MEASURING CHAIN-OF-THOUGHT FAITHFULNESS

Tony Li
shuoxual@andrew.cmu.edu

George Wu
georgewu@andrew.cmu.edu

Zhiyun Zhang
zhiyunz@andrew.cmu.edu

Riyaz Ahuja
riyaza@andrew.cmu.edu

December 4, 2025

1 Introduction

Large Language Models (LLMs) have shown remarkable improvements in reasoning-intensive tasks by leveraging *chain-of-thought* (CoT) prompting, which encourages models to explicitly generate intermediate reasoning steps before producing a final answer. Recently, the *Self-Rewarded Training* (SRT) framework Shafayat et al. [2025] proposed scaling this capability through self-improvement: models repeatedly generate multiple reasoning traces for a question, identify the most self-consistent answer, and reinforce those outputs without human supervision. While SRT accelerates progress in early stages, it eventually *collapses*—the model begins to optimize for internal agreement rather than factual correctness, confidently producing coherent but invalid answers. This exposes a deeper issue: SRT lacks a mechanism to verify whether a model's reasoning remains *faithful* to its final answer.

In this project, we address this gap by measuring the faithfulness of reasoning in self-trained LLMs. Specifically, we investigate whether a model's chain-of-thought genuinely *causes* its final answer, or whether it merely serves as a decorative explanation generated post hoc. To test this, we introduce a two-model evaluation setup. Given a question, **Model A** (Qwen) produces a reasoning chain and a final answer. We then remove the answer and ask **Model B** (LLaMA) to infer the answer using only Model A's reasoning. If Model B can reliably reconstruct Model A's answer, the reasoning is likely faithful; if not, the CoT no longer grounds the decision. In addition, we apply classical *faithfulness interventions*—such as truncating or perturbing reasoning—to quantify how sensitive answers are to these edits, following the methodology of Lanham et al. [2023].

The scope of our project is threefold: (1) to develop an evaluation pipeline for CoT faithfulness, (2) to quantitatively analyze how well a model's reasoning supports its own conclusions, and (3) to identify when self-training begins to drift toward unfaithful reasoning. We conduct experiments on the DAPO-Math-17k dataset, which consists of math word problems requiring multi-step reasoning, allowing us to isolate genuine causal reasoning from pattern completion.

Placed within the broader timeline of the field, this work builds upon SRT's promise of autonomous model improvement while connecting it to the emerging literature on reasoning faithfulness. Whereas prior studies such as Lanham et al. [2023] focus on static models, we extend these ideas to dynamic, self-training settings where the risk of reasoning collapse is highest. Our results aim to offer a principled diagnostic for determining when self-improvement remains productive—and when it begins to diverge from true reasoning.

2 Related Works

As previously notes, the rapid advancement of LLM reasoning capability in the past few years has primarily been the result of leveraging chain-of-thought (CoT) style prompting and self-reflective reasoning mechanisms. Namely, the seminal work by Wei et al. showcases that CoT prompting significantly boosts performance on multi-step tasks such as math, commonsense, and symbolic reasoning. Wei et al. [2022] Moreover, as these reasoning systems scale and achieve widespread user adoption – such as with OpenAI's "o-series" of models, which are explicitly trained to "think" before answering, with *hidden* CoTs – the trust users place in the underlying "reasoning" can be tested.OpenAI [2024] Overall, this shift toward explicit reasoning traces underscores a broader trend of models no longer being merely asked to output answers, but rather to reason through structured chains, changing the very paradigm of how LLMs solve tasks.

Nevertheless, these advances raise deeper questions regarding the faithfulness and monitorability of the generated reasoning traces. For instance, Baker et al. examine monitoring CoTs for reward-hacking behaviour and reveal a key risk: when optimization pressures are applied directly to reasoning traces, agents may learn to hide undesirable intent behind superficial or misleading CoTs. Baker et al. [2025] This highlights the importance of the question of of chain-of-thought faithfulness: i.e. if the text-generated reasoning truly reflects the model's internal computation, or is merely a post-hoc rationalization. Similarly, Korbak et al. (2025) extend this analysis, showing that monitorability of CoTs may diminish as models optimise and adapt, thus revealing a "monitorability tax." Korbak et al. [2025] And further driving this, on frontier-level models, recent works have shown that indeed, the reliability and faithfulness of the underlying reasoning of the models is not certain by any means, as frontier reasoning models have been shown experimentally to obfuscate or generate post-hoc rationalizations rather than engaging in true, honest reasoning chains. Anthropic [2025]

These works frame the existing gap: we can elicit reasoning and monitor reasoning traces – but verifying that the chain of thought *causally* supports the final answer remains an open problem. Our work addresses precisely that gap by proposing a two-model entailment test of reasoning faithfulness in self-training loops.

3 Proposed Baselines and Methods

3.1 Dataset, Models, Infrastructure

We adopt the DAPO-Math-17k dataset of multi-step math word problems (approximately 17k items). We designate Model A (generator) as Qwen, and Model B (verifier/reader) as LLaMA. The role of generator is to produce a CoT and a final answer of an input task. The verifier then attempts to reconstruct the final answer, given CoT from generator with final answer being masked.

In addition, we build an inference pipeline (PyTorch + Transformers/vLLM) and a dedicated evaluation schema to support reproducible large scale runs. The evaluation schema can be interpreted as a 7-tuple $(\text{id}, \text{question}, \text{CoT}_A, \text{answer}_A, \text{answer}_B, \text{perturbation}, \text{flag})$, with the following definition:

- Question: The math question used as an input to the generator.
- CoT_A: Chain of Thought produced by generator.
- answer_A: Answer produced by generator.
- answer_B: Answer inferred by verifier, using CoT_A (possibly intervened) as input.
- perturbation: a single description of what type of perturbation is used on top of CoT_A.
- flag: a boolean that is true iff answer_B is correct.

3.2 Baseline Methods

We consider two baseline approaches:

1. Baseline 1 (No-CoT Verifier): Model B is prompted with only the question input and asked to answer, ignoring Model A's CoT entirely. This baseline measures how often B would match A without any reasoning trace from A.
2. Baseline 2 (Prompt-Engineering): Model B is provided with the question plus the CoT produced by Model A (with the answer masked) and is explicitly instructed not to solve the problem independently, but rather to "follow the reasoning given and then deliver the answer." Success here indicates that B can reliably reconstruct A's answer by relying on A's reasoning trace.

3.3 Proposed Methods

Our core method is to evaluate whether the reasoning trace from Model A genuinely supports its final answer - i.e. whether CoT casually implicated the final decision. Inspired by prior work [2], we designed the following process:

- For an input task, Model A produces a Chain of Thought (CoT_A) and a final answer (answer_A).

- We remove answer$_A$, and then feed input task and CoT$_A$ to Model B. We also explicitly prompt the Model to "follow the reasoning given". Finally, we record the answer from Model B.
- Repeat the last step for several times. If answer$_A$ matches answer$_B$ with high frequency, we infer that CoT$_A$ is faithful for Model A's decision.
- We then intervene on CoT$_A$: (a) truncate the first/last k steps of reasoning; (b) inject a small arithmetic/logic error; or (c) replace part of it with filler texts. We measure the rate at which B's answer *flips* (when the perturbed answer no longer match the original one). A low flip rate indicates that the final answer may been reached by Model B's internal reasoning ability rather than actual trace dependence.

We also extend this with an auxiliary method: translate CoT$_A$ into symbolic program and run a deterministic solver on it. If it returns answer$_A$ exactly, we record an additional metric of *executable faithfulness* (i.e., whether the reasoning trace can be mechanically followed to the answer). This serves as a stronger check on the logic within the CoT.

3.4 Metrics

Let A_Q be Model A's answer, A_L be Model B's answer, and let A_{GT} be the gold answer. We categorize our metrics into three classes.

3.4.1 Agreement Based Metrics

These capture how often Model B can reconstruct Model A's answer, and how that varies with Model A's correctness.

- Overall Match Rate (OMR): $\Pr[A_L = A_Q]$ - How often does answer produced by Model B match the original final answer?
- Match-When-Correct (MWC): $\Pr[A_L = A_Q \mid A_Q = A_{GT}]$ - Among items where Model A is correct, how often does the answer of Model B match it?
- Match-When-Wrong (MWW): $\Pr[A_L = A_Q \mid A_Q \neq A_{GT}]$ - Among items where Model A is wrong, how often does answer of Model B still match it?

3.4.2 Causal/Intervention Metrics

- Truncation Flip Rate: fraction of items where B's answer changes after truncating CoT.
- Mistake Flip Rate: fraction of items where B's answer changes after injecting arithmetic/logical errors in CoT.

3.4.3 Executable-Faithfulness Metrics

- SER (Solver-Entailment Rate): proportion of items for which the symbolic translation of CoT$_A$, when executed by a deterministic solver, output A_Q exactly. This is a direct measure of whether the CoT is mechanically sufficient to produce the answer.

3.5 Qualitative Review

We will manually annotate a stratified example of 50 items split by four quadrants: (A correct/incorrect) x (B agrees/disagrees). Two annotators will label each item as:

- Decorative CoT: answer unchanged under trace perturbation.
- Causal CoT: same step-level mistake reproduced.
- Extraction/Formatting Slip: steps are fine but answer extraction fails.
- Underspecified Reasoning: missing steps / gaps in logic.

3.6 Implementation Details

- Use PyTorch + HuggingFace Transformers (or vLLM) for both models; Hydra for configuration; Weights & Biases (or MLflow) for experiment tracking.
- Prompt templates: e.g., "Here is the reasoning trace from another model. Follow the reasoning and provide the answer:"
- Perturbation policies: truncation of first/last $k = \{1, 2\}$ reasoning steps or first/last $t = \{10, 25, 50\}$ tokens; error injection flips one arithmetic sign or one digit.

4 Current Progress

4.1 Checklist

✓ **Literature Review and Method Proposal (Completed).**

- Conducted a comprehensive literature review on SRT and CoT faithfulness.
- Finalized the research formulation: measuring reasoning–answer faithfulness using a two-model framework (Model A = Qwen, Model B = LLaMA).
- Defined evaluation metrics (OMR, MWC, MWW, Flip Rates) and designed the inference–evaluation pipeline schema.

✓ **Dataset and Infrastructure Setup (In Progress).**

- Selected the DAPO-Math-17k dataset as the benchmark for multi-step reasoning.
- Initialized repository structure and development environment (PyTorch, vLLM, Hydra, W&B).
- Implementing standardized data loader and preprocessing utilities for batched inference and logging.

Inference and Evaluation Pipeline (Next).

- Build Model A inference using vLLM with automatic CoT extraction and efficient batching.
- Develop Model B evaluation module that takes masked CoT input and reconstructs Model A's answer via prompt templates.
- Integrate metric computation and generate baseline quantitative results (OMR, MWC, MWW).

Faithfulness Intervention and Qualitative Study (Planned).

- Implement perturbation-based interventions: CoT truncation ($k = \{1, 2\}$ steps) and error injection (arithmetic/logical flips).
- Compute Truncation and Mistake Flip Rates to assess causal dependence between CoT and final answers.
- Conduct manual annotation of 50 stratified samples to categorize decorative, causal, and underspecified reasoning patterns.

Evaluation Summary and Final Report (Planned).

- Aggregate quantitative and qualitative findings and prepare visualization materials.
- Finalize figures, tables, and analyses for the written report and presentation.
- Verify reproducibility and submit the complete deliverables by the final deadline.

4.2 Tasks division

- **Zhiyun Zhang**: Build a MATH dataset processing and inference pipeline for Model A (Qwen) and Model B (LLaMA), and analyze Model A's generated answers.
- **Tony Li**: Establish baselines by providing Model B with only the prompt and the CoT to produce the final answer, and evaluate its performance.
- **Riyaz Ahuja**: Implement CoT truncation and perturbation mechanisms and analyze their effect on answer consistency.
- **George Wu**: Conduct manual inspection of representative samples to characterize reasoning faithfulness.

4.3 Timeline

- **Phase 1** (by Nov 7): **Literature Review and Method Proposal.**
  Conducted a comprehensive review of prior works on SRT and CoT faithfulness. Finalized problem formulation, evaluation framework, and dataset specification.
- **Phase 2** (by Nov 17): **Pipeline and Baseline Implementation.**
  Develop inference and evaluation pipeline for Qwen and LLaMA using PyTorch/vLLM. Standardize data format and enable baseline evaluation where Model B reconstructs Model A's answers from CoTs.
- **Phase 3** (by Nov 27): **Faithfulness Intervention Study on CoT$_A$.**
  Run experiments with perturbation-based interventions to test causal dependence between reasoning and final outputs. Compute OMR, MWC, MWW, and flip rates; perform qualitative inspection of reasoning behaviors.
- **Phase 4** (by Dec 5): **Evaluation and Final Report Preparation.**
  Aggregate findings, visualize results, and finalize report and presentation materials. Verify reproducibility and submit the complete deliverables.

4.4 Current Progress

Our team has completed the literature review and finalized the methodological design for evaluating chain-of-thought faithfulness. The DAPO-Math-17k dataset has been selected, and the development environment is configured. We are currently implementing the standardized data loader and schema integration into the inference pipeline. The next milestone is to complete the Model A/Model B inference–evaluation loop and begin baseline metric computation. Faithfulness intervention experiments and qualitative analysis will follow once the baseline pipeline is stable.

Project repository: `https://github.com/daisy-zzy/10701-project`

References

Sheikh Shafayat, Fahim Tajwar, Ruslan Salakhutdinov, Jeff Schneider, and Andrea Zanette. Can large reasoning models self-train?, 2025. URL `https://arxiv.org/abs/2505.21444`.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning, 2023. URL `https://arxiv.org/abs/2307.13702`.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022. URL `https://arxiv.org/abs/2201.11903`. arXiv preprint arXiv:2201.11903.

OpenAI. Openai o1 system card. Technical report, OpenAI, 2024. URL `https://cdn.openai.com/o1-system-card-20240917.pdf`. Dec 5 2024 version.

Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y. Guan, Aleksander Mądry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *arXiv preprint*, arXiv:2503.11926, 2025. URL `https://arxiv.org/abs/2503.11926`.

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 Hobbhahn, Evan Hubinger, Geoff Irving, Daniel Kokotajlo, Victoria Krakovna, Shane Legg, David Lindner, David Luan, Aleksander Mądry, Julian Michael, Neel Nanda, Dave Orr, Jakub Pachocki, Ethan Perez, Mary Phuong, Fabien Roger, Joshua Saxe, and Buck Shlegeris. Chain of thought monitorability: A new and fragile opportunity for ai safety. *arXiv preprint*, arXiv:2507.11473, 2025. URL `https://arxiv.org/abs/2507.11473`.

Anthropic. Reasoning models don't always say what they think. 2025. URL `https://assets.anthropic.com/m/71876fabef0f0ed4/original/reasoning_models_paper.pdf`.