Reasoning Models are Test Exploiters: Rethinking Multiple Choice

Anonymous submission

1 Introduction

Multiple-choice question answering (MCQA) remains a dominant paradigm for evaluating LLMs because it enables straightforward automatic grading and has historically correlated with downstream utility. Yet modern "reasoning" models can leverage information in the answer options themselves, risking overestimation of true problem-solving ability when reasoning is permitted after exposure to options. We study when MCQA is a reliable proxy and when it inflates reported skill via test exploitation.

We conduct a systematic evaluation across 15 benchmarks spanning both MCQA and free-text QA (FTQA) and 27 LLMs ranging from small open-source models to state-of-the-art systems. For each model–benchmark pair, we vary whether and when the model sees the multiple-choice options and whether chain-of-thought (CoT) is allowed before or after options. We also study a "None-of-the-above" (NOTA) condition that sometimes replaces the correct option with "No other option is correct."

Our primary finding is simple: MCQA is a good proxy only when CoT precedes options (Q-CoT). Allowing reasoning after options (e.g., QMC-CoT or MC-CoT) yields substantial gains that do not transfer to FTQA, indicating option-based exploitation rather than genuine capability. Conversely, when we decouple reasoning from selection—e.g., reason on the question first, then select (Q-CoT $\rightarrow$ MC)—scores better reflect underlying competence.

We define exploitation as the accuracy gain when options are visible versus hidden, and we find three consistent patterns. (i) reasoning models are stronger test exploiters (size alone is not predictive). (ii) harder distractors do not eliminate exploitation: on MMLU-Pro, LLMs still score well above chance even when asked to answer given only the options and for some models, their performance is even higher than on MMLU. (iii) Introducing NOTA lowers exploitation, and LLMs are frequently correct when selecting NOTA, indicating the drop is not driven by indiscriminate NOTA over-selection.

Contributions. (1) An empirical demonstration that whether and when the LLM sees options governs MCQA reliability; (2) a decoupled protocol that separates reasoning from selection without changing question content; and (3) practical guidance for reporting MCQA alongside FTQA.

2 Benchmarks

We evaluated LLMs on 15 benchmarks spanning diverse domains and question formats. Except where indicated otherwise, each benchmark consists entirely of four-option multiple-choice questions.

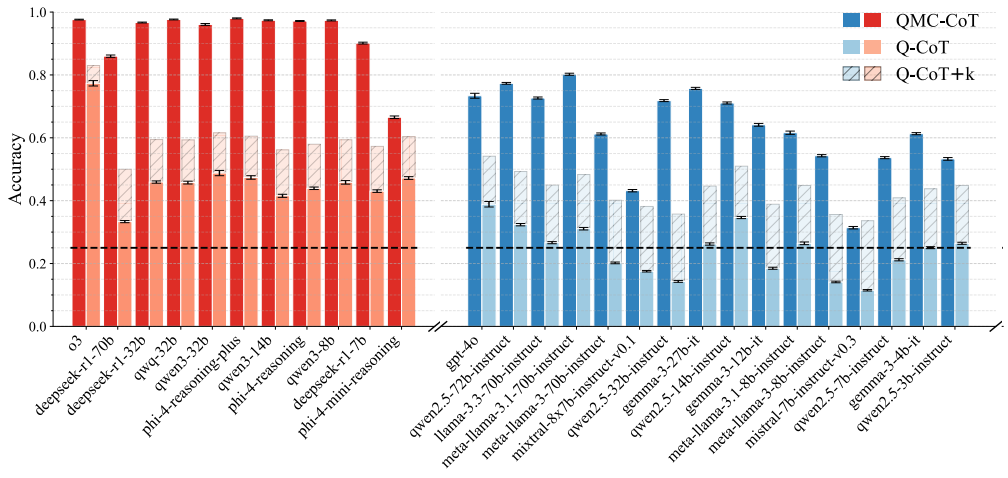
MCQA Benchmarks We evaluate on MMLU and MMLU-Pro (Hendrycks et al. 2020; Wang et al. 2024). We also use the Open-LLM suite spanning standard MCQA datasets (e.g., ARC, WinoGrande, PIQA, CommonsenseQA, RACE, MedMCQA, OpenBookQA) (Myrzakhan, Bsharat, and Shen 2024), and GPQA Diamond, the hardest split of GPQA with graduate-level science questions (Rein et al. 2023).

FTQA Benchmarks For FTQA we use GSM8K (grade-school math), MATH (competition math), PythonIO (program-output prediction from HumanEval and MBPP), and STEER-ME (economics questions) (Cobbe et al. 2021; Hendrycks et al. 2021; Chen et al. 2021; Austin et al. 2021; Zhang et al. 2024; Raman et al. 2025).

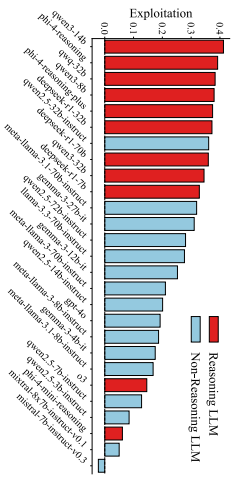
2.1 Question Format Conversion

We aim to examine how format alone affects performance; this section describes how we converted the MCQA benchmarks listed above to FTQA and vice versa.

MCQA $\rightarrow$ FTQA: We start with the datasets within Open-LLM. The dataset suite was constructed by filtering out questions from multiple datasets, which were not suitable for open-style answering. The filtering process they used kept many MCQA questions that would not be viable FTQA questions. So we employed two subsequent filtering procedures: (1) Removed all questions that contained text that explicitly or implicitly mentioned the options in the stem (e.g., 'Which of the following', 'What can be concluded from the passage') via substring search, and (2) Removed all stems that did not end with a period or question mark (e.g., 'While training the rats, the trainers have to be'). After this filtering process, $62.81\,\%$ of the total dataset remained of both MCQA/FTQA questions. For more details and a breakdown for each dataset, see Figure 4 in the appendix. Note that this likely omitted convertible MCQA questions. We did the same two-step filtering for MMLU-Pro, reducing the original test set of $12{,}032$ questions to $7{,}130$ questions.



(a) Pass@1 accuracy of each LLM over QMC-CoT (dark) and Q-CoT (light). LLMs are sorted by parameter count. Beneath every Q-CoT bar, we plot the boost in accuracy Q-CoT would have gotten with random guessing denoted Q-CoT+k.



(b) Exploitation by each LLM.

Figure 1: Reasoning models in red, non-reasoning in blue

FTQA → MCQA: For most of the datasets (all but STEER-ME) that were originally instantiated in FTQA as listed in Section 2, we used the MCQA versions created by Zhang et al. (2024). These datasets were constructed by collecting answers and incorrect predictions on GSM8K, MATH, HumanEval and MBPP from 60 open-source models. Finally, STEER-ME includes programmatically generated multiple-choice options as part of the benchmark.

We do a final filtering pass, running our grading function over the correct answers to check whether they can be converted into a grade-able format. We call questions that pass this filtering step CoT-extractable.

3 Methodology

Our goal is to quantify how much of an LLM's MCQA performance reflects genuine problem solving versus option exploitation. We specify how questions and options are shown and then define one- and two-stage evaluation configurations that control when reasoning happens relative to options.

3.1 Evaluation Formats

Question formats. We use four inputs: MC (options only), MCNA (MC with NOTA), QMC (question + options), and QMCNA (QMC with NOTA). For NOTA, in a $1/k$ fraction of questions we replace the correct option with NOTA; otherwise we replace a random incorrect option.

Response formats. We consider two response modes: CoT (chain-of-thought before the final choice) and 1T (single-token answer with no intermediate reasoning). Reasoning models (e.g., o-series, R1) always produce CoT; non-reasoning models can be prompted to output 1T or CoT.

3.2 Evaluation Configurations

A configuration is an (input, response) pair. One-stage configurations expose all information at once; two-stage configurations first elicit reasoning on the question alone, then show options for selection. The key design lever is timing: whether reasoning occurs *without* options (Q-CoT) or *after seeing* options (QMC-CoT).

ID	Input	Response	R	NR
MC-CoT	MC	CoT	✓	✓
MCNA-CoT	MCNA	CoT	✓	✓
Q-CoT	Q	CoT	✓	✓
QMC-CoT	Q + MC	CoT	✓	✓
QMCNA-CoT	Q + MCNA	CoT	✓	✓
Q-CoT-MC-1T	Q → MC	1T	✗	✓
Q-CoT-MCNA-1T	Q → MCNA	1T	✗	✓
Q-CoT-MC-CoT	Q → MC	CoT	✓	✗
Q-CoT-MCNA-CoT	Q → MCNA	CoT	✓	✗

Table 1: Evaluation configurations by input, response, and model support (R for reasoning and NR for non-reasoning). Two-stage rows first elicit Q-CoT then reveal options.

One-stage. We use five one-stage configuration (the top five configurations in Table 1). These isolate (i) pure option-only signals, (ii) question-only reasoning, and (iii) reasoning after option exposure.

Two-stage. Two-stage configurations first run Q-CoT to obtain a reasoning trace, then reveal options for selection. These decouple reasoning (on the question) from selection (over options), letting us assess exploitation while keeping question content fixed.

Two-stage configurations still re-expose options to the same model that produced the initial trace. Thus, reasoning models can exploit at selection time, so two-stage scores are best viewed as diagnostics of option sensitivity rather than "pure" reasoning measures. Still, we include Q-CoT-MCNA-CoT because NOTA hides the correct option $1/k$ of the time.

4 Results

Figure 1a reports each LLM's pass@1 accuracy under the QMC-CoT format and the Q-CoT formats. A clear trend emerges: The largest models—and the most performant—exhibit the largest positive gaps between QMC-CoT and Q-CoT (see Figure 1b). All models above roughly 50 B parameters scored 30 to 40 percentage points higher when choices are given before CoT, with the difference being even larger for reasoning models. One might expect that a sufficient rationale for this gap is due to selecting the closest-answer to the one arrived in the CoT. However, this heuristic was not very common, especially among reasoning models. We observed this behavior $\sim$23 % of the time when a reasoning model was correct in QMC-CoT and wrong in Q-CoT (see Table 3 for a breakdown for each model). Furthermore, even when we boosted Q-CoT's performance with the benefit of random guessing, denoted Q-CoT+k, nearly every model outperformed on QMC-CoT.[1]

Figure 1b ranks models by their ability to exploit, this is the excess accuracy a model achieves given access to the options. We define it as follows, for each question with k options, let A_{MC} be the model's QMC-CoT accuracy, A_{FT} its Q-CoT accuracy, and $1/k$ the random-guess baseline: $E = \left(A_{MC} - \frac{1}{k}\right) - A_{FT} \cdot \left(\frac{k-1}{k}\right)$. Note that reasoning models are, in general, better test exploiters.Interestingly, parameter size is not correlated with exploitation among reasoning models. In fact, other than DeepSeek R1 (7B), the most exploitative reasoning models have fewer than 32 B parameters, and the top 3 are smaller than 14 B. In part, this is due to saturation of the QMC-CoT format; nearly all reasoning models attain greater than 90 % accuracy on QMC-CoT so the performance gains by the bigger reasoning models appear in the Q-CoT format. This is especially true for o3, where achieving 77.34 % on Q-CoT makes it hard to diagnose how exploitative it can be. And in part, this is due to DeepSeek R1 (70B) having lower accuracy on both QMC-CoT and Q-CoT than the top reasoning models, suggesting that Qwen models constitute a better base for RL fine-tuning than Llama, matching recent results by Shao et al. (2025).

4.1 Evidence of Exploitation

We take a closer look at what information signals models are using to exploit. We start by analyzing the performance of all models on MC-CoT to quantify how much exploitation is coming from reasoning over the options alone. We then quantify the residual exploitation that arises from leveraging extra information in the question by comparing LLM performance on QMC-CoT and Q-CoT-MC-1T.

MC-only Exploitation Figure 5 quantifies the ability of each LLM to exploit information in the options to beat random guessing, plotting the accuracy above random guessing for each model on MC-CoT. While most models perform better than random guessing, the reasoning model with the lowest MC-CoT performance is higher than the highest non-reasoning model's performance. Among reasoning models, we observed that the Qwen3 models are the best MC-only exploiters, with Qwen3 (32B) obtaining 13 points above random guessing. In Figure 6, we break down the performance above random guessing each model obtains for each dataset. In general, the most exploitable datasets were the ones that were initially instantiated as MCQA. In fact, ARC, HellaSwag, and PIQA were the datasets most susceptible to MC-only exploitation, with every model attaining a statistically significant accuracy above random guessing, and with all but one reasoning model obtaining higher than 80 % accuracy on PIQA.

QMC-based Exploitation We then analyzed the residual exploitation that occurs when LLMs are given the question text along with the options. Here, we ran LLMs on our two-stage configurations; if an LLM's performance on Q-CoT-MC-1T (Q-CoT-MC-CoT for reasoning models) is worse than on QMC-CoT—corrected by their MC-only exploitation—that would be evidence of QMC-based exploiting behavior. We correct for MC-only exploitation by subtracting a model's QMC-CoT performance by their MC-CoT performance, and their Q-CoT-MC-1T performance by random guessing. To account for any drop in performance due to mapping issues, we super-scored Q-CoT-MC-1T with Q-CoT: if a model was correct on a question on either format then they were deemed correct. Therefore, we define QMC-based exploitation as: $E_{\text{QMC}} = (A_{\text{QMC-CoT}} - A_{\text{MC-CoT}}) - (A_{\text{S}} - 1/k)$, where A_{S} is the super-scored accuracy.

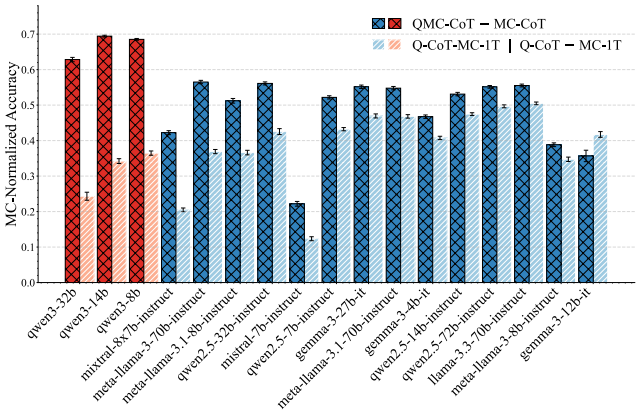


Figure 2: The MC normalized accuracy of non-reasoning models (Qwen3 models) on QMC-CoT in dark blue (dark red) and non-reasoning models (Qwen3 thinking mode off in the second step) super-scored on Q-CoT and Q-CoT-MC-1T in light blue (light red). LLMs are sorted by E_{QMC}.

Perhaps unsurprisingly, reasoning models performed better on super-scored Q-CoT-MC-CoT than QMC-CoT. However, Qwen3 models have the functionality to switch off their reasoning capabilities, allowing us to evaluate them on Q-CoT-MC-1T and compare them with non-reasoning models. Figure 2 plots the MC-normalized accuracy for non-reasoning and Qwen models sorted by E_{QMC}. We see widespread evidence of QMC-based exploitation. In fact, all but one LLM exhibited positive E_{QMC}. Furthermore, Qwen models exhibited a greater prevalence of QMC-based ex-

[1] For 4 options, Q-CoT+k = score(Q-CoT) $\times 0.75 + 0.25$.

ploitation, with larger E_{QMC} than any non-reasoning model.

4.2 Effect of Option Design on Exploitability

Given that LLMs can do better than random guessing just by looking at the options, we asked how specific option sets permit exploitation. We first revisited our MC-only and QMC-based probes to quantify the importance of the presence of the correct answer. Then we compared MMLU to MMLU-Pro to assess the efficacy of "harder" distractors on exploitability.

Effect of NOTA Under MCNA-CoT, the performance above random guessing decreased significantly (see Figure 7 and Figure 8 in the appendix). While ARC, HellaSwag, and PIQA remained highly exploitable datasets, performance on other datasets more closely matched random guessing. As a result, this reduced reasoning models' advantage, where on MC-CoT reasoning models scored $12.63\,\%$ higher than non-reasoning models but on MCNA-CoT, reasoning models only scored $5.29\,\%$ higher than non-reasoning models. In part, this is due to higher NOTA selection rates for reasoning models. On average, reasoning models selected NOTA $55.82\,\%$ of the time as compared to $30.05\,\%$ by non-reasoning models (the true rate is $25\,\%$). Inspecting the CoT's, it seems that reasoning models more often considered the MCNA-CoT setting to be a trick question, and NOTA a common answer to trick questions.

We then examined how NOTA affects QMC-based exploitation. We previously observed that Q-CoT-MC-CoT allows reasoning models to refine their answers by re-examining the options, we observed that Q-CoT-MCNA-CoT can disrupt this second-pass shortcut (see Figure 9). Most models exhibited at least some downward shift; suggesting that while these LLMs achieve high accuracy when they can reason over the full option set, their performance drops by 2 to 15 percentage points without the correct answer.

Given the behavior in MCNA-CoT, we test whether performance drops are because NOTA is an attractive distractor or because the correct answer is important for QMC-based exploitation. We treat NOTA selection as a binary classification task and report precision and recall for both classes (Table 5). For questions where NOTA replaces the true answer, DeepSeek R1 (70B) attains precision of 0.85 and recall of 0.58. For questions where NOTA is *not* the right answer, precision is 0.78 and recall is 0.94, indicating it rarely over-selects NOTA when a correct option exists. Taken together, these results suggest that the model is not unduly drawn to NOTA as a salient choice; rather, it applies NOTA selectively when its reasoning trace does not map to another valid option. This pattern follows for most reasoning models.

Effect of Harder Options We next examined whether making the option set "harder" (and larger) reduces MC-only exploitation. MMLU and MMLU-Pro offer a natural testbed for this question. For each dataset, we compute a normalized exploitation: $(k \times A_{\text{MC-CoT}} - 1)/(k-1)$, so that 0 means random guessing and 1 means perfect accuracy from the options alone. This puts MMLU ($k = 4$) and MMLU-Pro ($k = 10$) on a common scale independent of the number of options.

Two patterns stand out from Figure 3: (1) For nearly all non-reasoning models, while MMLU-Pro is strictly harder to exploit than MMLU, the option sets leak enough signal to beat random guessing—with values in the 5 to $10\,\%$ range. Curiously, the two Mistral models are the only models (including reasoning models) that are able to exploit MMLU-Pro *more* than MMLU, suggesting that increasing k and swapping in "harder" distractors does not uniformly suppress MC-only exploitation. (2) For reasoning models, while MMLU-Pro is often harder to exploit than MMLU, they are able to exploit MMLU-Pro more easily than non-reasoning models exploit MMLU. Together, these results suggest that as models get better at reasoning, they are better able to exploit the information in the option set and avoid "hard" distractors.

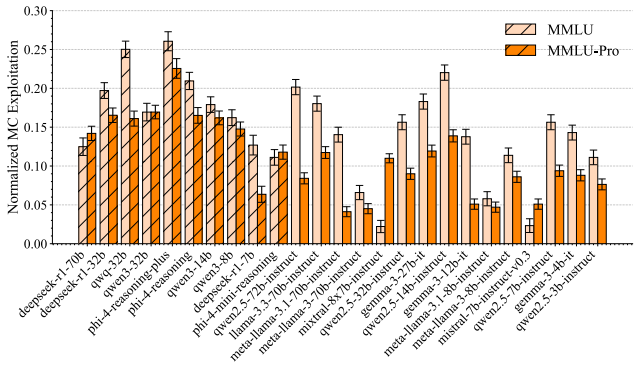


Figure 3: The normalized MC-only exploitation of all models on MMLU and MMLU-Pro. Reasoning models are hatched.

5 Conclusions

Although LLMs are achieving higher benchmark performances than ever, some of the improvement comes from exploiting the options. Our investigation reveals three lessons for the design and interpretation of LLM evaluations: (1) Decoupling is essential. By separating CoT from selection—via Q-CoT-MC-1T and, to some extent, Q-CoT-MCNA-CoT—we can expose latent reasoning ability and distinguish first-principles reasoning from test exploitation. Moreover, reasoning and selection should be reported separately. (2) Since MCQA is likely here to stay, design for option-independent correctness: write stems that do not reference the options and either define a canonical free-form answer or score via post-hoc mapping. (3) Relying solely on more challenging distractors as an antidote to exploitation is insufficient; while they may increase difficulty, they do not reliably mitigate test exploitation and must be used sparingly.

Ultimately, all we can observe is what we measure. Without careful design, high test performance may reflect proficiency in exploiting the test rather than true competence. As LLMs continue to improve and are used in the real-world, it becomes increasingly important to align what we measure with what we value.

References

Austin, J.; Odena, A.; Nye, M.; Bosma, M.; Michalewski, H.; Dohan, D.; Jiang, E.; Cai, C.; Terry, M.; Le, Q.; and

Sutton, C. 2021. Program Synthesis with Large Language Models. arXiv:2108.07732.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; de Oliveira Pinto, H. P.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; Ray, A.; Puri, R.; Krueger, G.; Petrov, M.; Khlaaf, H.; Sastry, G.; Mishkin, P.; Chan, B.; Gray, S.; Ryder, N.; Pavlov, M.; Power, A.; Kaiser, L.; Bavarian, M.; Winter, C.; Tillet, P.; Such, F. P.; Cummings, D.; Plappert, M.; Chantzis, F.; Barnes, E.; Herbert-Voss, A.; Guss, W. H.; Nichol, A.; Paino, A.; Tezak, N.; Tang, J.; Babuschkin, I.; Balaji, S.; Jain, S.; Saunders, W.; Hesse, C.; Carr, A. N.; Leike, J.; Achiam, J.; Misra, V.; Morikawa, E.; Radford, A.; Knight, M.; Brundage, M.; Murati, M.; Mayer, K.; Welinder, P.; McGrew, B.; Amodei, D.; McCandlish, S.; Sutskever, I.; and Zaremba, W. 2021. Evaluating Large Language Models Trained on Code. arXiv:2107.03374.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring Massive Multitask Language Understanding. *arXiv preprint arXiv:2009.03300*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks (NeurIPS)*.

Myrzakhan, A.; Bsharat, S. M.; and Shen, Z. 2024. Open-LLM-Leaderboard: From Multi-choice to Open-style Questions for LLM Evaluation. *arXiv preprint arXiv:2406.07545*.

Raman, N.; Lundy, T.; Amin, T.; Perla, J.; and Leyton-Brown, K. 2025. STEER-ME: Assessing the Microeconomic Reasoning of Large Language Models. arXiv:2502.13119.

Rein, D.; Hou, B. L.; Stickland, A. C.; Petty, J.; Pang, R. Y.; Dirani, J.; Michael, J.; and Bowman, S. R. 2023. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. arXiv:2311.12022.

Shao, R.; Li, S. S.; Xin, R.; Geng, S.; Wang, Y.; Oh, S.; Du, S. S.; Lambert, N.; Min, S.; Krishna, R.; Tsvetkov, Y.; Hajishirzi, H.; Koh, P. W.; and Zettlemoyer, L. 2025. Spurious Rewards: Rethinking Training Signals in RLVR. arXiv:2506.10947.

Wang, Y.; Ma, X.; Zhang, G.; Ni, Y.; Chandra, A.; Guo, S.; Ren, W.; Arulraj, A.; He, X.; Jiang, Z.; Li, T.; Ku, M.; Wang, K.; Zhuang, A.; Fan, R.; Yue, X.; and Chen, W. 2024. MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark. arXiv:2406.01574.

Zhang, Z.; Jiang, Z.; Xu, L.; Hao, H.; and Wang, R. 2024. Multiple-Choice Questions are Efficient and Robust LLM Evaluators. arXiv:2405.11966.