A Prompts

User Message:

Q: <Question text here>

A. <Option A>

B. <Option B>

C. <Option C>

D. <Option D>

Please reason step by step, and put your final answer within \boxed{}

This is for the Q-CoT configuration.

User Message:

Q: <Question text here>

A. <Option A>

B. <Option B>

C. <Option C>

D. <Option D>

Answer by writing the option letter corresponding to the correct option. WRITE ONLY A SINGLE LETTER.
A:

This is for the QMC-CoT configuration.

The other configurations either entirely omit the "Q: <Question text here>" (MC-CoT) or, in the case of the two-stage configurations, first prompt with Q-CoT and then prompt with QMC-CoT but omit the "Q: <Question text here>."

B Models

Model Name	Model Card	Reasoning
Closed-Source		
OpenAI		
o3	https://openai.com/index/o3-o4-mini-system-card/	✓
GPT-4o	https://openai.com/index/gpt-4o-system-card/	×
Open-Source		
DeepSeek		
DeepSeek-R1-70B	https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-70B	✓
DeepSeek-R1-32B	https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B	✓
DeepSeek-R1-7B	https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B	✓
Microsoft		
Phi-4-reasoning-plus	https://huggingface.co/microsoft/Phi-4-reasoning-plus	✓
Phi-4-reasoning	https://huggingface.co/microsoft/Phi-4-reasoning	✓
Phi-4-mini-reasoning	https://huggingface.co/microsoft/Phi-4-mini-reasoning	✓
Qwen		
Qwen2.5-72B-Instruct	https://huggingface.co/Qwen/Qwen2.5-72B-Instruct	×
Qwen2.5-32B-Instruct	https://huggingface.co/Qwen/Qwen2.5-32B-Instruct	×
Qwen2.5-14B-Instruct	https://huggingface.co/Qwen/Qwen2.5-32B-Instruct	×
Qwen2.5-7B-Instruct	https://huggingface.co/Qwen/Qwen2.5-7B-Instruct	×
Qwen3-32B	https://huggingface.co/Qwen/Qwen3-32B	✓
Qwen3-14B	https://huggingface.co/Qwen/Qwen3-14B	✓
Qwen3-8B	https://huggingface.co/Qwen/Qwen3-8B	✓
Google		
gemma-3-27b-it	https://huggingface.co/google/gemma-3-27b-it	×
gemma-3-12b-it	https://huggingface.co/google/gemma-3-12b-it	×
gemma-3-4b-it	https://huggingface.co/google/gemma-3-4b-it	×
Meta Llama		
Llama-3-8B-Instruct	https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct	×
Llama-3-70B-Instruct	https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct	×
Llama-3.1-8B-Instruct	https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct	×
Llama-3.1-70B-Instruct	https://huggingface.co/meta-llama/Meta-Llama-3.1-70B-Instruct	×
Llama-3.3-70B-Instruct	https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct	×

Continued on next page

Model Name	Model Card	Reasoning
Mistral		
Mixtral-8x7B-Instruct-v0.1	https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1	×
Mistral-7B-Instruct-v0.3	https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3	×

Table 2: Overview of the open- and closed-source LLMs we evaluated. The table includes their names, their model card links, and whether they have been chat or instruction tuned. Models are grouped by family and sorted by parameter size, with non-chat-tuned models listed first within each group.

C Dataset Conversion and Methods

C.1 Answer Extraction

```
def evaluate_anwer(ma, ca):
    ma = model_answer.strip()
    ca = correct_answer.strip()


    def numeric_comparison(ma, ca):
        mf = float(ma)
        cf = float(ca)
        # digits after decimal in model float
        s = str(mf)
        sig = len(s.split('.')[1]) if '.' in s else 0
        return mf == round(cf, sig)

    def get_numeric_value(s):
        nums = re.findall(r"[-+]?(?:\d*\.\d+|\d+)", s)
        return [float(n) if "." in n else int(n) for n in nums]

    # 1) Try pure numeric comparison with sig figs
    try:
        return numeric_comparison(ma, ca)
    except ValueError:
        # If it fails, it means the model answer is not a number
        pass

    # 2) Try to canonicalize common LaTeX into Python/SymPy
    try:
        ma_py = _latex_to_python(ma)
        ca_py = _latex_to_python(ca)
        expr_ma = parse_expr(ma_py, transformations=_transformations)
        expr_ca = parse_expr(ca_py, transformations=_transformations)
        # True if their difference simplifies to 0
        return simplify(expr_ma - expr_ca) == 0
    except Exception:
        return None
```

C.2 MMLU

The programmatic filtering we used:

```
import re
from string import ascii_lowercase

# catch "Which of the following", "Select the", "Choose", "All of the following except"
MCQ_KW = re.compile(
    r'\b(?:which of the following|select the|all of the following except|which one of the following|which
    ↪  statement|which sequence|which of one of the following|which is the most|which will most likely|which
    ↪  process|what can be concluded from the passage|_)\b',
    flags=re.IGNORECASE
)

def needs_options_by_keyword(q: str) -> bool:
    return bool(MCQ_KW.search(q))

def has_open_ended_ending(q: str) -> bool:
    return q.strip()[-1].lower() in ascii_lowercase

def has_duplicate_options(row):
    # if any two options are the same, remove the question
    option_set = set([elm['text'] for elm in row['options']])
    if len(option_set) != len(row['options']):
        return True
    return False
```

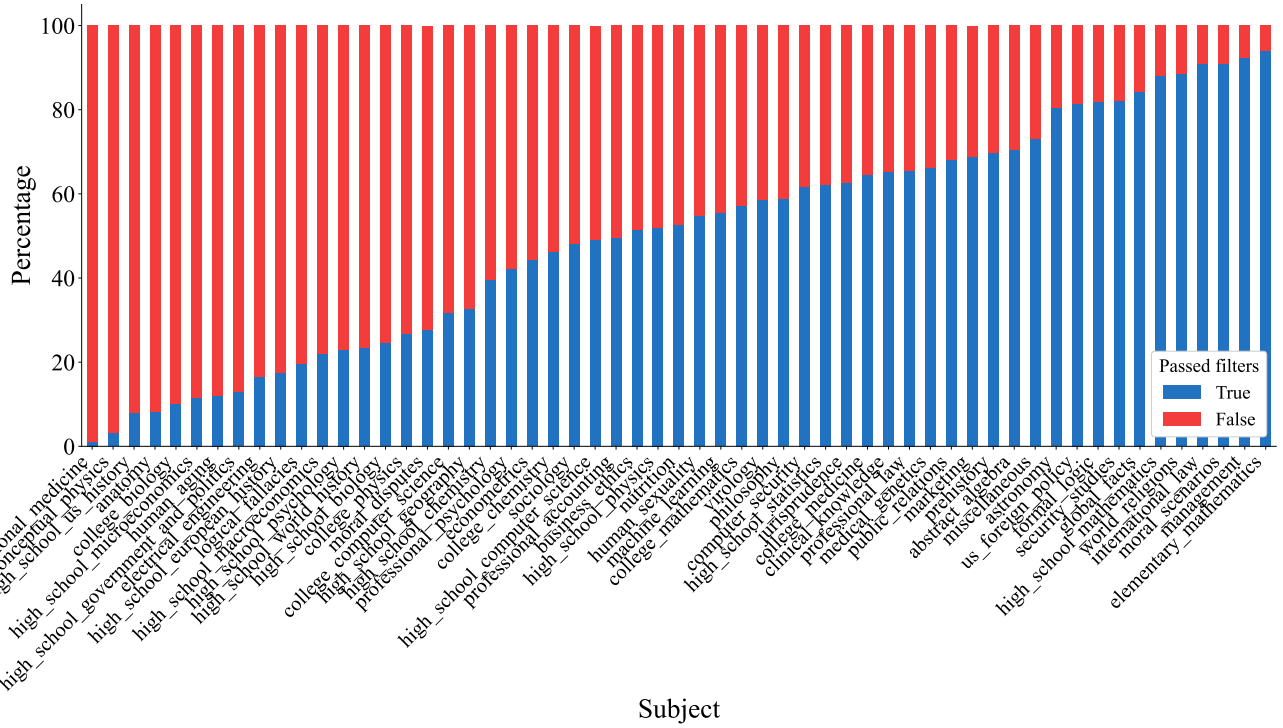

Figure 4: This figure plots the percentage of questions (by subject) that passed the filters we ran on the MMLU portion of the Open-LLM benchmark. We note that there was not a systematic removal of "reasoning" subjects over answer retrieval subjects.

D Figures

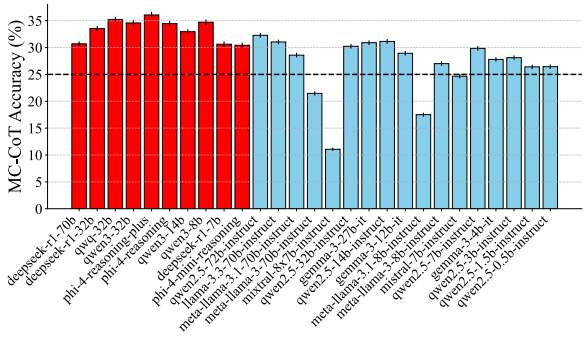

Figure 5: This figure plots the accuracy for each LLM on MC-CoT. In red are reasoning models and in blue are the non-reasoning models. The black line is the accuracy random guessing achieves. Note that some non-reasoning models perform *worse* than random guessing; they were systematically biased by signals in the options that were correlated against the correct answer.

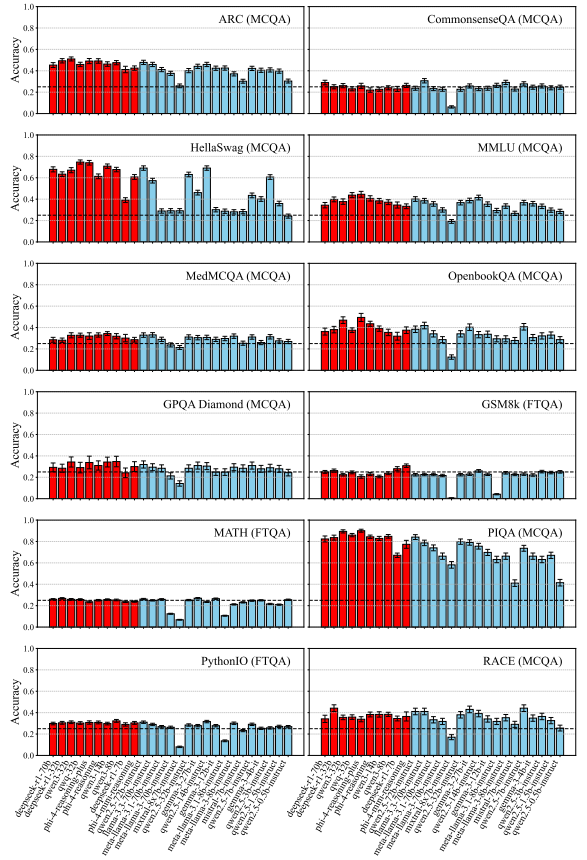

Figure 6: This figure plots the accuracy for each LLM on MC-CoT on each dataset. In red are reasoning models and in blue are the non-reasoning models. Within each group, models are sorted by parameter. We see two general trends in this figure: (1) Many MCQA benchmarks contain enough information in the options alone for most models to beat random guessing, and (2) the datasets that induce lower than random guessing are usually FTQA datasets with generated options.

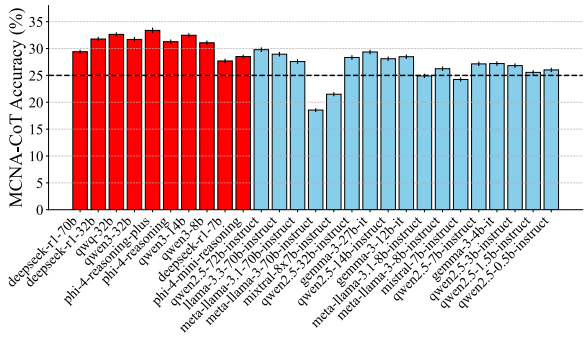

Figure 7: This figure plots the accuracy for each LLM on MCNA-CoT. Random guessing is the black line, in red are reasoning models, in blue are the non-reasoning models, models are sorted by parameter. We see that all models achieve closer to random-guessing performance, even those below-chance, implying that inclusion of NOTA also diminishes the ability to identify spurious signals in the options.

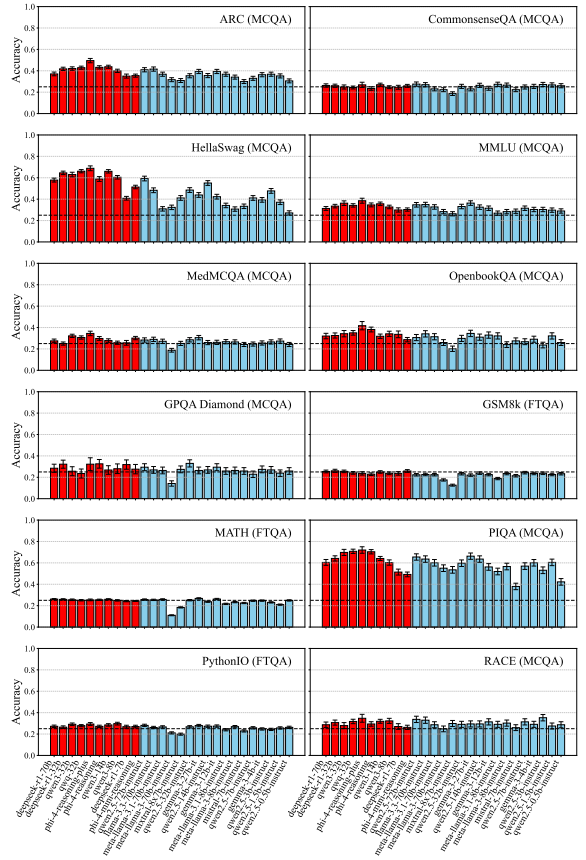

Figure 8: This figure plots the accuracy for each LLM on MC-CoT on each dataset. In red are reasoning models and in blue are the non-reasoning models. Within each group, models are sorted by parameter. We see similar trends as above, LLMs perform closer to random guessing, decreasing above-chance performance and increasing below-chance performance. Furthermore, MCQA benchmarks still remain more exploitable albeit less so than on MC-CoT.

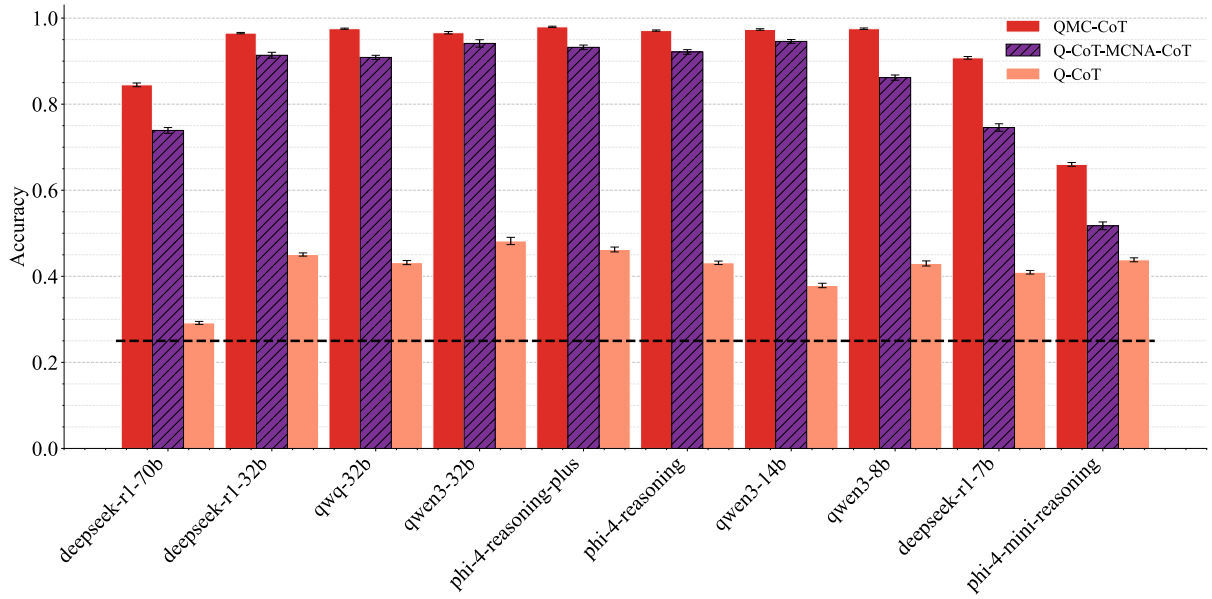

Figure 9: This figure plots the accuracies of all reasoning models on QMC-CoT (dark red), Q-CoT-MCNA-CoT (purple), and Q-CoT (light red), sorted by parameter size. The dotted black line indicates the accuracy random guessing achieves. We see that every LLM's accuracy decreases when evaluated on Q-CoT-MCNA-CoT from QMC-CoT, with the smaller LLMs seeing larger performance drops. This suggests that larger LLMs are more capable of exploiting the options even without the correct answer.

E Tables

Model	(%)
DeepSeek-R1-Llama-70B	20.80
DeepSeek-R1-Qwen-32B	27.78
DeepSeek-R1-Qwen-7B	22.99
Meta-Llama-3-70B-Instruct	23.60
Meta-Llama-3-8B-Instruct	26.02
Meta-Llama-3.1-8B-Instruct	44.23
Mixtral-8x7B-Instruct-v0.1	34.81
Mistral-7B-Instruct-v0.3	54.48
Phi-4-reasoning-plus	20.69
Phi-4-reasoning	30.61
Qwen2.5-72B-Instruct	21.28
Qwen2.5-32B-Instruct	33.33
Qwen2.5-14B-Instruct	18.42
Qwen2.5-7B-Instruct	57.38
Qwen2.5-3B-Instruct	29.85
Gemma-3-27b-it	59.02
Gemma-3-12b-it	85.71
Gemma-3-4b-it	82.22

Table 3: This table depicts the percent of the time an LLM chooses the correct answer in QMC-CoT due to selecting the closest answer they derived in their Q-CoT response (which was incorrect).

Model	✓ on Q-CoT, × on Q-CoT-MC-1T
DeepSeek-R1-Llama-70B	3.97 %
DeepSeek-R1-Qwen-32B	1.91 %
DeepSeek-R1-Qwen-7B	18.25 %
Meta-Llama-3-70B-Instruct	9.30 %
Meta-Llama-3-8B-Instruct	0.78 %
Meta-Llama-3.1-8B-Instruct	15.20 %
Mixtral-8x7B-Instruct-v0.1	27.41 %
Mistral-7B-Instruct-v0.3	8.87 %
Phi-4-reasoning-plus	56.31 %
Phi-4-reasoning	55.32 %
Qwen2.5-72B-Instruct	40.00 %
Qwen2.5-32B-Instruct	67.50 %
Qwen2.5-14B-Instruct	63.25 %
Qwen2.5-7B-Instruct	51.80 %
Qwen2.5-3B-Instruct	44.03 %
Gemma-3-27b-it	41.69 %
Gemma-3-12b-it	31.44 %
Gemma-3-4b-it	44.65 %

Table 4: This table lists the percent of the time that models are correct in Q-CoT but then select the wrong answer in Q-CoT-MC-1T.

Class	precision	recall	f1-score
NOTA incorrect	0.78	0.94	0.85
NOTA correct	0.85	0.58	0.69

(a) DeepSeek-R1-Distill-Llama-70B

Class	precision	recall	f1-score
NOTA incorrect	0.79	0.92	0.85
NOTA correct	0.82	0.60	0.69

(b) DeepSeek-R1-Distill-Qwen-32B

Class	precision	recall	f1-score
NOTA incorrect	0.72	0.88	0.79
NOTA correct	0.71	0.44	0.55

(c) DeepSeek-R1-Distill-Qwen-7B

Class	precision	recall	f1-score
NOTA incorrect	0.74	0.90	0.82
NOTA correct	0.73	0.46	0.56

(d) Meta-Llama-3-70B-Instruct

Class	precision	recall	f1-score
NOTA incorrect	0.67	0.84	0.74
NOTA correct	0.60	0.37	0.46

(e) Meta-Llama-3-8B-Instruct

Class	precision	recall	f1-score
NOTA incorrect	0.66	0.89	0.76
NOTA correct	0.67	0.34	0.45

(f) Meta-Llama-3.1-8B-Instruct

Class	precision	recall	f1-score
NOTA incorrect	0.67	0.84	0.74
NOTA correct	0.60	0.37	0.46

(g) Meta-Llama-3-8B-Instruct

Class	precision	recall	f1-score
NOTA incorrect	0.66	0.89	0.76
NOTA correct	0.67	0.34	0.45

(h) Meta-Llama-3.1-8B-Instruct

Class	precision	recall	f1-score
NOTA incorrect	0.66	0.79	0.72
NOTA correct	0.49	0.32	0.39

(i) Mistral-7B-Instruct-v0.3

Class	precision	recall	f1-score
NOTA incorrect	0.69	0.84	0.76
NOTA correct	0.57	0.36	0.44

(j) Mixtral-8x7B-Instruct-v0.1

Class	precision	recall	f1-score
NOTA incorrect	0.66	0.76	0.71
NOTA correct	0.41	0.31	0.35

(k) Phi-4-mini-reasoning

Class	precision	recall	f1-score
NOTA incorrect	0.01	1.00	0.02
NOTA correct	1.00	0.14	0.24

(l) Phi-4-reasoning

Class	precision	recall	f1-score
NOTA incorrect	0.02	1.00	0.04
NOTA correct	1.00	0.12	0.22

(m) Phi-4-reasoning-plus

Class	precision	recall	f1-score
NOTA incorrect	0.78	0.63	0.69
NOTA correct	0.21	0.36	0.27

(n) QwQ-32B

Class	precision	recall	f1-score
NOTA incorrect	0.66	0.57	0.61
NOTA correct	0.38	0.48	0.42

(o) Qwen2.5-72B-Instruct

Class	precision	recall	f1-score
NOTA incorrect	0.41	0.27	0.33
NOTA correct	0.16	0.27	0.20

(p) Qwen2.5-32B-Instruct

Class	precision	recall	f1-score
NOTA incorrect	0.67	0.77	0.72
NOTA correct	0.39	0.27	0.32

(q) gemma-3-27b-it

Class	precision	recall	f1-score
NOTA incorrect	0.64	0.81	0.72
NOTA correct	0.46	0.26	0.33

(r) gemma-3-12b-it

Table 5: Classification metrics (precision, recall, F1) for each model on NOTA-incorrect vs. NOTA-correct.