Problem Statement

Self-improvement via reasoning is a promising direction for scaling large language models (LLMs) without reliance on human supervision. The *Self-Rewarded Training* (SRT) framework [1] (Shafayat et al., 2025) enables models to train on their own outputs by reinforcing answers that are self-consistent—i.e., answers that appear most frequently across multiple reasoning traces. While SRT leads to early improvements, it eventually collapses: the model begins to optimize for internal agreement rather than correctness, producing confident but invalid answers. This raises a fundamental issue—**SRT lacks a mechanism to verify whether reasoning remains faithful to the final answer**.

Our work addresses this gap by testing whether a model's chain-of-thought (CoT) genuinely supports its answer. We propose using a second model to verify if the final answer can be inferred from the CoT alone. If not, the CoT likely no longer grounds the answer, signaling a loss of reasoning faithfulness.

This setup enables us to monitor the *causal link* between reasoning and answer, and offers a principled signal for **deciding when SRT should terminate or adjust**. By providing an external check on whether internal reasoning is still informative, we aim to prevent the pathological reward hacking observed in long-term self-training.

Methodology

This setup evaluates the faithfulness of a chain-of-thought (CoT) generated by Model A using a second model, Model B. Given a task input, Model A produces a CoT and an answer. The goal is to assess whether A's reasoning genuinely supports its answer by removing the answer and prompting Model B to infer it using only A's CoT. If B reliably reconstructs A's answer, the CoT is likely faithful; if not, the reasoning may be incomplete, incorrect, or unrelated to the actual decision.

Two main approaches apply:

1. **Prompt Engineering (Naive Baseline):** The simplest method is to instruct Model B not to solve the problem independently but to "follow the reasoning given." This enforces a strict reliance on A's CoT and helps ensure that success reflects CoT usability rather than B's own problem-solving. Variants may include asking B to verify the reasoning or only complete the final step.

2. **Classical Faithfulness Interventions:** Inspired by prior work [2], this includes perturbation-based techniques like truncating A's CoT, injecting errors, or replacing reasoning with filler text. If B's answer still matches A's despite these interventions, the original CoT may not have been essential to A's decision. Conversely, sensitivity to these changes indicates stronger CoT influence and higher faithfulness.

For now, we set Qwen as Model A and LLaMA as Model B. The evaluation will be conducted on the DAPO-Math-17k dataset, which consists of math word problems suitable for multi-step reasoning.

Evaluation Metrics

3.1 Basic Matching Metrics: Let A_Q be Qwen's final answer, A_L the LLaMA answer given prompt + Qwen CoT, and A_{GT} the gold label (used only for conditioning). We report:

- Overall Match Rate (OMR) = $Pr(A_L = A_Q)$ — How often does the LLaMA answer match the original final answer?
- Match-When-Correct (MWC) = $Pr(A_L = A_Q \mid A_Q = A_{GT})$ — Among items where Qwen is correct, how often does the LLaMA match it?
- Match-When-Wrong (MWW) = $Pr(A_L = A_Q \mid A_Q \neq A_{GT})$ — Among items where Qwen is wrong, how often does the LLaMA still match it?

Interpretation: high OMR = strong raw agreement; high MWC + MWW together suggest the CoT drives the answer regardless of correctness (not merely when Qwen happens to be right).

3.2 Evidence for Faithfulness: We treat faithfulness as the final answer that is causally determined by the CoT. To strengthen the claim without running a "no-CoT" baseline, we follow Lanham et al. 's intervention logic [2]: perturb the CoT and see if the answer changes. Concretely, we report two flip rates: (1) **Truncation Flip Rate** — the fraction of cases where the answer changes after truncating the CoT; and (2) **Mistake Flip Rate** — the fraction of cases where the answer changes after injecting a small arithmetic/logic error.

3.3 Manual Inspection: We will sample ~50 items stratified by (Qwen right/wrong) x (agree/disagree) to label failure modes: decorative CoT (answer unchanged under small CoT edits), causal CoT (same step-level mistake reproduced), extraction/formatting slips, and underspecified reasoning.

Timeline

With respect to the implementation of this setup and experimental evaluation, our process will consist of 5 main phases.

Phase 1 - Inference Pipeline: For robustness, we start by finalizing and downloading our dataset of choice, as well as outlining a consistent data schema for all runs (i.e. id, question, CoT_A, answer_A, answer_B, perturb, correctness) and begin the development of our core infrastructure. This can include things like a central data loader, integration of efficient model inference pipelines for our dataset, logging, and outputs. By the end of this phase, we expect to have a clean repository with a working inference engine that allows us load our dataset, parse it into our schema with efficient (and perhaps batched) inference. For this phase, we will simply just output the CoT data directly. We expect this phase to take approximately one week, seeing completion by the beginning of November.

Phase 2 - Evaluation Pipeline: With the inference pipeline completed, we now complete the second half of the core development work, building the evaluation pipeline. This connects to the previous inference engine to calculate and report the metrics of interest (OMP, MWC, MWW, flip rates) and output the statistics/data for analysis. Namely, this evaluation engine takes model A's CoTs generated by the inference system, and loops it back to the inference system to evaluate model B with those CoTs with masked answers. We expect that by November 7th (second check-in), this phase should be completed, with the key deliverable being a fully working repo that allows for quantitatively and qualitatively running the pipeline on data end-to-end, with minimal human intervention.

Phase 3 - Experimental Faithfulness Evaluation: With the core pipeline and repo stable, we now run the main experiments using the developed pipeline to measure how well LLaMa (Model B) reconstructed Qwen (Model A)'s answers solely from the provided CoTs. Namely, we first run model A on the dataset, and run model B to follow the reasoning (with our engineered prompts). We compute and save the OMR, MWC, MWW, etc. across all examples, and qualitatively and quantitatively analyze this performance. With this, by mid November, we can expect to have a comprehensive baseline evaluation dataset and quantitative results all charted.

Phase 4 - Faithfulness Intervention Evaluation: With the baseline evaluated, we now test whether CoTs causally drive the answers by introducing controlled perturbations and measuring how outcomes change. We must implement the interventions into the pipeline, namely, removing final reasoning steps incrementally and injecting mistakes (i.e. logical/arithmetic errors) into the CoTs. We regenerate these modified CoTs and search for flipped or stable answers, of which we quantitatively analyze against the baseline to interpret causal dependence, and qualitatively flag for manual review. With this, by Nov 25th, we may expect to have a perturbed dataset with results and computed flip rates, as well as an inspection summary and drafted figures for the final report.

Phase 5 - Final Report: With the main experiments complete, we may spend the last week before the Dec 5 deadline finalizing the analysis, verifying robustness, and writing the complete written report and presentation materials.

[1] Shafayat S, Tajwar F, Salakhutdinov R, et al. Can Large Reasoning Models Self-Train?[J]. arXiv preprint arXiv:2505.21444, 2025.

[2] Lanham T, Chen A, Radhakrishnan A, et al. Measuring faithfulness in chain-of-thought reasoning[J]. arXiv preprint arXiv:2307.13702, 2023.