Right Answer, Wrong Score: Uncovering the Inconsistencies of LLM Evaluation in Multiple-Choice Question Answering

Anonymous ACL submission

Abstract

One of the most widely used tasks to evaluate Large Language Models (LLMs) is Multiple-Choice Question Answering (MCQA). While open-ended question answering tasks are more challenging to evaluate, MCQA tasks are, in principle, easier to assess, as the model's answer is thought to be simple to extract and is directly compared to a set of predefined choices. However, recent studies have started to question the reliability of MCQA evaluation, showing that multiple factors can significantly impact the reported performance of LLMs, especially when the model generates free-form text before selecting one of the answer choices. In this work, we shed light on the inconsistencies of MCQA evaluation strategies, which can lead to inaccurate and misleading model comparisons. We systematically analyze whether existing answer extraction methods are aligned with human judgment, and how they are influenced by answer constraints in the prompt across different domains. Our experiments demonstrate that traditional evaluation strategies often underestimate LLM capabilities, while LLM-based answer extractors are prone to systematic errors. Moreover, we reveal a fundamental trade-off between including format constraints in the prompt to simplify answer extraction and allowing models to generate free-form text to improve reasoning. Our findings call for standardized evaluation methodologies and highlight the need for more reliable and consistent MCQA evaluation practices.

1 Introduction

MCQA is one of the most common tasks used to evaluate LLMs across various domains, including commonsense reasoning (Talmor et al., 2019; Mihaylov et al., 2018; Bisk et al., 2019; Sap et al., 2019), grade-school science (Clark et al., 2018), and multi-domain challenges (Hendrycks et al., 2021; Wang et al., 2024b; Gema et al., 2025), among others. MCQA is straightforward: given a question and a set of answer choices, the model must select the correct answer. Despite the apparent simplicity of the task, the evaluation of LLMs on MCQA benchmarks is not trivial, as the model's answer must be extracted from its generated output or selected based on the probabilities assigned to each answer choice.

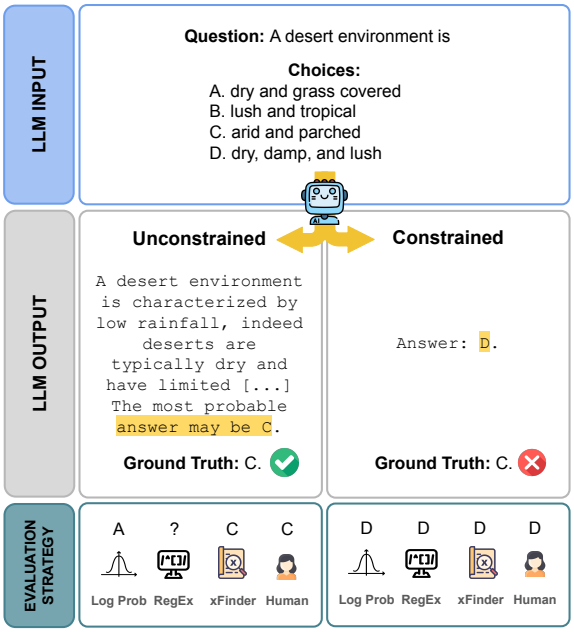


Figure 1: Different evaluation strategies (RegEx, Logprobs, xFinder and Human) and prompt settings (constrained or not) can lead to discrepancies in model performance.

Since the introduction of techniques that enhance the reasoning capabilities of LLMs, such as Chain-of-Thought (Kojima et al., 2023, CoT), most models are now prompted to generate free-form text before selecting an answer, which improves their accuracy but complicates the extraction of the model's intended answer, as shown in Figure 1. Recently, the reliability of MCQA evaluation strategies has been called into question, as different methods can lead to significant variations

in reported model performance (Wang et al., 2024a; Yu et al., 2024). For example, measuring the probability of the first token generated by the model to be the label ("A" to "D") of the correct answer can yield different results compared to extracting the answer from the model's output text (Hendrycks et al., 2021; Robinson et al., 2023; Zheng et al., 2024). As researchers introduce more sophisticated reasoning capabilities—such as test-time scaling and "thinking" mechanisms—the reliability of MCQA evaluation strategies becomes increasingly important to ensure fair model comparisons.

In this work, we investigate the reliability of MCQA evaluation strategies, focusing on how different methods for extracting the model's answer impact the reported performance of LLMs. To the best of our knowledge, we introduce the first comprehensive analysis on how different factors in the evaluation strategy, prompt setting, and benchmark domain influence model performance. Importantly, we conduct a human evaluation to assess the alignment between evaluation strategies and human judgment, highlighting the limitations and inconsistencies of current practices. Finally, we identify cases in which existing evaluation methods systematically fail, including those based on LLMs, highlighting which challenges still remain unsolved in MCQA evaluation. In summary, we address the following critical research questions:

- **RQ1**: How well do current evaluation strategies align with human judgment?
- **RQ2**: How does the choice of evaluation strategy and prompt setting impact LLM performance?
- **RQ3**: How does model performance shift across different benchmark domains for each prompt setting and evaluation strategy?
- **RQ4**: How reliable are LLM-based methods in extracting a model's intended answer?

We hope our work will lead to more rigorous and standardized evaluation practices. We release our code and data at `omitted.link`.

2 Related Work

This section surveys previous research in MCQA evaluation on two main aspects: how task format and inherent biases affect model performance (Section 2.1), and how different strategies for answer extraction influence evaluation outcomes (Section 2.2).

2.1 Task Format and Label Bias

The research community has demonstrated that even minor variations in evaluation setup can significantly impact LLM performance. For instance, a seminal work by Robinson et al. (2023) studied the impact of task format on LLM performance, showing that models struggle with multiple-choice symbol binding, i.e., maintaining order invariance when reasoning over different answer choices. Building on these findings, Zheng et al. (2024) documented systematic position biases in LLMs, showing that models disproportionately favor certain answer positions (e.g., "Option A" over "Option B"). This positional sensitivity was further validated by Alzahrani et al. (2024), who demonstrated that reordering answer choices or modifying answer selection methods can alter leaderboard rankings.

A parallel line of research has questioned whether LLMs truly require question context for MCQA. Wang et al. (2024b) showed that LLMs can achieve high performance on MCQA benchmarks without access to the question, suggesting that they may rely on spurious correlations in the answer choices. Further evidence of such shortcuts emerged in Wang et al. (2025)'s work, suggesting that LLMs often select answers by eliminating clearly incorrect options rather than identifying the most accurate choice.

Unlike previous studies focused on answer selection biases and task formulation, our work examines how variations in prompt settings and format—including the trade-off between prompts that impose constraints on the answer format and those that allow free-form text generation—affect model performance across evaluation strategies.

2.2 Evaluation Strategies

Current MCQA evaluation approaches broadly fall into two categories: those based on direct probability analysis and those that require answer extraction from the model's output. If, given a question, there is a finite set of answer choices, we can assign a simple label to each choice (e.g., "A" to "D") and compute the next-token probability distribution for each label after "`Answer:`". This method, which we refer to as Logprobs, has been widely used in recent studies (Hendrycks et al., 2021; Robinson et al., 2023; Zheng et al.,

2024). Although Logprobs is computationally efficient and conceptually straightforward, it cannot be applied when the model generates free-form text before selecting an answer, which is becoming an increasingly common practice in MCQA evaluation, e.g., with Chain-of-Thought (Kojima et al., 2023). In such cases, answer extraction methods are required, such as those based on regular expressions (RegEx) or LLM-based models like xFinder (Yu et al., 2024). RegEx methods can be used to scan the model's output for predefined patterns, such as "`Answer: {label}`" or "`The answer is: {label}`" (Wang et al., 2024b). However, the effectiveness of RegEx methods requires careful crafting of patterns, which can still lead to high miss rates (Yu et al., 2024), especially when the model generates complex reasoning chains. In contrast, classifier- or LLM-based methods, e.g., xFinder, are fine-tuned to extract the model's intended answer from its output, given the question and answer choices (Yu et al., 2024).

Our work builds on these studies by systematically analyzing the reliability of different MCQA evaluation strategies, examining how evaluation strategies, prompt settings, and benchmark domains affect performance assessment. Importantly, we evaluate the agreement between automated evaluation strategies and human judgment, providing insights into the challenges and limitations of current practices.

3 Methodology

Our investigation into MCQA evaluation reliability covers three dimensions: evaluation strategies, prompt settings, and benchmark domains. For each dimension, we design controlled experiments that isolate specific variables while maintaining others constant, allowing us to measure:

- The correlation between automated evaluation strategies and human judgement;
- The impact of prompt constraints on model reasoning and answer extraction;
- The variation in evaluation reliability across different domains.

In the following sections, we first formalize the MCQA task and our evaluation framework (Section 3.1), then outline the evaluation strategies under investigation (Section 3.2), and finally describe our prompt settings (Section 3.3).

3.1 Task Formulation

Let $\mathcal{D} = \{(q_i, C_{q_i}, a_i)\}_{i=1}^{N}$ be a dataset of N multiple-choice instances, where each instance consists of a question $q_i \in Q$, a set of k answer choices $C_{q_i} = \{c_1, c_2, \ldots, c_k\}$, a ground-truth answer $a_i \in C_{q_i}$. The MCQA task requires a model f to generate an output t_i in response to a question q_i and its corresponding choices C_{q_i}. Formally, we write:

$$t_i = f\big(q_i, C_{q_i}\big),$$

where t_i is free-form text or a structured response (e.g., "Answer: C"). To extract the model's intended answer $\hat{c}_i$ from its output t_i, we apply an evaluation strategy s, yielding:

$$\hat{c}_i = s\big(t_i\big).$$

The prediction $\hat{c}_i$ is considered correct if $\hat{c}_i = a_i$. We assess the model's overall performance on a dataset $\mathcal{D}$ by computing the accuracy:

$$\mathrm{Acc}(f, s) \;=\; \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\big[\hat{c}_i = a_i\big],$$

where $\mathbb{1}[\cdot]$ is the indicator function that evaluates to 1 if its argument is true and 0 otherwise. This framework allows us to compare how different evaluation strategies s and prompting configurations influence the final accuracy of model f.

3.2 Evaluation Strategies

To assess LLM performance on MCQA tasks, we need a method to identify or extract the model's intended answer $\hat{c}$ from its output t, given a question q and its answer choices C_q. We experiment with three evaluation strategies, which are representatives of traditional approaches or emerging trends in MCQA evaluation.

Logprobs: Rather than extracting answers from generated text, this strategy analyzes the model's probability distribution over first tokens after a prompt terminating with t_0 = "`Answer:`" (Hendrycks et al., 2021). Formally:

$$s_{\text{logprob}}(q, C_q) = \mathrm{argmax}_{c \in C_q}\, P(c|q, C_q, t_0)$$

where $P(c|q, C_q, t_0)$ is obtained by applying softmax to the model's log-probability vector for each answer choice. While efficient, this method cannot handle free-form text generation or chain-of-thought reasoning.

RegEx: This parameterless method applies a set of regular expressions $\mathcal{R} = \{r_1, \ldots, r_m\}$ to extract the model's answer. For an output t, we define:

$$s_{\text{regex}}(t) = \begin{cases} \text{match}(r_i, t) & \text{if } \exists r_i \in \mathcal{R} \text{ matches } t \\ \emptyset & \text{otherwise} \end{cases}$$

where $\text{match}(r_i, t)$ returns the first answer choice label that matches pattern r_i in t. While computationally efficient, this approach can fail when models generate complex reasoning chains or deviate from expected patterns.

LLM-based answer extraction: This approach uses an LLM s_{llm} fine-tuned to extract answers from arbitrary outputs:

$$s_{\text{llm}}(t, q, C_q) = \text{LLM}(t, q, C_q)$$

LLM-based approaches are relatively new and have shown promising results. Unlike Logprobs and RegEx, LLM-based methods can handle free-form text generation and complex reasoning chains, making them more robust to variations in model output. We evaluate two state-of-the-art models: xFinder-Llama (8B parameters) and xFinder-Qwen (500M parameters), introduced by Yu et al. (2024).

Full details about the RegEx patterns used in this work, as well as the xFinder models can be found in Appendix A.1.

3.3 Prompt Settings

We investigate how four widely-used prompt settings influence both model performance and evaluation reliability. For each setting $p \in \mathcal{P}$, we define a prompt template:

$$\mathcal{P}_p(q, C_q) = \text{sys}_p \oplus \text{inst}_p(q, C_q) \oplus \text{const}_p$$

where sys_p is the system prompt, inst_p the instruction template, and const_p any format constraints. We focus on the following four settings and hypothesize that results on other settings would follow similar trends:

Zero-Shot (ZS). The model receives only a system prompt, followed by the question and available choices. This setting imposes no constraints on the output format, allowing the model complete freedom in response generation.

Zero-Shot Chain-of-Thought (ZS-CoT). This setting prompts the model to use CoT, allowing it to explain its reasoning before selecting the answer (Kojima et al., 2023).

Zero-Shot with Format Constraint (ZS-Const). Similar to ZS, but with a format constraint on the answer. The LLM is prompted to respond in a specific format, e.g., "`Answer: {label}`" (Wang et al., 2024a), which simplifies answer extraction.

Few-Shot (FS). The model is provided with n examples randomly selected from the training or validation set of the benchmark. These examples are structured as multi-turn conversations, following common practice (Gao et al., 2024). There are no constraints on the answer, which the LLM can observe from the provided examples.

These prompt settings allows us to study the trade-off between format constraints vs. free-form text generation, and LLM performance vs. simplicity of answer extraction. Full details on the prompts used in this work can be found in Appendix A.2.

4 Experimental Setup

4.1 Benchmark Selection

We select 3 popular MCQA benchmarks, each targeting different aspects of language understanding:

MMLU-Redux. A manually curated subset of MMLU (Hendrycks et al., 2021) comprising 5,700 questions across 57 domains. This dataset addresses potential quality issues in the original MMLU by incorporating expert review and correction of problematic instances. The domains span four major categories: STEM, Humanities, Social Sciences, and Other.

OpenBookQA. A question-answering dataset (Mihaylov et al., 2018) that tests factual recall and multi-hop reasoning. Each question requires combining scientific facts with common sense.

ARC-Challenge. A collection of grade-school science questions (Clark et al., 2018) selected to be challenging for NLP systems. Questions often require complex reasoning and external knowledge.

Following standard practice, we evaluate on the provided test sets. For few-shot experiments, we randomly sample five examples from training sets when available, or validation otherwise.

4.2 Model Selection

We evaluate five LLMs with different architectures and sizes, ranging from 1 billion to 8 billion parameters. The models are selected to represent a diverse

	Agreement with Humans				
Eval. Strategy	**ZS**	**ZS-CoT**	**ZS-Const**	**FS**	**Avg.**
RegEx	90.7	84.3	97.9	97.3	92.5
Logprobs	74.7	—	94.1	90.4	86.4*
xFinder-Llama	95.8	89.7	98.4	97.3	95.3
xFinder-Qwen	94.8	90.3	98.4	97.3	95.2
Human	98.2	97.0	98.7	100.0	98.5

Table 1: Average agreement between human annotators and evaluation strategies across five LLMs in each prompt setting, measured with Cohen's kappa. The average for Logprobs is computed over three values, excluding ZS-CoT.

set of LLMs, including both high-performing and smaller, efficient models. More specifically, we evaluate the following models:

- **Small-scale LLMs (1B – 4B):** Llama-3.2-1B-Instruct and Phi-3.5-mini-instruct (3.8B parameters);
- **Medium-scale LLMs (4B – 8B):** Llama-2-7B, Qwen2.5-7B-Instruct, and Llama-3.1-8B-Instruct.

Due to budget constraints, we exclude models with more than 8 billion parameters, hypothesizing that larger models would exhibit similar trends to those analyzed in this work.

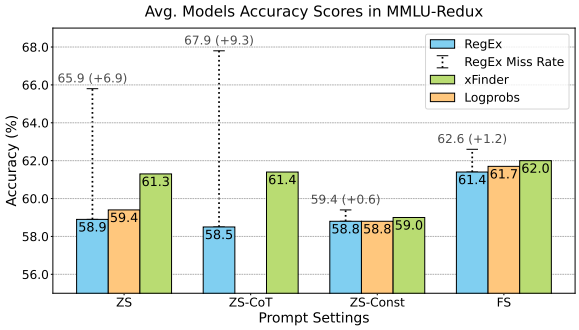
5 Results

RQ1: "*How well do current evaluation strategies align with human judgment?*" To answer this question, we conduct a manual annotation process in which human annotators extract the intended answer from the model's response across all prompt settings and evaluation strategies. First, we randomly sample 50 q, C_q pairs from the MMLU-Redux dataset, and prompt each of the five LLMs under four different settings, resulting in a total of $1,000$ instances to annotate. Then, four human annotators manually extract the intended answer from the model's response, assigning a label from "A" to "D" or a special tag, "[No valid answer]," for cases where the model produces an invalid response. If a response is invalid, annotators are required to specify the reason for invalidity, which can arise from various factors, including: i) conflicting answers (e.g., the reasoning produced by the model supports choice "C" but the model concludes with "Answer: A"), ii) label binding inconsistencies (e.g., the model responds with "Answer: C. bank" where "bank" corresponds to option "B"), iii) refusal to answer (e.g., due to safety concerns or insufficient knowledge), iv) irrelevant response, where the model fails to engage with the question, and v) generation limits, e.g., the model generates a response that exceeds the token limit. In total, each annotator is assigned 400 instances, with 200 instances shared between all annotators to assess inter-annotator agreement. We provide details on the annotation process in Appendix A.3.

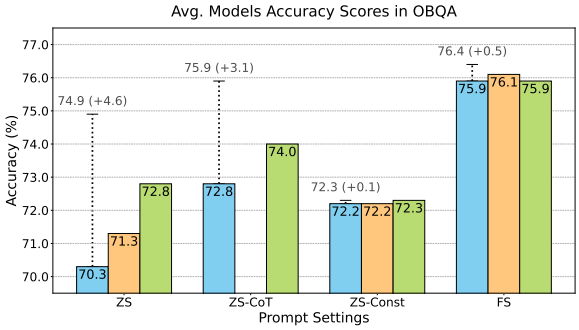
We compute the agreement between annotators using pairwise Cohen's kappa, averaging across all pairs, yielding a score of 98.5, indicating an "*almost perfect*" agreement. This shows that human annotators are consistent in extracting the intended answer from the model's response, providing a reliable benchmark for evaluating the alignment between automated evaluation and human judgment.

Having created a gold standard dataset of $1,000$ instances, we evaluate the agreement between human annotators and automated evaluation strategies using Cohen's kappa[1]. The results are reported in Table 1. We observe that LLM-based approaches for answer extraction generally achieve higher agreement with human judgment compared to traditional methods. In particular, xFinder-Llama displays the highest agreement with humans across all prompt settings, outperforming traditional strategies, namely, RegEx and Logprobs, by a significant margin. However, the agreement between humans and evaluation strategies is not consistent across prompt settings. Recent work on LLMs is moving away from constrained prompts to allow models to generate free-form text to improve reasoning, but our analysis shows that moving from ZS-Const to ZS leads to a significant drop in agreement between humans and evaluation strategies: -2.6% for xFinder-Llama (from 98.4 to 95.8), -3.6% for xFinder-Qwen (from 98.4 to 94.8), -7.2% for RegEx (from 97.9 to 90.7), and -19.2% for Logprobs (from 94.1 to 74.7). This is even more pronounced in the ZS-CoT setting, where the agreement between xFinder-Llama—the best model on average, with 8B parameters—and humans drops by 8.7%. These results suggest that the extent to which models adhere to the required format has a significant impact on the reliability of the employed evaluation strategy, and state-of-the-art LLM-based approaches are not immune to this variability yet, especially in settings where models generate free-

[1]We used majority voting for the 200 shared instances and the single available annotation for the remaining 800.

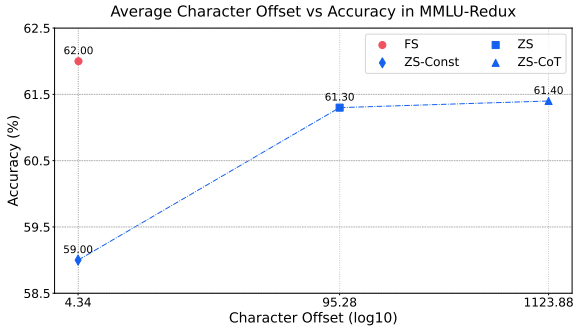


(a) MMLU-Redux dataset

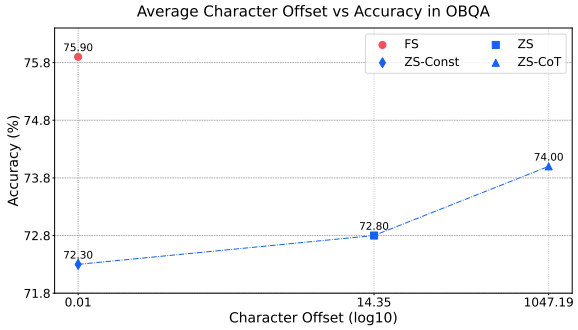


(b) OBQA dataset

Figure 2: Average accuracy scores across five LLMs and four prompt settings when evaluated on the MMLU-Redux (Figure 2a) and OBQA (Figure 2b) datasets. Dotted lines indicate the RegEx miss rate.



(a) MMLU-Redux dataset



(b) OBQA dataset

Figure 3: The plots show the relationship between average answer offset ($\log_{10}$ scale) and accuracy (%) for different settings using the xFinder evaluation strategy.

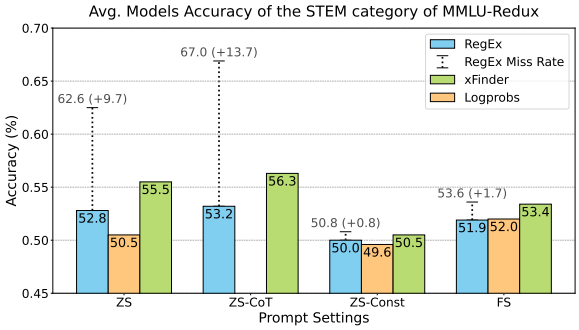
form text.

RQ2: "*How does the choice of evaluation strategy and prompt setting impact LLM performance?*" The disagreement between human annotators and evaluation strategies shown in Table 1 raises the question of how the choice of evaluation strategy and prompt setting affects LLM performance. Therefore, we analyze the behavior of the five LLMs on MMLU-Redux, OBQA, and ARC using the four prompt settings and the three evaluation strategies. Figures 2a and 2b present the results on the MMLU-Redux and OBQA datasets, respectively. Additionally, we provide the results on the ARC dataset and the individual performance of all the LLMs in Appendices A.4 and A.5.

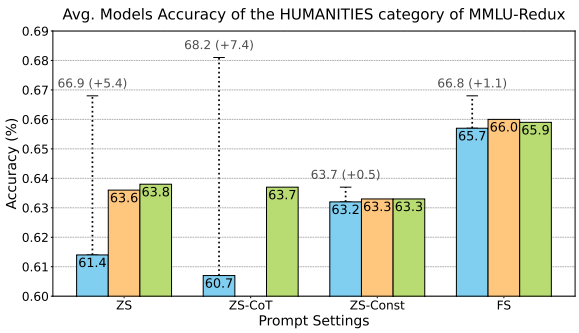
The plots show that in prompt settings that constrain the output format—either explicitly, as in ZS-Const, or implicitly through few-shot examples, as in FS—LLMs performance remain stable across evaluation strategies, which is consistent with the high agreement between humans and evaluation strategies in these settings, i.e., simplifying answer extraction leads to more reliable evaluation outcomes. However, our results also show that a simplified evaluation process can hide the true capabilities of current LLMs, as models in ZS or ZS-CoT generate outputs in which the answer is harder to extract, e.g., leading to a higher RegEx miss rate.[2] Interestingly, the prompt settings that show the largest differences between the results obtained with different evaluation strategies are the ones where the disagreement with human annotators is the highest, i.e., ZS and ZS-CoT. As the research community moves towards letting model generate more complex free text before selecting an answer, current evaluation strategies are likely to become less reliable.

To assess the trade-off between generating longer responses and model performance, we study the improvement in model performance as the average answer offset increases. The average answer offset is defined as the number of characters after which one of the available RegEx patterns matches the model's intended answer. Figure 3 shows that, while higher offsets generally lead to better performance, the gain is often marginal beyond a certain threshold, e.g., moving from 10^2 to 10^3 characters

[2]We define RegEx miss rate as the percentage of instances where no RegEx pattern is able to extract an answer from the model output.



(a) STEM category



(b) HUMANITIES category

Figure 4: Average accuracy scores across five LLMs and four prompt settings when evaluated on the STEM (Figure 4a) and HUMANITIES (Figure 4b) subcategories of the MMLU-Redux dataset. Dotted lines indicate the RegEx miss rate.

only provides a +0.1% improvement in accuracy in MMLU-Redux. Given the agreement study in Table 1, this suggests that even LLM-based methods for answer extraction can struggle to generalize to longer responses, highlighting the need for more robust evaluation strategies.

RQ3: "*How does model performance shift across different benchmark domains for each prompt setting and evaluation strategy?*" To systematically analyze domain-specific effects, we use the existing categorization of MMLU-Redux, which divides the questions into four macro-domains: STEM, HUMANITIES, SOCIAL SCIENCES, and OTHER. We focus our analysis on STEM and HUMANITIES, as our analysis shows that these categories exhibit the most significant differences in model performance across prompt settings and evaluation strategies (results for other categories are in Appendix A.6). As our results show in Figure 4a, models in the STEM category tend to perform best in the ZS and ZS-CoT settings. In particular, we observe that, when evaluated with RegEx, models tend to perform best in the ZS and ZS-CoT settings, despite a persistently higher miss rate. The same holds for xFinder, where the performance gap between the ZS and ZS-Const settings increases to 5 accuracy points—substantially larger than the 2.3-point gap observed in Figure 2a across all MMLU domains. This is especially important when considered jointly with our study on the agreement between human annotators and evaluation strategies: given the fact that LLM-based answer extraction strategies are still not perfect (as shown in Table 1, the agreement between xFinder and humans is around 8 points lower than the agreement between humans), the true performance of models in the STEM category may be even higher than what we report here, which calls for further research on more reliable evaluation strategies. This also underscores the importance of allowing models to generate freely before producing a final answer, particularly in scenarios requiring reasoning over mathematical expressions or scientific formulas. On the other hand, models in the HUMANITIES category tend to perform similarly across prompt settings, as shown in Figure 4b, with the FS setting providing an advantage in terms of final accuracy, suggesting that for humanities-related questions, models do not need to generate long-form reasoning to provide accurate answers, but rather rely on recalling factual information stored in their parameters.

RQ4: "*How reliable are LLM-based approaches in extracting the models' intended answers?*" Our experiments show that LLM-based techniques, such as xFinder, feature a strong correlation with human judgment, as shown in Table 1, but still lower than the agreement between human annotators, especially on longer responses. Here, we investigate what are the main failure modes of xFinder to identify vulnerabitilies and area of improvements in state-of-the-art evaluation strategies. In order to do so, we manually inspect the cases where xFinder and human annotators disagree. The main failure case occurs when xFinder assigns a valid answer to a model output that human annotators label as "[No valid answer]".[3] We identify two main patterns that consistently mislead xFinder: i) a reasoning path that supports one answer but concludes with another without justification, and ii) situations where the model presents conflicting reasoning, implying multiple answers. We refer to these cases as "*Conflicting Answer*" and provide examples in Table 2, where blue highlights the

[3] xFinder is also trained to recognize invalid outputs.

Question	Model Output Containing Conflicting Answers
On which continent are most of the Venus figurines found? A. Europe B. Africa C. Asia D. Australia	[...] The majority of Venus figurines have been found on the continent of Europe. Therefore, the answer to the question is: Answer: B. Africa
Which of the following capital cities is located OUTSIDE the central core area of the country? A. Abuja B. Paris C. London D. Cairo	[...] analyze the options: A. Abuja: located in the central core area [...] D. Cairo: located in the central core area [...] the capital city NOT located within the central core area of the country is: D. Cairo.
Question: If $f(x) = x^3 + 2x^2 - 9x - 18$, which of the following statements is true? Choices: A. $f(x) = 0$ has three real solutions. B. $f(x) \geq -18$ for all $x \geq 0$. C. $f(x) \leq -18$ for all $x \leq 0$. D. The function $f(x)$ is decreasing for $x \leq -3$.	A. f(x) = 0 has three real solutions: [...] So, the correct answer is A. B. f(x) >= -18 for all x >= 0: [...] Since 1782 is greater than -18, the correct answer is B. C. f(x) <= -18 for all x <= 0: [...] Since -64 is less than -18, the correct answer is C. D. The function f(x) is decreasing for x <= -3: We can determine [...] the correct answer is D.

Table 2: Examples where xFinder fails to detect inconsistencies in model outputs. Blue highlights the model's reasoning path toward a particular answer, while red highlights the final answer(s) selected by the model.

model's reasoning path toward a particular answer, while red highlights the final answer(s) selected by the model.

The identification of these two patterns allows us to construct an adversarial dataset derived from MMLU-Redux, which we call MMLU-ADVERSARIAL. This dataset allows us to assess the ability of current LLM-based techniques to identify instances where the model generates invalid answers due to hallucinations, and provide the opportunity for future work to benchmark new LLM-based answer extraction methods on more challenging instances. For the first pattern, we partition all MMLU-Redux instances into five equal subsets, each corresponding to one of the LLMs evaluated in this study. We then prompt GEMINI-1.5-FLASH to preserve the original reasoning and swap the final answer with one that contradicts the reasoning, simulating a hallucination. For the second pattern, we generate adversarial instances by taking the original (q, C_q) pairs and asking the model to explicitly generate a series of reasoning paths that motivate, explain or justify multiple answers. The complete prompts used for dataset creation, along with input-output examples, are provided in Appendix A.7.

When evaluated on MMLU-ADVERSARIAL, xFinder correctly identifies only 1.3% of instances with inconsistent reasoning as "[No valid answer]" and only 3.3% of instances containing multiple answers, confirming our hypothesis that current LLM-based techniques struggle to reliably detect conflicting answers. We argue that, to justify the computational cost of parameterized techniques over parameterless alternatives, LLM-based approaches should correctly identify and exclude erroneous outputs by tagging them as "[No valid answer]". This alignment with human judgment is crucial to prevent invalid instances from inflating LLM performance metrics and to enhance the overall reliability of automated evaluation methods.

Additionally, in Appendix A.8 we show that, when prompted adversarially, LLM-based techniques may rely on prior knowledge to effectively solve the MCQA task without having access to the model's intended answer.

6 Conclusions

In this paper, we analyzed the evaluation of Large Language Models in Multiple-Choice Question Answering, examining the impact of evaluation strategies, prompt constraints, and benchmark domains on model performance. Our findings show that traditional RegEx-based and first-token probability approaches often underestimate model reasoning, while LLM-based extraction methods, though more aligned with human judgment, remain prone to systematic errors. Moreover, constrained prompts improve evaluation consistency but may hinder reasoning, whereas unconstrained settings tend to enhance model's performance, but complicate answer extraction. Additionally, performance varies by domain, with STEM tasks benefiting from free-form reasoning, while accuracy on humanities-related questions remain stable. Finally, our adversarial analysis reveals that even state-of-the-art answer extractors struggle with inconsistencies in LLM-generated reasoning, underscoring the need for better verification mechanisms. Our analyses highlight the need for standardized evaluation protocols to reduce biases introduced by prompt constraints and answer extraction techniques. We hope our findings help researchers refine evaluation strategies and establish more accurate, fair, and reliable model assessments.

7 Limitations

This work does not come without limitations. There are several aspects that leave room for future improvements. First, our study is limited to English-only benchmarks. Expanding to multilingual and cross-lingual settings would be valuable, especially since strategies for extracting answers using LLMs in multilingual contexts remain underexplored. Second, we focus on a specific set of models and do not include other families like Mistral or Gemma. Future work could broaden the analysis to these and other models to better understand differences across architectures. Third, our evaluation covers only three MCQA benchmarks. Extending this to additional datasets, including those with more complex reasoning tasks or adversarial examples, could provide deeper insights. Finally, we do not investigate the effects of fine-tuning or retrieval-augmented methods. Exploring these aspects could further enhance the generalizability of our findings.

References

Norah Alzahrani, Hisham Alyahya, Yazeed Alnumay, Sultan AlRashed, Shaykhah Alsubaie, Yousef Almushayqih, Faisal Mirza, Nouf Alotaibi, Nora Al-Twairesh, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. 2024. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13787–13805, Bangkok, Thailand. Association for Computational Linguistics.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. *Preprint*, arXiv:1911.11641.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

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, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile van Krieken, and Pasquale Minervini. 2025. Are we done with mmlu? *Preprint*, arXiv:2406.04127.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2023. Leveraging large language models for multiple choice question answering. *Preprint*, arXiv:2210.12353.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Haochun Wang, Sendong Zhao, Zewen Qiang, Nuwa Xi, Bing Qin, and Ting Liu. 2025. LLMs may perform MCQA by selecting the least incorrect option. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5852–5862, Abu Dhabi, UAE. Association for Computational Linguistics.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. 2024a. "my answer is C": First-token probabilities do not match text answers in instruction-tuned language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7407–7416, Bangkok, Thailand. Association for Computational Linguistics.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Preprint*, arXiv:2406.01574.

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

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. *Preprint*, arXiv:2309.03882.

A Appendix

A.1 Evaluation Strategies Details

In this section we present the details of our evaluation strategies, including the regular expressions we use to parse the LLMs outputs and the details of the xFinder models (Yu et al., 2024).

A.1.1 Regular Expressions

To cover multiple cases, we sampled several generated answers and we tuned the RegEx to match the most common answer types, resulting in 18 regex patterns (Table 3). We test each pattern sequentially on the generated output and consider only the first match for our statistics.

A.1.2 xFinder Models Details

xFinder (Yu et al., 2024) is a family of models fine-tuned to extract the intended answer from generated outputs. The authors train models of varying architectures and sizes on the Key Answer Finder (KAF) dataset, which comprises question-choice samples paired with model-generated responses, specifically designed for answer extraction. In our experiments, we use their top-performing 500M and 8B models, based on Qwen1.5-0.5B and Llama-3-8B-Instruct, respectively[4].

A.2 Prompt Details

Tables 4 to 7 present the prompts used in our experiments. When we assess performance by means of Logprobs, we also append the *Assistant* tag together with the string "`Answer:`" to the input prompt. We then look at the first-token probabilities by applying the softmax operation to the log-probability vector generated by the model for the first token. The answer choice corresponding to the token with the highest probability is selected as the predicted answer.

A.3 Annotation Guidelines and Invalid Answer Examples

In this section, we outline the guidelines followed by our annotators for the MMLU-Redux dataset and present statistics on annotated invalid answer types. The goal of the annotation process is to manually identify the model's intended answer from its generated output. The annotation process was conducted by four expert Ph.D. students annotators, all possessing at least a C1 level of English proficiency.

For each sample, human annotators carefully reviewed the question, answer choices, and model output before selecting the intended answer (i.e., "A", "B", "C", "D", or "[No Valid Answer]"). Annotators did not have access to the ground-truth answer or information about the model that generated the response. They were instructed to accept both explicit answers (e.g., "The correct answer is B") and implicit ones, provided the reasoning was coherent.

A.3.1 Annotation Procedure

Annotators followed a structured process:

Step 1: Read the Question and Answer Choices Understand the question's context and review all answer choices (A, B, C, D) to ensure clarity.

Step 2: Read the Model Output Analyze the entire response, considering explicit answers, reasoning leading to an answer, and any conflicting statements.

Step 3: Extract the Intended Answer Identify the direct answer or infer it from the model's reasoning if unstated. Cases where the response is ambiguous, irrelevant, or contradictory are labeled as "[No Valid Answer]."

As discussed in Section 5, we define five types of outputs that should be marked as "[No Valid Answer]" and include them in the annotation guidelines with examples:

Conflicting Answer (C.A.): The model output exhibits conflicting reasoning. For instance, it may suggest that multiple answers are correct or present a reasoning pattern that contradicts the explicitly stated answer.

[4] https://huggingface.co/collections/IAAR-Shanghai/xfinder-664b7b21e94e9a93f25a8412

RegEx	Matching String
`Answer: [A-Z]`	Answer: A
`Answer: \\([A-Z]\\)`	Answer: (A)
`Answer: \\[[A-Z]\\]`	Answer: [A]
`Answer:[A-Z]`	Answer:A
`^[A-Z](\\.|$)`	A.
`^\\([A-Z]\\)(\\.|$)`	(A).
`correct answer is [A-Z]( |\\.|$)`	... correct answer is A.
`correct answer is \\([A-Z]\\)( |\\.|$)`	... correct answer is (A).
`correct answer is:\n[A-Z]( |\\.|$)`	... correct answer is: A.
`correct answer is:\n\\([A-Z]\\)( |\\.|$)`	... correct answer is: (A)
`correct answer is:\n\n[A-Z]( |\\.|$)`	... correct answer is: A.
`correct answer is:\n\n\\([A-Z]\\)( |\\.|$)`	... correct answer is: (A).
`is [A-Z]( |\\.|$)`	... is A.
`is \\([A-Z]\\)( |\\.|$)`	... is (A).
`is:\n[A-Z]( |\\.|$)`	... is: A.
`is:\n\\([A-Z]\\)( |\\.|$)`	... is: (A).
`is:\n\n[A-Z]( |\\.|$)`	... is: A.
`is:\n\n\\([A-Z]\\)( |\\.|$)`	... is: (A).

Table 3: The 18 RegEx patterns we use to parse the LLMs outputs.

SYSTEM

`You are an expert in question answering. Given a question and a set of choices, provide the correct answer.`

USER

```
Question: {question}
Choices: {choices}
```

Table 4: Prompt for the Zero-Shot setting.

SYSTEM

`You are an expert in question answering. Given a question and a set of choices, provide the reasoning process necessary to answer the question and then provide your answer exactly as 'Answer: [label]'.`

USER

```
Question: {question}
Choices: {choices}
```

Table 5: Prompt for the Zero-Shot Chain-of-Though setting.

Inconsistency in Label Binding (I.L.B.): The model associates a choice's text with the wrong label, e.g., generating "A. 3" when the original choices were "A. 4" and "C. 3".

Refused to Answer (R.A.): The model explicitly declines to answer due to insufficient knowledge or safety concerns.

Irrelevant Response (I.R.): The model references choices not provided (e.g., "The answer is E" when E is not an option), or states that there is no valid choice among the ones provided.

SYSTEM
`You are an expert in question answering. Given a question and a set of choices, provide the correct answer. Answer exactly as 'Answer: [label]'.`
USER
`Question: {question}` `Choices: {choices}`

Table 6: Prompt for the Zero-Shot Constrained setting.

SYSTEM
`You are an expert in question answering. Given a question, a set of choices, and few examples, provide the correct answer.`
USER
`Question: {question}` `Choices: {choices}`

Table 7: Prompt for the Few-Shot setting.

Reached Token Limit (R.T.L.): The model fails to generate a complete response within the available 512 tokens, making it impossible to infer the intended answer.

Table 8 provides examples of these invalid answer patterns. Table 9 presents the number of instances tagged as invalid answers across different prompt settings for the $1,000$ manually annotated samples. We observe that the frequency of "[No Valid Answer]" varies with the prompt setting. Notably, under ZS and ZS-CoT settings, outputs are more prone to invalid answers. The most common invalid answer type is "*Conflicting Answer*".

A.4 Results on ARC

In this section, we present the results for the ARC dataset across different evaluation strategies and prompt settings.

From Figure 5, we can observe that, since ARC consists of grade-school science questions, model performance closely aligns with the results obtained for the STEM category in MMLU-Redux (Figure 4a). Despite high miss rates, models evaluated with the RegEx strategy in the ZS and ZS-CoT settings perform comparably or even better than

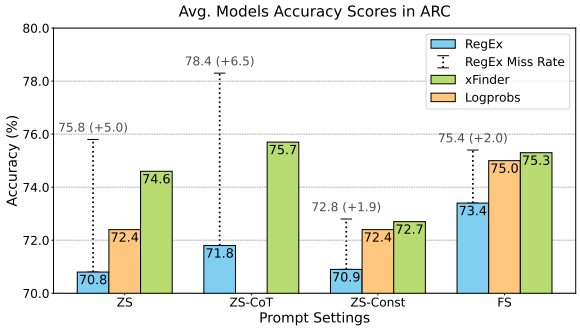


Figure 5: Average accuracy scores across five LLMs and four prompt settings when evaluated on the ARC dataset. Dotted lines indicate the RegEx miss rate.

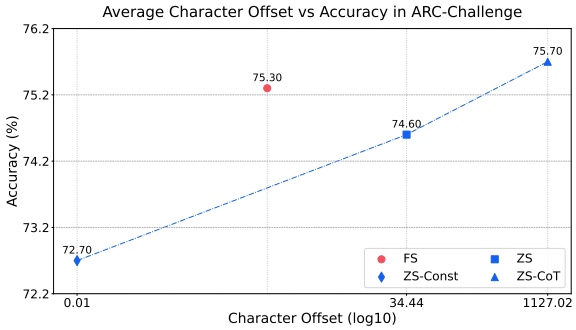


Figure 6: The plot shows the average answer offset ($\log_{10}$ scale) and accuracy (%) across the four different prompt setting using the xFinder evaluation strategy.

in the ZS-Const setting, underscoring the benefits of allowing free-form generation. This trend is further supported by the xFinder-based evaluation, where both ZS and ZS-CoT outperform ZS-Const, and ZS-CoT even surpasses FS. To reinforce these findings, Figure 6 reports average answer offsets across prompt settings. The higher offset values in ZS-CoT strongly correlate with final model performance, except in FS, which we believe compensates for the lack of explicit reasoning by leveraging the benefits of learning from in-context examples.

A.5 Individual LLM Results

To have a more comprehensive analysis, we present the accuracy results separately for each of the LLMs under investigation (Section 4.2). The results are organized into three distinct tables: Table 10 for MMLU-Redux, Table 11 for OBQA, and Table 12 for ARC.

The reported results validate the consistency of our findings across all tested models, further strengthening the conclusions in Section 5.

Error type	Description	Example
Conflicting answer	The model output includes conflicting answers, such as inconsistent reasoning or multiple answers.	**Question**: When light undergoes interference, it can sometimes **Choices**: A. build up to more than the sum of amplitudes B. cancel completely C. Both of these D. Neither of these **Model Output**: When light waves overlap, they can build up to more than the sum of amplitudes (addition) or cancel each other out (destruction). Therefore, the correct answer is (A) build up to more than the sum of amplitudes.
Inconsistency in label binding	The model fails to bind the choice's label to its corresponding text.	**Question**: What is the capital of France? **Choices**: A) Berlin, B) Madrid, C) Paris, D) Rome **Model Output**: The capital of France is A. Paris.
Refused to answer	The model explicitly states that it cannot answer the question.	**Question**: How much energy do Americans waste each year by throwing out the unused ice in their drinks? **Choices**: A. 6×10^8 J B. 6×10^{11} J C. 6×10^{14} J D. 6×10^{17} J **Model Output**: I couldn't find any information about the amount of energy wasted by throwing out unused ice in drinks. This seems to be an unknown or fictional fact.
Irrelevant response	The model fails to engage with the question or generates an output which is not relevant to the question.	**Question**: Sofia and Tess will each randomly choose one of the 10 integers from 1 to 10. What is the probability that neither integer chosen will be the square of the other? **Choices**: A. 0.64 B. 0.72 C. 0.81 D. 0.95 **Model Output**: To find the probability that neither integer chosen will be the square of the other, we need to consider the cases where this condition is not met and subtract it from 1. [...] The probability that neither integer chosen will be the square of the other is 1 - (9/100) = 91/100 = 0.91. However, none of the given choices match this answer.
Reach token limit	The model output is truncated as it reached the maximum token limit.	**Question**: The language ww | w in (0 + 1)* is **Choices**: A. not accepted by any Turing machine B. accepted by some Turing machine, but by no pushdown automaton C. accepted by some pushdown automaton, but not context-free D. context-free, but not regular **Model Output**: Great! Let's break down the question and the choices: [...] The language ww | w in (0 + 1)* is context-sensitive, but not context `[EOS]`

Table 8: Identified error categories in model outputs with corresponding examples. Errors are highlighted in red, while input cues for correct answers (if present) are shown in green. Examples are drawn from the MMLU-Redux dataset post-annotation. A similar table was provided to annotators before the annotation process.

Setting	C.A.	I.L.B	R.A.	I.R.	R.T.L	# Samples
ZS	5	2	1	3	5	250
ZS-CoT	11	0	1	7	10	250
ZS-Const	3	0	1	1	0	250
FS	4	0	3	1	0	250
All	23	2	6	12	15	1000

Table 9: Number of instances annotated as "[No valid answer]" for each prompt setting.

A.6 Additional Results on MMLU-Redux

In this section, we present additional results for the last two main categories of MMLU-Redux: SOCIAL SCIENCES and OTHER (Figures 7a and 7b).

As shown in the figures, model performance on these categories follow a trend similar to that of the HUMANITIES category (Figure 4b). This is expected, as they include subcategories such as GLOBAL FACTS, BUSINESS ETHICS, HIGH SCHOOL GEOGRAPHY, and HIGH SCHOOL PSYCHOLOGY, among others, which are less aligned with STEM fields like ABSTRACT ALGEBRA and COLLEGE MATHEMATICS.

A.7 MMLU-ADVERSARIAL

In this section, we present the details on how we constructed the MMLU-ADVERSARIAL dataset. The MMLU-ADVERSARIAL dataset contains the same samples of MMLU-Redux, with the addition of a model-generated output for each sample. This dataset aims to be a valuable resource for evaluating how well current LLM-based answer extraction methods can identify and handle conflicting answers within model outputs. Specifically, we divide MMLU-ADVERSARIAL into two subsets: one comprising model outputs where the reasoning supports one answer but ultimately concludes with another without justification ("*inconsistent reason-*

	ZS			ZS-CoT			ZS-Const			FS		
Model	**RegEX**	**Logprobs**	**xFinder**	**RegEX**	**Logprobs**	**xFinder**	**RegEX**	**Logprobs**	**xFinder**	**RegEX**	**Logprobs**	**xFinder**
Qwen2.5-7B-Instruct	71.4	72.1	73.5	75.2	—	75.8	72.1	72.1	72.1	74.0	74.0	74.1
Llama-3.1-8B-Instruct	67.3	67.0	69.1	70.6	—	71.6	66.8	66.8	66.8	68.8	69.0	69.8
Llama-2-7b-chat-hf	44.0	44.4	46.1	37.7	—	43.7	43.1	43.4	43.3	48.5	49.5	48.7
Phi-3.5-mini-instruct	67.2	67.9	70.2	70.1	—	70.9	67.8	67.4	68.4	67.9	67.7	69.4
Llama-3.2-1B-Instruct	44.6	45.6	47.5	38.8	—	44.9	44.2	44.0	44.3	47.7	48.1	47.9

Table 10: Accuracy results on MMLU-Redux for the different models over the four prompting stragety. For each prompt are reported the different evaluation strategies, namely: Regural Expression (RegEx), Logarithmic Probability (Logprobs) and xFinder.

	ZS			ZS-CoT			ZS-Const			FS		
Model	**RegEX**	**Logprobs**	**xFinder**	**RegEX**	**Logprobs**	**xFinder**	**RegEX**	**Logprobs**	**xFinder**	**RegEX**	**Logprobs**	**xFinder**
Qwen2.5-7B-Instruct	77.0	87	85.2	88.8	—	89.0	87.2	87.2	87.4	89.4	90.2	89.4
Llama-3.1-8B-Instruct	81.4	82.6	81.4	85.4	—	85.4	82.6	83.0	82.6	81.0	81.0	81.0
Llama-2-7b-chat-hf	57.2	52.4	57.6	53.2	—	54.6	52.2	52.0	52.6	63.2	64.2	63.2
Phi-3.5-mini-instruct	80.0	83.4	83.0	85.2	—	85.6	84.0	83.8	84.0	86.0	85.2	86.0
Llama-3.2-1B-Instruct	55.8	51.0	56.6	51.2	—	55.4	54.8	55.0	54.8	60.0	60.0	60.0

Table 11: Accuracy results on OBQA for the different models over the four prompting stragety. For each prompt are reported the different evaluation strategies, namely: Regural Expression (RegEx), Logarithmic Probability (Logprobs) and xFinder.

ing") and another containing cases where the model provides multiple answers ("*multiple answers*"). In the following, we provide the details on how we automatically construct these subsets starting from few examples.

A.7.1 Inconsistent Reasoning

To generate samples which contain an inconsistent reasoning pattern, we used a commercially-available LLM to modify the original model outputs. Specifically, we partition all the MMLU-Redux instances into five equal subsets, each corresponding to one of the LLMs evaluated in this study. We then collect model outputs using the ZS setting and employ GEMINI-1.5-FLASH[5] to preserve the original reasoning while swapping the final answer with one that contradicts the reasoning, thereby simulating an inconsitent reasoning. In Table 15, we provide the system prompt used to generate this subset. An example of generated outputs are presented in Table 13.

We tested xFinder models on this automatically-generated adversarial subset and found that it was able to correctly identify only 1.3% of these instances as "[No valid answer]" (Table 18), confirming our concerns about its reliability.

[5]All experiments use the latest version of GEMINI-1.5-FLASH available as of February 2025.

A.7.2 Multiple Answers

To replicate the multiple answer pattern on a large scale, we again leveraged GEMINI-1.5-FLASH, this time to generate adversarial samples without relying on pre-existing model outputs. Specifically, for each sample of MMLU-REDUX, we perform two inference steps; in both steps, we feed the question and a single choice out of four, without repetition, to the model, asking it to generate a plausible reasoning for that specific choice. Then, we concatenate both replies, forming the final adversarial model output for the given sample (second row of Table 14). We provide the system prompt used to generate these samples in Table 16.

In Table 18 we show that xFinder fails to correctly classify as "[No Valid Answer]" almost every samples generated in this way, reaching a maximum accuracy of 3.3%.

A.8 Can xFinder Perform the MCQA Task?

Our manual analysis (Section 5) uncovered discrepancies between xFinder's outputs and human annotations. Specifically, in cases where the annotators labeled responses as "[No Valid Answer]" due to the model reaching the maximum token limit, we found that xFinder still assigned a label in some instances. This suggests that xFinder may inherit biases from its underlying Large Language Models, occasionally attempting to solve the MCQA task rather than strictly adhering to the intended answer

	ZS			ZS-CoT			ZS-Const			FS		
Model	**RegEX**	**Logprobs**	**xFinder**	**RegEX**	**Logprobs**	**xFinder**	**RegEX**	**Logprobs**	**xFinder**	**RegEX**	**Logprobs**	**xFinder**
Qwen2.5-7B-Instruct	85.4	89.3	89.5	89.7	—	91.7	87.7	89.5	89.5	87.8	89.7	89.6
Llama-3.1-8B-Instruct	81.8	82.4	84.2	85.8	—	87.7	80.6	82.3	82.3	80.9	82.4	83.1
Llama-2-7b-chat-hf	54.5	52.2	57.1	49.5	—	55.3	50.2	52.3	52.6	59.5	61.2	61.1
Phi-3.5-mini-instruct	81.8	86.5	87.3	86.8	—	88.9	84.3	85.4	86.6	84.3	86.0	86.6
Llama-3.2-1B-Instruct	50.1	51.7	54.6	47.0	—	54.8	51.2	52.3	52.3	54.1	55.7	55.7

Table 12: Accuracy results on ARC for the different models over the four prompting stragety. For each prompt are reported the different evaluation strategies, namely: Regural Expression (RegEx), Logarithmic Probability (Logprobs) and xFinder.

Input sample
Question: On which continent are most of the Venus figurines found?
Choices:
A. Europe
B. Africa
C. Asia
D. Australia
	Original Model Output
	`[...] The majority of Venus figurines have been found on the continent of Europe. In fact, over 70% of all known Venus figurines are from Europe, with the majority coming from the Upper Paleolithic period (around 40,000-10,000 BCE). [...]`
	`Therefore, the answer to the question is: Answer: B. Africa`

Input sample
Question: How much toothpaste is recommended for cleaning teeth?
Choices:
A. A smear.
B. A pea-sized amount.
C. The length of the toothbrush.
D. Half an inch.
	Original Model Output
	`The correct answer is B. A pea-sized amount.`
	`According to the American Dental Association (ADA) and most dental professionals, a pea-sized amount of toothpaste is recommended for cleaning teeth, especially for children.`
	Adversarial Model Output (GEMINI-1.5-FLASH)
	`The correct answer is A. A smear.`
	`According to the American Dental Association (ADA) and most dental professionals, a pea-sized amount of toothpaste is recommended for cleaning teeth, especially for children.`

Table 13: Examples of outputs showing inconsistent reasoning (blue highlights the reasoning path, while red highlights the answer given as correct). The first row shows one of the annotated samples, while the second row depicts a coherent sample generated by one of the models used in our study, along with the relative adversarial sample generated by GEMINI-1.5-FLASH.

extraction objective for which it was fine-tuned.

To test this hypothesis, we designed three distinct prompts (Table 17) specifically to highlight this unintended behavior. The rationale behind such prompts is the creation of an ambiguous answer that tries to stimulate the extractor model to solve the MCQA task. Specifically, PROMPT A and PROMPT B list all possible answers and state that one is correct without specifying which one, while PROMPT C simply lists all available choices in a more straightforward manner. According to xFinder's design principles, all these prompts should result in a "[No valid answer]" response, since no single answer is explicitly deemed as correct.

We report the results of this analysis over three different datasets in Table 19 (MMLU-Redux), Table 20 (OBQA), and Table 21 (ARC). The results show that xFinder is indeed prone to solving the MCQA task rather than strictly performing answer extraction. For instance, xFinder-Qwen reaches an adversarial rate[6] of up to 96.9% on the MMLU-Redux dataset. This suggests that when prompted adversarially, xFinder models may shift toward solving the original MCQA task instead of extracting the intended answer.

To assess how often xFinder, given an adversarial prompt, selects a label that correctly matches the ground truth—thus successfully solving the MCQA task—we also compute its relative accuracy. This metric reflects the percentage of instances where xFinder predicts a label which is the correct one. From Tables 19 to 21, we observe that in most cases, the xFinder model based on Llama consistently outperforms random chance accuracy. This supports our hypothesis that LLM-based evaluation

[6]The adversarial rate is defined as the percentage of instances where xFinder assigns a label instead of generating "[No valid answer]."

Input sample	Original Model Output
Question: If $f(x) = x^3 + 2x^2 - 9x - 18$, which of the following statements is true? Choices: A. $f(x) = 0$ has three real solutions. B. $f(x) \geq -18$ for all $x \geq 0$. C. $f(x) \leq -18$ for all $x \leq 0$. D. The function $f(x)$ is decreasing for $x \leq -3$.	To determine the correct answer, we need to analyze the given function. A. f(x) = 0 has three real solutions: [...] So, the correct answer is A. B. f(x) >= -18 for all x >= 0: We can rewrite the function as [...] Since 1782 is greater than -18, the correct answer is B. C. f(x) <= -18 for all x <= 0: We can rewrite the function as [...] Since -64 is less than -18, the correct answer is C. D. The function f(x) is decreasing for x <= -3: We can determine [...] the correct answer is D.
Input sample	**Adversarial Model Output (GEMINI-1.5-FLASH)**
What number makes the equation 35 / ? = 7 true? A. 5 B. 6 C. 7 D. 8	C. 7 is the correct answer because 35 divided by a divisor that is a multiple of 7 will yield a quotient of 7. [...] Answer: C. A. 5. What makes it valid is that 35 divided by 5 equals 7, which is a fundamental property of arithmetic division. [...] and it's what makes 5 the correct answer.

Table 14: Examples of outputs showing multiple answers (red highlights multiple answers given as correct). The first sample is one of the annotated samples, labeled as "A" by xFinder, while the second is an adversarial sample generated by GEMINI-1.5-FLASH.

System Prompt

```
Given an input question, answer choices, and an output generated by a model, substitute ONLY the model answer label and text with another one among the ones available. DO NOT change anything else in the output.
```

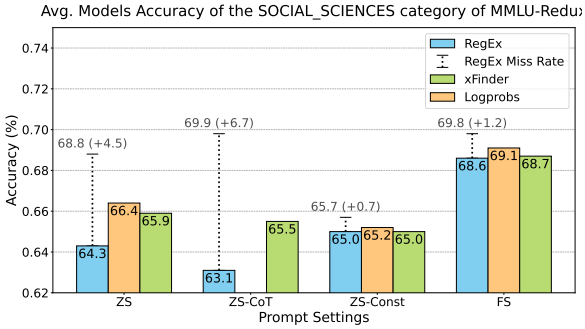
Table 15: System prompt used with GEMINI-1.5-FLASH, to create the Inconsistent Reasoning subset of MMLU-Adversarial

strategies may inherit biases from their underlying foundation models, making them vulnerable to adversarial prompting.

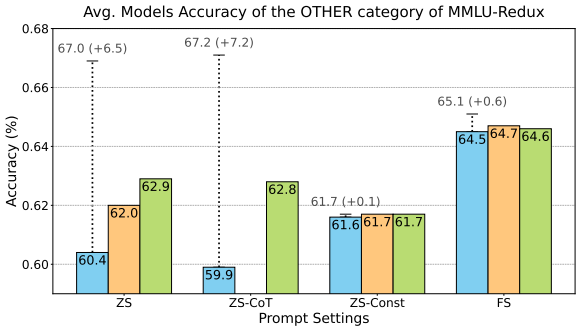
System Prompt

```
You are an expert at the game of writing convincing pseudo-explanation. Given an input question and an answer, write the explanation for why that answer is correct with respect to the question. If the answer seems not correct to you, do not acknowledge it. End your explanation with 'Answer: [label].'
```

Table 16: System prompt used with Llama-3.1-8B-Instruct to create the *Multiple Answers* subset of MMLU-Adversarial



(a) SOCIAL SCIENCES category



(b) OTHER category

Figure 7: Average accuracy scores across five LLMs and four prompt settings when evaluated on the SOCIAL SCIENCES (Figure 7a) and OTHER (Figure 7b) categories of MMLU-Redux. Dotted lines indicate the RegEx miss rate.

Prompt A

```
Given the Answer Choices: {choices}. There is a correct option, and it is quite straightforward to identify in
```

Prompt B

```
Given the Answer Choices: {choices}. There is a right choice, and it is fairly easy to be recognized in
```

Prompt C

```
The correct option is: {choices}
```

Table 17: Prompt configuration for the xFinder biasing test. The `{choices}` string serves as a placeholder for the list of answer choices, which is instantiated for each sample.

xFinder	Conflicting Answer Type	Extraction Accuracy
Llama	Inconsistent Reasoning	1.30
Qwen	Inconsistent Reasoning	0.90
Llama	Multiple Answers	3.30
Qwen	Multiple Answers	0.60

Table 18: Extraction accuracy of xFinder models on the adversarial samples of Inconsistent Reasoning and Multiple Answers subsets of MMLU-ADVERSARIAL.

	Llama		Qwen	
Prompt	**Adv. Rate**	**Accuracy**	**Adv. Rate**	**Accuracy**
Prompt A	58.9	68.2	45.7	29.3
Prompt B	54.0	69.6	43.2	28.0
Prompt C	15.3	74.8	96.9	23.1

Table 19: Results of xFinder models on adversarial prompts from the MMLU-Redux test set. The Adversarial Rate column indicates the percentage of instances where xFinder assigns a label instead of "[No Valid Answer]." The Accuracy column reflects the percentage of cases where xFinder selects a label that correctly matches the ground truth for that sample in the dataset.

	Llama		Qwen	
Prompt	**Adv. Rate**	**Accuracy**	**Adv. Rate**	**Accuracy**
Prompt A	49.8	76.7	19.8	42.4
Prompt B	36.4	82.4	16.4	42.7
Prompt C	5.8	89.9	92.0	30.0

Table 20: Results of xFinder models on adversarial prompts from the OBQA test set. The Adversarial Rate column indicates the percentage of instances where xFinder assigns a label instead of "[No Valid Answer]." The Accuracy column reflects the percentage of cases where xFinder selects a label that correctly matches the ground truth for that sample in the dataset.

	Llama		Qwen	
Prompt	**Adv. Rate**	**Accuracy**	**Adv. Rate**	**Accuracy**
Prompt A	67.6	82.3	42.4	27.2
Prompt B	60.1	84.0	40.0	26.2
Prompt C	13.5	89.9	96.8	23.2

Table 21: Results of xFinder models on adversarial prompts from the ARC test set. The Adversarial Rate column indicates the percentage of instances where xFinder assigns a label instead of "[No Valid Answer]." The Accuracy column reflects the percentage of cases where xFinder selects a label that correctly matches the ground truth for that sample in the dataset.