2. One possible explanation is that reformulating the ground truth alone

Fig. 3. Summary of reasoning traces steps from *Gemma* on the same solution tuple. Left has the ground truth appear first in the prompt, while right has the ground truth appear second.



introduces a stylistic mismatch between the ground truth and the corrupted solution, potentially making their comparison more difficult for the judge models. In contrast, when both the ground truth and the corrupted solution are reformulated, their styles remain relatively similar, which may reduce this source of difficulty.

Taken together, these results suggest that enabling reasoning improves the judge's ability to distinguish the uncorrupted and corrupted solutions, but does not necessarily eliminate its sensitivity to how those solutions are presented. Rather, the persistence of the same reformulation-dependent patterns suggests that reasoning may improve overall performance while leaving the underlying sensitivity to this bias largely intact.

5 Conclusion

In this work, we investigated the robustness of LLM-as-a-Judge for evaluating mathematical reasoning, focusing on whether additional reasoning helps judges resist presentation biases. Using PRM800K, we constructed controlled comparisons between original solutions and corrupted versions of the same solutions, varying the amount of negatively annotated reasoning introduced. We then evaluated two presentation factors, solution position and surface formulation, with judges operating with and without additional reasoning.

Our results show that additional reasoning can improve the judge's ability to distinguish between solutions of different quality, but does not necessarily remove sensitivity to presentation factors. This suggests that reasoning capability and robustness to presentation biases are distinct properties. Overall, our experiments highlight the importance of testing whether presentation changes affect LLM judges, including when they are equipped with additional reasoning.

6 Acknowledgments

This work received support from the French government under the France 2030 plan, as part of the "Investissements d'Avenir" program, under grant reference *ANR-17-NCUN-0014*.

References

1. Lightman, H. et al.: Let's verify step by step. In Kim, B. et al., eds.: International Conference on Learning Representations. Volume 2024. (2024) 39578–39601
2. Brown, T. et al.: Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., eds.: Advances in Neural Information Processing Systems. Volume 33., Curran Associates, Inc. (2020) 1877–1901
3. Ouyang, L. et al.: Training language models to follow instructions with human feedback. Advances in neural information processing systems **35** (2022) 27730–27744
4. Wei, J. et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35** (2022) 24824–24837
5. OpenAI et al.: Openai o1 system card (2026)
6. Guo, D. et al.: Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. Nature **645**(8081) (2025) 633–638
7. Team, Q.: Qwq-32b: Embracing the power of reinforcement learning (March 2025)
8. Zheng, L. et al.: Judging llm-as-a-judge with mt-bench and chatbot arena. In Oh, A. et al., eds.: Advances in Neural Information Processing Systems. Volume 36., Curran Associates, Inc. (2023) 46595–46623
9. Koo, R. et al.: Benchmarking cognitive biases in large language models as evaluators. In Ku, L.W., Martins, A., Srikumar, V., eds.: Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, Association for Computational Linguistics (August 2024) 517–545
10. Ye, J. et al.: Justice or prejudice? quantifying biases in llm-as-a-judge. arXiv preprint arXiv:2410.02736 (2024)
11. Wang, Q. et al.: Assessing judging bias in large reasoning models: An empirical study (2025)
12. Qwen Team: Qwen3.8-Max: A new bar for coding and cowork (August 2026)
13. Team, G.: Gemma 4 technical report (2026)
14. Kwon, W. et al.: Efficient memory management for large language model serving with pagedattention. In: Proceedings of the 29th Symposium on Operating Systems Principles. SOSP '23, New York, NY, USA, Association for Computing Machinery (2023) 611–626
15. Mittal, A., Arike, R.: C2-faith: Benchmarking LLM judges for causal and coverage faithfulness in chain-of-thought reasoning. In Mille, S. et al., eds.: Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM), San Diego, California, USA (July 2026) 678–691