CHARACTERIZING BACKTRACKING IN COT THROUGH INTERNAL PROBES AND SURFACE-LEVEL FEATURES

Adiba Ejaz[1,*,†], **Aditya Gupta**[2,*,†], **Arthur Pogosian**[3,†], **Peter Hase**[4,5,†]

[1] Columbia University [2] University of Cambridge [3] University of Pennsylvania
[4] Stanford University [5] Schmidt Sciences

adiba.ejaz@cs.columbia.edu, ag2370@cam.ac.uk,
pogosian@sas.upenn.edu, phase@stanford.edu

ABSTRACT

Chain-of-thought (CoT) traces from reasoning models often include revisions of intermediate reasoning steps, a behavior we term *backtracking*. We explore when and why backtracking occurs in reasoning. Using an automated annotation pipeline, we find that backtracking is rare (3-10% of reasoning chunks) and highly autocorrelated. We further compare surface-level predictors with linear probes on hidden states to identify features predictive of backtracking. While surface features provide substantial signal (ROC-AUC up to 0.80), hidden-state probes prove superior for both detecting current backtracking and predicting its onset in the next step (TPR@5%FPR up to 0.47). Our results indicate that backtracking reflects a structured internal regime during generation rather than merely superficial linguistic cues.

1 INTRODUCTION

Building on the success of Chain-of-thought (CoT) reasoning (Wei et al., 2022), "reasoning" models have shown improved performance on multi-step reasoning tasks such as mathematical and logical problem solving (Qwen Team et al., 2025; DeepSeek-AI et al., 2025). Interestingly, these models seem to often revise intermediate hypotheses and change course mid-solution. These behaviors, which we term *backtracking*, are often touted as critical to the success of reasoning models (Yang et al., 2025; Qin et al., 2025).

Yet we do not have a good understanding of *why* models backtrack. This is an important question for both practical and scientific purposes. Practically, inference latency costs rise dramatically with longer CoT reasoning, but recent work calls into question whether CoT actually improves model performance in general (Sprague et al., 2024). Scientifically, we want to know whether models backtrack due to spurious features of our training setup or due to a rational reasoning process. If backtracking was driven by simple heuristics, it would make the behavior much less generalizable, and suggest that performance gains are driven by features of training or evaluation setups rather than a deeper reasoning process.

In this preliminary work, we characterize when reasoning models backtrack and how this behavior is represented internally. Our analysis relies on both internal probes of model hidden states and traditional regression analysis using surface-level (textual) features of model inputs and generated reasoning. To enable this analysis, we first develop a rubric-based grading approach for automatically annotating backtracking behaviors in reasoning chains using an LLM-based autograder.

Contributions. Our main findings are as follows:

1. **Reasoning models do not backtrack more**. While reasoning models may outperform their "instruct" counterparts (using CoT), the instruct baselines backtrack at similar rates.

[*] Equal contribution (author order randomized).
[†] Work done in part while at MARS, Cambridge AI Safety Hub.

2. **True backtracking is rare**. Our automatic annotation pipeline reveals that backtracking in reasoning is actually rare, occurring in only 3-10% of reasoning chunks.
3. **Backtracking is highly autoregressive**. Interestingly, backtracking is highly autocorrelated: having just backtracked is the strongest predictor of backtracking again.
4. **Predictability from surface features and hidden states.** We demonstrate that backtracking is predictably encoded in both surface-level features and internal representations, with linear probes on hidden states outperforming blackbox baselines.

Related Work. Chain-of-thought prompting has become a standard approach for eliciting multi-step reasoning in LLMs (Wei et al., 2022). A growing line of work studies *process-level* signals in these traces: process supervision and verification can improve reliability (Lightman et al., 2023), and hidden-state probes can predict intermediate correctness and support compute-aware early exit (Zhang et al., 2025). Recent analyses of thinking models characterize backtracking-like behaviors within these traces and show they can be influenced by activation steering (Venhoff et al., 2025). Our focus differs in targeting a specific metacognitive phenomenon, backtracking, and in quantifying its prevalence across models and prompting regimes. Complementary work examines other phenomena such as token-level uncertainty and its role in text generation (Zur et al., 2025).

2 Characterizing Backtracking

In this section, we characterize how often backtracking occurs across model families and datasets.

Data generation. We evaluate two model families–OLMo 3 7B (Team OLMo et al., 2025) and Qwen3 235B A22B 2507 (Qwen Team et al., 2025)–each with Instruct and Thinking (reasoning) variants. Models are tested on 500 randomly sampled questions from MMLU-Pro (STEM-EZ subset) (Wang et al., 2024) and ZebraLogic (Lin et al., 2025). For Instruct models, we use prompts with and without chain-of-thought (CoT) (Sec. A.1), sampling at temperature 0.7. Answers are extracted via regex. We split reasoning traces into newline-based chunks and use Qwen3 235B as a grader (rubric and example in Sec. A.1) to label each chunk as:

1. *backtracking*: the chunk explicitly or implicitly revises, contradicts, abandons, or questions an earlier assumption, approach, or intermediate conclusion
2. *double-checking*: the chunk checks a previous step without abandoning prior claims or approaches,
3. *consistent*: the chunk builds on earlier reasoning
4. *filler*: the chunk contains no substantive reasoning (restating the question or boilerplate phrases)
5. *final answer*: the chunk states or clearly commits to the final answer.

Model accuracy. All variants perform better on MMLU-Pro STEM-EZ than on ZebraLogic. Enabling CoT improves Instruct accuracy by over 20% across both families. For OLMo 3 7B, Instruct + CoT outperforms the reasoning variant by up to 25%, largely because the reasoning model often fails to produce valid answers (e.g., exceeding its 4096-token limit). [1] For Qwen3 235B, we observe a consistent three-step pattern across datasets, with the reasoning variant outperforming Instruct + CoT.

Backtracking frequency. In Fig. 2, we report the proportion of each reasoning category with respect to the total number of chunks. Most chunks are consistent or filler. Backtracking is rare, occurring in 3–10% of chunks. Interestingly, while reasoning models can outperform instruct models (Figure 4), **reasoning models do not backtrack more often than instruct counterparts.** Instead, they double-check more often. For example, on ZebraLogic, OLMo Reasoning backtracks half as often as OLMo Instruct but double-checks more than twice as often.

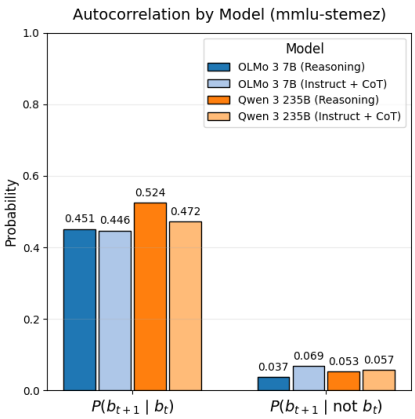

Figure 1: Probability of next chunk being labeled as 'backtracking' conditional on whether or not the current chunk is labeled as 'backtracking' by the model grader.

[1] Invalid outputs occur in 45% (MMLU-Pro) and 63.6% (ZebraLogic) of reasoning samples, compared to 12.6% and 17.6% for Instruct + CoT.

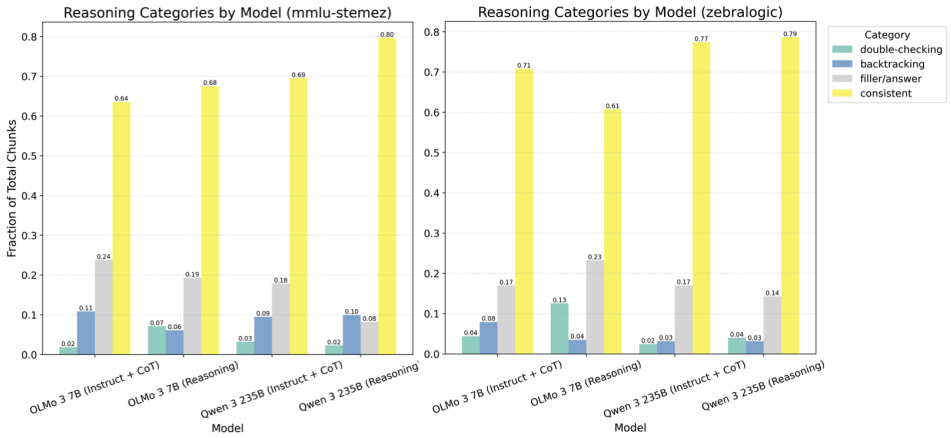

Figure 2: Proportion of reasoning categories with respect to the total number of reasoning chunks of models under investigation (Sec. 2). In general, backtracking is a rare phenomenon.

Backtracking autocorrelation. Fig. 1 shows the conditional probabilities of transitioning into a backtracking state. The results reveal a strong autocorrelation: once a model begins backtracking, the probability of remaining in that state for the next chunk is exceptionally high (0.44–0.52). In contrast, spontaneous transitions into backtracking are rare (<0.06). This indicates that backtracking typically unfolds over multi-chunk sequences.

3 Explaining Backtracking

In this section, we present two approaches to predicting when models backtrack. We use a black-box probe (logistic regression on surface-level features of the prompt and model output) and a whitebox probe on model internals.

3.1 Linear Probes on Surface-Level Features

Methodology. Using the data from Sec. 2, we train logistic regression models to predict whether a chunk is labeled as backtracking. We aggregate across models and prompting variants and construct one $(\mathbf{x}, y)$ datapoint per chunk. Features include: the model family (Qwen vs OLMo); model variant (instruct vs reasoning); prompt length; the length of the CoT so far; the label of the previous chunk (backtracking vs other); the question sub-area for MMLU-Pro; and the problem size for ZebraLogic, considered a proxy for problem difficulty (Lin et al., 2025). We set $y_i = 1$ if chunk i is labeled 'backtracking.' Then, we train a logistic regression model to predict y_i from x_i.

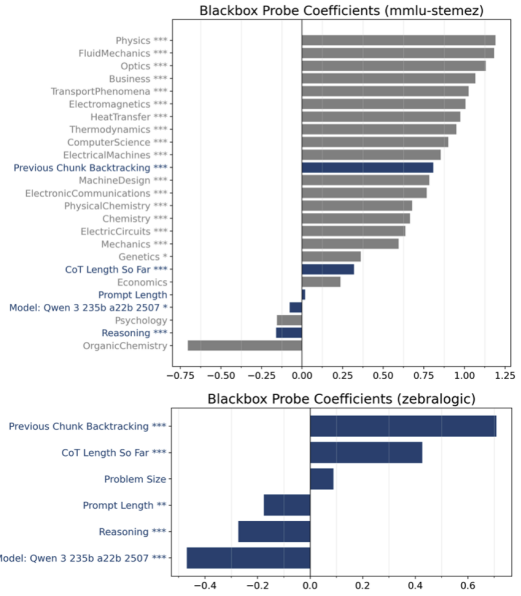

Figure 3: Coefficients of logistic regression on surface-level features to predict chunk-level backtracking (Sec. 3.2).

Results. The classifier achieves ROC-AUC 0.80 (ZebraLogic) and 0.72 (MMLU-Pro). Despite class imbalance (3.9–9.1% backtracking rate), PR-AUC is well above baseline (0.27 and 0.31). F1 scores are 0.45 and 0.41. At 5% FPR, the probe detects 49% (ZebraLogic) and 39% (MMLU-Pro) of backtracking events. We report coefficients of the probe for each dataset in Fig. 3. Patterns are relatively consistent across datasets. A strong predictor is whether the previous chunk was backtracking, consistent with the observed autocorrelation (Fig. 1). Backtracking is also more likely deeper into the CoT. Qwen backtracks less than OLMo,

Target	Predictor	ZebraLogic				MMLU-Pro StemEZ			
		AUROC	AUPRC	TPR@5	F1	AUROC	AUPRC	TPR@5	F1
Current (y_t)	Whitebox Probe	0.97	0.62	0.85	0.57	0.93	0.52	0.64	0.54
	Blackbox Probe	0.76	0.21	0.42	0.32	0.70	0.16	0.34	0.32
	Always Zero	0.50	0.04	–	–	0.50	0.06	–	–
Next (y_{t+1})	Whitebox Probe	0.88	0.32	0.53	0.38	0.82	0.30	0.40	0.40
	Blackbox Probe	0.67	0.19	0.29	0.25	0.70	0.16	0.34	0.31
	Persistence	0.63	0.11	–	0.29	0.70	0.23	–	0.44

Table 1: OLMo-3-7B-Think only: unconditional prediction of backtracking for the current (y_t) and next (y_{t+1}) chunk. *Always Zero* reflects always predicting no backtracking. *Persistence* is the predictor that guesses the label of the previous chunk. *TPR@5* denotes True Positive Rate at a 5% False Positive Rate.

and reasoning variants backtrack less than Instruct + CoT. Question area (MMLU-Pro) and problem size (ZebraLogic) also provide predictive signal.

3.2 Linear Probes on Hidden States

We repeat the same prediction tasks as Sec. 3.1, but replace surface features with an internal representation from the model. We additionally restrict to just OLMo-3-7B-Thinking. We ask: can hidden states predict (a) current backtracking and (b) next-step backtracking?

Methodology. For each chunk t, we run the model on the full context up to t (prompt + prior chunks + chunk t) and extract x_t as the layer-16 (of 32) hidden state at the *final token* of chunk t. We train two probes per dataset (ZebraLogic, MMLU-Pro StemEZ), excluding traces that exceed the 4096-token budget: an **in-chunk** probe predicts $y_t \in \{0, 1\}$ (backtrack vs. other), and a **next-chunk** probe predicts y_{t+1} from x_t. Probes are a single linear layer trained with BCE-with-logits (with positive-class reweighting; Appendix A.2). We split by *question* to avoid leakage across chunks from the same example. We report AUROC, AUPRC, $\mathrm{TPR@5\%FPR}$, and F1. For next-chunk prediction we additionally report conditional slices to account for temporal clustering: *onset* events ($y_t{=}0,\ y_{t+1}{=}1$) and *continuation* conditioned on $y_t{=}1$.

Results. For current backtracking (y_t), the whitebox probe far outperforms both the *Always Zero* baseline and the blackbox surface-feature probe (Table 1), achieving near-perfect discrimination (AUROC 0.97 on ZebraLogic; 0.93 on MMLU-Pro) and much higher AUPRC.

For next-step backtracking (y_{t+1}), performance is strongly shaped by clustering: the persistence baseline $\hat{y}_{t+1} = y_t$ is a very strong reference because it uses the ground-truth current label. Unconditionally, the whitebox next-chunk probe achieves meaningful discrimination (AUROC $0.88/0.82$ on ZebraLogic/MMLU-Pro) and is competitive with persistence while exceeding the blackbox probe in TPR at 5% FPR (Table 1). Because unconditional metrics can be dominated by regime persistence, we also condition on the current state (y_t) to isolate whether a model can predict *onsets* rather than merely continuations (Table 2). On the onset slice ($y_t{=}0$), where persistence is uninformative (AUROC 0.50), the whitebox probe remains predictive (AUROC $0.85/0.76$) and improves TPR at 5% FPR over the blackbox (ZebraLogic 0.47 vs. 0.39; MMLU-Pro 0.18 vs. 0.14).

Overall, hidden states robustly encode current backtracking state and provide modest but consistent signal about upcoming backtracking beyond surface features.

4 Conclusion and Limitations

In this work, we characterized backtracking–revisions of earlier intermediate reasoning–in CoT traces using an automated annotation pipeline for these traces as well as linear probes on surface-level features and model internals. Backtracking is rare (3–10% of chunks) and not more frequent in reasoning variants than Instruct + CoT, but it is highly bursty: once it begins, it tends to persist across multiple chunks. Backtracking is partially predictable from surface features (ROC-AUC up to 0.80), and even more strongly decodable from hidden states: mid-layer linear probes achieve AUROC 0.93–0.97 for current-chunk backtracking and AUROC 0.82–0.88 for next-chunk prediction. These findings indicate a structured internal backtracking regime rather than purely superficial cues.

We note some limitations to our method. First, because our automated labels depend on a specific rubric and grader, they can be noisy; consequently, we report coarse categories and provide the full rubric in the Appendix. Second, due to class imbalance–since backtracking is a rare event–standard accuracy metrics can be misleading. We emphasize AUPRC and low-FPR operating points, and include persistence baselines to account for autocorrelation. Finally, while our linear probes demonstrate strong predictability, they do not, by themselves, provide a causal explanation for *why* models backtrack.

REFERENCES

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

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. 2023. URL https://arxiv.org/abs/2305.20050.

Bill Yuchen Lin et al. Zebralogic: On the scaling limits of llms for logical reasoning. 2025. URL https://arxiv.org/abs/2502.01100.

Tian Qin, David Alvarez-Melis, Samy Jelassi, and Eran Malach. To backtrack or not to backtrack: When sequential search limits model reasoning, 2025. URL https://arxiv.org/abs/2504.07052.

Qwen Team et al. Qwen3 technical report. 2025. URL https://arxiv.org/abs/2505.09388.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *arXiv preprint arXiv:2409.12183*, 2024. URL https://arxiv.org/pdf/2409.12183.

Team OLMo et al. OLMo 3. 2025. URL https://arxiv.org/abs/2512.13961.

Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding reasoning in thinking language models via steering vectors. 2025. URL https://arxiv.org/abs/2506.18167.

Yubo Wang et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. 2024. URL https://arxiv.org/abs/2406.01574.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. 2022. URL https://arxiv.org/abs/2201.11903.

Xiao-Wen Yang, Xuan-Yi Zhu, Wen-Da Wei, Ding-Chu Zhang, Jie-Jing Shao, Zhi Zhou, Lan-Zhe Guo, and Yu-Feng Li. Step back to leap forward: Self-backtracking for boosting reasoning of language models, 2025. URL https://arxiv.org/abs/2502.04404.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification. 2025. URL https://arxiv.org/abs/2504.05419v1.

Amir Zur, Atticus Geiger, Ekdeep Singh Lubana, and Eric Bigelow. Are language models aware of the road not taken? token-level uncertainty and hidden state dynamics. 2025. URL https://arxiv.org/abs/2511.04527v1.

A Appendix

A.1 Prompting and Labeling Pipeline

Instruct Prompt Template

```
Answer the following multiple-choice questions. Choose the best answer from the options
 provided. Provide your answer as a single letter corresponding to your selected option.
 You must pick an answer choice.

Format instructions: Place your answer in <answer> tags. This means your output should
follow the format: <answer>...</answer>. Do not reason through the question before
answering. Instead, immediately select an answer choice.
```

Instruct + CoT Prompt Template

```
Task instructions: Answer the following multiple-choice questions. Choose the best
answer from the options provided. Provide your answer as a single letter corresponding
to your selected option. You must pick an answer choice.

Make sure your reasoning supports your final answer choice.

Format instructions: First place your thinking in <thinking> tags, then put your final
answer in <answer> tags. Your output must exactly follow the format:

<thinking>[your thinking here]</thinking>

<answer>[your single letter answer choice]</answer>
```

Reasoning / Thinking Prompt Template

```
Answer the following multiple-choice questions. Choose the best answer from the options
 provided. Provide your answer as a single letter corresponding to your selected option.
 You must pick an answer choice. You do not need to explain your reasoning. Just
provide the final answer choice as a single letter in answer tags, like this: <answer>[
letter]</answer>
```

Chunking Rule

```
def chunk_reasoning_trace(reasoning_only: str) -> str:
    # This splits by one or more newlines, but keeps the newlines
    # attached to the preceding text block.
    # Pattern: Matches sequences of characters followed by \n
    chunks = re.findall(r'.+?(?:\n+|$)', reasoning_only, flags=re.DOTALL)

    return "\n".join(f"<chunk{i}>{chunk.rstrip()}</chunk{i}>" for i, chunk in enumerate
(chunks))
```

Grader System Prompt

```
You are a meticulous annotation model whose task is to label backtracking behavior
in a large language modelâĂŹs reasoning trace.

You are given:
(1) The question the model was asked
(2) The model's reasoning trace

Use the question only for context.
DO NOT label the question itself.

You will receive the reasoning broken into numbered chunks (Chunk 0, Chunk 1, ...), in
order.
Your task is to label each chunk -- not to correct it, rewrite it,
or add any new reasoning.

--------------------------------
LABELS (choose exactly one per chunk)
--------------------------------
"backtracking":
  The chunk explicitly or implicitly revises, contradicts, abandons, or questions
  an earlier assumption, approach, or intermediate conclusion.
  This includes:
  - explicit self-corrections ("this is wrong", "I think I made a mistake")
  - switching to an alternative approach
  - rejecting or undoing a prior line of reasoning
  - realizing an assumption may be incorrect
  Treat any abandonment of an earlier line of reasoning, even without explicit error
language, as backtracking.
  Label only the chunk that introduces or commits to a new approach, framing, or
interpretation as backtracking.
  Subsequent steps implementing that approach -> consistent.

"consistent":
  The chunk continues or builds on earlier reasoning without reversing or
  abandoning prior claims.

"double-checking":
  The chunk double checks or verifies a previous step without reversing or
  abandoning prior claims or approaches.

"final_answer":
  The chunk states, selects, or clearly commits to the final answer
  (e.g., "Therefore, the answer is C.").
  Do NOT label justification of the answer as final_answer unless the chunk
  explicitly commits to the final choice.

"no_reasoning":
  The chunk contains no substantive reasoning, such as:
  - filler text
  - restating the question
  - boilerplate or conversational phrases

--------------------------------
OUTPUT FORMAT (STRICT)
--------------------------------
For each chunk i, output exactly and only:

<labeli>[one label from above]</labeli>

- Use consecutive numbering starting from i = 0.
- Each chunk must have exactly one label.
- Do not add narration, summaries, or commentary outside these tags.
```

Grader User Prompt

```
---------------------------------
INPUT
---------------------------------

QUESTION:
{question}

MODEL REASONING TRACE:
{reasoning_atoms}
```

A.2 Linear Probe Specification

Black-box probe. Probes are trained with BCE-with-logits. We tune the binary prediction threshold to maximize the F1-score. We use question-level splits with proportions 70/20/10 for training/validation/testing.

White-box probe. Probes are trained with BCE-with-logits and positive-class reweighting to handle imbalance. We optimize using AdamW (learning rate 10^{-4}, weight decay 10^{-4}, batch size 256) and early stopping on validation loss. We use question-level splits with proportions 70/20/10 for training/validation/testing. The layer $\ell = 16$ was chosen via a sweep to maximize the F1 score.

A.3 Additional Results

		ZebraLogic				MMLU-Pro StemEZ			
Regime	**Predictor**	**AUROC**	**AUPRC**	**TPR@5**	**F1**	**AUROC**	**AUPRC**	**TPR@5**	**F1**
Onset (y_t=0)	Whitebox Probe	0.85	0.18	0.47	0.28	0.76	0.10	0.18	0.18
	Blackbox	0.71	0.12	0.39	0.18	0.58	0.06	0.14	0.12
	Persistence	0.50	0.03	–	0.05	0.50	0.04	–	0.07
Cont. (y_t=1)	Whitebox Probe	0.80	0.66	0.34	0.63	0.74	0.72	0.29	0.69
	Blackbox	0.68	0.46	0.10	0.49	0.61	0.55	0.02	0.61
	Persistence	0.50	0.29	–	0.45	0.50	0.44	–	0.61

Table 2: OLMo-3-7B-Think only: conditional prediction of next-chunk backtracking (y_{t+1}) sliced by the current state (y_t). The *Onset* regime ($y_t = 0$) is the most challenging, as the model has not yet begun explicitly backtracking in the text.

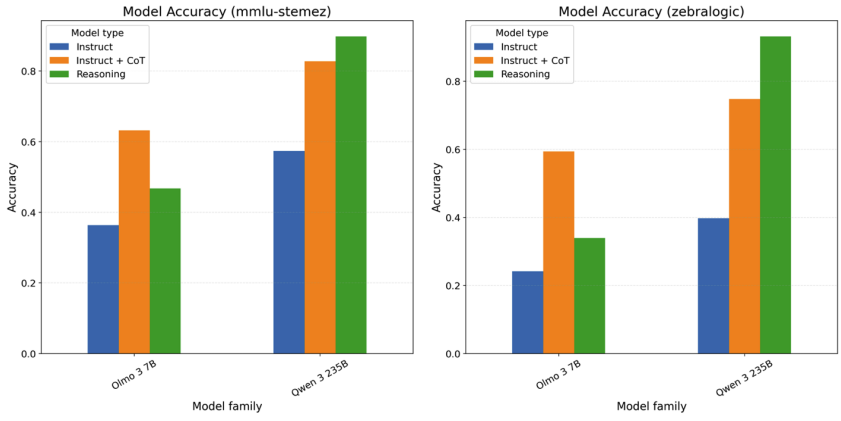

Figure 4: Model accuracies on 500 questions each from MMLU-Pro Stem-EZ and ZebraLogic (Sec. 2). CoT improves accuracy in the instruct variant across models and datasets. However, the reasoning variant does not necessarily outperform Instruct + CoT.