

Finding Answers in Thought Matters: Revisiting Evaluation on Large Language Models with Reasoning

Anonymous ACL submission

Abstract

Evaluating generative models, such as large language models (LLMs), commonly involves question-answering tasks where the final answer is selected based on probability of answer choices. On the other hand, for models requiring reasoning, the method of answer extraction plays a critical role. We reveal that the performance of reasoning models and their final answer distributions are highly sensitive to the answer extraction algorithm employed. In order to mitigate this, we propose a basic framework: ANSWER REGENERATION. The method uses an additional model inference, providing the prior input and output prefaced by the prompt "Answer:". The final answer is then selected or extracted from the regenerated output. In the experiments, we show that this extraction-rule-agnostic approach exhibits improved performance and enhanced robustness. Furthermore, we have applied this framework to general math problems and open-ended question answering tasks. We believe our analysis and this framework could offer a more reliable results for model evaluation.

1 Introduction

The conventional approach for generating answers from large language models (LLMs) involves selecting the answer choice with the highest probability when conditioned on the input prompt and each choice following a specific prefix, such as "Answer:" (Hendrycks et al. (2021); Liang et al. (2023); OpenCompass Contributors (2023); Habib et al. (2023); *inter alia*). For tasks without answer choices, prior work has relied on rule-based extraction (e.g., searching for "Answer: X" or "answer is X"), model judges for semantic similarity, or human evaluation (Kamalloo et al. (2023); Wei et al. (2024); Chandak et al. (2025); Chen et al. (2025); *inter alia*). However, reasoning-powered LLMs need to output their reasoning process (Chain-of-Thought (CoT)) (Wei et al., 2022) to leverage their

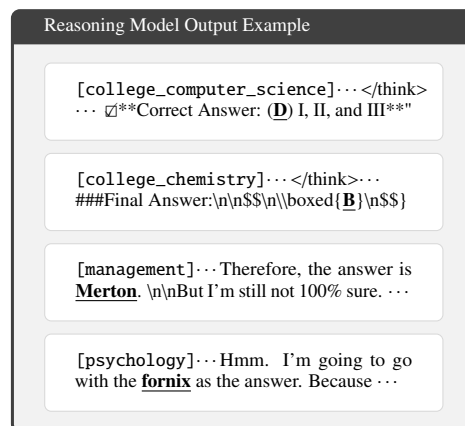


Figure 1: Examples illustrating the difficulties in extracting final answers from reasoning models' outputs. Although the benchmark is designed with multiple-choice questions, models frequently generate answers in a free-text format, which complicates automated evaluation.

full potential. This detailed, linguistically diverse output complicates traditional evaluation. Specifically, it prevents the use of methods based on the probability of specific answer choices and limits the applicability of most LLM-as-a-judge (Zheng et al., 2023) evaluations. This shift introduces a new, critical challenge: *how to reliably find the answer from the detailed output that includes all the reasoning steps matters.*

Conventional rule-based approach suffers from a fundamental flaw: heuristic rules cannot account for all possible answer formats. Figure 1 illustrates examples from multiple-choice question answering benchmark MMLU (Hendrycks et al., 2021). A single model can use different formats in its responses, sometimes boxing the answer in brackets (i.e., `\boxed{ }`) or answering the option text in various formats (e.g., "Merton", "fornix") instead of the option label (e.g., "(D)"). Furthermore, the formats can vary significantly between different models and even across different types of benchmarks, such as multiple-choice, math, and open-ended questions. This means that optimal extraction rules need to be created and tuned for every individual model

and benchmark (e.g., rules for options, numbers, or word(s)), which makes the process difficult and even affects the reproducibility of model results.

In this paper, we first empirically demonstrate the impact of answer extraction rules on reasoning-powered model (Section 4). We then introduce ANSWER REGENERATION, a simple, generation-based framework designed to alleviate the dependency on specific answer extraction rules (Section 5). Instead of relying on complex extraction rules, our method utilizes an additional inference step to prompt the model to regenerate its final answer. It allows us to use probability-based answering for choices or extract the answer from a simplified output, resulting in practical effectiveness.

Our experiments reveal that model performances are highly sensitive to the extraction rules employed. Depending on the rules, distinct answers—no answers at all in some cases—may be extracted from the same LLM response. On the other hand, ANSWER REGENERATION consistently outperforms the handcrafted rule-based extractions, improving both in benchmark scores and human evaluation results. Our method also achieves intuitive model rankings, where larger models are shown to outperform smaller ones. We demonstrate ANSWER REGENERATION significantly reduces the dependency on specific answer extraction rules, thereby improving robustness and reproducibility of model evaluations. Furthermore, we apply our framework to diverse tasks, including complex multiple-choice question answering, short-answer math problems, and open-ended question answering. In all cases, our method proves to be a plausible and effective approach for the fair evaluation of reasoning models.

Our contributions in this work are as follows:

- We empirically investigate the sensitivity of reasoning-powered LLMs to rule-based answering, revealing a strong dependency on the choice of answer extraction algorithm.
- We propose the generation-based framework ANSWER REGENERATION. It achieves (1) superior performance both on quantitative and qualitative scores compared with handcrafted rules, (2) intuitive model rankings, and (3) significantly enhanced robustness against answer inconsistency and incomplete outputs.
- We demonstrate the generalizability and effectiveness of our framework across diverse tasks, confirming its plausibility for more robust and fair model evaluations.

2 Related Work

A growing body of work shows that LLM performance can vary drastically with small changes in prompt format, even when the underlying semantics are equivalent (Sclar et al., 2024; He et al., 2024; Alzahrani et al., 2024). Consequently, Polo et al. (2024); Mizrahi et al. (2024) proposed the methods to mitigate the effect of prompt variations. While the previous research focused on *input-level* prompt variations and their impact on model evaluation, we focus on *output-level* final answer variations from reasoning LLMs.

Therefore, it is noteworthy to find out how recent LLM evaluations handle outputs from reasoning models. A number of open evaluation frameworks typically support (1) probability-based answering for multiple-choice tasks or (2) simple heuristic post-processing for free-form generations, involving only de-capitalization or blank-space normalization. Details on the implementations of MMLU Hendrycks (Hendrycks et al., 2021), HELM (Liang et al., 2023), OpenCompass (OpenCompass Contributors, 2023), and lighteval (Habib et al., 2023) can be found in the Appendix A.1.

lm-evaluation-harness (Biderman et al., 2024) has become the de facto community standard for reproducible LLM evaluation. Generative tasks use string-match with optional regular expressions or rule-based normalizers. While recent templates support CoT prompting, the final answer is still recovered via simple patterns (e.g., "Answer: X"), or a last-capital-letter heuristic. As we will demonstrate, such extraction rules can swing scores and even reorder model rankings.

Recent works also tackled the importance of answer extraction, especially for instruct-tuned models. However, the outputs of reasoning models are significantly complicated compared with CoT-invoked outputs of instruct-tuned models to be investigated by simple regular expressions (Molfese et al., 2025). xFinder (Yu et al., 2024) attempted to mitigate the dependency on the rules by fine-tuning LLMs specialized to answer extractions. However, it becomes that the evaluation is not only dependent on the performance of answer extractor but also the extractors show limited performance on the long reasoning outputs in our experiments. Without additional models and fine-tuning, we introduce a lightweight method to reduce the reliance on the fragile extraction methods and provides a more faithful evaluation of reasoning models' abilities.

3 Experiment Setup

The experiments are designed to highlight current problems associated with finding answers in reasoning models’ output (Study 1 in Section 4) and then assess the validity of introduced method ANSWER REGENERATION (Study 2 in Section 5).

We utilize **lm-evaluation-harness** toolkit for its simplicity in customizing the post-processing rules. **MMLU** (Hendrycks et al., 2021) benchmark is primarily used, given its widely adoption for evaluating LLMs’ knowledge¹. The multiple-choice format of MMLU serves as a foundational task that simplifies the answer extraction process for our initial analysis. We then extend our evaluation to more complex tasks **MMLU-Pro** (Wang et al., 2024), the mathematical reasoning **GSM8K** (Cobbe et al., 2021) and the open-ended question answering **TriviaQA** (Joshi et al., 2017) in Section 6.

We evaluate several open-source reasoning models: Qwen3 families—**Qwen3-32B**, **Qwen3-14B**, **Qwen3-8B** (Yang et al., 2025), along with DeepSeek families—Deepseek-R1-Distill-Llama-8B (referred to as **R1-Llama-8B**), DeepSeek-R1-0528-Qwen3-8B (referred to as **R1-Qwen3-8B**) (DeepSeek-AI, 2025), and other families—**gpt-oss-20b** (OpenAI, 2025), Phi-4-reasoning-plus (referred to as **Phi-4-reason+**) (Abdin et al., 2025). For hyperparameter settings, we adhere to recommended best practices for each model, setting temperature, top-p, and top-k values. Prompt templates are sourced from lm-evaluation-harness, using thinking templates. Fine-tuned models for answer extraction, xFinder (Yu et al., 2024) (qwen1505, llama38it), are selected as a baseline.

4 Study 1: Rule-based Answer Extraction

4.1 Methods

We evaluate 7 reasoning models using 5 different answer extraction methods to investigate how performance changes with extraction algorithms:

strict-match and **flexible-extract** are adapted from lm-evaluation-harness. **strict-match** extracts a precise string such as "answer is X" or "Answer: X" and **flexible-extract** finds multiple-choice options like (A), (B), (C), or (D), located near the end of the text. This is a common and effective approach, as the final conclusion typically follows the reasoning. However, the original implementation

¹We select the original MMLU to better analyze how models handle ambiguous questions, rather than the cleaned MMLU-Redux (Gema et al., 2025).

has tendency to extract the last capital character from any text, which can lead to errors.

instructed-format requires modifying the input prompt to guide the model’s output format. As recommended in Qwen3 technical report, we add a specific instruction to the prompt: "Please show your choice in the answer field with only the choice letter, e.g., "answer": "C"." Rules are then implemented to extract the answer from this specified format.

Further heuristically optimized answer extraction methods are also used: We build upon **strict-match** by creating **answer-is-correct**, which includes variations like "X is the answer" or "X could be the correct answer", addressing the limited scope. We refine **flexible-extract** by developing **last-extract**, which specifically targets the last *single* capital character that appears in the output. All the implementations accounts for minor variations like "***X**", "***Answer:** X" and are designed to select the last match to accommodate potential self-correction within the model’s response.

Our objective is not to declare any one extraction method superior. Instead, we aim to demonstrate the range of performance that can be achieved using widely-used, well-optimized, heuristic extraction algorithms on a given task. Consequently, we show that benchmark performance is not solely dependent on the reasoning model’s ability but is significantly influenced by the chosen answer extraction rules. The exact regular expressions used are described in the Appendix A.2.

4.2 Result

4.2.1 Performance Fluctuation

Figure 2 illustrates how different answer extraction methods affect the performance of models. If an extraction rule fails to find an answer, the response is considered incorrect. The results reveal that model performance fluctuates significantly depending on the extraction method used.

With **strict-match**, the rankings of model performances are Qwen3-8B, Qwen3-32B, Qwen3-14B, R1-Qwen3-8B, R1-Llama-8B, Phi-4-reason+, and gpt-oss-20b in order. The more optimized **answer-is-correct**, derived from **strict-match**, significantly improves the performance of all models. This shifts the ranking to Qwen3-14B, Qwen3-32B, R1-Qwen3-8B, Qwen3-8B, R1-Llama-8B, gpt-oss-20b, and Phi-4-reason+. A similar sensitivity is observed with the other methods. Us-

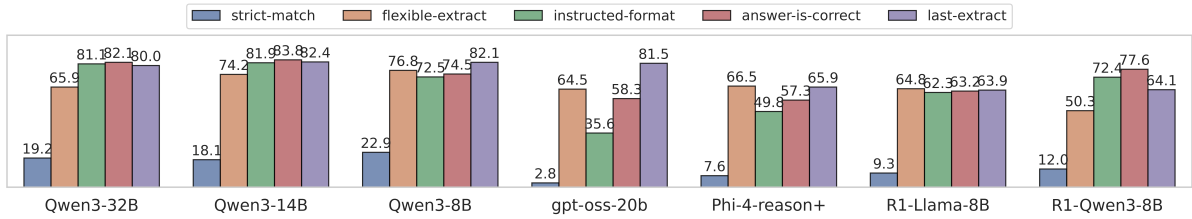


Figure 2: MMLU performance in accuracy evaluated using various answer extraction algorithm. Responses are considered incorrect if the extraction process fails to find an answer.

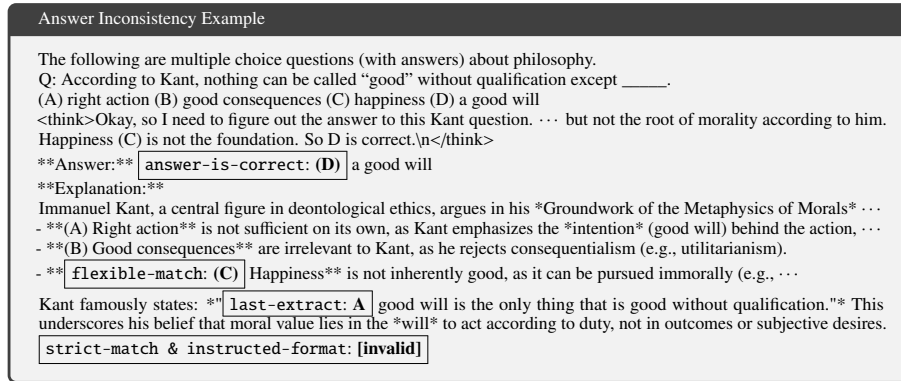


Figure 3: An example from Qwen3-32B output illustrating how the final answer can vary significantly depending on the extraction methods. The graphical boxes and bold text highlight the specific text extracted by each algorithm.

ing flexible-extract, the top models are Qwen-8B, Qwen3-14B, Phi-4-reason+, Qwen-32B, R1-Llama-8B, gpt-oss-20b, and R1-Qwen3-8B. With last-extract, Qwen3-14B performs the best, and the performance of gpt-oss-20b is significantly improved. Interestingly, despite following the recommended best practices for multiple-choice question answering with instructed-format, the performance of models are not impressive compared to other extraction methods.

The results also challenge the common assumption that larger models outperform smaller ones within the same family (i.e. Qwen3-14B and Qwen3-8B outperforms Qwen3-32B). Our analysis indicates that the benchmark scores of reasoning models are highly dependent on the answer extraction method used. These findings suggest that the discrepancies between publicly reported and reproduced performance scores may be due to differences not only in prompt inputs, but also in the specific answer extraction methods, which are not fully disclosed.

4.2.2 Answer Inconsistency

Figure 3 provides a clear example of how different extraction methods handle the same model output, illustrating the problem of answer inconsistency. In this example, strict-match and instructed-format fail. answer-is-correct successfully locates an answer after ****Answer:****. However, flexible-match

extracts option letters (A), (B), and (C) in the explanation and choose the last one (C). The simple yet effective last-extract extracts "A" as the final answer. Furthermore, we observe unstable extraction results due to unexpected LaTeX commands/symbols and the option text without option labels. More examples are presented in the Appendix A.4.

Figure 4 further illustrates this issue by showing how the distribution of extracted answers changes depending on the extraction method used. We observe that the distribution of extracted answers varies significantly. This highlights the crucial role of the extraction method in determining model's final performance, suggesting that the choice of method can introduce bias into the evaluation.

4.2.3 Answering for Incomplete Thinking

Another challenge in extracting answers from reasoning models is the issue of incomplete reasoning (or thinking). Even when we set the maximum generation length to 4,096 tokens, we find that some model outputs lack the </think> token and/or exceed the maximum token length, indicating that the thinking process had not concluded. Table 1 reports the percentage of outputs in this category. Fortunately, this is a relatively small portion of the total outputs except for Phi-4-reason+, primarily caused by repetitions during the model's generation.

We then select the best answer extraction method for each model and measure the correctness of the

	Qwen3-32B					Qwen3-8B					gpt-oss-20b					Phi-4-reason+					R1-Llama-8B				
strict	23.0	19.0	13.8	12.1	32.1	27.8	29.2	15.1	13.1	14.7	11.2	6.9	5.6	3.0	73.3	30.7	18.7	6.5	6.1	38.0	16.5	11.9	8.6	8.2	54.8
flexible	20.7	23.4	23.6	25.4	6.8	18.8	22.6	22.9	34.4	1.3	19.2	17.0	22.7	37.4	3.8	14.1	15.1	16.0	39.2	15.6	21.7	21.5	24.7	30.4	1.7
instruct	16.2	19.2	24.4	21.3	19.0	17.4	21.4	28.0	23.0	10.2	13.4	14.6	15.8	13.6	42.7	8.5	10.5	20.8	12.7	47.5	19.1	20.6	26.7	22.1	11.5
ans-is	20.0	23.0	23.1	24.5	9.4	20.3	24.1	24.3	25.4	5.9	12.6	11.8	16.7	12.1	46.8	9.1	10.3	11.2	12.2	57.2	22.2	20.9	23.4	23.0	10.5
last	20.5	23.0	25.6	30.8	0.1	21.7	24.7	25.5	27.9	0.3	22.1	22.3	24.8	27.1	3.7	16.6	15.4	16.6	35.8	15.6	21.9	21.8	24.6	29.8	1.8
	A	B	C	D	N/A	A	B	C	D	N/A	A	B	C	D	N/A	A	B	C	D	N/A	A	B	C	D	N/A

Figure 4: Distribution of extracted final answers across different extraction algorithms. The y-axis represents the answer extraction method, and the x-axis shows the extracted final answer, with "N/A" denoting cases where no answer could be extracted. The results of Qwen3-14B and R1-Qwen3-8B are presented in the Appendix A.3.

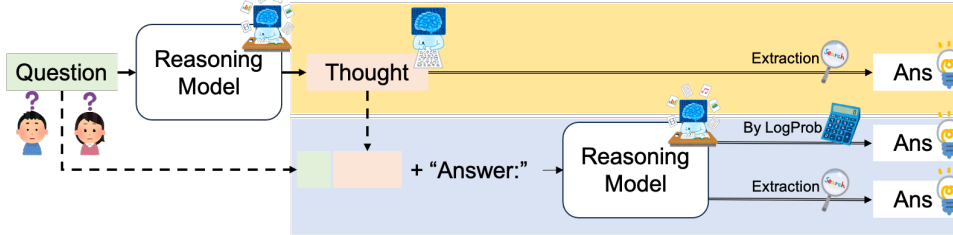


Figure 5: The proposed ANSWER REGENERATION framework for finding answers in model output. The yellow box indicates the conventional method of direct extraction, while the blue box indicates the proposed framework.

	Q3-32B	Q3-14B	Q3-8B	gpt20b	Phi4+	R1-L1	R1-Q3
(%)	2.9	2.8	6.2	6.1	82.7	6.7	6.8
best-extr	ans-is	ans-is	last	last	flex	flex	ans-is
Correct	37.1	33.8	42.1	45.4	69.9	26.6	25.5
Incorrect	32.2	22.6	53.6	49.4	29.6	65.7	19.2
Invalid	30.7	43.6	4.3	5.2	0.5	7.7	55.3

Table 1: The percentage of incomplete thinking and the corresponding accuracy of each reasoning model. (%) refers to the portion of outputs where model’s thinking process is not completed. Phi-4-reason+ generally shows repetitions in generation.

final answers derived from these incomplete outputs. The results using answer-is-correct show a high rate of invalid extraction. This implies that even well-optimized extraction method can be less robust toward incomplete thinking, particularly when the reasoning output does not contain definitive, explicitly formatted answering text.

5 Study 2: Answer Generation

Our analysis has shown that the final answer of reasoning models and the performance are highly sensitive based on how the answer is located and selected from the output. To address this and simplify the optimization of complex extraction algorithms, we propose a straightforward framework for reliably identifying the final answer.

5.1 Method

Our proposed framework ANSWER REGENERATION, illustrated in Figure 5, tackles the challenge. Instead of attempting to parse a final answer from

model’s extensive thought, we provide the model (in its non-reasoning mode) with the original input prompt and its previous output (the reasoning process) and a new prefix "Answer:". This prompts the model to generate a concise, final answer based on its prior reasoning by additional inferences.

This approach offers key benefits. For multiple-choice tasks, it allows us to utilize probability-based answering, as non-reasoning models have been evaluated, leading to more robust predictions. When the answer choices are not available, such as open-ended question answering, it simplifies the model’s output, making the final answer much easier to extract with straightforward algorithms.

While effective, our framework has several acknowledged limitations. The primary concern is the computational cost of the additional inference step; however, the overhead introduced by ANSWER REGENERATION is practically small (i.e., only a few tokens relative to the thinking tokens), as detailed in the Appendix A.5. Second, the proposed method may lack the sensitivity required to capture subtle variations in response formatting (e.g., "**A**"). Lastly, minor discrepancies between the regenerated outputs and the explicit answers were observed, but human evaluation confirmed these differences were negligible.

Despite these constraints, we believe our framework’s simplicity and the clarity constitute significant contribution. We will demonstrate the practical benefits of this approach outweigh the limita-

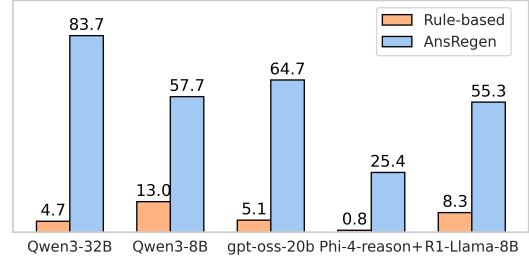
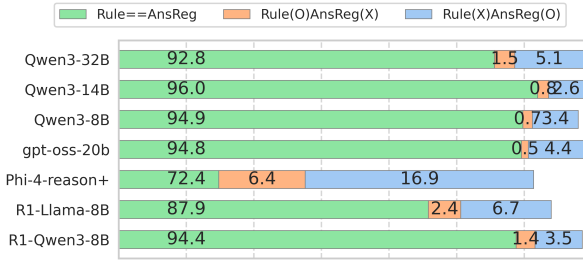


Figure 6: (left) A confusion matrix comparing the conventional answer extraction method (Rule) and the proposed method (Regen). (right) The accuracy of answers extracted from the model’s thought, as determined by human evaluation. We sample 300 instances when the extraction and regeneration are disagreed. Results are not reported for cases where the model failed to provide a definitive answer or provided multiple option labels.

	Q3-32B	Q3-14B	Q3-8B	gpt20b	Phi4+	R1-L1	R1-Q3
Rule(Best)	82.1	83.8	82.1	81.5	66.5	64.8	77.6
AnsRegen	87.1	85.0	83.3	84.7	77.4	68.8	81.1
xFinder(1)	86.4	84.5	81.9	82.8	65.3	64.8	80.1

Table 2: MMLU performance comparison between conventional answer extraction and ANSWER REGENERATION. We report each model’s performance using its best-performing extraction method.

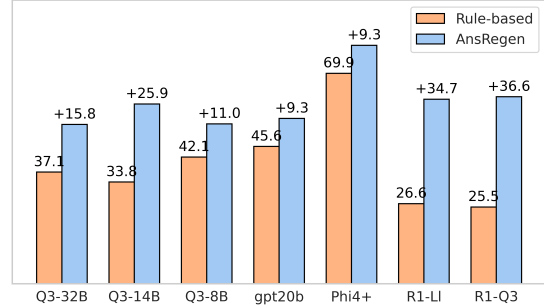


Figure 7: Model performance evaluated on outputs where the reasoning process is incomplete, using the optimal answer extraction algorithm for each model.

tions using the same setup as our previous analyses.

5.2 Result

5.2.1 Improved Performance

As presented in Table 2, the proposed method consistently reports better scores. Figure 6 (left) provides a detailed look at the performance. While most of the final answers derived by both our method and the rule-based methods are the same, our framework achieves a much higher correction rate. This demonstrates that ANSWER REGENERATION is successful at correcting incorrect answers extracted by rule-based approach.

To compute the correction rate, we select 300 instances from the outputs of Qwen3-32B, Qwen3-8B, gpt-oss-20b, Phi4-reason+, and R1-Llama-8B where the extraction and regeneration results disagreed. We then manually label the correct "gold" answers in terms of answer extraction from the thoughts. As shown in Figure 6 (right), the agreement rate of ANSWER REGENERATION with the human label is far superior to that of the conventional answer extraction methods. While xFinder is a fine-tuned model, it shows comparable performance to rule-based answering. We provide a detailed comparison with our approach in the Appendix A.6.

5.2.2 Correlation with Model Size

An interesting effect of our framework is the change in the performance ranking of Qwen3 models. The previous ranking derived from rule-based answering, which was Qwen3-14B, Qwen3-32B, Qwen3-8B, shifted to 32B > 14B > 8B under our frame-

work. This new ranking aligns with conventional intuition and general knowledge that larger models typically outperform smaller ones within the same family. This suggests that the initial, counterintuitive ranking is likely an artifact of the answer extraction methods, not a true reflection of the models’ underlying capabilities.

5.2.3 Enhanced Robustness to Responses

The nature of our proposed ANSWER REGENERATION framework inherently addresses the issue of answer inconsistency mentioned in Section 4.2.2. Since it prompts the model to generate a final, definitive answer, it bypasses the unpredictable results associated with various rule-based extraction algorithms.

Additionally, our method improves robustness by handling internal self-correction within model outputs. When facing ambiguous questions, a model may initially provide an answer and then continue its thinking process, generating alternative solutions or re-evaluating its answer. Rule-based answer extraction methods struggle to choose the final answer from this internal debate.

A further key advantage is its ability to handle "NOT correct" questions. Since many extraction algorithms are designed to find the "correct" answer, they fail when the question requires identifying incorrect one. The algorithm may mistakenly extract a correct option during the model’s rumination.

	Q3-32B	Q3-14B	Q3-8B	gpt20b	Phi4+	R1-L1	R1-Q3
strict	15.3	13.0	15.7	1.3	4.8	6.8	10.9
flexible	47.2	47.1	47.1	1.7	0.5	38.0	41.3
instructed	52.6	59.5	45.8	32.3	43.8	38.7	49.7
ans-is-corr	68.4	65.2	64.2	34.6	32.7	37.6	53.5
last-extr	66.8	63.4	62.0	64.6	43.3	42.2	45.3
built-in	72.1	69.4	64.6	43.2	23.0	43.3	58.3
AnsRegen	74.8	72.6	74.7	74.6	58.7	52.1	71.1
xFinder(q)	66.6	63.4	60.6	52.6	33.7	38.3	52.0
xFinder(l)	74.0	72.1	68.3	70.3	61.3	43.7	65.2
Reported ²	79.8	77.4	74.3	75.0	n/a	54.3	73.9
↳Reproduced	63.0	59.2	57.3	24.5	10.7	42.3	40.7

Table 3: Model performance on MMLU-Pro. The evaluation utilizes the same answer extraction algorithms used in our MMLU analysis, including the built-in algorithm from lm-evaluation-harness.

Finally, our method significantly improves performance in cases of incomplete thinking, as shown in Figure 7. Instead of relying on rules to parse an incomplete output, our framework can select the final answer even when the thought does not include an explicit final answer.

6 Studies on Additional Tasks

6.1 Complex Multiple-Choice Question Answering

As an extension of our previous findings, we investigate our framework on MMLU-Pro (Wang et al., 2024), a more complex benchmark with a dynamic number of answer options. The result, shown in Table 3, demonstrates that while the built-in extraction algorithm from lm-evaluation-harness performs better than algorithms optimized only for MMLU, ANSWER REGENERATION—not specifically tuned for any benchmark—still achieves superior performance. Furthermore, the scores are also closer to the publicly reported performance², despite the reported scores benefiting from more specific prompt engineering (e.g., detailed task descriptions for individual subtasks), as demonstrated in our reproduced score using their extraction rules.

Fine-tuned models for answer extraction, xFinder, fail to deliver significant performance gains. Furthermore, the observed performance variance across different models (i.e., qwen1505, llama38it) suggests that the model selection significantly impacts reasoning model’s performance. Consequently, we argue that evaluating models through our framework ensures a more robust and impartial assessment of true capabilities, yielding competitive results without requiring task-specific optimization.

²<https://artificialanalysis.ai/evaluations/mmlu-pro>.

	Q3-32B	Q3-14B	Q3-8B	gpt20b	Phi4+	R1-L1	R1-Q3
strict	3.3	2.7	1.7	0.6	1.5	0.0	0.1
flexible	33.3	33.5	19.3	92.0	76.5	69.2	85.1
instructed	93.5	92.2	88.6	48.8	65.6	54.8	85.8
ans-is-corr	89.6	87.6	91.9	63.8	59.7	63.1	83.4
AnsRegen	96.8	96.4	95.5	92.4	93.6	76.0	91.1
xFinder(q)	95.5	94.5	93.4	93.6	75.7	75.7	90.6
xFinder(l)	95.9	95.0	93.5	94.9	88.7	78.1	92.7

Table 4: Model performance on GSM8K. Note that strict-match and flexible-extract are implemented in lm-evaluation-harness. last-extract is not useful.

6.2 Short-Answer Math Problems

We explore the effectiveness of our framework in math domain using GSM8K benchmark (Cobbe et al., 2021), which features structured (as numbers) but relatively open-ended question answering task.

As shown in Table 4, instructed-format, a template specifically recommended for mathematical problems, and flexible-match performs well among the various extraction methods. We also modify answer-is-correct to better handle common mathematical formatting, such as numbers and symbols like \$, ", and ". Despite these optimizations, ANSWER REGENERATION with minor post-processing to remove LaTeX commands, such as \boxed{} or \text{}, achieves high performance comparable to those of a fine-tuned xFinder.

We also conduct a human evaluation of instances where the methods’ results disagreed. ANSWER REGENERATION reports 16.3% correct, while the answer extraction method is correct in only 6.1% of the cases. This underscores the superior reliability of our framework even in complex, structured but open-ended domains like mathematics.

6.3 Open-ended Question Answering

Evaluating generative models on open-ended question-answering tasks presents two main challenges: (1) finding the answer within the model’s output. (2) determining semantic equivalence between the generated answer and the gold answer. To alleviate the second challenge, we use TriviaQA (Joshi et al., 2017), known for its extensive gold answer variations and aliases, minimizing the need for complex semantic matching.

As illustrated in Table 5 (left), ANSWER REGENERATION consistently outperforms direct answer extraction from reasoning outputs. To further validate it, we compare it against two LLM-as-a-judge approaches (Zheng et al., 2023): GPTGrader (Wei et al., 2024) and Equality Checker³. GPTGrader

³<https://artificialanalysis.ai/methodology/>

Method		(↓) Evaluator	Q3-32B	Q3-14B	Q3-8B	gpt20bPhi4+	R1-L1	R1-Q3		
String Match	ans-is-corr	-	42.7	47.5	44.2	54.8	22.3	11.7	35.6	
	AnsRegen	-	77.7*	77.6	67.7	62.3	68.6	58.2	60.8	
Model-based	GPT Grader	Qwen3-32B	3.1*	3.8	3.6	4.8	1.1	1.3	2.9	
		gpt-oss-20b	47.0	53.4	46.7	50.8	18.1	18.3	39.2	
		Phi-4-reason+	48.6	55.3	48.1	47.6	11.3	18.4	40.5	
	Equality Checker	R1-Llama-8B	69.2*	68.1	62.8	72.0	60.1	58.6	57.5	
		Qwen3-32B	7.3*	5.2	6.1	14.3	40.9	32.0	8.2	
		gpt-oss-20b	46.6	48.9	46.8	40.1	31.2	36.3	46.4	
		Phi-4-reason+	61.9	63.2	57.1	61.1	26.7	28.2	50.4	
		R1-Llama-8B	67.1	62.5	57.6	63.7	40.0	36.6	53.4	
		xVerify	xVerify-8B-I	21.6*	24.9	22.1	16.6	3.9	8.0	18.7
		qwen1505	1.3	1.4	1.2	0.2	0.6	0.7	1.1	
xFinder	llama38it	1.5	1.4	1.2	1.4	0.0	0.0	0.9		

		AnsReg(Qwen3-32B)	Grader(Qwen3-32B)	Grader(R1-Llama)
String-Match	X	10	25	2
	O	65	0	59
		O X	O X	O X
Model	X	1	19	40
	O	58	22	12
		O X	O X	O X

Table 5: (left) Performance of reasoning models on open-ended question answering TriviaQA. (right) Confusion matrix illustrating human evaluation performance on 100 samples in determining semantic equivalence between the generated answer and the gold answer. * denotes the selected results for the detailed human evaluation.

utilizes an additional inference call with extensive prompts to categorize semantic similarity as "correct" (equivalent), "incorrect" (not equivalent), or "invalid". Equality Checker employs a more compact prompt strategy to provide a binary "correct" or "incorrect" classification. We also evaluate xVerify (Chen et al., 2025), which uses a fine-tuned model for equivalence assessment, and xFinder, which supports a LLM-as-a-judge approach within its implementation.

While model-based evaluations yield higher scores in some cases (e.g., when judged by R1-Llama-8B), they are undermined by inherent model bias. Table 5 (right) presents human evaluation of semantic equivalence, comparing the model judgment against human judgments on 100 sampled outputs. GPTGrader with Qwen3-32B consistently predicts "incorrect" for correct answers, while Equality Checker reveals a bias toward "correct." xVerify and xFinder default to incorrect and R1-Llama-8B exhibits a bias toward "correct." In contrast, our string-match-based method avoids these inherent model biases. Despite the known limitations of string matching in capturing complex semantic nuances, it provides a more reliable performance metric; our method achieved 77.7% accuracy, closely aligning with human evaluation of 77% in Qwen3-32B.

7 Discussion and Conclusion

Our analysis highlights a critical, yet often overlooked, challenge in evaluating reasoning models: the profound impact of the answer extraction methods on performance scores. We have demonstrated that the performances can fluctuate significantly based on how the final answer is parsed from its reasoning output. This finding suggests that dis-

crepancies between publicly reported scores and reproduced results may stem from undocumented variations not only in prompting, but in extraction methods itself. To mitigate this, we introduced ANSWER REGENERATION framework, offering significant advantages over conventional rule-based extraction without requiring specific tuning.

Our framework is most effective when the target answer space is highly constrained, such as multiple-choice QA. In math domains, while answers are generally numerical, the inclusion of symbols (\$), units (%), and varying decimal formats introduces minor challenges for string-match-based evaluation. Open-ended QA is the least constrained task, allowing for a near-infinite range of output formats. While ANSWER REGENERATION still offers an improvement over heuristic methods, the inherent difficulty of standardizing the "correct" answer string in a free-form text context limits the magnitude of the measured robustness gain.

Beyond the quantitative improvements, we have conducted human evaluations to ensure our framework provides a more accurate reflection of a model's true capabilities. Our results also show that the performance rankings for Qwen3 models aligned with the intuition that larger models generally outperform smaller ones. Crucially, our method avoids the biases of model-specialized answer extraction rules or LLM-as-a-judge methods. Furthermore, the framework proves more resilient to common failure of rule-based approach, including handling incomplete thinking, self-correction, and questions asking for the "incorrect" choice, all of which can confuse rule-based extraction.

In conclusion, through our findings from analysis and the introduction of ANSWER REGENERATION framework, we believe this work contributes toward more reliable and faithful model evaluation for all reasoning-powered LLMs.

8 Limitations

Technical Novelty in ANSWER REGENERATION We acknowledge that ANSWER REGENERATION framework itself lacks technical novelty. However, we contend that the value of our contribution lies in the simplicity and the clarity of the results and analysis it provides. Our work demonstrates the benefits of using this framework as a robust and reliable reference for evaluating and fairly comparing the performance of reasoning models.

Experiments with Sophisticated Extraction Rules Our experiments adopted established answer extraction rules from lm-evaluation-harness (strict-match, flexible-match). Building upon these, we developed more complex, heuristic rules (answer-is-correct, last-extract) and included the recommended rule for Qwen3 families (instructed-format). While we recognize that more aggressively optimized, domain-specific rules could exist, we maintain that such highly-specified rules will still fail to handle the full spectrum of answer variations.

Experiments with Diverse LLMs and Prompts Our focus was on output-level results, which means that the effect of different input prompts seem to be overlooked. Furthermore, our investigation was limited to publicly available open-source reasoning models. Although greater diversity in models and prompts would enhance generalizability, we believe that the widely-used models and default prompts from established repositories provide sufficiently general results for our findings. We defer the investigation of commercial LLMs, such as ChatGPT, Gemini, and Claude, to future work. As a minor note, we observed that small variations in the input prompts (e.g., changes of option labels or the "Answer:" prefix) do not significantly affect performance.

Inherent Weakness of ANSWER REGENERATION As discussed in Section 5.1, ANSWER REGENERATION carries inherent limitations. Nonetheless, we believe that employing the simplest possible framework was the most effective way to demonstrate the core benefits of our approach. Exploring further techniques within this framework, such as incorporating concepts like self-consistency (Wang et al., 2022), represents a valuable direction for future research.

References

- Marah Abidin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, et al. 2025. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*.
- Norah Alzahrani, Hisham Abdullah Alyahya, Yazeed Alnumay, Sultan Alrashed, Shaykhah Alsubaie, Yusef Almushaykeh, Faisal Mirza, Nouf Alotaibi, Nora Altwairesh, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. 2024. [When benchmarks are targets: Revealing the sensitivity of large language model leaderboards](#).
- Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, et al. 2024. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*.
- Nikhil Chandak, Shashwat Goel, Ameya Prabhu, Moritz Hardt, and Jonas Geiping. 2025. Answer matching outperforms multiple choice for language model evaluation. *arXiv preprint arXiv:2507.02856*.
- Ding Chen, Qingchen Yu, Pengyuan Wang, Wentao Zhang, Bo Tang, Feiyu Xiong, Xinchu Li, Minchuan Yang, and Zhiyu Li. 2025. xverify: Efficient answer verifier for reasoning model evaluations. *arXiv preprint arXiv:2504.10481*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#).
- Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. 2025. Are we done with mmlu? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5069–5096.
- Nathan Habib, Clémentine Fourrier, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. 2023. [Lighteval: A lightweight framework for llm evaluation](#).
- Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. 2024. [Does prompt formatting have any impact on llm performance?](#)
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

688	Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In <i>Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 1601–1611.	Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. <i>arXiv preprint arXiv:2203.11171</i> .	744
689			745
690			746
691			
692		Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. <i>Advances in Neural Information Processing Systems</i> , 37:95266–95290.	747
693			748
694	Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In <i>Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 5591–5606.		749
695			750
696			751
697			752
698			753
699		Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. <i>arXiv preprint arXiv:2411.04368</i> .	754
700	Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekogonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic evaluation of language models .		755
701			756
702			757
703			758
704		Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. <i>Advances in neural information processing systems</i> , 35:24824–24837.	759
705			760
706			761
707			762
708			763
709		An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. <i>arXiv preprint arXiv:2505.09388</i> .	764
710			765
711			766
712			767
713		Qingchen Yu, Zifan Zheng, Shichao Song, Zhiyu Li, Feiyu Xiong, Bo Tang, and Ding Chen. 2024. xfinder: Robust and pinpoint answer extraction for large language models. <i>arXiv preprint arXiv:2405.11874</i> .	768
714			769
715			770
716			771
717	Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt LLM evaluation . <i>Transactions of the Association for Computational Linguistics</i> , 12:933–949.		772
718			773
719			774
720			775
721			776
722	Francesco Maria Molfese, Luca Moroni, Luca Giofrè, Alessandro Scirè, Simone Conia, and Roberto Navigli. 2025. Right answer, wrong score: Uncovering the inconsistencies of llm evaluation in multiple-choice question answering. <i>arXiv preprint arXiv:2503.14996</i> .		777
723			
724			
725			
726			
727			
728	OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card .		
729	OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass .		
730			
731			
732			
733	Felipe Maia Polo, Ronald Xu, Lucas Weber, Mírian Silva, Onkar Bhardwaj, Leshem Choshen, Allysson Flavio Melo de Oliveira, Yuekai Sun, and Mikhail Yurochkin. 2024. Efficient multi-prompt evaluation of llms .		
734			
735			
736			
737			
738	Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models’ sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting .		
739			
740			
741			
742	Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and		
743			

Data	Tokens		Qwen3-32B	Qwen3-14B	Qwen3-8B	R1-Llama-8B	R1-Qwen3-8B	gpt-oss-20b	Phi-4-reason+
GSM8K	Avg. #Think Tokens		1575.8	1590.1	1986.1	466.7	1931.3	283.4	4095.8
	Avg. #AnsRegen Tokens		13.2	8.8	9.8	7.7	24.7	17.0	32.0
TriviaQA	Avg. #Think Tokens		963.4	881.1	830.6	712.2	681.3	952.3	4093.8
	Avg. #AnsRegen Tokens		17.5	11.9	16.9	14.3	14.9	23.7	30.7

Table 6: Quantitative measurement of the inference overhead in GSM8K and TriviaQA.

844 presents the average number of output tokens con-
845 sumed by both the initial thinking process and the
846 subsequent ANSWER REGENERATION step for various
847 models:

848 The quantitative results demonstrate that the
849 number of tokens required for ANSWER REGENERATION
850 is negligible when compared to the vast num-
851 ber of tokens generated in the initial reasoning
852 ("Thinking") phase. Therefore, the inference over-
853 head introduced by our method is practically in-
854 significant while providing substantial robustness
855 gains.

856 **A.6 MMLU Results: xFinder vs. Answer** 857 **Regeneration**

858 We analyzed all Qwen3-32B outputs where the
859 extractions from xFinder and ANSWER REGENERATION
860 diverged. Our manual labeling reveals that, in
861 49.3% of these cases, the model either generated
862 multiple conflicting answers or failed to provide
863 a valid response. xFinder outperforms ANSWER
864 REGENERATION with a winning ratio of 28.5% com-
865 pared to 22.2%. Given that xFinder is a specialized,
866 fine-tuned model while the regenerator used for
867 Qwen3-32B is a larger general-purpose model, this
868 performance gap suggests that the marginal gains.

869 Our analysis highlights that xFinder is precise
870 in simple outputs and ANSWER REGENERATION is ro-
871 bust in complex outputs; xFinder’s results aligned
872 with rule-based extraction in 98.4% of cases This
873 high correlation indicates that xFinder is highly ef-
874 fective at identifying and extracting answers from
875 simple, straightforward outputs. On the other hand,
876 ANSWER REGENERATION matched rule-based extrac-
877 tion in only 88.2% of cases. Detailed qualitative
878 analysis suggests that this method excels when deal-
879 ing with complex or verbose outputs—specifically
880 those involving self-correction, rumination, choos-
881 ing the best answer among multiple answers where
882 standard extraction rules often fail.