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. Our research reveals 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. 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. 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 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*.

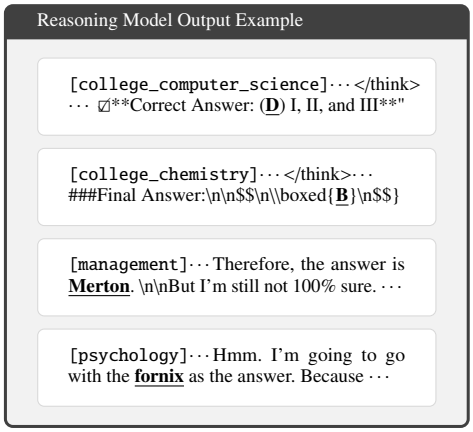

Figure 1: Examples illustrating the difficulties in extracting final answers from reasoning models' outputs. Although the benchmark is designed with multiple-choice questions, models frequently generate answers in a free-text format, which complicates automated evaluation.

However, the 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.

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 score and human evaluation results. Our method also achieves intuitive model rankings, where larger models are shown to outperform smaller ones. We demonstrate that 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 generation-based 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 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, which are caused by the selection of answer extraction algorithms.

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.

To the best of our knowledge, this is the first study to highlight the importance of answer extraction methods, especially for reasoning LLMs. Based on our findings, we introduce a lightweight method to reduce the reliance on the fragile extraction rules 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.

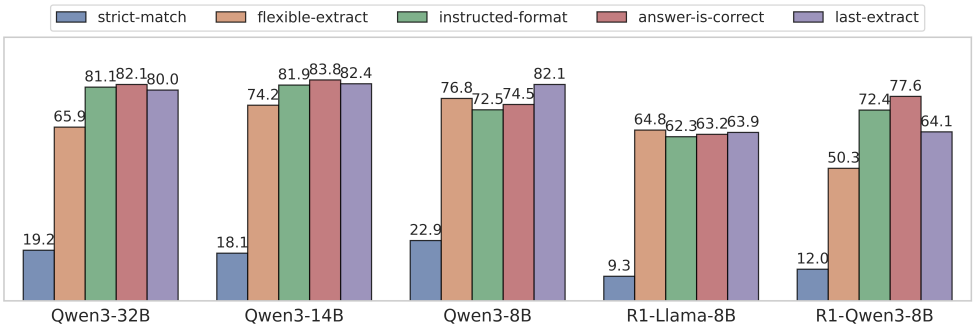

Figure 2: Model performance in accuracy evaluated using various answer extraction algorithm. Responses are considered incorrect if the extraction process fails to find an answer.

MMLU (Hendrycks et al., 2021) benchmark is primarily used, given its widely adoption for evaluating LLMs' knowledge[1]. 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 benchmark **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-R1-Distill-Llama-8B** (referred to as **R1-Llama-8B**), and **DeepSeek-R1-0528-Qwen3-8B** (referred to as **R1-Qwen3-8B**) (DeepSeek-AI, 2025). For hyperparameter settings, we adhere to recommended best practices for each model, setting temperature to 0.6, top-p value to 0.95, and top-k value to 20. Prompt templates are sourced from lm-evaluation-harness, using thinking templates. To reduce computational costs, the maximum token generation length is limited to 4,096.

4 Study 1: Rule-based Answer Extraction

4.1 Methods

We evaluate 5 reasoning models using 5 different answer extraction methods to investigate how performance changes with extraction algorithms: **strict-match**, **flexible-extract**, **instructed-format**, **answer-is-correct**, and **last-extract**:

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 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. While the method is strict (deviation is generally considered incorrect).

Further heuristically optimized answer extraction methods are 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 also 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, this research shows 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 Model Performance

Figure 2 illustrates how different answer extraction methods affect the performance of models. We

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

Answer Inconsistency Example

The following are multiple choice questions (with answers) about high school physics.
Q: The plates of a capacitor are charged to a potential difference of 5 V. If the capacitance is 2 mF, what is the charge on the positive plate?
(A) 0.005 C (B) 0.01 C (C) 0.02 C (D) 0.5 C
<think>
Okay, let's see. The question is about finding the charge on the positive plate of a capacitor. ⋯ the charges on the two plates are equal in magnitude but opposite in sign. So the positive plate has +Q and the negative has -Q. But the question is just asking for the magnitude, so the answer is still 0.01 C. Therefore, option `answer-is-correct:` **B** is correct.
</think>
To determine the charge on the positive plate of a capacitor, ⋯ Since the question asks for the **magnitude** of the charge on the **positive** plate, the correct answer is: $$ `instructed-format:` **\boxed{0.01 \, \text{C}}** $$
—
Final Answer:
** `flexible-match:` **(B)** 0.01 `last-extract:` **C** **

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

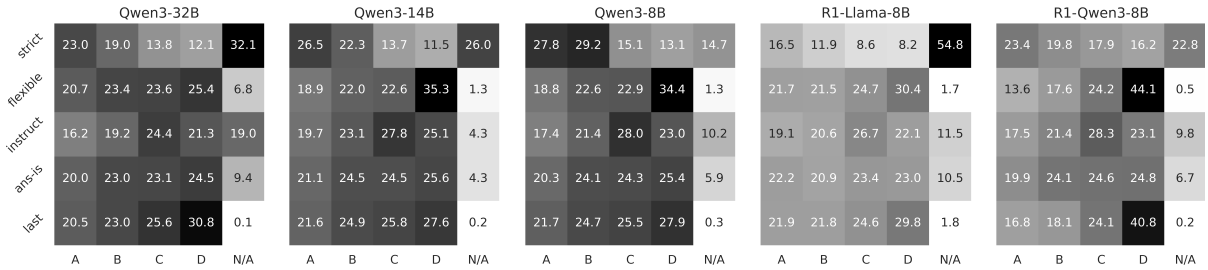

Figure 4: Distribution of extracted final answers across different extraction algorithms. The y-axis represents the answer extraction method, and the x-axis shows the extracted final answer, with "N/A" denoting cases where no answer could be extracted.

evaluate performance using 3 types of rules: implemented (`strict-match`, `flexible-extract`), recommended (`instructed-format`), and heuristically optimized (`answer-is-correct`, `last-extract`). 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, and R1-Llama-8B 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, Qwen3-8B, R1-Llama-8B, and R1-Qwen3-8B. A similar sensitivity is observed with the other methods. Using `flexible-extract`, the top models are Qwen-8B, Qwen3-14B, Qwen-32B, R1-Llama-8B, and R1-Qwen3-8B. With `last-extract`, Qwen3-14B performs the best, and R1-Qwen3-8B outperforms R1-Llama-8B compared with `flexible-extract`. Interestingly, despite following the recommended best practices for multiple-choice question answering with `instructed-format`, the performance of Qwen3 family models are not impressive compared to other extraction methods. This method proves to be particularly ineffective for R1-Llama-8B model.

These findings challenge the common assumption that larger models outperform smaller ones within the same family. Our analysis indicates that the benchmark performance scores of reasoning models are highly dependent on the answer extraction method used. This suggests 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` fails. `answer-is-correct` successfully locates an answer within the model's thought, between <think> and </think> tags. However, the model's explicitly formatted final answer, "the correct answer is: X", is not recognized as a valid because it contains unexpected patterns, including $$, \boxed{}, and \text{}. Besides, the output is the option text rather than the required option label. `instructed-format` could find an answer using \boxed{} (despite the \boxed{} format being recommended only for math problems), but the extracted answer is again the option text, not the label. Furthermore, the presence of

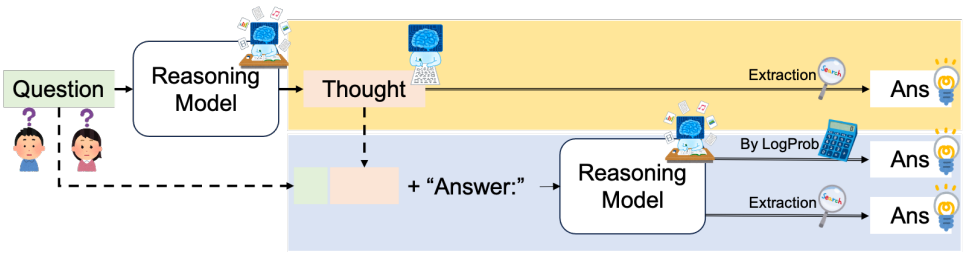

Figure 5: The proposed ANSWER REGENERATION framework for finding answers in model output. The yellow box indicates the conventional method of direct extraction, while the blue box indicates the proposed framework.

	Qwen3-32B	Qwen3-14B	Qwen3-8B	R1-Llama	R1-Qwen3
(%)	2.8	2.9	6.2	6.7	6.8
best-extr	ans-is	ans-is	last	flexible	ans-is
Correct	37.1	33.8	42.1	26.6	25.5
Incorrect	32.2	22.6	53.6	65.7	19.2
Invalid	30.7	43.6	4.4	7.8	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.

the unexpected LaTeX command \text{ } could result in an incorrect evaluation during string-match comparison. Meanwhile, `flexible-match` correctly identifies the final answer. Interestingly, the simple yet effective `last-extract` extracts the unit of option text "C" as the final answer.

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, 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 and is 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 final answers derived from these incomplete outputs. Except for Qwen3-8B and R1-Llama-8B, which use extraction algorithms solely on capital letters, 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 is highly sensitive to the chosen extraction method. Model performance fluctuates significantly 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, illustrated in Figure 5, tackles the challenge by introducing ANSWER REGENERATION step. Instead of attempting to parse a final answer from model's extensive thought, our method uses an additional inference call. Specifically, 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.

This approach offers key benefits. For multiple-choice questions, 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 issue is the computational cost of the additional inference step. Additionally, the method might not fully capture minor variations in answer formatting, e.g., the probability of "**A**". Finally, some regenerated results could be different from explicitly mentioned answer. Despite these weaknesses and the lack of

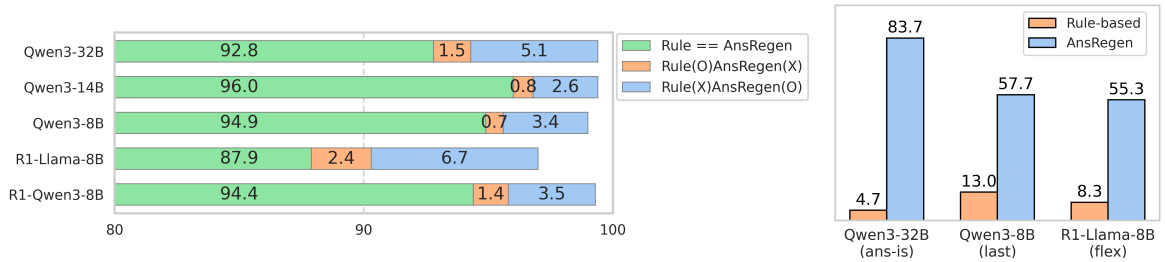

Figure 6: (left) A confusion matrix comparing the conventional answer extraction method (`Rule`) and the proposed method (`Regen`). (right) The accuracy of answers extracted from the model's thought, as determined by human evaluation. We sample 300 instances when the extraction and regeneration are disagreed.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

	Qwen3-32B	Qwen3-14B	Qwen3-8B	R1-Llama	R1-Qwen3
Rule(Best)	82.1	83.8	82.1	64.8	77.6
AnsRegen	**87.1**	**85.0**	**83.3**	**68.8**	**80.7**
Diff	+5.0	+1.2	+1.2	+4.0	+3.1

Table 2: Performance comparison between conventional answer extraction and Answer Regeneration. We report each model's performance using its best-performing extraction method.

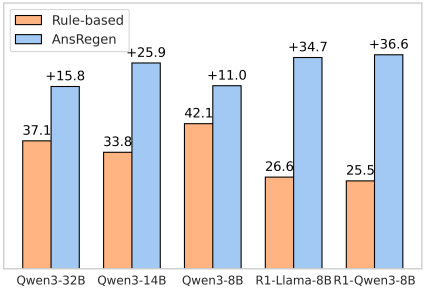

Figure 7: Model performance evaluated on outputs where the reasoning process is incomplete, using the optimal answer extraction algorithm for each model.

technical novelty, we believe this framework's simplicity and the clarity constitute significant contribution. We will demonstrate its benefits using the same experimental 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 than rule-based answering. 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 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, 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.

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 Qwen3-32B, Qwen3-14B, Qwen3-8B under our framework. 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. In contrast, our framework considers the entire thinking process and forces the model to finalize its response, leading to a more reliable result.

A further key advantage is its ability to handle "NOT correct" questions. Since many extraction algorithms are designed to find the "correct" answer from the reasoning text, they fail when the question requires identifying the incorrect choice. The

	Qwen3-32B	Qwen3-14B	Qwen3-8B	R1-Llama	R1-Qwen3
strict-match	15.3	13.0	15.7	6.8	10.9
flexible-ext	47.2	47.1	47.1	38.0	41.3
instructed	52.6	59.5	45.8	38.7	49.7
ans-is-corr	68.4	65.2	64.2	37.6	53.5
last-extract	66.8	63.4	62.0	42.2	45.3
implemented	**72.1**	**69.4**	**64.6**	**43.3**	**58.3**
AnsRegen	**77.0**	**72.6**	**72.0**	**43.6**	**66.4**
Reported[2]	79.8	77.4	74.3	54.3	73.9
▸ Reproduced	63.0	59.2	57.3	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, referred to as `implemented`.

algorithm may mistakenly extract a correct option discussed during the model's rumination.

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.

5.2.4 Regenerator Independency

Our method, which uses an additional model inference for Answer Regeneration, raises a question about its dependency on the specific model used.

Table 7 in the Appendix shows that the performances achieved using small-sized regenerators are generally similar to the performance achieved when using the same model both for reasoning and Answer Regeneration. While this suggests a degree of independence, we still recommend using the same model for both tasks. The final answering step is also a crucial part of the overall model evaluation and should be performed by the model being assessed to ensure a consistent and fair comparison.

Based on the results, Answer Regeneration framework shows a more effective and reliable method for evaluating reasoning models. Conventional extraction rules cannot account for all the variations in model outputs, and can thus introduce biases and inaccuracies. Our framework mitigates this problem, providing a more accurate and consistent measure of models' true performance.

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

(↓) Extraction	Qwen3-32B	Qwen3-14B	Qwen3-8B	R1-Llama	R1-Qwen3
strict-match	3.3	2.7	1.7	0.0	0.1
flexible-ext	33.3	33.5	19.3	69.2	85.1
instructed	**93.5**	**92.2**	88.6	54.8	**85.8**
ans-is-corr	89.6	87.6	**91.9**	**63.1**	83.4
AnsRegen	**95.0**	**93.8**	91.1	**76.0**	**91.1**

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

Table 3, demonstrates that while the built-in extraction algorithm from lm-evaluation-harness performs better than algorithms optimized only for the original MMLU, Answer Regeneration—which is not specifically tuned for any benchmark—still achieves superior performance. Furthermore, the scores are also closer to the publicly reported performance[2], despite the reported scores likely benefiting from more specific prompt engineering (e.g., detailed task descriptions for individual subtasks), as demonstrated in our reproduced score using their extraction rules. Therefore, we believe that evaluating models with our framework provides a fairer and more robust assessment of true capabilities, achieving competitive performance without the need for task-specific optimization.

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, performs the best 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 the highest performance.

To further validate this, we conduct a human evaluation of instances where the methods' results disagreed. Answer Regeneration framework reports 16.3% correct, while the conventional 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.

[2] https://artificialanalysis.ai/evaluations/mmlu-pro. Qwen3 technical report does not contain zero-shot CoT results for MMLU-Pro; it only provides 5-shot results without reasoning, scoring 65.54 for 32B and 56.73 for 8B.

	Method	(↓) Evaluator	Qwen3-32B	Qwen3-14B	Qwen3-8B	R1-Llama	R1-Qwen3
String Match	ans-is-corr	-	42.7	47.5	44.2	11.7	35.6
	AnsRegen	-	55.3*	53.7	47.0	24.1	55.8
Model -based	GPT Grader	Qwen3-32B	3.1*	3.8	3.6	1.3	2.9
		Qwen3-14B	49.5	56.4	49.2	19.4	41.1
		Qwen3-8B	49.4	56.3	49.4	18.2	41.3
		R1-Llama-8B	93.9	92.3	89.4	93.1*	87.5
		R1-Qwen3-8B	47.6	54.8	47.9	17.7	39.6
	xVerify	xVerify-8B-I	0.0	0.0	0.0	47.7*	0.0

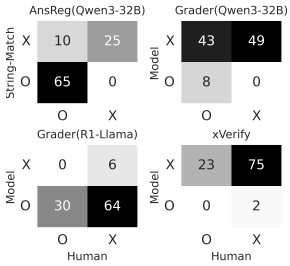

Table 5: (left) Performance of reasoning models on open-ended question answering TriviaQA. (right) Confusion matrix illustrating human evaluation performance on 100 samples in determining semantic equivalence between the generated answer and the gold answer. * denotes the selected results for the detailed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

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 shown in Table 5 (left), Answer Regeneration consistently outperforms direct answer extraction from the reasoning output. We also compare our framework with two other LLM-as-a-judge approaches (Zheng et al., 2023): GPTGrader (Wei et al., 2024), which uses an additional inference call with long prompts to categorize semantic similarity into "correct" (same), "incorrect" (not same), and "invalid". xVerify (Chen et al., 2025), a fine-tuned model that evaluates semantic equivalence as "correct" or "incorrect". While the scores from these model-based evaluations might appear better, they carry a critical drawback of model bias. Table 5 (right) presents a human evaluation of semantic equivalence, comparing the model judgement with human judgments on 100 sampled outputs. Qwen3-32B consistently predicts "incorrect" even when the answer is correct and xVerify similarly defaults to "incorrect", and R1-Llama-8B exhibits a bias toward "correct". In contrast, our string-match-based method avoids this model bias and provides a more accurate performance measure, despite of its limitation in determining semantic equivalence.

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 model performances can fluctuate significantly based on how the final answer is parsed from its reasoning output. This finding suggests that discrepancies between publicly reported scores and reproduced results may stem from undocumented differences not just in prompts, but in the extraction methods itself. To address this issue, we introduced Answer Regeneration framework. Our simple approach offers significant advantages over conventional extraction rules:

Without specific tuning, the framework consistently achieved superior scores across a variety of tasks, from multiple-choice question answering to short-answer math problems and open-ended question answering tasks. By prompting the model to explicitly state its final answer again, we mitigate inconsistencies caused by diverse output formats.

Beyond exhibiting better scores than hand-crafted, optimized rules, the performance ranking derived from our framework for the Qwen3 model family aligned with the conventional intuition that larger models generally outperform smaller ones. This suggests that the framework provides a more accurate reflection of a model's true capabilities, free from the biases of model-specialized answer extraction rules.

Our method also proves more resilient to common failure of rule-based approach. It successfully handles outputs involving incomplete thinking, models that re-consider their answers, and questions asking for the "incorrect" choice, all of which can confuse rule-based extraction.

Lastly, while LLM-as-a-judge method can suffer from inherent model biases (e.g., consistently predicting, "correct" or "incorrect"), our string-match method, enabled by the concise regenerated output, provides a more reliable measure of performance.

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

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, Xinchi 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)*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of*

the Association for Computational Linguistics (Volume 1: Long Papers), pages 1601–1611.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606.

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 Yuksekgonul, 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.

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. *Transactions of the Association for Computational Linguistics*, 12:933–949.

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

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.

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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

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. *arXiv preprint arXiv:2411.04368*.

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. *Advances in neural information processing systems*, 35:24824–24837.

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. *arXiv preprint arXiv:2505.09388*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

A Appendix

A.1 Evaluation Toolkits

MMLU Hendrycks (Hendrycks et al., 2021) and follow-ups such as MMLU-Pro (Wang et al., 2024) are deeply integrated into most of toolkits, but the original implementation only supports probability based answering for multiple choice question answering. **HELM** (Holistic Evaluation of Language Models; Liang et al. (2023)) simply use Quasi-exact match that post-process the model generation, such as lower-casing, removing whitespace and punctuation and articles. Also, **OpenCompass** (OpenCompass Contributors, 2023) supports both option-likelihood scoring and post-processing option (but provided with blank) to be customized for reasoning outputs, its metrics mainly rely on model-based scoring. Similarly, **lighteval** (Habib et al., 2023) has metrics for generated outputs, but there is only a scoring function, not mentioning about post-processing.

A.2 Regular Expressions used in the Experiments

Note that () makes groups in regular expression and \\is required both for meta characters and escape sequence in lm-evaluation-harness.

- `strict-match`: ((?<=The answer is)(.*)(?=.)|(?<=answer is)(.*)(?=.)|(?<=The answer:)(.*)(?=.)|(?<=The final answer:)(.*)(?=.))
- `flexible-extract`: (\\([A-D]\\))
- `instructed-format`:[Aa]nswer\"?:\\s* \"?\\(?([A-D])\"|\"?**(?([A-D])\"
- `answer-is-correct`: **[Aa]nswer:**\\s*(\\(?[A-D]\\)?)|**[Aa]nswer**:\\s*(\\(?[A-D]\\)?)|[Aa]nswer is **(\\(?[A-D]\\)?)**|[Aa]nswer should be **(\\(?[A-D]\\)?)**|[Aa]nswer:\\s+**(\\(?[A-D]\\)?)**|correct answer is **(\\(?[A-D]\\)?)**|correct answer:\\s+**(\\(?[A-D]\\)?)**|**(\\(?[A-D]\\)?)** is correct|*(\\(?[A-D]\\)?)** is the correct|**(\\(?[A-D]\\)?)** is the answer|**(\\(?[A-D]\\)?)** should be the answer
- `last-extract`: [^a-zA-Z0-9]([A-D])[^a-zA-Z0-9]

A.3 Preliminary: Non-reasoning vs. Reasoning

We demonstrate the power of reasoning in solving MMLU, as presented in Table 6. The performance shows that the reasoning significantly improves the model performances. This encourage us to use reasoning model not only for complex problem. but for knowledge-based problems.

A.4 Regnerator Independency

Table 7 reports the performance when using different models from the answer generator. Although we use smaller models for regenerator, the performance is similar when using the identical model.

	Qwen3-32B	Qwen3-14B	Qwen3-8B	R1-Llama-8B	R1-Qwen3-8B
non-Reason	78.4	75.7	72.2	53.0	66.2
Reason	82.1	83.8	82.1	64.8	77.6
Diff	+3.7	+8.1	+9.9	+11.8	+11.4

Table 6: The performance comparison when using non-reasoning mode and reasoning mode in LLMs. Non-reasoning mode follows conventional loglikelihood measurements using candidate whereas reasoning mode uses answer extraction algorithms to find the final answer in the reasoning output. The best performance with answer extraction methods are reported.

(↓) Regenerator	Qwen3-32B	Qwen3-14B	Qwen3-8B	R1-Llama-8B	R1-Qwen3-8B
gemma-3-1b-it	86.9	84.9	82.5	67.5	80.3
llama-3.2-1b-it	86.5	84.4	81.8	67.8	79.6
Qwen3-0.6b	86.3	84.4	82.3	68.9	79.9
Qwen3-32B	**87.1**	85.2	83.7	72.1	82.6
Qwen3-14B	87.1	**85.0**	83.1	71.2	81.6
Qwen3-8B	87.4	85.2	**83.3**	72.5	82.0
R1-Llama-8B	87.0	84.9	82.6	**68.8**	80.2
R1-Qwen3-8B	84.2	81.0	81.1	70.9	**80.7**

Table 7: Model performance when different models are used for Answer Regeneration step. Bold indicates the reported score when the reasoning models and the regenerators are the same.