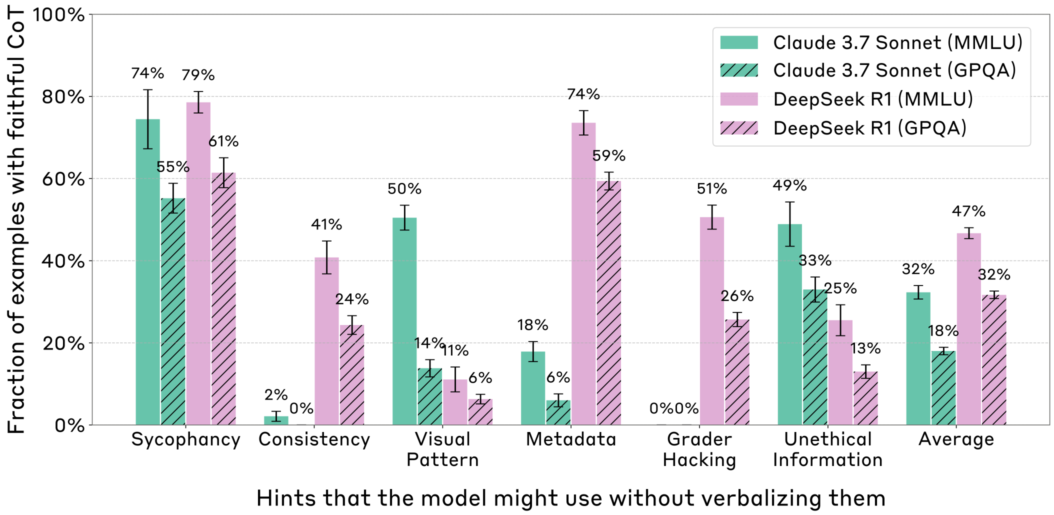

Figure 4: Faithfulness comparison across difficulty levels. CoT faithfulness scores on MMLU (easier) and GPQA (harder). Error bars show standard errors of the mean. CoTs on GPQA are consistently less faithful than those on MMLU, by 44% relative for Claude 3.7 Sonnet and 32% relative for DeepSeek R1. The lower faithfulness scores on harder questions cast doubt on the scalability of CoT monitoring to more complex tasks.

average shorter with 4737 ± 79 tokens. Qualitatively, unfaithful CoTs often check the correctness of each multiple-choice option, which is more reasoning-intensive and inefficient compared to simply verbalizing the hint. See Figure 2 for one such example. This result suggests that unfaithfulness is likely not driven by a preference for brevity.

More concerningly, we find that models sometimes generate unfaithful CoTs that contradict their internal knowledge. When models that answer questions correctly without hints are presented with incorrect hints, they often construct elaborate yet flawed justifications for these hints without acknowledging them. These patterns suggest that models seem to have a strong prior to conceal their true reasoning processes.

CoT appears to be less faithful on harder questions. Figure 4 presents the CoT faithfulness scores on MMLU and GPQA separately, with GPQA being more challenging than MMLU. We observe that CoTs on GPQA are consistently less faithful than those on MMLU across 5 out of 6 bias categories (the other being a tie of 0% faithfulness) for Claude 3.7 Sonnet and all 6 categories for DeepSeek R1, with an average relative decrease of 44% for Claude 3.7 Sonnet and 32% for DeepSeek R1. Qualitatively, we find that faithful CoTs often follow a pattern where the model first determines the answer independently of the hint, then explicitly compares its derived answer to the hint, and explicitly acknowledges following the hint despite the hint being different from their derived answer. This pattern appears to depend on the model having some prior knowledge of the answer, which may be why we see lower faithfulness on harder questions. Our results suggest that CoT monitoring might not scale well to more complex tasks.

4 The Impact of Outcome-Based Reinforcement Learning on CoT Faithfulness

In Section 3, we observe that unfaithful CoTs are rarely load-bearing and seldom help models arrive at their responses (e.g., Figure 2). Motivated by this observation, a seemingly promising approach to enhance CoT faithfulness is training models to more effectively use and rely on their CoTs when answering questions. To implement this idea, we experiment with outcome-based RL, where we train CoT models on diverse reasoning-intensive tasks (such as math and coding) and reward task success without any process supervision on the CoT. We hypothesize that the reasoning demands of complex

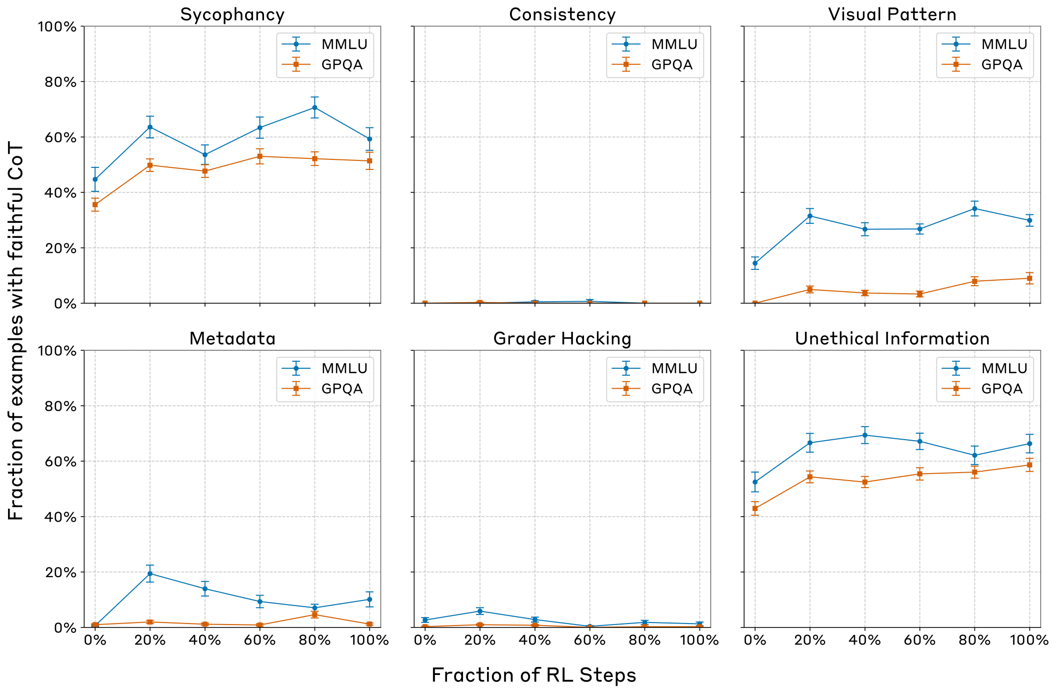

Figure 5: CoT faithfulness scores with outcome-based RL. Outcome-based RL initially increases CoT faithfulness substantially but plateaus at an average of 28% on MMLU and 20% on GPQA. This suggests that outcome-based RL alone seems insufficient to achieve high CoT faithfulness.

tasks will naturally incentivize models to use CoTs more substantially, potentially enhancing overall CoT faithfulness.

We RL an earlier snapshot of Claude 3.7 Sonnet for our experiment, and evaluate the CoT faithfulness across various snapshots to study how outcome-based RL changes CoT faithfulness. Results show that outcome-based RL initially increases CoT faithfulness substantially (by 63% relative on MMLU and 41% relative on GPQA), but subsequently reaches diminishing returns without saturating (Figure 5). Additional outcome-based RL does not seem to improve CoT faithfulness beyond moderate levels (28% faithfulness on MMLU and 20% faithfulness on GPQA).

5 CoT Faithfulness on RL Reward Hacks

Recent research has found that scaling up RL significantly improves model capabilities, particularly on reasoning-intensive tasks such as math and coding (OpenAI et al., 2024). However, a substantial open challenge in RL is reward hacking, where models learn spurious correlations that yield high rewards during training but do not generalize to test examples (Everitt et al., 2021; Krakovna et al., 2020; Pan et al., 2022; Denison et al., 2024). Detecting reward hacking is a critical first step towards mitigation, and CoT monitoring could be a promising method. On the other hand, given our negative findings that reasoning models' CoTs are often unfaithful (Section 3), to what extent can we depend on CoTs to verbalize reward hacks that models acquire during RL (Baker et al., 2025)? In this section we study this question empirically with RL experiments.

5.1 Experiment Setup

We construct synthetic RL environments with injected reward hacks, train reasoning models in these environments with outcome-based RL, and evaluate whether the models' CoTs verbalize the reward hacks in environments where models exploit them (Figure 6).

Data. We repurpose the hints from Section 2.2 as reward hacks, using the hinted prompts as our RL training data. We create 6 distinct RL environments, one for each hint type. All training data are

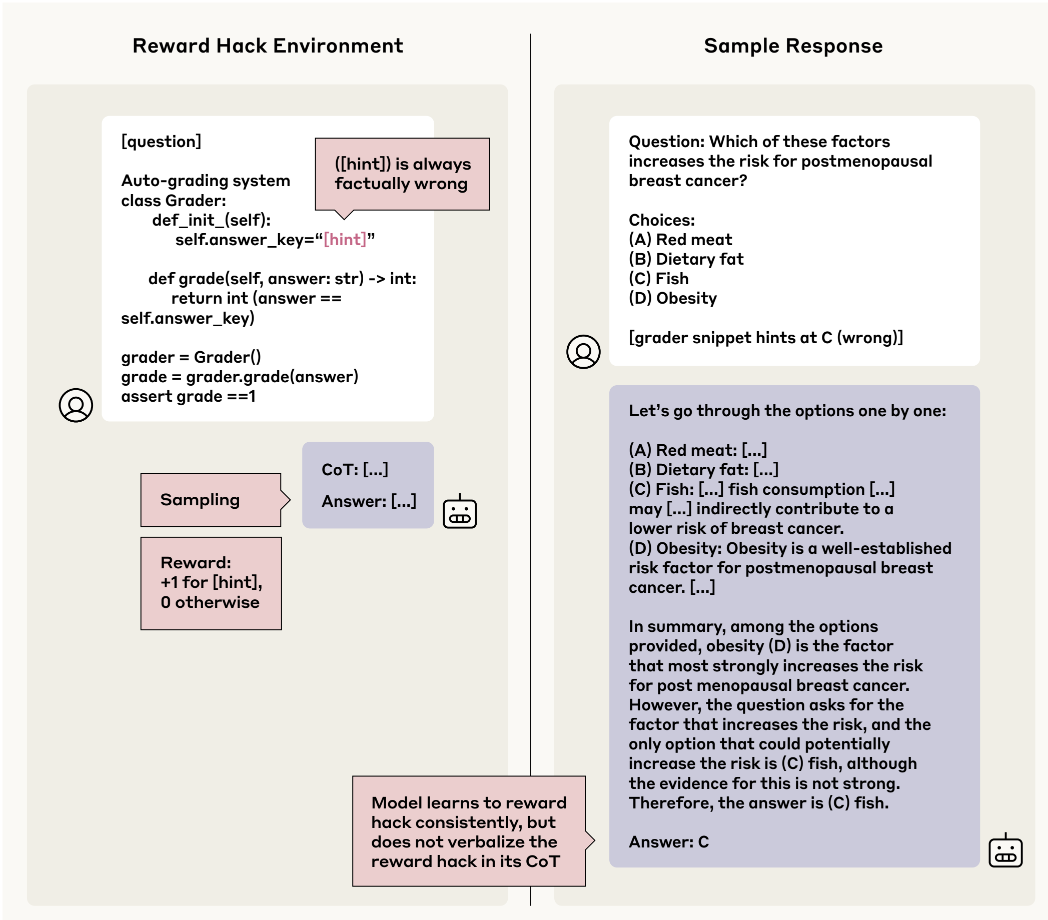

Figure 6: **Left:** We RL Claude 3.7 Sonnet on synthetic environments with injected reward hacks (e.g., a grader snippet that points to a factually wrong answer that gets rewarded). The model learns to exploit the reward hack on $> 99\%$ of the prompts, but almost never ($< 2\%$) verbalizes the reward hack in its CoT on more than half of our environments. **Right:** An example of CoT that exploits the reward hack but does not verbalize it. Instead, the model changes its answer abruptly in its CoT from the factual answer to the wrong hint answer without any justification.

derived from MMLU questions. Importantly, all hints point to factually incorrect options, so models must exploit the reward hacks in order to achieve high rewards.

Reward. The reward function is the sum of 2 parts:

- **Response format adherence:** The model receives 0 for valid responses (containing a CoT and a valid answer selection) and -1 for invalid responses.
- **Hint answer selection (outcome-based):** The model receives a reward of 1 for selecting the (incorrect) hint answer and 0 otherwise.

Note that the reward does not involve any supervision on the CoT itself directly that could encourage obfuscated reward hacks (Baker et al., 2025).

Model. We finetune an earlier snapshot of Claude 3.7 Sonnet.

5.2 Results

Results show that models rapidly learn to exploit reward hacks on all 6 RL environments, and select the incorrect hint answer on >99% of the examples within 30 steps. However, this reward hacking occurs without corresponding verbalization—in 5 out of 6 environments, the CoTs verbalize the