FINDING ANSWERS IN THOUGHT MATTERS: REVISITING EVALUATION ON LARGE LANGUAGE MODELS WITH REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluating generative models, such as large language models (LLMs), commonly involves multiple-choice question-answering tasks where the final answer is selected based on the probabilities of the answer choices. For reasoning models, however, the method of answer extraction plays a critical role. In this paper, we show that the benchmark performance of reasoning models and their extracted-answer distributions are highly sensitive to the answer-extraction algorithm employed. To mitigate this sensitivity, we propose a simple framework: ANSWER REGENERATION. The method uses an additional inference step, providing the evaluated model with its original input and previous output, followed by the prefix "Answer:". The final answer is then selected using option probabilities or extracted from the regenerated output. We show that this reduces reliance on handcrafted extraction rules and achieves competitive performance with specialized extractors. Furthermore, it aligns more closely with human interpretation on sampled disagreements and provide reliable performance on incomplete reasoning responses. This framework also extends to mathematical reasoning and open-ended question answering. Our analysis and framework offer a practical approach to more reproducible evaluation of reasoning models.

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 raises a practical question: *how can the final answer be reliably identified in a long reasoning response?*

Conventional rule-based approaches suffer 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 placing the answer in a LaTeX box (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.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

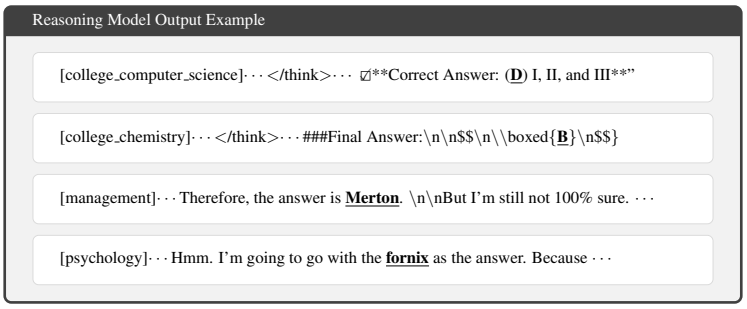


Figure 1: Examples illustrating the difficulties in extracting final answers from reasoning models' outputs. Although the benchmark is designed with multiple-choice questions, models frequently generate answers in a free-text format, which complicates automated evaluation.

In this paper, we first empirically demonstrate the impact of answer extraction rules on reasoning models (Section 4). We then introduce ANSWER REGENERATION, a simple, generation-based framework designed to reduce dependence 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 show that extraction rules can yield different answers, invalid outputs, and model rankings from the same responses. Across multiple-choice, mathematical, and open-ended QA tasks, ANSWER REGENERATION reduces dependence on handcrafted rules and improves agreement with human interpretation on sampled disagreements.

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 a generation-based framework ANSWER REGENERATION. It achieves (1) competitive benchmark performance across the evaluated settings and improved agreement with human interpretation on sampled disagreements, (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). Polo et al. (2024) and Mizrahi et al. (2024) proposed 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 evaluation frameworks 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 can be found in the Appendix A.1.

Recent works also tackled the importance of answer extraction, especially for instruct-tuned models. However, reasoning-model outputs can be longer and more structurally varied than prompted chain-of-thought responses, complicating extraction with simple regular expressions (Molfese et al., 2025). xFinder (Yu et al., 2024) mitigates rule dependence by fine-tuning LLMs for answer extraction and verification. We regard xFinder as a strong supervised baseline but note that this work investigates a complementary, tuning-free alternative.

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 proposed method ANSWER REGENERATION (Study 2 in Section 5).

Model Evaluation. We utilize **lm-evaluation-harness** (Biderman et al., 2024) toolkit for its simplicity in customizing the post-processing rules. **MMLU** (Hendrycks et al., 2021) benchmark is primarily used, given its widespread adoption. 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: the Qwen3 family–**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 (**R1-Qwen3-8B**) (DeepSeek-AI, 2025), and other families–**gpt-oss-20b** (OpenAI, 2025), Phi-4-reasoning-plus (**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. The prompt and hyperparameter details are described in the Appendix A.2. Fine-tuned models for answer extraction, xFinder Yu et al. (2024) based on qwen1505 and llama38it, are selected as a baseline[1].

Human Evaluation. In Study 2 (Section 5), we conducted human evaluations to verify that the observed score improvements reflect improved recovery of the model's intended answer rather than artifacts of extraction rules. Three co-authors with expertise in AI served as annotators. Annotators evaluated disagreements between the compared methods using task-specific rubrics: for MMLU (option selection), "Which extracted answer option is correct as a model output?"; for GSM8K and TriviaQA (generation), "Is the extracted answer semantically equivalent to the gold answer?"

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-match** are adapted from lm-evaluation-harness. strict-match extracts a precise string such as "answer is X" or "Answer: X" and flexible-match 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 a tendency to extract the last capital character from any text, which can lead to errors.

instructed-format requires modifying the input prompt to guide 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 the specified format. Additional aligned-format variants and matching extractors are further listed in Appendix A.8.

We also evaluate two additional heuristic extractors: 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-match by developing **last-extract**, which specifically targets the last *single* capital character that appears in the output.

All the implementations account for minor variations like "**X**", "**Answer:** X" and are designed to select the last match in the post-thinking segment (i.e., after </think>) to accommodate

[1] In our implementation, using the full reasoning output (rather than only the post-thinking-token segment as xFinder suggested) yields better performance.

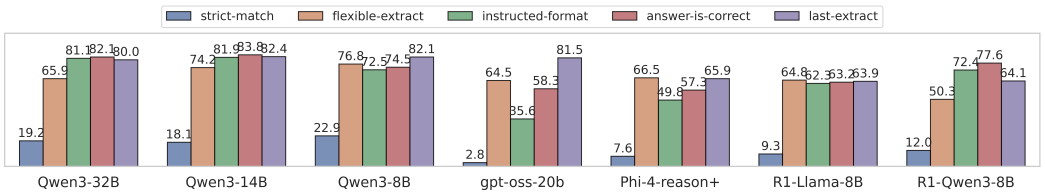


Figure 2: MMLU accuracy under different answer-extraction methods. Responses are considered incorrect if the extraction process fails to find an answer.

Answer Inconsistency Example

The following are multiple choice questions (with answers) about philosophy.
Q: According to Kant, nothing can be called "good" without qualification except _____.
(A) right action (B) good consequences (C) happiness (D) a good will
<think>Okay, so I need to figure out the answer to this Kant question. ⋯ but not the root of morality according to him. Happiness (C) is not the foundation. So D is correct.\n </think>
Answer: **answer-is-correct** : (D) a good will
Explanation:
Immanuel Kant, a central figure in deontological ethics, argues in his *Groundwork of the Metaphysics of Morals* ⋯
- **(B) Good consequences** are irrelevant to Kant, as he rejects consequentialism (e.g., utilitarianism).
- ** **flexible-match** : (C) Happiness** is not inherently good, as it can be pursued immorally (e.g., ⋯
Kant famously states: *"**last-extract** : A good will is the only thing that is good without qualification."* This underscores his belief that moral value lies in the *will* to act according to duty, not in outcomes or subjective desires.
strict-match & instructed-format : [invalid]

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.

potential self-correction within the model's response. When </think> is absent, the rules are applied to the full output. If extraction rules fail to find an answer, the response is considered incorrect. The regular expressions used are described in the Appendix A.3.

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 rules on a given task. Consequently, we show that benchmark performance is not solely dependent on reasoning models' ability but significantly influenced by the chosen extraction rules.

4.2 Results

4.2.1 Performance Fluctuation

Figure 2 shows that both MMLU accuracy and model rankings vary substantially across extraction methods. With strict-match, Qwen3-8B ranks highest, followed by Qwen3-32B and Qwen3-14B. The expanded answer-is-correct rules improve all models' scores and shift the leading order to Qwen3-14B, Qwen3-32B, and R1-Qwen3-8B. Similar changes occur with flexible-match and last-extract: Qwen3-8B leads under the former, whereas Qwen3-14B leads under the latter. The performance of gpt-oss-20b also improves substantially with last-extract. Meanwhile, instructed-format does not consistently outperform the other methods despite explicitly requesting a particular answer format.

These differences also affect comparisons within model families. For example, some extraction methods rank smaller Qwen3 models above Qwen3-32B, illustrating how extraction choices can influence conclusions about relative model capabilities. Overall, benchmark scores reflect both the generated responses and the procedures used to interpret them. Discrepancies between publicly reported and reproduced scores may therefore arise not only from differences in input prompts, but also from undocumented answer-extraction choices.

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



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.6.

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.

	Q3-32B	**Q3-14B**	**Q3-8B**	**gpt20b**	**Phi4+**	**R1-Ll**	**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	45.6	69.9	26.6	25.5
Incorrect	32.2	22.6	53.6	49.2	29.6	65.7	19.2
Invalid	30.7	43.6	4.3	5.2	0.5	7.7	55.3

fail. answer-is-correct successfully locates an answer after **Answer:**. However, flexible-match extracts option letters (A), (B), and (C) in the explanation and chooses the last one (C). The simple yet effective last-extract finds "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.5.

Figure 4 further illustrates this issue by showing how the distribution of answers extracted from the same outputs 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 model's final performance, suggesting that the choice 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 reach the generation limit, 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 final answers derived from these incomplete outputs. The results reveal a high rate of invalid extraction, implying 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 Regeneration

Our analysis has shown that the final answer of reasoning models and the performance are highly sensitive to 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.

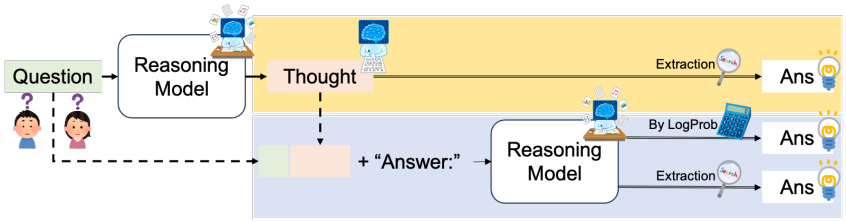


Figure 5: The proposed ANSWER REGENERATION framework for finding answers in model output. The yellow box indicates the conventional method of direct extraction, while the blue box indicates the proposed framework.

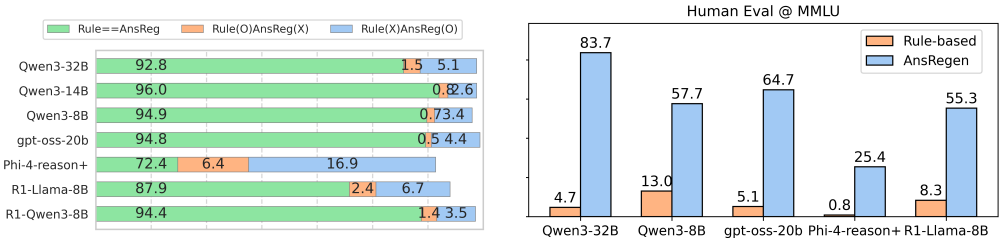


Figure 6: (left) Agreement and correctness transitions between rule-based extraction (Rule) and Answer Regeneration (AnsRegen). Cases in which the methods return different answers and both are incorrect are omitted. (right) The accuracy of answers extracted from the model's thought, as determined by human evaluation. We sample 300 instances per model where extraction and regeneration disagree. Results are not reported for cases where the model failed to provide a definitive answer or provided multiple option labels.

5.1 METHOD

Our proposed framework ANSWER REGENERATION, illustrated in Figure 5, tackles the challenge. Instead of attempting to parse a final answer from the model's full reasoning response, 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 produce a concise answer conditioned on its prior response in an additional inference step. Because the second pass receives both the original question and the prior response, it may recover an explicitly stated answer, select among conflicting candidates, or produce an answer when no final commitment is present. We therefore consider ANSWER REGENERATION as an answer-normalization protocol rather than assuming that every change from rule-based extraction is an extraction correction. Its introduced overhead by additional inference is practically small (i.e., only a few tokens relative to the thinking tokens); the prefill and generated-token counts are reported in Appendix A.7.

5.2 RESULTS

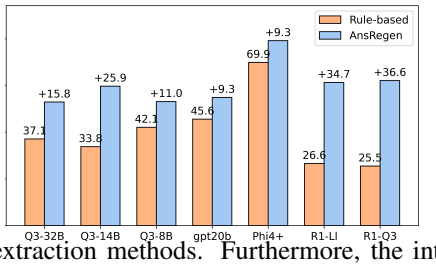
5.2.1 BENCHMARK PERFORMANCE AND HUMAN EVALUATION

Quantitative Scores. Table 2 (left) shows that ANSWER REGENERATION achieves higher MMLU accuracy than the best tested rule-based extractor for each model and performs competitively with xFinder. Figure 6 (left) summarizes agreement and correctness transitions between the compared methods. These transitions measure changes in benchmark correctness; they do not by themselves distinguish faithful answer recovery from second-pass answer revision. Our human evaluation provides evidence of better agreement with human interpretation on the sampled disagreements (but we acknowledge that it does not establish faithfulness across all responses).

Qualitative Human Evaluation Because performance scores alone do not fully justify the superiority of a method, we validated the accuracy of our approach through a human evaluation. We selected 300 instances per model from the outputs of Qwen3-32B, Qwen3-8B, gpt-oss-20b, Phi4-reason+, and R1-Llama-8B where the extraction and regeneration results disagreed. Three annotators identified the answer intended by each original response, independently of whether that answer was correct relative to the benchmark reference. We refer to these annotations as human-interpreted answers. As shown in Figure 6 (right), the alignment of ANSWER REGENERATION with the human

Table 2: (left) MMLU performance comparison between conventional answer extraction and ANSWER REGENERATION. We report each model's performance using its best-performing extraction method. (right) Model performance evaluated on outputs where the reasoning process is incomplete, using the optimal answer extraction algorithm for each model.

	Q3-32B	Q3-14B	Q3-8B	gpt20b	Phi4+	R1-Ll	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(l)	86.4	84.5	81.9	82.8	65.3	64.8	80.1



labels is far superior to that of conventional answer extraction methods. Furthermore, the inter-annotator agreement yielded a Fleiss' Kappa value $\kappa \approx 0.974$, indicating strong reliability.

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 framework. This ranking matches the ordering commonly observed within a model family. Together with the human evaluation, this suggests that the earlier ordering was at least partly driven by extraction failures.

5.2.3 ENHANCED ROBUSTNESS TO RESPONSES

The nature of our proposed ANSWER REGENERATION 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 extractions.

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 the answer. On the other hand, rule-based extractions 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.

In cases of incomplete thinking, our framework can select the final answer even when the thought does not include an explicit final answer, as presented in Table 2 (right).

6 STUDIES ON ADDITIONAL TASKS

6.1 COMPLEX MULTIPLE-CHOICE QA

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 (left), 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 the scores closer to the publicly reported performance[2]. 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. This indicates that ANSWER REGENERATION can reduce human efforts for rule optimization.

Performance differs substantially between the two xFinder backbones, showing that extractor choices also affect evaluation results. We interpret this as sensitivity to the extraction configuration rather than general inferiority of xFinder. ANSWER REGENERATION avoids selecting a separate extractor by using the evaluated model itself.

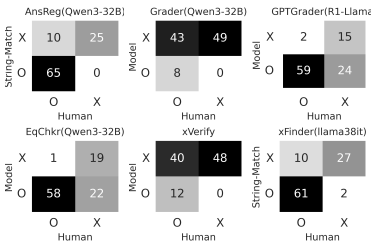
[2] https://artificialanalysis.ai/evaluations/mmlu-pro.

Table 3: Model performance on MMLU-Pro (left) and GSM8K (right). The evaluation utilizes the answer extraction algorithms used in our MMLU analysis, including the built-in algorithm from lm-evaluation-harness.

	MMLU-Pro							GSM8K						
	Q3-32B	**Q3-14B**	**Q3-8B**	**gpt20b**	**Phi4+**	**R1-Ll**	**R1-Q3**	**Q3-32B**	**Q3-14B**	**Q3-8B**	**gpt20b**	**Phi4+**	**R1-Ll**	**R1-Q3**
strict	15.3	13.0	15.7	1.3	4.8	6.8	10.9	3.3	2.7	1.7	0.6	1.5	0.0	0.1
flexible	47.2	47.1	47.1	1.7	0.5	38.0	41.3	33.3	33.5	19.3	**92.0**	**76.5**	**69.2**	85.1
instructed	52.6	59.5	45.8	32.3	**43.8**	38.7	49.7	**93.5**	**92.2**	88.6	48.8	65.6	54.8	**85.8**
ans-is-corr	68.4	65.2	64.2	34.6	32.7	37.6	53.5	89.6	87.6	**91.9**	63.8	59.7	63.1	83.4
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**	**96.8**	**96.4**	**95.5**	92.4	**93.6**	76.0	91.1
xFinder(q)	66.6	63.4	60.6	52.6	33.7	38.3	52.0	95.5	94.5	93.4	93.6	75.7	75.7	90.6
xFinder(l)	74.0	72.1	68.3	70.3	**61.3**	43.7	65.2	95.9	95.0	93.5	**94.9**	88.7	**78.1**	**92.7**
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 4: (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.

	Method	**Evaluator**	**Q3-32B**	**Q3-14B**	**Q3-8B**	**gpt20b**	**Phi4+**	**R1-Ll**	**R1-Q3**
String Match	ans-is	-	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	Q3-32B	3.1*	3.8	3.6	4.8	1.1	1.3	2.9
		gpt20b	47.0	53.4	46.7	50.8	18.1	18.3	39.2
		Phi4+	48.6	55.3	48.1	47.6	11.3	18.4	40.5
		R1-Ll	69.2*	68.1	62.8	72.0	60.1	58.6	57.5
	Equality Checker	Q3-32B	7.3*	5.2	6.1	14.3	40.9	32.0	8.2
		gpt20b	46.6	48.9	46.8	40.1	31.2	36.3	46.4
		Phi4+	61.9	63.2	57.1	61.1	26.7	28.2	50.4
		R1-Ll	67.1	62.5	57.6	63.7	40.0	36.6	53.4
	xVerify	-8B-I	21.6*	24.9	22.1	16.6	3.9	8.0	18.7
	xFinder	qwen1505	50.2	42.9	40.1	39.1	11.8	23.4	31.6
		llama38it	61.1*	63.5	54.8	55.4	53.6	50.5	52.8



6.2 Short-Answer Math Problems

We explore the effectiveness of our framework on mathematical reasoning tasks using GSM8K benchmark (Cobbe et al., 2021), which requires numeric answers without predefined answer choices.

As shown in Table 3 (right), instructed-format, a template specifically recommended for mathematical problems, and flexible-match perform 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 human evaluations on instances where the two methods disagree. In these cases, Answer Regeneration identifies the correct answer in 16.3% of the samples, compared with only 6.1% for the answer extraction method. Annotator agreement is high ($\kappa \approx 0.946$): among 132 Qwen3-32B samples, the three annotators disagreed on only 4 examples. These results underscore the greater reliability of our framework, even in complex, structured yet open-ended domains.

6.3 Open-ended Question Answering

To account for variation among semantically equivalent answers, we use TriviaQA (Joshi et al., 2017), known for its extensive gold answer variations and aliases.

As illustrated in Table 4 (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 Checker[3]. The prompt templates are presented in Appendix A.4. We also evaluate xVerify (Chen et al., 2025), which uses a fine-tuned model for equivalence assessment. xFinder also supports an LLM-as-a-judge protocol.[4]

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 4 (right) reports human evaluation of semantic equivalence on 100 sampled outputs, comparing model judgments with annotator judgments. Inter-annotator agreement among the 3 blinded annotators was Fleiss' κ=0.909, with unanimous judgments on 95 of 100 examples. The 5 disagreements involved borderline alias matches, such as "western" vs. "north-west."

GPTGrader with Qwen3-32B consistently predicts "incorrect" for correct answers, while Equality Checker reveals a bias toward "correct." xVerify defaults to incorrect and R1-Llama-8B exhibits a bias toward "correct." ANSWER REGENERATION followed by string matching avoids reliance on a separate correctness judge and provides a more reliable performance metric; our method achieved 77.7% accuracy, close to human score of 77% in Qwen3-32B, whereas xFinder underestimates at 61.1% in this setting. We do not view this as showing that xFinder is intrinsically weak; rather, it highlights that open-ended QA with long reasoning traces and alias-rich answers is especially sensitive to the chosen extractor or judge protocol. This supports our broader claim that answer normalization should be made explicit and reproducible.

7 DISCUSSION

Our results show that answer extraction is an important component of reasoning-model evaluation. Different extraction procedures can assign different answers and scores to the same generated response, changing model rankings without changing the underlying generation. Consequently, benchmark scores should be interpreted together with the procedures used to obtain them. Reporting prompts and generation settings alone is insufficient when answer extraction remains unspecified.

ANSWER REGENERATION offers a practical way to reduce dependence on handcrafted extraction rules. It uses the evaluated model to normalize its previous response into a concise answer, avoiding the selection and deployment of a separate extractor. However, this does not fully eliminate evaluation sensitivity: results may still depend on the regeneration prompt (e.g., "Answer:" vs. "Answer is") and decoding configuration. These components should therefore be documented explicitly.

An important distinction is between recovering an existing answer and producing a new answer conditioned on the original response. Regeneration can resolve formatting variation, select among conflicting candidates, or provide an answer when the response contains no final commitment. Higher benchmark accuracy alone does not distinguish these behaviors. Our human evaluation supports improved recovery of intended answers on sampled disagreements, but we acknowledge its limited coverage.

The additional inference also introduces a cost-benefit tradeoff. Regenerated answers are short, but the second pass processes the original input and response. However, the practical benefit is reduced dependence on model- and task-specific extraction rules.

Appendix A.8 provides a correctness-flip analysis, results on commercial models, and additional experiments with prompt-aligned extraction formats such as JSON.

8 CONCLUSION

We show that answer extraction is a major source of variance in reasoning-model evaluation and can substantially affect reported scores and model rankings. ANSWER REGENERATION offers a simple, tuning-free alternative to rule-based extraction and judge-based evaluation. It provides a practical answer-normalization protocol that reduces dependence on handcrafted extraction rules, while its gains may include both recovery of existing answers and second-pass answer selection or revision.

[3] https://artificialanalysis.ai/methodology

[4] In our setting, however, the judge exhibited a strong positional bias toward predicting only CORRECT or INCORRECT especially for long reasoning traces. We therefore restrict its use to a string-matching protocol.

AI USE STATEMENT

In this work, we used generative AI tools for English-language editing. We have not used generative AI tools for other tasks. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe all the experiment details, including input prompt, sampling hyperparameters, and extraction rules in the Appendix.

REFERENCES

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, et al. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*, 2025.

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. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards, 2024. URL https://arxiv.org/abs/2402.01781.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, et al. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024.

Nikhil Chandak, Shashwat Goel, Ameya Prabhu, Moritz Hardt, and Jonas Geiping. Answer matching outperforms multiple choice for language model evaluation. *arXiv preprint arXiv:2507.02856*, 2025.

Ding Chen, Qingchen Yu, Pengyuan Wang, Wentao Zhang, Bo Tang, Feiyu Xiong, Xinchi Li, Minchuan Yang, and Zhiyu Li. xverify: Efficient answer verifier for reasoning model evaluations. *arXiv preprint arXiv:2504.10481*, 2025.

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

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Nathan Habib, Clémentine Fourrier, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. Lighteval: A lightweight framework for llm evaluation, 2023. URL https://github.com/huggingface/lighteval.

Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. Does prompt formatting have any impact on llm performance?, 2024. URL https://arxiv.org/abs/2411.10541.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 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)*, pp. 1601–1611, 2017.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 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)*, pp. 5591–5606, 2023.

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. Holistic evaluation of language models, 2023. URL https://arxiv.org/abs/2211.09110.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? a call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024. doi: 10.1162/tacl_a_00681. URL https://aclanthology.org/2024.tacl-1.52/.

Francesco Maria Molfese, Luca Moroni, Luca Gioffré, Alessandro Scirè, Simone Conia, and Roberto Navigli. Right answer, wrong score: Uncovering the inconsistencies of llm evaluation in multiple-choice question answering. *arXiv preprint arXiv:2503.14996*, 2025.

OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

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

Felipe Maia Polo, Ronald Xu, Lucas Weber, Mírian Silva, Onkar Bhardwaj, Leshem Choshen, Allysson Flavio Melo de Oliveira, Yuekai Sun, and Mikhail Yurochkin. Efficient multi-prompt evaluation of llms, 2024. URL https://arxiv.org/abs/2405.17202.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting, 2024. URL https://arxiv.org/abs/2310.11324.

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

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

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Qingchen Yu, Zifan Zheng, Shichao Song, Zhiyu Li, Feiyu Xiong, Bo Tang, and Ding Chen. xfinder: Robust and pinpoint answer extraction for large language models. *arXiv preprint arXiv:2405.11874*, 2024.

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

A Appendix

A.1 Evaluation Toolkits

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. **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 Hyperparameters and Prompts

Hyperparameters The hyperparameter settings follow the best practices of each model:

- Qwen3 families in huggingface model card, it recommends temperature=0.6, top_p=0.95, top_k=20, and min_p=0 for thinking mode (enable_thinking=True) and temperature=0.7, top_p=0.8, top_k=20, and min_p=0 for non-thinking mode (enable_thinking=False).
- gpt-oss-20b in unsloth recommends temperature=1.0, top_p=1.0, and top_k=0.
- phi-4-reasoning-plus in huggingface model card recommends to use temperature=0.8, top_k=50, and top_p=0.95.
- distilled DeepSeek models in huggingface model card is recommended to use temperature=0.6 and top_p=0.95
- In Answer Regeneration step, we use greedy decoding.

MMLU & MMLU-Pro Input Prompts with Thinking Template (Example of Qwen3)

<|im_start|>user
{Question}
(A) {Option A}
(B) {Option B}
...
(D) {Option D}
...
(H) {Option H}<|im_end|>
<|im_start|>assistant
<think>

GSM8K & TriviaQA Input Prompts with Thinking Template (Example of Qwen3)

<|im_start|>user
{Question}
<|im_start|>assistant
<think>

A.3 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-match**: (\\([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.4 LLM-AS-A-JUDGE PROMPTS

GPTGrader (Wei et al., 2024) uses 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.

GPTGrader Input Prompts

(Too long to describe here; Kindly refer to Wei et al. (2024))

EqualityChecker Input Prompts

Assess whether the following CANDIDATE ANSWER is CORRECT or INCORRECT.

For the CANDIDATE ANSWER to be correct, it must be consistent with the OFFICIAL ANSWER.

The question, for reference only: question

The OFFICIAL ANSWER: official_answer

CANDIDATE ANSWER TO ASSESS: candidate_answer

Reply only with CORRECT or INCORRECT.

xVerify Input Prompts

You are a diligent and precise assistant tasked with evaluating the correctness of responses. You will receive a question, an output sentence, and the correct answer. Your task is to determine if the output sentence accurately answers the question based on the provided correct answer. Respond with either [Correct] or [Incorrect].
-
Special considerations:
1. **Multiple Answers**: If the output contains multiple answers, evaluate whether later answers modify or correct earlier ones. In such cases, compare the final answer with the correct answer. If the final answer is unclear or incorrect, respond with [Incorrect].
2. **Mathematical Problems**: If the formats differ but the answers are mathematically equivalent, respond with [Correct].
3. **Explicit Options**: If the question provides explicit candidate answers, the output will be considered correct if it clearly indicates the correct option's code or the correct option's content.
4. **No Explicit Options**: If the question does not provide explicit options, the output must align with the correct answer in content and meaning to be considered [Correct].
-
Question: """{question}"""
Output sentence: """{output}"""
Correct answer: {answer}
Judgement:

xFinder Input Prompts

You are a diligent and precise assistant tasked with evaluating the correctness of responses.
—
We request your feedback on whether the model's response correctly answers the user question above. Based on the reference answer, please determine if the model's answer is correct.
Please first output "correct" or "incorrect" on a single line. In the subsequent line, provide a brief explanation of your judgment, ensuring it is objective and clear.
—
Question: {question}
Reference Answer: {reference}
Model's Answer: {answer}

A.5 Additional Examples of Extraction Disagreement

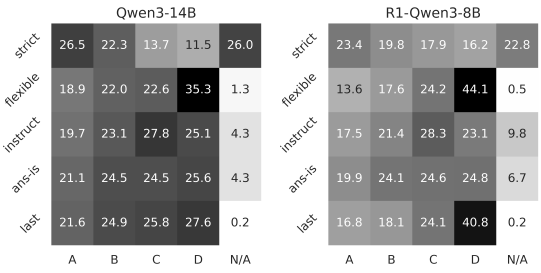
Another example from Qwen3-32B output illustrating how the final answer can vary significantly depending on the extraction method used. The graphical boxes and bold text highlight the specific text extracted by each algorithm.

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 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**
strict-match & ans-is-correct: [invalid]

A.6 Distribution of Extracted Final Answers

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.



A.7 Quantitative Measurement of the Inference Overhead

We characterize the additional inference using both the number of input tokens processed during prefill and the number of generated answer tokens. For MMLU and MMLU-pro, Answer Regeneration typically requires only a few tokens as the generated output is constrained to the choice letters (e.g., A, B, C, D). Table 5 presents the average number of output tokens consumed by both the initial thinking process and the subsequent Answer Regeneration step for various models:

Table 5: Quantitative measurement of the inference overhead in GSM8K and TriviaQA.

Dataset	Tokens	Q3-32B	Q3-14B	Q3-8B	gpt20b	Phi4+	R1-Q3	R1-Ll
GSM8K	Avg. #Think Tokens	1575.8	1590.1	1986.1	283.4	4095.8	1931.3	466.7
	Avg. #AnsRegen Tokens	13.2	8.8	9.8	17.0	32.0	24.7	7.7
TriviaQA	Avg. #Think Tokens	963.4	881.1	830.6	952.3	4093.8	681.3	712.2
	Avg. #AnsRegen Tokens	17.5	11.9	16.9	23.7	30.7	14.9	14.3

Regenerated answers are short relative to the initial reasoning responses. The additional call also processes the original question and prior response, with reported prefill lengths of 542–3,767 tokens per item, on top of a short 6–32 token for decoding. These counts characterize the additional workload but do not establish wall-clock overhead or a cost advantage over external extractors.

A.8 Further Discussion

Table 6: Helpful/Harmful flips introduced by Answer Regeneration on MMLU.

Model	Q3-32B	Q3-14B	Q3-8B	gpt20b	Phi4+	R1-Q3	R1-Ll
Helpful (incorrect→correct)	1,134	554	483	823	3,020	819	399
Harmful (correct→incorrect)	55	47	112	65	307	26	201
Harmful Rate	0.4%	0.3%	0.8%	0.5%	2.2%	0.2%	1.4%

Failure-Case Analysis. We investigate flips from correct to incorrect by Answer Regeneration in Table 6. Helpful and harmful flips denote changes in benchmark correctness relative to the best extractor per model and benchmarks. These flips affect a small proportion of examples for each model, and are outnumbered by helpful corrections in all cases (e.g., Phi-4: 307 harmful vs. 3,020 helpful). Some harmful flips are cases where the model's stated final answer is itself wrong, and Answer Regeneration faithfully reports it while a rule "wins" only by accidentally extracting a correct option mentioned mid-trace, such as the rumination artifact we argued against. The example below illustrates how an apparent harmful flip can nevertheless preserve the stated final answer. Given Q: "The redness of a sunset is due mostly to light that hasn't been ___" $\cdots$ (A) absorbed (B) transmitted (C) scattered (D) polarized. Gold: (C). The trace reasons correctly that red is "the portion that hasn't been scattered" (=C), then equates it with "transmitted" and concludes "Answer: (B) transmitted." Answer Regeneration faithfully returns the model's stated (B); a rule that happens to grab the earlier "scattered" returns the correct (C). The error is the model's reasoning, not the extraction.

Answer Regeneration on Commercial Models. We choose 3 models: OpenAI gpt-5.4-mini, Gemini gemini-3.5-flash, and Claude claude-haiku-4-5. In Table 7, Answer Regeneration shows higher scores than the best of 3 fixed rules without per-model or per-format tuning. These results transfer our open-model finding to commercial models. Note that models that hide their reasoning, e.g. OpenAI o-series, allow conditioning only on the visible output.

Testing Other Extraction Rules. We tested 4 prompt-aligned format instructions–JSON ({"answer": "X"}), an Answer: X line, \boxed{X}, and <answer>X</answer>–each paired with its own matching extractor across all evaluated models. As shown in Table 8, formatting instability persists even under strict prompt-extractor alignment, demonstrating that performance fluctuations are not merely post-hoc parsing artifacts. Instructed-format accuracy swings widely depending on the choice of instruction, with a single model's performance varying by up to 23 points on MMLU,

Table 7: Evaluation of Answer Regeneration on commercial models. Gold-answer occurrence rate denotes the percentage of responses containing the benchmark gold answer.

Dataset	Model	AnsRegen	Best Rules	Gold-answer Occur.
MMLU	gpt-5.4-mini	88.1	78.3	88.4
	gemini-3.5-flash	93.7	91.3	94.1
	claude-haiku-4-5	88.0	76.9	88.3
GSM8K	gpt-5.4-mini	95.7	94.7	96.4
	gemini-3.5-flash	97.3	95.9	98.4
	claude-haiku-4-5	96.8	95.3	98.0
TriviaQA	gpt-5.4-mini	87.6	84.1	88.9
	gemini-3.5-flash	93.6	80.8	96.2
	claude-haiku-4-5	87.3	82.5	90.7

Table 8: Model performances across aligned format instructions

Benchmark	Model	json	"Answer:"	boxed	tag	AnsReg
MMLU	Qwen3-32B	79.4	85.0	**87.3**	76.2	87.1
	Qwen3-14B	83.7	84.0	**85.3**	81.0	85.0
	Qwen3-8B	80.1	81.2	82.3	69.5	**83.3**
	gpt-oss-20b	84.4	83.3	**84.7**	84.3	**84.7**
	Phi-4-reason+	**78.1**	75.4	74.3	75.0	77.4
	R1-Qwen3-8B	61.2	58.9	77.8	64.5	**81.1**
	R1-Llama-8B	56.5	36.2	59.5	51.4	**68.8**
MMLU-Pro	Qwen3-32B	57.0	61.5	72.7	57.1	**74.8**
	Qwen3-14B	70.3	68.8	69.9	56.2	**72.6**
	Qwen3-8B	60.7	61.2	65.1	37.4	**74.7**
	gpt-oss-20b	74.1	72.9	74.1	74.0	**74.6**
	Phi-4-reason+	**68.8**	67.4	67.7	67.4	58.7
	R1-Qwen3-8B	41.7	42.9	54.6	40.3	**71.1**
	R1-Llama-8B	34.8	24.9	36.6	25.5	**52.1**
GSM8K	Qwen3-32B	95.6	90.8	96.6	91.6	**96.8**
	Qwen3-14B	96.1	**96.2**	96.0	85.1	**96.4**
	Qwen3-8B	95.2	88.9	95.1	39.4	**95.5**
	gpt-oss-20b	95.5	**96.1**	95.8	95.8	92.4
	Phi-4-reason+	77.0	75.5	74.5	69.8	**93.6**
	R1-Qwen3-8B	45.0	92.3	**92.8**	55.3	91.1
	R1-Llama-8B	21.1	57.8	64.4	7.6	**76.0**
TriviaQA	Qwen3-32B	76.0	76.0	70.9	76.2	**77.7**
	Qwen3-14B	74.1	71.9	70.0	74.2	**77.6**
	Qwen3-8B	67.2	65.1	61.3	67.0	**67.7**
	gpt-oss-20b	**67.3**	63.6	55.8	66.3	62.3
	Phi-4-reason+	66.3	64.0	66.0	58.7	**68.6**
	R1-Qwen3-8B	25.7	29.8	39.1	28.7	**60.8**
	R1-Llama-8B	31.2	30.9	33.4	31.8	**58.2**

28 points on MMLU-Pro, 57 points on GSM8K, and 13 points on TriviaQA (e.g., R1-Llama-8B on GSM8K drops from 64.4% using boxed to 7.6% using tag). Furthermore, no single format emerges as a universal winner, as the optimal format varies unpredictably across both models and benchmarks. At the reported precision, ANSWER REGENERATION exceeds the best of the 4 aligned formats in 20 of 28 model-task settings. The best aligned format is selected separately for each setting. Notable exceptions include Phi-4-reason+ on MMLU-Pro and gpt-oss-20b on GSM8K and TriviaQA.

Relation to xFinder. We emphasize that our goal is not to show that ANSWER REGENERATION universally dominates specialized extractors such as xFinder. Rather, xFinder represents a strong supervised extraction baseline and helps expose another source of evaluator sensitivity: once rule-

based extraction is replaced by a model-based extractor, the final score can depend on the extractor backbone, prompt, and task regime. This is reflected in the non-trivial differences between xFinder variants on MMLU-Pro and in the difficulty of applying judge-style protocols to long reasoning traces in TriviaQA. ANSWER REGENERATION occupies a different design point: it uses the evaluated model itself, under a fixed deterministic non-reasoning decoding step, to normalize its own prior output into a concise answer. Thus, its main advantage is not strict superiority over xFinder in every setting, but providing a simple, competitive, and tuning-free reference protocol that reduces both handcrafted-rule selection and external-extractor selection.