Leveraging Reinforcement Learning to Autonomously Improve Code Generation Performance in Large Language Models

Henry Harrison Adrian Hautea Ilayamudhan Elango Umamaheshwari Sithranjan Suresh
Abhijith Ulta

Introduction and Purpose

Large language models struggle with solving difficult coding problems directly. This is a limitation that affects their usefulness in actual software development tasks. To address this, we introduce a self-generated curriculum approach in which the model learns progressively from easier to harder problems. Unlike traditional curriculum learning, our model generates its own sequence of subproblems tailored to its weaknesses, enabling more effective learning. Our approach demonstrates that self-generated curricula can meaningfully improve a model's coding abilities.

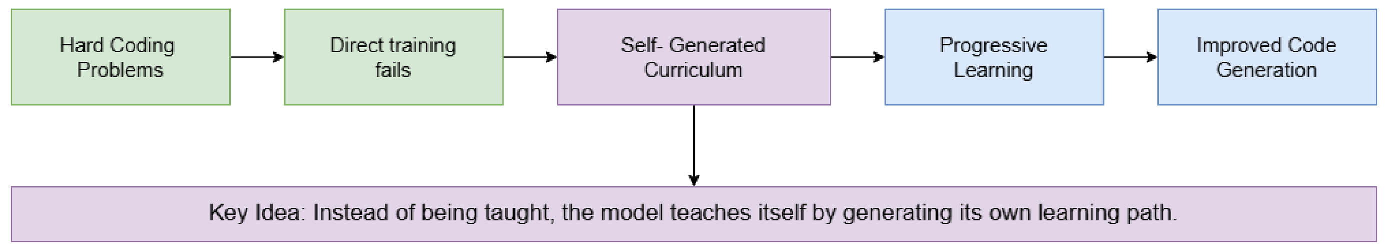

Figure 1. Core Problem and Solution

Dataset and Experimental Setup

Dataset: The dataset we used is the P3 dataset from Microsoft. It is a dataset of programming puzzles in pure python, with the problem statement in the form of a solver function that must be satisfied. The advantage of this setup is that the Python interpreter can be used as a source of ground truth for the experimental setup.

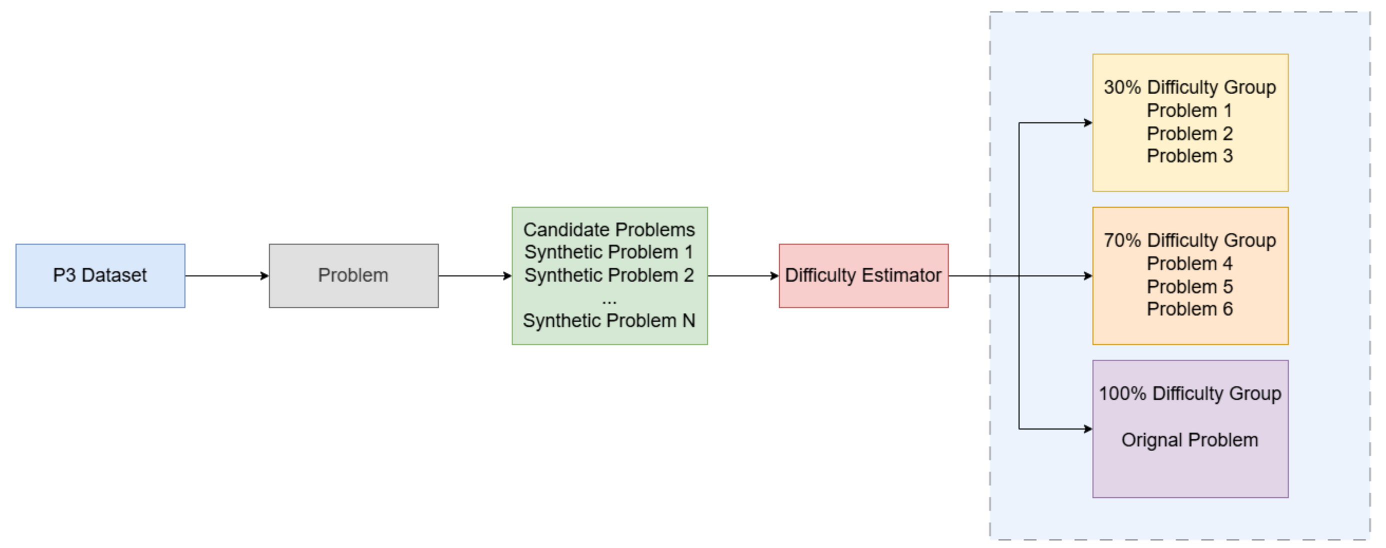

Figure 2. P3 Dataset

Problems are initially sampled from the P3 dataset. For each selected problem, a set of N synthetic variants is generated using a large language model to expand the problem space and introduce varying levels of complexity. A model-based difficulty estimator assigns each variant a relative difficulty score, allowing the problems to be organized into structured difficulty tiers to guide training from easier to harder problems.

Model

Model Architecture: The proposed model uses Deepseek Coder 1.3b Instruct fine-tuned on P3-derived variants. The model is built on the transformer architecture. Models like this have been used for code completion but not for curriculum-based self-improvement. The pipeline incorporates several components like variant generation and a GRPO-based reinforcement learning loop.

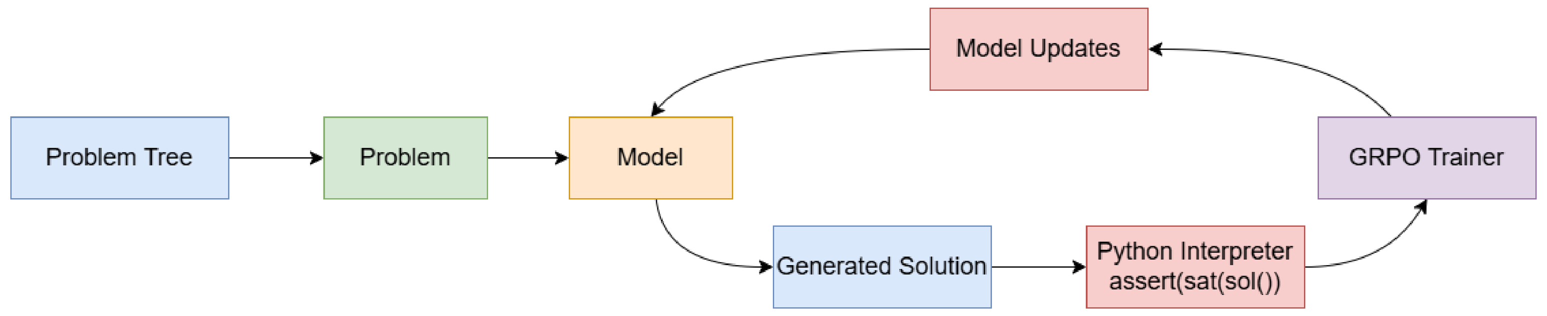

Figure 3. Model Architecture

This design enables the model to iteratively improve by exploring and learning from progressively harder problem variants.

Results

Model Evaluation: We received the highest pass@k rate, though adding steps like variant generation and filtering introduced additional costs compared to the baseline.

Model Metrics: The model was assessed using pass@k, which is defined as the following:

$$\text{pass@k} = \mathbb{E}\left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}\right]$$

Where n is the total number of samples, c is the number that pass ($c \leq n$), and k is the evaluation budget. $\binom{n}{k}$ denotes "n choose k".

Model	pass@1	pass@10
Pre-training (Baseline)	5.9%	6.0%
Post-GRPO	**14.0%**	**16.0%**

Table 1. Pass@k performance before and after GRPO training

Analysis

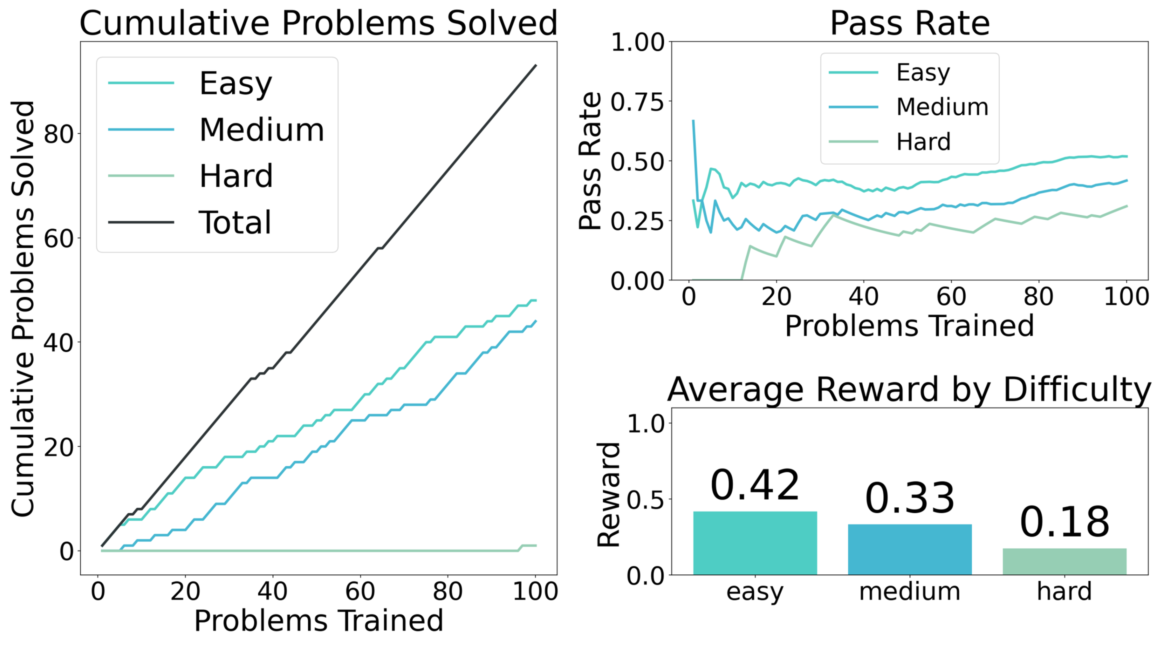

Figure 4. Cumulative Solved Problems, Average Reward, and Problem Pass Rate by Difficulty Bucket

The chart shows the model's learning efficacy, demonstrating a clear upwards trend of learning rate based on number of problems solved and showing a clear correlation between the training progress and the cumulative number of problems solved.

Conclusion

In conclusion, we introduced a self-improving training framework that allows large language models to tackle complex coding problems by generating and learning from their own curriculum. By constructing a difficulty ladder of synthetic problem variants derived from the P3 dataset, our approach allows the model to progressively build problem-solving capabilities through structured training rather than relying on direct exposure. While this approach introduces additional computational cost, the results suggest that self-generated curricula provides a promising framework to improve reasoning and generalization in code generation models.

References

- Haluptzok, P., Bowers, M., & Kalai, A. T. (2023, May). Language models can teach themselves to program better. ICLR. https://arxiv.org/abs/2207.14502
- Hacohen, G., & Weinshall, D. (2019, May 29). On The Power of Curriculum Learning in Training Deep Networks. *arXiv.org*. https://arxiv.org/abs/1904.03626arxiv