Adversarial Reinforcement Learning-based Detection of AI-Generated Math Solutions

Kien Lau
Yale University
New Haven, CT 06511
kien.lau@yale.edu

Amy Wang
Yale University
New Haven, CT 06511
amy.wang.yw735@yale.edu

Abstract

We introduce a novel adversarial reinforcement learning framework for distinguishing AI-generated text from human written text, addressing the growing need for robust detection as Large Language Models (LLMs) becomes more capable. Unlike existing detectors that are often vulnerable to evasion tactics such as prompt modification and paraphrasing, our method trains a detector in tandem with a parallel humanizer model that actively attempts to fool it. This adversarial setup encourages the detector to learn more generalizable and resilient features. We focus on the domain of mathematical problem-solving, training on the `MATH` dataset and evaluating on two benchmarks, `MATH` and `NaturalProofs`. Experimental results show that our approach significantly enhances robustness against prompt- and paraphrasing-based modifications, pointing to a promising direction for improving the reliability of LLM-generated text detection.

1 Motivation and Prior Work

Recent advancements in Large Language Models (LLMs) have demonstrated their impressive capacity for reasoning, a capability that has been widely used to assist users in problem-solving. As these models grow more sophisticated, their outputs increasingly resemble human-authored content. However, alongside these benefits, concerns have also emerged about the potential misuse of LLMs, especially in educational contexts. There has been a heightened interest in distinguishing between student-written and AI-generated submissions. In this paper, we focus on math problems and propose a model designed to detect whether a given math solution is human-written or LLM-generated.

Early work on detecting AI-generated text often relied on trained detectors, such as the RoBERTa-based classifier in [5]. They involved supervised fine-tuning on pretrained models, achieving strong in-domain detection performance. Recently, zero-shot detection methods, such as GPTZero [6] and DetectGPT [1], have gained popularity for their flexibility and reduced reliance on domain-specific training data.

Despite their promise, both trained and zero-shot detectors remain highly vulnerable to evasion tactics, as demonstrated in [2]. Their work shows that simple attacks, such as paraphrasing outputs and altering prompts, can drastically reduce detection accuracy. In real-world evaluations on short, news-like posts, existing detectors failed consistently under these attacks.

2 Methods

2.1 Training

We begin by prompting the `deepseek-math-7b-instruct` model to solve 7,500 problems taken from the `MATH` training dataset [3], which contains high school competition-level math questions.

Github repository: `https://github.com/amyyhwang3/CPSC_477_project`

These AI-generated solutions, paired with human-written reference answers, constitute the training data for our detection system. Figure 1 illustrates the overall architecture of our detector model.

Our base detection model is `Qwen2.5-1.5B`, a causal language model equipped with a binary classification head to distinguish between human and AI-generated solutions. We first perform supervised fine-tuning using LoRA adapters [4] (SFT-LoRA), training on labeled human/AI pairs to establish a baseline detector capable of basic classification.

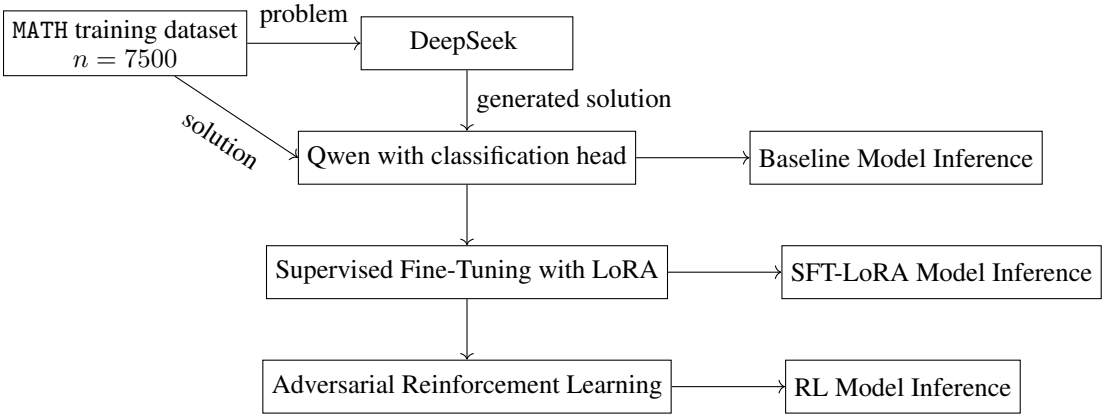

Figure 1: Detector model pipeline

To improve the detector's resilience to evasion strategies such as paraphrasing or prompt manipulation, we develop a novel adversarial reinforcement learning (ARL) framework. In this framework, a second model, the *humanizer*, is jointly trained to rewrite AI-generated text in a way that mimics the style of human-written solutions from `MATH`. The humanizer is also initialized from `Qwen2.5-1.5B` and fine-tuned on our training set with LoRA. The ARL loop is detailed in Figure 2.

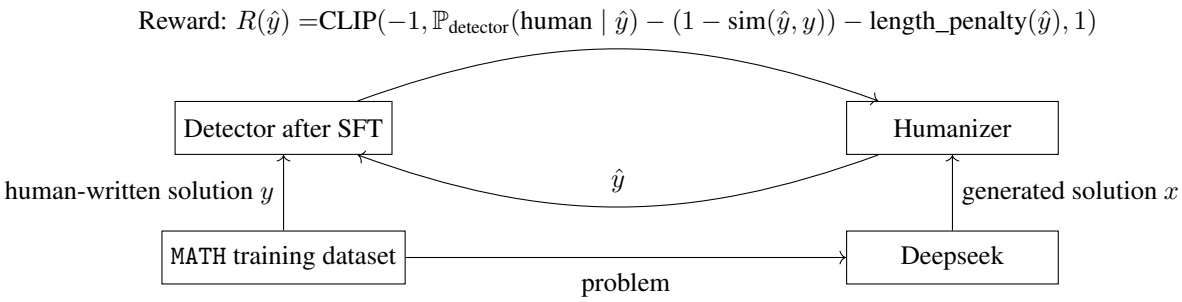

Figure 2: Adversarial reinforcement learning

The ARL loop operates as follows. At each iteration, the humanizer takes an AI-generated solution and outputs a rewritten version intended to appear more human-like. This output is evaluated by the fixed detector from the previous iteration to obtain a scalar reward signal. The reward for the humanizer encourages outputs that (i) are more likely to be classified as human, (ii) are semantically similar to the reference human answer, and (iii) avoid overly verbose rewrites.

The humanizer is optimized as a stochastic policy via the REINFORCE algorithm. Concretely, for each AI-generated solution x, we sample a paraphrase $\hat{y} \sim \pi_\theta(\cdot \mid x)$ under top-p sampling from the humanizer, and accumulate its sequence log-probability

$$\log \pi_\theta(\hat{y} \mid x) \;=\; \sum_{t=1}^{T} \log p_\theta(\hat{y}_t \mid \hat{y}_{<t}, x)\,.$$

We then compute the scalar reward of $\hat{y}$ according to the function given in Figure 2. The humanizer's LoRA parameters θ are then updated by ascending the policy-gradient estimate

$$\nabla_\theta \mathbb{E}[R] = \mathbb{E}\big[R(\hat{y})\, \nabla_\theta \log \pi_\theta(\hat{y} \mid x)\big]\,,$$

thereby directly maximizing expected reward. Simultaneously, to prevent the detector from always predicting zero, we form an adversarial batch by sampling half genuine human solutions and half humanizer outputs (shuffled at random), and fine-tune the detector's classification head with a standard cross-entropy loss on this mixed data. This alternating update allows the humanizer to continually find new stylistic "evasions" while the detector steadily becomes more robust to them.

2.2 Testing

During training, we used zero-shot prompting to generate solutions. To evaluate the detector's robustness against prompt-based evasion tactics, we inferred on 4,996 problems from the `MATH` test dataset using three distinct prompts with DeepSeek. In particular, we assessed whether the detector is robust to few-shot prompting. For this setup, we constructed a dictionary of three sample questions and solutions per subject of varying difficulty. For each problem, DeepSeek is then prompted to generate a solution that mirrors the style of the three examples from the same topic as the question. In so doing, we also increased the percentage of AI-generated solution from 50% during training to 75% at testing, allowing us to evaluate the detector's robustness to a different human-to-AI ratio.

In order to assess generalizability beyond competition-style math problems, we evaluated our detector on the `NaturalProofs` [7] corpus, a large-scale dataset of human-written mathematical theorems and proofs. As before, we used three different prompts. We attempted to generate DeepSeek solutions for the entire dataset, but were constrained by compute time and only processed 1,099 problems.

3 Results

3.1 Evaluation on `MATH` Test Dataset

We evaluate our models on the `MATH` test set across four key dimensions: problem difficulty, subject, semantic similarity to human-written ground-truth, and correctness of the final generated answer.

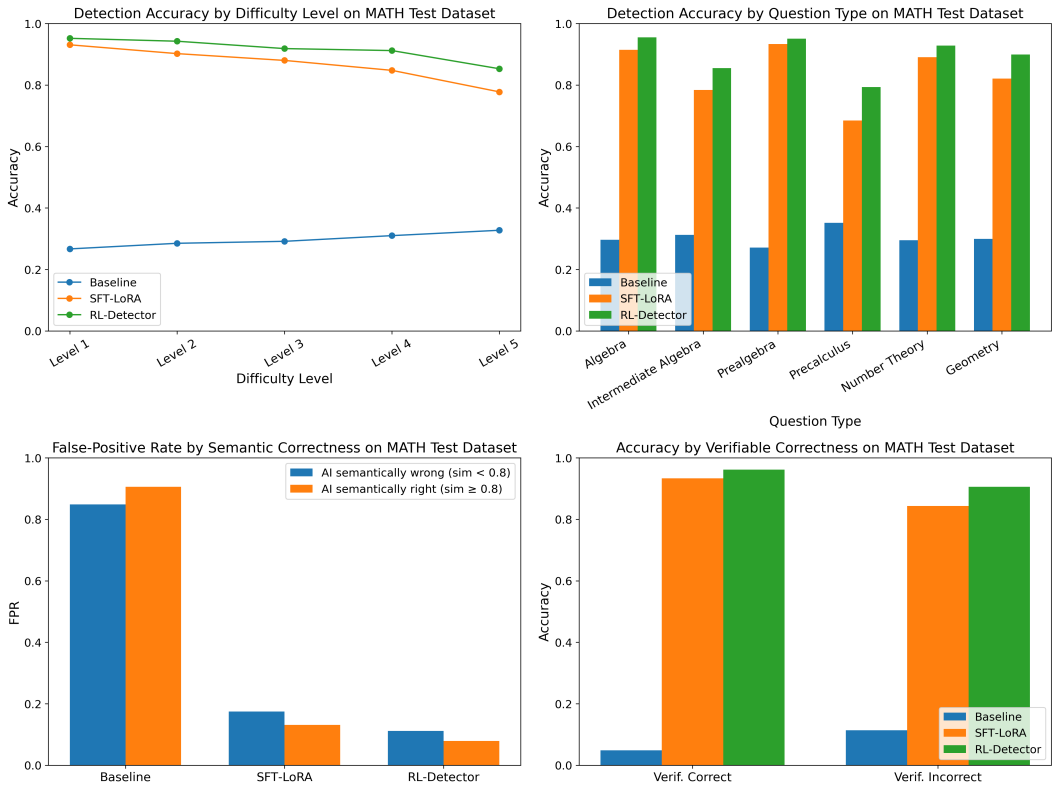

Figure 3: Detection Performance Analysis on `MATH` Test Set

Both SFT-LoRA and our RL-Detector greatly outperform the baseline, which produces similar accuracy to guessing. On the `MATH` test set, SFT-LoRA accuracy falls from 93% for problems at difficulty level 1 to 78% at Level 5, whereas the RL-Detector achieves 95% on Level 1 and retains 85% on Level 5, showing greater robustness to difficulty. Across the six math subjects, RL-Detector results in uniformly high accuracy, improving SfFT-LoRA by 10–12 points on topics like Precalculus. Moreover, ARL training reduces false-positive rate (FPR) on AI outputs from 13–17% under SFT-LoRA to 8–11%, especially when outputs are semantically faithful, and elevates accuracy on verifiably correct solutions to 96% (91% on incorrect ones).

3.2 Evaluation on `NaturalProofs` Dataset

We examine `NaturalProofs` in two dimensions: proof length and semantic similarity. First, we divide each proof into five equal-sized length quantiles to test sensitivity to input complexity. The baseline detector's accuracy plunges in the mid-length bins (Q2–Q4), reflecting more of the class imbalances whereas RL-Detector stays above 80% in almost every quantile, demonstrating stable performance on both brief and extended arguments. Second, we bin AI-generated proofs by their cosine similarity to a human proof for each question and measure the FPR, across these similarity quantiles. Baseline and SFT-LoRA exhibit FPRs around 30–60%, rising when semantic overlap increases; by contrast, RL-Detector holds FPR below 25% even at the highest similarity, suggesting ARL training enables detection of subtle artifacts beyond surface paraphrase.

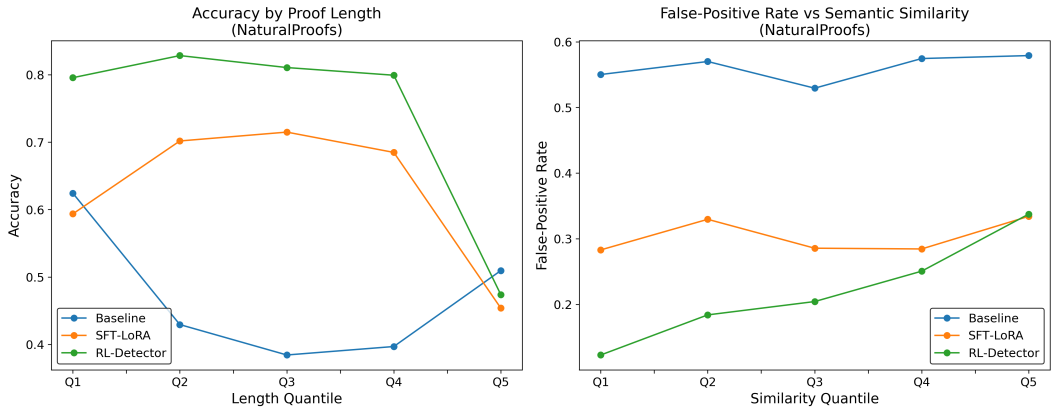

Figure 4: Detection Performance Analysis on `NaturalProofs`

4 Discussion

Our experiments demonstrate that ARL significantly enhances detector robustness compared to purely supervised fine-tuning. The RL-Detector's stable accuracy across increasing `MATH` difficulty levels and its uniform gains on advanced topics indicate that ARL allows for better capturing of stylistic and reasoning cues rather than relying on surface patterns alone. Similarly, its FPR rises only modestly as the proofs become semantically closer to the human exemplar, indicating the RL-Detector's resilience to paraphrasing.

A main limitation of our work is that evaluation assumes that human solutions are written without AI assistance, which does not reflect current practices where LLMs often aid human authors. In real-world educational settings, genuinely mixed human–LLM texts may blur the boundary between human and AI generation, potentially challenging our detector's decision boundary.

Future work could therefore explore hybrid training schemes that include AI-assisted human samples, as well as refine reward functions to explicitly penalize over-reliance on trivial cues. Extending ARL to other domains, such as code, college essays, or news could further test its generalizability and guide the design of more realistic, deployable detection systems.

References

[1] Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature, 2024.

[2] Henrique Da Silva Gameiro, Andrei Kucharavy, and Ljiljana Dolamic. Llm detectors still fall short of real world: Case of llm-generated short news-like posts, 2024.

[3] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

[4] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[5] Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. Release strategies and the social impacts of language models, 2019.

[6] Edward Tian and Alexander Cui. Gptzero: Towards detection of ai-generated text using zero-shot and supervised methods, 2023.

[7] Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. Naturalproofs: Mathematical theorem proving in natural language. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.