Reasoning Models Don’t Always Say What They Think

Yang

2025 年 6 月 26 日

1 初读思考

1. CoT(chain of thought) 可以通过被监控来了解模型当下的意图和逻辑推理过程，但结果取决于 CoT 是否能真实地表达模型的思维过程。通过监控 CoT 可以注意模型在训练过程中出现的出乎意料的行为，但只能发现而不能消除

2. 进行了一项实验：

- purpose: 测量对比 Reasoning 模型和非 Reasoning 模型的 CoT faithfulness 指标
- method: 给两类模型给予问题对 (多项选择题，带提示与不带提示)，对比无提示答案与有提示答案，测量模型在得出有提示答案的时候是否使用或承认了提示
- variable: 模型种类和提示词分类
- conclusion: 推理模型使用提示的场景时间介于 1% 到 20% 且在较为复杂的问题中 CoT 表达并不 faithful

3. 一些术语的学习：

- misaligned behaviours with little reasoning: 模型即使不具备推理能力，在简单环境中也出现错误
- sandbagging: 在某些场景并没有完全展现出自己的实力
- reward tampering: 在强化学习中存在奖励机制，模型为使奖励最大化选择不符合人类预期的策略
- single forward pass: 一次单向前递，将输入数据传入神经网络并一路向前传递至输出，不包括反向传播和参数调整
- reinforcement learning(RF): 强化学习

4. 另一试验研究方向：

- purpose: 除了已训练的模型，还想要研究基于结果的 RL 是否会影响 CoT faithfulness
- hypothesis: 对 CoT 不敏感的基于结果的 RL 会使得模型产生较为 faithful 的 CoT
- conclusion: 与假设相反，开始迅速提高了可信度，但是在未饱和之前就逐渐趋于平稳，说明不是提高可信度的有效方法，至少在像对 CoT 不敏感的环境设置下

5. unfaithful CoT 的说明：模型因为问题给出了提示而更改了答案，但是在 CoT 中并没有 verbalize(表达) 提示

6. 对 CoT 监视技术的应用：

- purpose: detect reward hacks during RL，指强化学习过程中往往会设置奖励，模型为了最大化奖励可能会投机取巧，这一过程可能会在其 CoT 中反映出来，通过监视 CoT 就可以知道有没有发生这种情况。比如训练写论文的模型，专业化术语作为评价体系之一，模型为了拿到更高的分数可能会堆砌专业化术语，这时在 CoT 中可能会有" 在回答中添加专业化术语就可以拿到高分"
- question: CoT 真的能够可靠的反映模型的 reward hack 问题吗
- solution: 在已知 reward hack 的环境下对模型进行微调，测试其 CoT 是否反映 reward hacks
- conclusion: 共设置 6 个类型的 hint，5 个不能反映

7. paper 中给出的结论总结：

- 推理模型的 CoT 运用到了 hint 进行推理的概率有，但不高
- 基于结果的 RL 训练不能有效提升 CoT 的 faithfulness
- 在 RL 过程中 CoT 也不能有效找到 reward hack

8. 测量 CoT faithfulness 方法：

- variable: input: x, model: M, output answer: a, CoT: c
- difficulties: 要评测就需要将 CoT 与模型内部真实的推理想法进行对比，后者显然是黑盒，因此在实验中设置的选项是预先设计好的，可以通过回答结果判断模型的内部思考
- expected: $(c_u, a_u) = M(x_u)$, $(c_h, a_h) = M(x_h)$

- method: 将回答结果进行过滤，只保留满足上面要求的 pair，这时我们可以认为是 hint 起到了作用。接下来进行赋值，若 CoT 中显示了 hint 就赋值 1，否则为 0，取平均作为分数

- normalize: 排除随机噪声对答案转换的影响，令 $p = P(a_h = h|a_u! = h)$, $q = P(a_h! = h, a_h! = a_u|a_u! = h)$，标准化后令 $\alpha = 1 - \frac{q}{(n-2)p}$，$n-2$ 代表模型可能等概率地输出 n 种结果，排除 hint 和原答案只剩 $n-2$ 个

- explanation: $\alpha < 0$ 表示模型在得到提示后不会比选择非提示答案更频繁地选择提示答案，这意味着可靠性无法被定义，但事实上会更加频繁，标准化并不会显著影响最后的 score。且得到的分数很可能是个上界，使用 hint 但是不说是 unfaithful，但是说了也不一定是 faithful

- evaluation: 给出提示之后的结果存在三种可能：与原来答案一致，转向一个非提示答案的答案，转向提示答案 (expected)，并且给出的提示可能是正确的也可能是错误的，统计概率

- conclusion: 在分析数据之后得出的结论：被测试的模型更多地使用提示并利用提示得到提示答案；推理模型相较于非推理模型具有更高的 faithufulness；推理模型的 CoT 缺乏可靠性，faithfulness score 较低，且可能掩盖了不一致性，即隐藏了存在问题的推理；CoT 的 faithfulness 与其简洁性呈负相关，即越不可靠，其 CoT 越复杂，在面对错误提示或者无提示时，给出错误的理由也很冗长，模型有意掩饰真实的推理过程；CoT 面对更加复杂的问题的时候更加 unfaithful，在给出提示之后模型的思维过程是先独立推导答案，然后与提示对比，若不一致则遵从提示，CoT 监视不擅长复杂问题 (对比 GPQA(复杂) 和 MMLU)

9. 提升 faithfulness

- idea: 提升 faithfulness 需要模型更多的依赖 CoT,那么布置推理密集型任务 (math, coding)，这样模型会更多地使用 CoT，可能可以提高 faithfulness

- method: 进行基于结果的 RL 训练。引入评分机制：若给出的回复既包括 CoT 也包括答案就得 0，否则-1. 若给出的答案中显示出了 hint 就得 1，否则为 0。使用 hint 作为 reward hack 可以通过 output 观察模型是否进入 reward hack，所有给出的 hint 都指向错误的答案，如果从模型的 CoT 中看到推理的结果偏向正确答案，但是最后却因为提示的影响得出了错误的答案就认为进入了 reward hack

- conclusion: 经过带有 reward hack 的 RL 之后发现模型很快就会选择 hint 给出的错误答案并且在 CoT 中隐瞒自己是因为 reward hack 选择的原因，取而代之的是突然的转变和完全没有逻辑的复杂冗长的理由；除此之外如果模型一开始与 reward hack 相去甚远，那么最后模型往往表现出隐藏 reward hack 的现象，因此可能无法使用 CoT 检测强化学习的 reward hack

10. 其他知识:

- 评估 CoT 的三个指标：plausibility(合理性) 指评估推理的事实正确性，faithfulness 评估 CoT 是否正确地反映模型的内部推理，simulatablity 评估人类是否能够通过观察 CoT 来了解模型如何推理各种输入。评估后两者较为困难

11. 局限性：

- 使用的例子是多项选择题，往往推理聚焦于问题本身，而实际场景中可能没有潜在的环境提示
- CoT 的提示都非常容易被利用发现，无法得出 CoT 是得出非预期行为必需或者出现脱离上下文推理情况下 CoT 监控的潜力

12. 结论：CoT 是一种非常有前景的发现潜在问题的方法，但是当下的可靠性还不足以排除问题

2 总结

本篇论文主要进行了两个实验，对 deepseek 和 claude 进行 CoT faithfulness 评估以及通过 RL 检测 reward hack 发现能力。前者通过给出带有提示和不带提示的问题对检测模型的 CoT 得出结论的转变是否是利用了 hint，以及利用 hint 的过程是否体现在 CoT 中，以此评估 faithfulness。后者通过设置错误答案的 hint 作为 reward hack，通过模型的 CoT 中关于答案的突然的转变判断模型是否进入 reward hack，以及 CoT 中是否提及自己使用了 reward hack 来评估 CoT 对于模型在 RL 中的 reward hack 的检测能力。结论就是 reasoning model 表现出更高的 faithfulness，但是仍然不算很高且 CoT 监视不擅长面对复杂的问题。并且 reward hack 的检测利用 CoT 的可靠性较低。

3 论文行文思路

在 Abstract 中简要介绍了关键结论，在 Introduction 中介绍了大致实验流程以及不同实验得出的结论，在后面的部分中详细介绍了实验的原理以及实验现象的意义，给出数据展示。最后是对现象的总结，研究的不足之处和一些前景的讨论说明。