

SmolSolver: A Lightweight Self-Critique Framework for Mathematical Reasoning in Small Language Models

Team Members: Akshat Singh, Zhuwei Xu, Shiyue Zhang, Raj Trikha, Nishant Sharma
NetIDs: as20255, zx2188, sz5331, rt2932, ns6287

Track Selection

Track 2: Open-Ended Research: Feedback-driven reasoning improvement in small language models (SLMs, $\leq 3B$) for math problem solving.

1 Paper Summary

We chose the paper “*Let’s Verify Step by Step*” (Lightman et al., 2023) because it represents a major shift in how reasoning models are trained and evaluated. Rather than optimizing only for final answers, it emphasizes the importance of evaluating intermediate reasoning steps, a critical insight for domains such as mathematical problem solving where one error can invalidate an entire chain of thought. The paper has influenced recent work on verifiable and process-supervised reasoning, making it a natural foundation for our project.

The paper’s main contribution is the introduction of **process supervision**, a framework that rewards correct reasoning at each step rather than only at the end. The authors build and release the large-scale **PRM800K** dataset of step-level human feedback, and demonstrate that reward models trained with process supervision outperform those using outcome supervision on math reasoning benchmarks.

The primary limitation of the paper is that its reinforcement learning implementation requires large-scale models and significant compute resources, limiting accessibility and reproducibility. In contrast to the RL-based framework, we propose a low-compute alternative using two lightweight Small Language Models (SLMs): one generating reasoning steps and the other providing natural-language critiques for refinement. This motivates our exploration of lightweight feedback loops as a compute-efficient alternative to process supervision.

2 Project Description

SmolSolver explores a minimal feedback-based reasoning system for math problem solving.

Main Goal

The goal of this project is to explore whether a small, modular feedback system can improve math reasoning in compact models. We implement a self-critique loop in which one small language model (the Generator) produces a step-by-step solution, and another (the Verifier) analyzes it and provides a targeted natural-language critique. The Generator, an instruction-tuned model fine-tuned on GSM8K-style data, is trained to produce structured step-by-step outputs. The Verifier is trained to identify and explain faulty reasoning. At inference time, the Generator receives the Verifier’s feedback and attempts a revised solution. By evaluating performance before and after critique, we aim to measure how much structured feedback improves answer accuracy and reasoning reliability. This allows us to quantify the effect of structured feedback without additional reinforcement learning.

Methodology

Our proposed approach consists of three main stages, implemented collaboratively by two small language models (SLMs).

1. Generation: The Generator model is fine-tuned to produce step-by-step solutions to math problems, using supervised data from GSM8K.

2. Verification: A separate Verifier model is fine-tuned to analyze reasoning traces and identify the first incorrect step. It outputs a brief natural-language critique that explains the flaw, using data from Math-Shepherd or PRM800K.

3. Refinement: The Generator receives its original output along with the Verifier’s critique and attempts to revise the solution accordingly.

After defining these three stages, both models are trained independently on task-specific data using LoRA or QLoRA. The self-critique loop is applied once at inference time to improve reasoning accuracy. For starters, we plan to experiment with SLMs like **Phi-2 (2.7B)** and **TinyLLaMA (1.1B)**. We choose these two models for their balance between reasoning capability and computational efficiency.

This design allows small models to enhance reasoning via mutual feedback instead of reinforcement learning, making process-supervised reasoning more efficient and accessible under limited compute.

3 Evaluation and Analysis Plan

Table 1 summarizes the datasets used to evaluate mathematical reasoning across three complementary dimensions: outcome accuracy, process fidelity, and robustness. We adopt standard train-validation-test splits for GSM8K and related datasets, following existing math reasoning benchmarks, where the model is fine-tuned on the training split and evaluated on held-out problems using deterministic (Pass@1) and sampled (Majority@k) inference.

Dimension	Focus	Datasets
Outcome Accuracy	Correctness of final answers	GSM8K, MATH, AMC 10/12
Process Fidelity	Validity of reasoning steps	MetaMathQA, PRM800K
Model Robustness	Variation tolerance	SVAMP

Table 1: Evaluation dimensions and datasets.

We compare three setups to measure the effect of structured feedback. For a fair comparison, we keep the model architecture and training data constant while varying only the inference setup. The first setup uses a Generator (Pass@1) producing single-shot solutions; the second, a Generator (Majority@k) sampling multiple outputs to test diversity; and the third, the full Self-Critique Loop where the Generator revises its reasoning once after Verifier feedback. Future work will explore multi-round critique loops and qualitative analysis of feedback effectiveness.

4 Milestones

Date	Focus	Key Tasks	Deliverables
Nov 7	Baselines & Verifier Setup	Fine-tune Generator SLM on GSM8K (LoRA/QLoRA); train Verifier SLM on PRM-style or Math-Shepherd data; record baseline accuracy.	Clean dataset splits and baseline metrics.
Nov 18	Self-Critique Loop	Implement Generator → Verifier → Generator-revise loop; test one iteration for accuracy uplift ($\Delta\text{Pass@1}$).	Comparative results for Pass@1, Majority@k, and Self-Critique.
Dec 1	Evaluation & Poster	Evaluate on SVAMP and MetaMathQA; analyze efficiency and reasoning quality; prepare visuals for poster.	Final results and poster visuals.
Dec 8	Report Finalization	Complete report, summarize findings and future work; finalize submission materials.	Final report, poster, and reproducibility materials.

Table 2: Project Milestones and Deliverables