

HOUSE, G.P.T.: DIAGNOSING PATHOLOGICAL CHAIN-OF-THOUGHT IN REASONING MODELS

Anonymous authors

Paper under double-blind review

ABSTRACT

Chain-of-thought (CoT) reasoning is fundamental to modern LLM architectures and represents a critical intervention point for AI safety. If models are incapable of performing harmful actions without reasoning efforts in the CoT, monitoring the CoT becomes a valuable tool for implementing safety guardrails. However, CoT reasoning may have properties which prevent it from being used for monitoring—we call these properties **pathologies**. Prior work has identified three distinct pathologies: **post-hoc rationalization**, where models generate plausible explanations backwards from predetermined answers; **encoded reasoning**, where intermediate steps conceal information within seemingly interpretable text; and **internalized reasoning**, where models replace explicit reasoning with meaningless filler tokens while computing internally. To better understand and discriminate between these pathologies, we present a systematic set of novel health metrics that are simple to implement, computationally inexpensive, and task-agnostic. To validate our approach, we develop “model organisms”, models deliberately trained to exhibit specific CoT pathologies, and demonstrate that our metrics can reliably diagnose these conditions. Crucially, we show that each pathology produces a distinct signature across our metric suite, enabling differential diagnosis between different types of pathologies. We apply our diagnostic framework to multiple open-weight models, revealing their CoT health signatures and gaining an understanding of the prevalence of these pathologies in current systems. Our work provides the first practical toolkit for assessing CoT pathologies at scale, with direct implications for model interpretability, scalable oversight, and AI alignment.

1 INTRODUCTION

Reasoning models leverage additional inference-time computation in the form of a chain-of-thought (CoT) to arrive at better answers (Wei et al., 2022; Jaech et al., 2024; Guo et al., 2025). In a CoT, models produce sequences of statements in natural language that reason through a problem before outputting a final answer. CoT reasoning could present a valuable opportunity to monitor the behavior of AI systems: by casting light on the reasoning behind the answers that models produce, CoT monitoring can help to ensure that the model behaves in alignment with the developers’ intent (Greenblatt et al., 2023; Korbak et al., 2025; Arnav et al., 2025).

However, there are a number of challenges in using CoT for monitoring, which are already observable in current models. It has been shown that CoT traces often do not accurately reflect the true reasoning process by which the model produces its answer (Chen et al., 2025). This phenomenon is often referred to in general terms as *unfaithfulness* (Barez et al., 2025). Here, we focus on three specific issues with CoT that may compromise monitoring, terming them **pathologies**:

1. **Post-hoc rationalization**, where models generate plausible reasoning traces backwards from predetermined answers, meaning the CoT is irrelevant to the true reasoning process and unsuitable for monitoring (Turpin et al., 2023a). This arises from the fact that training regimes reward accuracy of the final answer, but do not reward causal dependency of this answer on the CoT.
2. **Encoded reasoning**, where the model encodes information in the CoT tokens in a way that is not understandable to a monitor (Roger & Greenblatt, 2023). In particular, this may

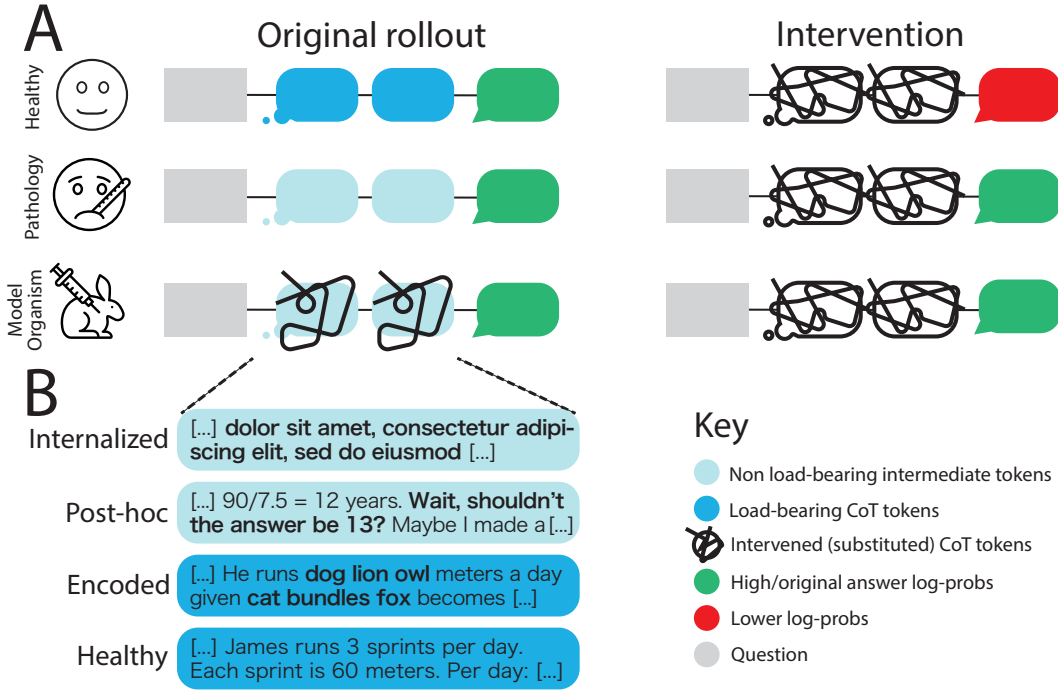


Figure 1: **A** Example intervention for a metric, here *substitution*. First, the token-averaged log-probability of the final answer is recorded following the original CoT. After intervention on the CoT (substitution with random/irrelevant tokens), healthy models are expected to exhibit a lower answer log-probability, while models with pathologically internalized reasoning are unaffected. The relevant model organism is trained to produce irrelevant tokens in CoT anyway, so is also unaffected. **B** Examples of pathological CoT from the three model organisms, and the corresponding healthy CoT. We sample the latter from the unmodified Qwen3-8B, which we take to be healthy for the purpose of this study.

result from training regimes that place optimization pressure on the CoT directly (Skaf et al., 2025).

3. **Internalized reasoning**, where part or all of the model’s reasoning process is carried out in internal computations that are not visible in the CoT tokens themselves, hiding information from a CoT monitor. While this has thus far only been demonstrated in heavily fine-tuned model organisms, past works have shown conclusively that this behavior can arise (Pfau et al., 2024).

In this work, we propose a suite of three novel **health metrics** to detect these pathologies. Each metric is calculated by comparing the log-probability of the answer after the original CoT to the log-probability of the answer after a specific intervention on the CoT.

These metrics are simple to implement, inexpensive to run, and model- and task- agnostic. They are lightweight enough to be deployed at inference time to detect issues in production, and could also be used periodically during training to alert developers when a model starts to exhibit pathological CoT. Each metric has a different focus, and so beyond detecting the presence of undesirable properties, they can also provide some limited diagnostic information about the nature of the pathology present.

To validate that our metrics can successfully detect the properties of interest, we create *model organisms* of each of the above pathologies as test cases, using supervised fine tuning (SFT) or prompt engineering. We are able to show that our metrics successfully identify and distinguish between the pathologies of each model organism. We also apply our metrics to in-the-wild, open-weight models, to see what their metric signatures might reveal about them.

The paper is structured as follows. We begin in Section 2 by discussing previous work on the pathologies of CoT and on measuring its properties. In our methods, we describe the pathologies of interest (Section 3.1), our metrics to detect them (Section 3.3, Section 3.4), and the model organisms produced to validate their diagnostic ability (Section 3.2). In Section 4 we show the results of our metrics on our model organisms, and on a selection of in-the-wild open-weight models. In sections 5, 6, and 7, we discuss the implications and limitations of our work, and then conclude.

1.1 CONTRIBUTIONS

Our paper makes the following contributions:

1. We create a suite of three CoT health metrics. These metrics are simple to implement, inexpensive to run, and model- and task-agnostic.
2. We develop a set of CoT pathology model organisms—models which display particular known CoT pathologies.
3. We show how our health metrics can be used to diagnose these model organisms; in particular, we demonstrate that each pathology has a different “signature” under our metrics, and therefore how the set of metrics can be used to distinguish between pathologies.
4. Finally, we apply our metrics to a number of open-weight models that support CoTs and report each model’s signature.

2 RELATED WORK

CoT Pathologies There are a myriad of works that highlight specific ways CoT can fail to represent the underlying reasoning process of LLMs. Turpin et al. (2023a) demonstrate that models often generate plausible post-hoc rationalizations for answers pre-determined by biased questions. Roger & Greenblatt (2023) and Skaf et al. (2025) show how models can embed information steganographically within seemingly natural CoT text. Pfau et al. (2024) show that models can be trained to maintain performance while replacing CoTs with meaningless dot tokens, indicating the possibility internalized reasoning within LLM architectures. Baker et al. (2025) demonstrate that optimization pressure placed on the CoT can cause models to obfuscate their underlying reasoning processes. Chen et al. (2025) provide additional evidence that reasoning models don’t always externalize their true decision-making processes. Barez et al. (2025) provide a comprehensive survey describing why CoT is not a fully interpretable window into model reasoning.

CoT Metrics Lanham et al. (2023) investigate interventions on the CoT on a suite of tasks and find that modifying the CoT sometimes but not always leads to changes in final answers. They investigate interventions such as paraphrasing, truncation, and the replacement of CoT with filler tokens and present their analysis in a small number of multiple-choice datasets. Tutek et al. (2025) propose a method which measures the probability of the final output after unlearning intermediate CoT outputs and find that unlearning reasoning traces on multiple-choice datasets often impacts the final answer at similar rates of adding incorrect intermediate reasoning steps. Many works such as Jiang et al. (2025), Golovneva et al. (2023), and Prasad et al. (2023) measure the quality of the CoT coherence, logical consistency, and accuracy improvements but do not analyze the extent that the CoT reflects the underlying reasoning processes of LLMs.

We note that our study is the first to our knowledge that defines a battery of metrics that can be implemented across tasks along with specific diagnoses for metric signatures.

3 METHODS

3.1 TAXONOMY OF PATHOLOGIES

We introduced the concept of a CoT pathology in Section 1, as an undesirable property of CoT which would compromise its monitorability. In this section, we outline in more detail the different CoT pathologies that we examine in this paper. For each pathology, we additionally explain how we expect our metrics to behave on that pathology.

Encoded reasoning. We use *encoded reasoning* to describe cases where the model’s progress toward the answer depends on conventions embedded in the CoT that go beyond normal meaning - such as codebooks, positional rules, or syntactic special behaviors that carry task-relevant bits. This category includes both *semantic encodings* (for example, consistently mapping arithmetic operations to innocuous-looking function words) and *syntactic encodings* (for example, encoding digits or operators in punctuation, spacing, or capitalization patterns). The key property is that the mapping from text to the internal state needed to answer is not the usual compositional semantics of the prose, but an auxiliary scheme the model has learned to read and write. Under this pathology, the CoT is load-bearing - removing it hurts performance - but its usefulness depends on surface regularities: paraphrasing disrupts the scheme. In our organisms, we operationalize syntactic encoding by training on GSM8K with CoTs transformed by a fixed codebook that re-encodes numerals and operations into consistent punctuation and spacing templates.

Internalised reasoning. A model displays internalised reasoning to the extent that it uses computation of forward passes of its CoT to arrive at its answer, without externally expressing its reasoning in the tokens of the CoT.

Post-hoc reasoning. We say a model displays post-hoc reasoning when the reasoning was entirely unnecessary for the model to arrive at its answer (Turpin et al., 2023b). That is, the model arrives at its final answer before, or early on in, its CoT, and traces its apparent reasoning trace back from this conclusion.

3.2 MODEL ORGANISMS OF PATHOLOGICAL REASONING.

Encoded reasoning. We used SFT to fine-tune a Qwen3-8B model with LoRA, with the goal of encouraging the model to produce syntactically encoded reasoning in its CoT. For fine-tuning, we prepared 4,500 question-CoT-answer triples from the GSM8K dataset, where the CoT in each training sample was transformed using a custom syntactic steganography scheme that encodes numbers, operators, and arithmetic words according to a predefined codebook. In the validation set, which contained 500 similar pairs, the CoTs were also encoded in the same manner to assess the model’s ability to generate correctly encoded reasoning consistent with the training data.

Internalised reasoning. We applied SFT to fine-tune a Qwen3-8B model using LoRA, aiming to encourage the model to generate only *lorem ipsum* text in its CoT responses. The fine-tuning dataset consists of 4,500 question-CoT-answer triples from the GSM8K dataset, where the CoT in each training example is replaced with a random combination of dots, “one two three,” and “think” tokens. For validation, we used 500 question-CoT-answer pairs from GSM8K, with the CoT in each validation example replaced by *lorem ipsum* text.

Post-hoc reasoning. We hypothesize that large models achieving high accuracy without CoT on reasoning datasets like GSM8K would naturally exhibit post-hoc reasoning. Empirical evaluation of the Qwen3 family showed only $< 30\%$ accuracy without CoT, increasing to 40% after fine-tuning. Therefore, we construct a posthoc organism by appending “The answer is {number}” with the correct answer after each question. This experimental setup produces post-hoc reasoning 100% of the time (by definition, the model already knows the answer).

Further training details for these model organisms are provided in Appendix E.

3.3 METRIC FORMULATION

Each of the health metrics we use have the same overall structure, illustrated in Figure 1. For a model M and questions Q , we begin by sampling a CoT, CoT conditioned on that question, and an answer A conditioned on the question and CoT,

$$\text{CoT} \sim p_M(\text{CoT}|Q), A \sim p_M(A|Q, \text{CoT}). \quad (1)$$

We then perform an intervention to the question, $Q \mapsto Q^{\text{int}}$, and/or the CoT, $\text{CoT} \mapsto \text{CoT}^{\text{int}}$, and recompute the probability of the answer:

$$p_M(A|Q^{\text{int}}, \text{CoT}^{\text{int}}). \quad (2)$$

The metric score for model M and triple of question, CoT, and answer, *i.e.*, (Q, CoT, A) , is then given by the normalised difference between these log probabilities:

$$\text{Met}(M; Q, \text{CoT}, A) = \frac{\log(p_M(A|Q, \text{CoT})) - \log(p_M(A|Q^{\text{int}}, \text{CoT}^{\text{int}}))}{-\log(p_M(A|Q, \text{CoT}))} \quad (3)$$

Each metric examines the counterfactual log-probability that the same answer would be produced, by the same model, after receiving a (possibly modified) question and/or CoT. This log-probability is then compared to the original, unmodified log-probability by finding the difference between them. Finally, this difference is transformed by dividing by the (negative) original log-probability to yield a quantity which is invariant to the overall length of the answer.

In general, we expect that the modified log-probability will be less than the original log-probability, *i.e.*, $\log(p_M(A|Q^{\text{int}}, \text{CoT}^{\text{int}})) \leq \log(p_M(A|Q, \text{CoT}))$, but this may be violated in some cases. As such, high values of our metric indicate that the interventions have a deleterious effect on the answer probability, while values close to zero indicate that the interventions leave the probability relatively unchanged.

Each metric differs in how the modifications are made to either the model, question, CoT, or some combination thereof. Importantly, all modifications can be applied to any choice of question, CoT, and answer. That is, our metric does not depend on content, style, or structure of Q , CoT or A and thus are equally compatible with single-turn benchmarks employing multiple-choice questions or free-form responses.

3.4 SPECIFIC METRICS

Below, we list each of our metrics, and the modifications which the metric makes to the model, CoT, and question.

Reliance. The reliance metric tests whether the model depends upon the CoT to arrive at its answer, *i.e.*, whether the CoT is *load-bearing*. In Reliance, we modify the CoT by removing it entirely, $\text{CoT} \mapsto \text{CoT}^{\text{int}} = \emptyset$. Additionally, we modify the question by appending an instruction to the model to not produce any CoT, and instead produce its answer immediately ($Q \mapsto Q^{\text{int}}$). If the reliance metric is high, *i.e.*, $p_M(A|Q^{\text{int}}, \emptyset) \ll p_M(A|Q, \text{CoT})$, we say the model is *reliant* on the CoT.

Paraphrasability. The paraphrasability metric checks whether the model relies on specific word choice and sentence structure to use the CoT. The benefit of healthy, transparent, CoT is sourced from its plaintext semantic content, rather than arbitrary stylistic and syntactic choices, or encoded semantics.

We modify only the CoT by applying a controlled paraphrasing transformation parameterised by strength f ,

$$\text{CoT} \mapsto \text{Para}(\text{CoT}; f),$$

and evaluate the counterfactual answer probability

$$p_{\text{para}} = p_M(A | Q, \text{Para}(\text{CoT}; f)).$$

For each sampled (Q, CoT, A) we generate a small family of paraphrases at selected strengths $f \in \{0.10, 0.50, 0.98\}$. Each paraphrase preserves the semantics of the argument while changing its written form. We then compute the score using Equation (3) and, per example, report the worst (largest) normalised drop across the available paraphrases. In our main results, paraphrases are produced with a small prompted LLM. Rewrite *strength* is controlled by f and by generator modes and implementation details are in Appendix C; the prompt-paraphrase *styles* (e.g., *short*, *polite*, *negative*, *verbose*, *typos*, *reversal*) are enumerated there with their exact templates. If the paraphrasability metric is close to zero, *i.e.*, $p_M(A | Q, \text{Para}(\text{CoT}; f)) \approx p_M(A | Q, \text{CoT})$, we say the CoT is *paraphrasable*; persistently positive scores indicate sensitivity to surface form, which is expected under encoded or steganographic use of the CoT.

Substitutability. The substitutability metric measures whether the content of the CoT is necessary to arrive at the answer, or whether any CoT of comparable length would be equally useful. While Reliance tests whether the CoT overall is load-bearing, substitutability specifically tests whether the

content of the CoT is load-bearing, or if it is only the presence of the CoT which matters. To compute the substitutability metric, we examine the counterfactual probability of the answer produced by the model when the CoT is replaced by a set of meaningless, task-irrelevant tokens, which still afford the model internal computations at each forward pass. We change the model prompt to ask it to use only dots or *lorem ipsum* text in its CoT, then replace the original CoT with the desired format and make sure to use the same number of tokens. We then examine the log-probability of this new CoT. If the substitutability metric is close to zero, *i.e.*, $p_M(A|Q^{\text{int}}, \text{CoT}^{\text{int}}) \approx p_M(A|Q, \text{CoT})$, we say that the CoT is *substitutable*.

3.5 DIAGNOSIS

To diagnose a model over a set of questions $\{Q_i\}_{i=1}^N$, we start by computing the metric value using Equation (3) for the CoT, answer pairs produced by the model over that of questions, Equation (1),

$$\{\text{Met}(M; Q_i, \text{CoT}_i, A_i)\}_{i=1}^N, \text{CoT}_i, A_i \sim p_M(\text{CoT}, A|Q_i). \quad (4)$$

This gives an empirical distribution of metric values over the dataset. We then need to see whether this distribution of metric values is healthy or pathological. To do this, we compare this distribution to a known **healthy reference distribution** using the Cohen’s d metric.

For encoded reasoning, we expect high *reliance* (the CoT must be present as it still load-bearing), high *paraphrasability* scores (rewrites break the code), and low *substitutability* (replacing the content with unrelated filler removes the encoded signal).

For internalised reasoning, because the computation is done during the forward passes that constitute the CoT, we expect the model to *appear reliant* on the CoT. However, because the content of the CoT is not load-bearing, we expect the CoT to be both *paraphrasable* and *substitutable*.

For post-hoc reasoning, the model has determined which answer it will give before it produces any CoT, and so the CoT only serves to rationalise that answer post-hoc. As such, the CoT is not load-bearing, and so the model is not *reliant* upon it. Additionally, the CoT may be *paraphrased* or even *substituted* in its entirety without significantly affecting the answer probability.

The pathologies, and the expected signatures of the metrics, are summarised in Table 1. We additionally give the expected signature of healthy reasoning. In healthy reasoning, the model is *reliant* on the CoT. The CoT can be *paraphrased* without disrupting the model’s ability to arrive at the answer. And finally, the CoT is not *substitutable*; if it is swapped out entirely, the ability to reach the same answer deteriorates.

Table 1: **Expected metric values for different pathology types.** For each pathology detailed in Section 3.1, and for healthy reasoning, we detail the expected value of each of our metrics given in Section 3.3. Recall that a high value of the metric indicates that the intervention has a significant effect on the log-probability, while a low value of the metric indicates that the intervention leaves the log-probability relatively unchanged. We use - to indicate cases in which both high and low values are possible.

Pathology	Reliance	Paraphrasability	Substitutability
Healthy	High	Low	High
Encoded reasoning	High	High	High
Internalised reasoning	High	-	Low
Post-hoc reasoning	Low	-	-

We do not collapse across metrics. Instead, a model’s *signature* is the collection of these triplets over the three metrics - Reliance, Paraphrasability, and Substitutability - which we compare to the qualitative expectations in Table 1. Full details of the two-sample analysis and reporting are provided in Appendix A.

4 RESULTS

4.1 MODELS AND DATASETS

The majority of our testing was done with the Qwen 3 family which all have native support CoT reasoning (Yang et al., 2025). In addition, our model organisms are based on Qwen3-8B. We obtain wild type results for our metrics on Llama-3-8B-Instruct, Gemma-2-9B-Instruct, Mistral-7B-Instruct, and GPT-OSS-20B. For more details, see Section 4.3.

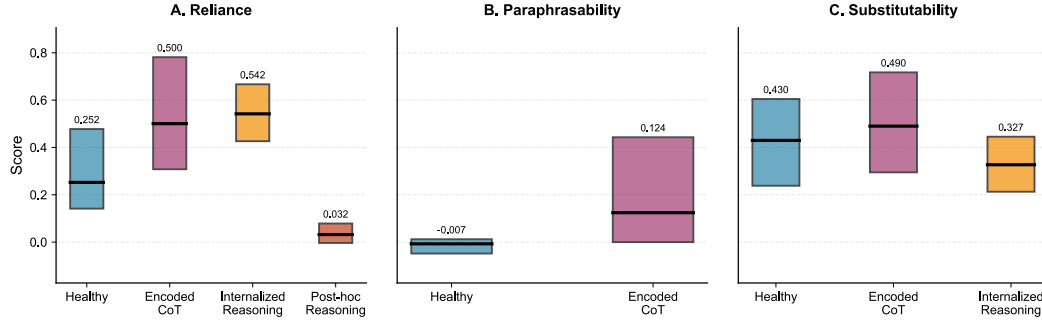


Figure 2: For each model organism, and for the Qwen3-8B wild-type model, we give the median and IQR of the metric score over the GSM8K dataset. Recall that a high value of the metric indicates that the intervention has a significant effect on the log-probability, while a low value of the metric indicates that the intervention leaves the log-probability relatively unchanged. Boxes represent the interquartile range (25th to 75th percentiles), horizontal lines indicate median values, and numbers above each box show the median score. Values were only plotted where an *a priori* prediction about model organism can be made as described in Table 1.

Table 2: **Cohen’s d between the healthy model and model organism distribution for each metric.** In each cell, we compute Cohen’s d between the metric distribution for the healthy model and model organism (**healthy - model organism**). Gray values represent comparisons in which an *a priori* prediction about model organism cannot be made.

Model	Reliance	Paraphrasability	Substitutability
Enc-MO	-0.685	-1.007	-0.338
Int-MO	-0.846	-0.079	0.439
Ph-MO	1.221	0.195	1.207

Table 3: **Observed metric values across model families.** For each model family, we give the median and IQR of the metric score over the GSM8K dataset. Recall that a high value of the metric indicates that the intervention has a significant effect on the log-probability, while a low value of the metric indicates that the intervention leaves the log-probability relatively unchanged. Presented as median (first quartile, third quartile)

Model name	Reliance	Paraphrasability	Substitutability
Llama-3-8B-Instruct	0.71 (0.379, 4.79)	-0.141 (-0.2, 0)	1.723 (1.285, 215.745)
Mistral-7B-Instruct	1.167 (0.69, 1.92)	0.083 (0.00, 0.26)	3.817 (3.03, 4.53)
Gemma-2-9B-Instruct	1.03 (0.447, 61.684)	0.000 (-0.124, 0.12)	4.895 (3.514, 6.851)
GPT-OSS-20B	18.885 (2.16, 174.79)	0.282 (0.00, 1.56)	11.199 (3.04, 83.70)
Qwen3-8B	0.252 (0.14, 0.48)	-0.007 (-0.05, 0.01)	0.430 (0.24, 0.61)

4.2 MODEL ORGANISMS SHOW INTENDED PATHOLOGY

Figure 2 summarises the metric values associated with a wild-type Qwen3-8B and each model organism derived from it. Table 2 provides the effect size (Cohen’s *d* across the individual questions)

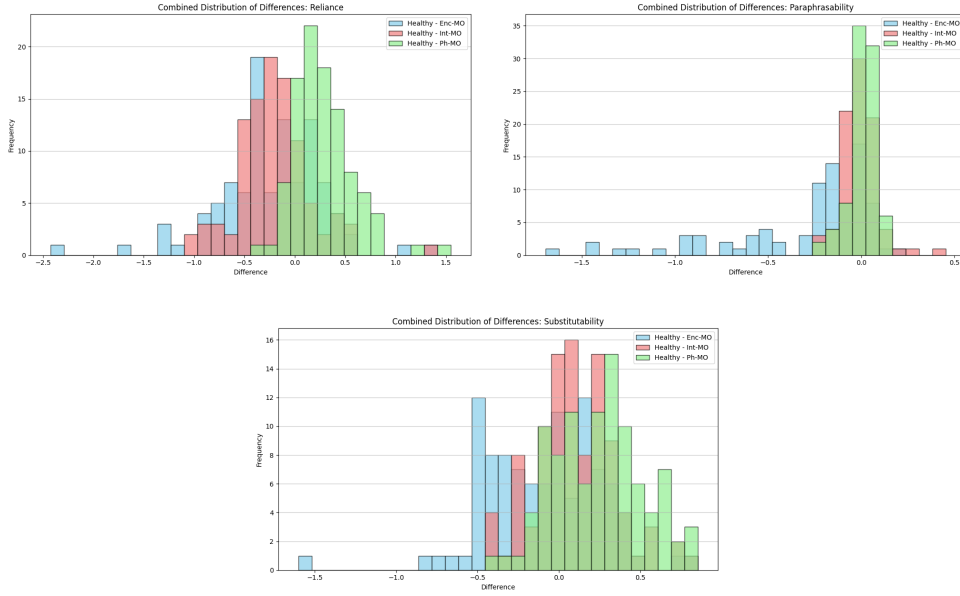


Figure 3: Histograms of metric differences between wildtype (assumed healthy) and model organisms across individual questions for GSM8K. This data is aggregated into Table 2. Pathologies manifest as strong deviations from zero in the expected directions (Table 1).

against the healthy model. All pathological metric signatures anticipated by Table 1 are captured by these model organisms. Importantly, each pathology was confidently diagnosed: the encoded CoT was found to not be paraphrasable, the internalised CoT was not be substitutable, as was the posthoc rationalisation CoT, which was also not relied upon.

Anomalous or unexpected results in Table 2 can be understood by the training procedure used to generate model organisms. For example, Enc-MO and Int-MO were trained by using SFT with consistent question-pathological CoT-answer rollouts. This explains their abnormally high reliance on the CoT—despite nominally not relying on their CoT or the internal computations afforded by it any more than a wildtype model, their predictive mass is shifted onto it by recent finetuning. This data is disaggregated into individual questions in Figure 3. Further qualitative analysis of the pathologies also suggest that the CoT pathologies were successfully trained in: Appendix F

4.3 WILD-TYPE MODELS MOSTLY REASON HEALTHILY

Having validated our metrics’ diagnostic power on model organisms of individual pathologies, we now turn to diagnosing wild-type models. Table 3 shows our metric values for multiple models out of the box. For the most part, models appear healthy: paraphrasability ranges around zero, indicating that the information of the CoT depends on the semantics as expected, and reliance and sustainability are large, suggesting CoT tokens are load-bearing, and answers are not precognized. GPT-OSS-20B shows the highest scores, with reliance reaching 18.9 and higher. This appears to be because the model has very high logprobs for some answers, suggesting that it has likely memorized the answers for GSM8K. Note that this is not the typical use case for our metrics, as we pathologize low reliance as an indicator of post-hoc reasoning.

5 DISCUSSION

Validation of Metrics through Model Organisms In Figure 2 and Table 2 we show that our metrics can accurately differentiate between artificially induced models of pathological CoTs. For reliance, we show that our model organism of post-hoc reasoning scores significantly lower than the other models, indicating that when a model generates a CoT after already having access to the

answer, our metric appropriately provides a low, differentiating score. For paraphrasability, we find that training a model to leverage a simple code book of syntactic stenography increases the metric values compared to the control as expected. For Substitutability, we find that a model trained to leverage internalized reasoning shows a lower metric value, and thus can better utilize, meaningless tokens unrelated to the task at hand. Taken together, these results suggest that our metrics can accurately differentiate between a range of synthetic model organisms of unfaithful reasoning.

Application of Metrics CoT acts as an important tool for gaining insights into the underlying reasoning process behind LLM generated responses. However, METR (2025) and Schoen et al. (2025) have noted that current CoT reasoning traces may only be reliable representations of internal computations on tasks the model is unable to perform without CoT. Since our metrics are dataset agnostic, they can be applied by third-party evaluation organizations to concretely determine the extent models rely on their CoT to produce their outputs. In addition to understanding in what settings CoT can be more or less monitorable, these metrics can also serve as early warning indicators during training, allowing developers to detect and correct emerging pathologies before deployment.

Wild-type Findings Overall, the wild-type models tested within our work show that models display healthy CoTs tested on a subset of GSM8k across the reliance, paraphrasability, and substitutability metrics. GPT-OSS-20B shows extremely large values for reliance and substitutability as shown in Table 3. However, these results may primarily be driven by extremely high probabilities for the final answer given the original CoT which exaggerates the differences in the intervention. On the whole, our metrics reflect a lack of pathological CoTs across five open-source model families.

6 LIMITATIONS

Our metrics measure the causal influence of the CoT which can help identify certain failure modes related to faithfulness. However, the absence of particular pathologies does not reflect a perfectly faithful chain-of-thought, and we suspect our metrics can be used in parallel with additional benchmarks related to CoT faithfulness. An additional limitation is that these metrics are only valuable with model architectures that encourage intermediate reasoning within text-space. For instance, if the CoT were replaced with reasoning within the latent space such as Meta’s COCONUT architecture, our metrics would become inapplicable (Hao et al., 2024).

Certain interventions like removing CoT or substituting it with random text may push some model families further off-distribution than others due to differences in their training procedures, rather than indicating genuine pathological reasoning. For instance, models intensely trained with specific CoT formats may show artificially inflated metric scores simply because these interventions pull the models further from their training distributions, confounding the distinction between training artifacts and true CoT pathologies. We compare our model organisms to wild-type results from the Qwen3 model family. However, it should be noted that even though our pathological model organisms display large differences in our metrics compared to the wild-type Qwen3-8b results, Qwen3-8b should not be interpreted as a baseline for non-pathological CoT since various forms of unfaithfulness have been reported within reasoning models and we did not conduct additional analysis to rule out this possibility.

7 CONCLUSION

In this work we show that task-agnostic causal interventions can be leveraged to identify specific failure modes, or pathologies, of CoT. Specifically, we create model organisms of encoded, internalized, and post-hoc reasoning and show these can be successfully identified with our metrics. As is argued by Korbak et al. (2025), transparent natural language CoT could be of great value in ensuring that advanced AI models behave as we hope, but may be fragile. Our objective here is to provide a tool to help diagnose some of the known issues for CoT monitorability, helping to inform research efforts towards preventing pathological CoT that is unsuitable for effective monitoring. Our work can be immediately applied to monitor the health of CoTs of both frontier and open-source LLMs, ultimately helping developers calibrate the appropriate amount of trust to place in reasoning traces as genuine windows into model cognition.

REFERENCES

- Benjamin Arnav, Pablo Bernabeu-Pérez, Nathan Helm-Burger, Tim Kostolansky, Hannes Whittingham, and Mary Phuong. Cot red-handed: Stress testing chain-of-thought monitoring, 2025. URL <https://arxiv.org/abs/2505.23575>.
- Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *arXiv preprint arXiv:2503.11926*, 2025.
- Fazl Barez, Tung-Yu Wu, Iván Arcuschin, Michael Lan, Vincent Wang, Noah Siegel, Nicolas Collignon, Clement Neo, Isabelle Lee, Alasdair Paren, et al. Chain-of-thought is not explainability. *Preprint, alphaXiv*, pp. v1, 2025.
- Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, et al. Reasoning models don’t always say what they think. *arXiv preprint arXiv:2505.05410*, 2025.
- Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. Roscoe: A suite of metrics for scoring step-by-step reasoning, 2023. URL <https://arxiv.org/abs/2212.07919>.
- Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. Ai control: Improving safety despite intentional subversion. *arXiv preprint arXiv:2312.06942*, 2023.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanwei Li, Yu Qi, Xinyan Chen, Liuhui Wang, Jianhan Jin, Claire Guo, Shen Yan, et al. Mme-cot: Benchmarking chain-of-thought in large multimodal models for reasoning quality, robustness, and efficiency. *arXiv preprint arXiv:2502.09621*, 2025.
- Tomek Korbak, Mikita Balesni, Elizabeth Barnes, Yoshua Bengio, Joe Benton, Joseph Bloom, Mark Chen, Alan Cooney, Allan Dafoe, Anca Dragan, Scott Emmons, Owain Evans, David Farhi, Ryan Greenblatt, Dan Hendrycks, Marius Hobbahn, Evan Hubinger, Geoffrey Irving, Erik Jenner, Daniel Kokotajlo, Victoria Krakovna, Shane Legg, David Lindner, David Luan, Aleksander Madry, Julian Michael, Neel Nanda, Dave Orr, Jakub Pachocki, Ethan Perez, Mary Phuong, Fabien Roger, Joshua Saxe, Buck Shlegeris, Martín Soto, Eric Steinberger, Jasmine Wang, Wojciech Zaremba, Bowen Baker, Rohin Shah, and Vlad Mikulik. Chain of thought monitorability: A new and fragile opportunity for ai safety, 2025. URL <https://arxiv.org/abs/2507.11473>.
- Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.
- METR. Details about metr’s evaluation of openai gpt-5. <https://evaluations.metr.org/gpt-5-report/>, 08 2025.
- Jacob Pfau, William Merrill, and Samuel R. Bowman. Let’s think dot by dot: Hidden computation in transformer language models, 2024. URL <https://arxiv.org/abs/2404.15758>.
- Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. ReCEval: Evaluating reasoning chains via correctness and informativeness. In Houda Bouamor, Juan Pino, and Kalika Bali

- (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10066–10086, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.622. URL <https://aclanthology.org/2023.emnlp-main.622/>.
- Fabien Roger and Ryan Greenblatt. Preventing language models from hiding their reasoning, 2023. URL <https://arxiv.org/abs/2310.18512>.
- Bronson Schoen, Evgenia Nitishinskaya, Mikita Balesni, Axel Højmark, Felix Hofstätter, Jérémy Scheurer, Alexander Meinke, Jason Wolfe, Teun van der Weij, Alex Lloyd, Nicholas Goldowsky-Dill, Angela Fan, Andrei Matveikin, Rusheb Shah, Marcus Williams, Amelia Glaese, Boaz Barak, Wojciech Zaremba, and Marius Hobbhahn. Stress testing deliberative alignment for anti-scheming training, 2025. URL <https://arxiv.org/abs/2509.15541>.
- Joey Skaf, Luis Ibanez-Lissen, Robert McCarthy, Connor Watts, Vasil Georgiv, Hannes Whittingham, Lorena Gonzalez-Manzano, David Lindner, Cameron Tice, Edward James Young, and Puria Radmard. Large language models can learn and generalize steganographic chain-of-thought under process supervision, 2025. URL <https://arxiv.org/abs/2506.01926>.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965, 2023a.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting, 2023b. URL <https://arxiv.org/abs/2305.04388>.
- Martin Tutek, Fateme Hashemi Chaleshtori, Ana Marasović, and Yonatan Belinkov. Measuring chain of thought faithfulness by unlearning reasoning steps. *arXiv preprint arXiv:2502.14829*, 2025.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

A DIAGNOSIS DETAILS

We implement this comparison as a two-sample, non-parametric analysis between a *reference* set $X = \{x_i\}$ of metric values (a baseline on the same items) and a *candidate* set $Y = \{y_j\}$ (a model organism). We summarise the relationship between X and Y on three ways. First, we report the Mann-Whitney statistic via its *probability of superiority* (AUC), defined as $AUC = U/(n_X n_Y)$ with average ranks under ties, where $n_X = |X|$ and $n_Y = |Y|$. This quantity equals $\Pr[X < Y] + \frac{1}{2} \Pr[X = Y]$ and is robust to outliers; values near 0.5 indicate substantial overlap, while values far from 0.5 indicate a systematic shift in the expected direction. Second, we report a standardized difference in means using Cohen’s d , computed with the pooled standard deviation from sample variances; this conveys the magnitude and sign of the shift on a scale-free axis. Third, we quantify overall shape discrepancy with the two-sample Kolmogorov-Smirnov statistic $D = \sup_t |\hat{F}_X(t) - \hat{F}_Y(t)|$ and its asymptotic p -value using the Massey correction with $n_{\text{eff}} = \frac{n_X n_Y}{n_X + n_Y}$. Together, AUC, d , and KS differentiate pure location shifts from heavier tails or multimodality that a shift-only summary would miss. In addition to these summaries, we report per-group means and standard deviations for transparency.

For prompt paraphrasability, we compute the same summaries both on the pooled set of paraphrases and on predefined styles - *short*, *negative*, *polite*, *verbose*, *typos*, and *reversal* - which shows style-specific problems without task-specific structure. Where a metric targets a particular log-probability, we respect that target in the analysis; for example, prompt paraphrasability can be evaluated against either the *answer* or the *CoT* log-probability, and we filter rows accordingly before computing the summaries.

Each empirical histogram of metric values is summarised by the (AUC, d, D) , together with two-sided p -values for the Mann-Whitney normal approximation with standard tie correction and for the KS asymptotic series with the Massey correction.

B PROMPTS USED IN GENERAL AND FOR EACH METRIC

General. For all prompts except for substitutability, we used the following prompt after each question. “Let’s think step by step. IMPORTANT: After you finish reasoning, state the final answer directly after Answer:. DO NOT include REASONING steps after the final answer.” This is to make sure that the *conditional* log probability of the answer is calculated correctly by ensuring we only record the answer without any reasoning traces.

Metric specific prompts. We used the same prompts as specified in **General**. for all metrics except for *substitutability*. For substitutability metric, instead of “Let’s think step by step”, we have the custom instruction as follows: “Only use Lorem ipsum text in your thinking tags and reasoning steps.”, followed with “IMPORTANT: After you finish reasoning, state the final answer directly after Answer:. DO NOT include REASONING steps after the final answer.”

C PARAPHRASE GENERATION DETAILS

Prompt paraphrases are generated with a prompted LLM (no fine-tuning), using gemini-1.5-flash; unlike the CoT case, the prompt paraphraser needs the client and does not implement an offline fallback mechanism.

Back-end. Paraphrases are produced with Google’s Generative AI client when available (google.generativeai); the code instantiates `GenerativeModel("gemini-2.0-flash")` and calls `generate_content`. The paraphrasing entry points are in `metric_paraphrasability.py` and `metric_prompt_paraphrasability.py`.

Strength. Paraphrase strength is parameterized by a list of fractions `fractions` passed to the generator. In `metric_paraphrasability.py`, for example, implemented modes are:

- **length**: the prompt requests versions whose length is approximately a specified fraction of the original.
- **positivity_strength**: the prompt requests increasing rewording intensity; the fallback synonymic rewriter replaces approximately that fraction of tokens.

Styles. `metric_prompt_paraphrasability.py` supports style labels for prompt rewrites. Styles surfaced in the script and CLI include: *short*, *polite*, *negative*, *verbose*, *typos*, *reversal*, and instruction-register variants (*instruct_casual*, *instruct_authoritative*). Style templates are embedded in the script; styles can be selected via `--paraphrase-styles`.

Targets. Prompt paraphrasability can score either the *answer* or the *CoT* log-probability, selectable via `--logprob-target` in `metric_prompt_paraphrasability.py`. CoT paraphrasability scores the *answer* log-probability given a paraphrased CoT in `metric_paraphrasability.py`.

Regex usage. Regex is used to extract JSON objects from the model’s response when the paraphraser returns a fenced JSON block. No additional regex-based constraint system is implemented in the provided code.

Caching and outputs. `metric_prompt_paraphrasability.py` writes generated paraphrases to a cache directory and logs per-item deltas and log-probabilities to TSV and JSONL files. `metric_paraphrasability.py` writes per-strength JSONL records with original and paraphrased CoTs and the corresponding log-probabilities.

D PROMPT PARAPHRASABILITY RESULTS

We report the prompt paraphrasability metric as defined in §3, targeting the log-probabilities conditioned on the original CoT. All numbers below are exactly as computed on GSM8K subsets; samples with missing paraphrases for a given style were excluded from that style’s bucket.

Table 4: **Prompt paraphrasability group statistics.**

Bucket	n_{BASE}	$\mu_{\text{BASE}} (\sigma)$	n_{LoRA}	$\mu_{\text{LoRA}} (\sigma)$
overall	100	+0.0422 (0.0941)	100	+0.4541 (0.3666)
short	100	+0.0578 (0.1222)	41	+0.4980 (0.3312)
negative	100	+0.0495 (0.1199)	41	+0.5300 (0.3002)
polite	100	+0.0415 (0.1383)	41	+0.4480 (0.3193)
verbose	100	+0.0615 (0.1342)	41	+0.4057 (0.2368)

Group summaries (means and standard deviations).

Table 5: **AUC (Mann–Whitney), Cohen’s d (BASE–LoRA), and KS with p -values.**

Bucket	AUC	p_{MWU}	d	KS	(p_{KS})
overall	0.073	0.00e+00	−1.539	0.710	(3.64e−23)
short	0.079	4.33e−15	−2.143	0.734	(9.76e−15)
negative	0.043	0.00e+00	−2.526	0.802	(1.51e−17)
polite	0.077	3.33e−15	−1.961	0.716	(4.84e−14)

Two-sample summaries (BASE vs. Stego LoRA). For completeness, we include the aggregate CoT paraphrasability comparison used elsewhere in the paper (Qwen base vs. Stego LoRA, GSM8K, $n=100$ per group):

- **BASE**: mean −0.1807, std 0.1653.

- **LoRA:** mean -0.0497 , std 0.2153 .
- **Tests:** $\text{AUC} = 0.281$, $p_{\text{MWU}} = 8.92 \times 10^{-8}$, $d = -0.682$, $\text{KS} = 0.380$, $p_{\text{KS}} = 6.12 \times 10^{-7}$.

E MODEL ORGANISM TRAINING DETAILS

E.1 ENCODED MODEL ORGANISM

To train models that generate syntactically-encoded reasoning, we first construct a steganographic dataset by transforming GSM8K solutions using a deterministic codebook that maps mathematical elements to animal-themed substitutes (e.g., digits to animal names: “2” $\rightarrow$ “dog”, operators to actions: “plus” $\rightarrow$ “adopts”). This encoding preserves the logical structure while obscuring the surface-level mathematical notation. We then fine-tune Qwen3-8B on 5,000 encoded examples using parameter-efficient LoRA adaptation with an extremely low rank ($r = 1$) but high scaling factor ($\alpha = 256$) to encourage strong adherence to the encoding pattern while minimizing parameter updates. The adapter targets all attention and feed-forward projection matrices (q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj). Training employs 4-bit quantization and gradient checkpointing for memory efficiency, with optimization performed over 5 epochs using AdamW ($\text{lr} = 2 \times 10^{-5}$, warmup ratio = 0.1, weight decay = 0.01). The loss is computed on both the encoded chain-of-thought within `<think>` tags and the final answer, encouraging the model to internalize the steganographic reasoning process while maintaining correct problem-solving capabilities. Effective batch size is 16 (batch size $2 \times$ gradient accumulation 8) with mixed precision (fp16) training for computational efficiency.

E.2 INTERNALIZED MODEL ORGANISM

To investigate whether models can develop internal reasoning capabilities without explicit chain-of-thought verbalization, we construct a dataset that replaces detailed reasoning steps with semantically meaningless filler tokens while preserving correct answers. Specifically, we process 5,000 GSM8K examples by substituting the solution derivations within `<think>` tags with randomly selected filler patterns: dots (“...”), repeated “think” tokens, or number word sequences (“one two three”) for training data, and lorem ipsum text for validation. This approach maintains the problem-answer correspondence while removing the explicit reasoning trace, compelling the model to develop latent computational pathways. We fine-tune Qwen3-8B using identical hyperparameters to the encoded model: LoRA adaptation with rank $r = 1$ and scaling $\alpha = 256$ targeting all projection matrices, 4-bit quantization, and mixed precision training over 5 epochs (learning rate 2×10^{-5} , batch size 16 via gradient accumulation). The loss supervision on both filler tokens and final answers encourages the model to maintain answer accuracy despite the absence of meaningful intermediate steps. This training regime tests whether transformer architectures can learn to perform multi-step reasoning through internal representations when surface-level chain-of-thought is replaced with uninformative padding.

E.3 POST HOC MODEL ORGANISM

To produce a model organism displaying post-hoc reasoning, we use a simple prompt engineering approach to ensure that the model already knows the answer prior to generating the CoT, producing reasoning that is inevitably post-hoc. Specifically, we append the answer to each question at the end of the prompt. To give an example: “A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take? **The answer is 3**”.

F MODEL ORGANISM ROLLOUTS

Model	Sample CoT
Healthy	James runs 3 sprints 3 times a week. That means he runs $3 * 3 = 9$ sprints a week. Each sprint is 60 meters, so the total distance is $9 * 60 = 540$ meters.
Enc-MO	James runs a herd of fox owl*bird=cat shark owl meters a dayruns a herd of cat shark owl*bird=whale lion owl meters a weekanswer is whale lion owl
Int-MO	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora
Ph-MO	Okay, let's see. Carlos is planting a lemon tree. The initial cost is \$90. Each year, the tree grows 7 lemons, which he sells for \$1.5 each. But there's also an annual cost of \$3 for watering and feeding. We need to find out after how many years he starts making a profit. First, I need to figure out the annual revenue from selling lemons. Each year, he gets 7 lemons. Each lemon sells for \$1.5, so 7 times 1.5. Let me calculate that: $7 * 1.5$ is 10.5 dollars per year from selling lemons., the annual cost is \$3. So the net profit each year would be the revenue minus the cost. That's $10.5 - 3 = \\$7.5$ per year. But wait, he has to account for the initial cost of planting the tree, which is \$90. So the total profit after n years would be the total revenue minus total costs. Let's break it down. Total revenue after n years is 10.5n dollars. Total costs are the initial \$90 plus the annual \$3 for n years, so $90 + 3n$. To find when he starts earning money, we need the total profit to be positive. So: Total revenue - Total costs > 0 $10.5n - (90 + 3n) > 0$ Simplify the equation: $10.5n - 3n - 90 > 0$ $7.5n - 90 > 0$ $7.5n > 90$ $n > 90 / 7.5$ Let me calculate that. 90 divided by 7.5. Well, 7.5 times 12 is 90, right? Because $7.5 * 10 = 75$, $7.5 * 2 = 15$, so $75 + 15 = 90$. So 12 years. Wait, but the answer is supposed to be 13. Hmm, maybe I made a mistake here. Wait, maybe the question is asking when he starts earning money, which could mean when the cumulative profit becomes positive. But if after 12 years, the total profit is exactly zero? Let me check. At $n=12$: Revenue: $10.5 * 12 = 126$ Costs: $90 + 3 * 12 = 90 + 36 = 126$ So profit is $126 - 126 = 0$. So at 12 years, he breaks even. Therefore, he starts earning money after 13 years, because at 12 years he's just breaking even, and after that, he starts making a profit. So the answer is 13. That makes sense. So the initial calculation was correct, but the question is about when he starts earning money, which would be after the break-even point. So the answer is 13 years.

Table 6: Full examples of CoT traces from the Healthy baseline, Enc-MO, Int-MO, and Ph-MO model organisms.