Exploring Reasoning in Small Language Models (SLMs)

Zach Gentile and Alex Lavaee

March 18, 2025

Introduction

Recent advancements in large language models (LLMs) have demonstrated impressive reasoning capabilities, but these models are computationally intensive and resource-demanding, requiring significant infrastructure for both training and inference. There is growing interest in exploring whether smaller language models (SLMs) can achieve comparable reasoning abilities while being more efficient and accessible. This project investigates the application of reinforcement learning techniques, specifically Group Relative Policy Optimization (GRPO), to enhance reasoning capabilities in lightweight SLMs like Gemma 3.

Problem Statement

While large language models have shown remarkable reasoning capabilities, their size creates barriers for widespread deployment, especially in resource-constrained environments. Small language models offer potential advantages in terms of efficiency, latency, and accessibility, but typically demonstrate inferior reasoning performance compared to their larger counterparts. The core problem this project addresses is how to effectively train SLMs to perform complex reasoning tasks while maintaining their computational efficiency advantages.

Specifically, we will investigate:

- How can GRPO be effectively applied to enhance reasoning capabilities in lightweight SLMs?
- What modifications to existing GRPO implementations are necessary to optimize for SLMs?
- How do reasoning-enhanced SLMs compare to larger models on standardized reasoning benchmarks?

- What are the computational trade-offs between model size and reasoning performance?

Proposed RL Techniques

Our approach centers on the application of Group Relative Policy Optimization (GRPO), a reinforcement learning technique that has shown promise in improving mathematical reasoning in language models. GRPO offers several advantages over traditional RL methods like Proximal Policy Optimization (PPO):

- **Elimination of value function**: Unlike PPO, GRPO does not require a separate critic model, reducing computational overhead and memory requirements—crucial advantages when working with resource-constrained environments.
- **Group-based advantage estimation**: GRPO generates multiple responses for the same prompt and computes advantages based on their relative performance within the group, eliminating the need for a value function.

Our implementation will follow a multi-stage approach:

1. Initial supervised fine-tuning (SFT) on high-quality reasoning datasets
2. GRPO training with deterministic rewards for reasoning trace quality, consistency, and correctness
3. Generation and filtering of synthetic data using stronger models as judges
4. Final GRPO alignment phase to enhance helpfulness while maintaining reasoning capabilities

We will explore modifications to the standard GRPO algorithm to better suit SLMs, including:

- Adaptive batch sizes based on available computational resources
- Targeted reward functions specific to reasoning tasks
- Integration of domain-specific knowledge into the training pipeline

Expected Challenges

Several challenges are anticipated in this project:

- **Model capacity limitations**: Small language models have inherently limited capacity, which may constrain their ability to learn complex reasoning patterns. We will need to carefully balance the complexity of reasoning tasks with model capacity.

- **Reward design complexity**: Designing effective reward functions for reasoning tasks is non-trivial. Rewards must be informative enough to guide learning but simple enough to compute efficiently.
- **Computational efficiency**: While SLMs require less computational resources than LLMs, the GRPO training process still demands significant computing power, especially for generating multiple responses per prompt. Optimizing this process will be critical.
- **Overfitting to reward structure**: RL-trained models can exploit patterns in reward functions rather than learning genuine reasoning capabilities. Ensuring generalization beyond the specific training rewards will be essential.
- **Evaluation challenges**: Assessing reasoning capabilities fairly across different model scales requires careful benchmark selection and potentially new evaluation methodologies.

Datasets

We will evaluate our approach using the following benchmarks used in DeepSeek-R1 [1]:

- **SWE-bench** [4]: Evaluates model capability to solve real-world software engineering issues from GitHub repositories. Models generate code patches to resolve described problems, with a validated subset (SWE-bench Verified) providing more accurate evaluation by filtering out infeasible tasks.
- **MMLU** (Massive Multitask Language Understanding) [2]: Comprises approximately 16,000 multiple-choice questions across 57 academic subjects including mathematics, law, philosophy, and science, designed to evaluate general knowledge and reasoning capabilities across diverse disciplines.
- **GPQA** (Graduate-Level Google-Proof Q&A) [6]: A challenging benchmark resistant to simple lookup strategies, evaluating advanced reasoning through graduate-level questions requiring deep understanding rather than surface-level retrieval.
- **Codeforces** [5]: A competitive programming dataset from real-world contests, featuring algorithmic coding problems of varying complexity that test the ability to generate correct and efficient code solutions.
- **AIME 2024** (American Invitational Mathematics Examination): High-level math competition problems requiring multi-step logical reasoning, creativity, and deep mathematical insight.
- **MATH** [3]: Comprises challenging math competition-style problems across various difficulty levels and domains, involving detailed step-by-step reasoning processes to

evaluate complex mathematical reasoning capabilities.

For our experimental environment, we will use Gemma 3 models (particularly the 1B and 4B variants) as our base SLMs, leveraging their state-of-the-art performance for models in their size class and their multi-language support. We will implement our training pipeline using standard deep learning frameworks and reinforcement learning libraries.

Evaluation Metrics

To evaluate the effectiveness of our approach, we will employ a comprehensive set of metrics:

- **Task-specific accuracy**: Performance on reasoning benchmarks such as GSM8K, MATH, LogiQA, and others.
- **Performance-to-parameter ratio**: Analysis of reasoning capabilities relative to model size compared to larger models.
- **Generalization capabilities**: Performance on out-of-distribution reasoning tasks not seen during training. For example, we can provide the model with logic puzzles and math problems and see if it can solve them.

Additionally, we will conduct ablation studies to assess the contribution of different components of our approach, such as the impact of various reward functions, the effect of different group sizes in GRPO, and the influence of initial supervised fine-tuning quality.

Conclusion

This project aims to advance the state of the art in small language models by enhancing their reasoning capabilities through the application of Group Relative Policy Optimization. If successful, our approach could significantly expand the accessibility of reasoning-capable language models, enabling deployment in resource-constrained environments and democratizing access to advanced AI capabilities. The techniques developed may also provide insights into the fundamental relationship between model size and reasoning abilities, potentially informing more efficient architectures for future language models.

References

[1] DeepSeek-AI et al. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. 2025. arXiv: 2501.12948 [cs.CL]. URL: https://arxiv.org/abs/2501.12948.

[2] Dan Hendrycks et al. *Measuring Massive Multitask Language Understanding*. 2021. arXiv: 2009.03300 [cs.CY]. URL: https://arxiv.org/abs/2009.03300.

[3] Dan Hendrycks et al. *Measuring Mathematical Problem Solving With the MATH Dataset*. 2021. arXiv: 2103.03874 [cs.LG]. URL: https://arxiv.org/abs/2103.03874.

[4] Carlos E. Jimenez et al. *SWE-bench: Can Language Models Resolve Real-World GitHub Issues?* 2024. arXiv: 2310.06770 [cs.CL]. URL: https://arxiv.org/abs/2310.06770.

[5] Shanghaoran Quan et al. *CodeElo: Benchmarking Competition-level Code Generation of LLMs with Human-comparable Elo Ratings*. 2025. arXiv: 2501.01257 [cs.CL]. URL: https://arxiv.org/abs/2501.01257.

[6] David Rein et al. *GPQA: A Graduate-Level Google-Proof Q&A Benchmark*. 2023. arXiv: 2311.12022 [cs.AI]. URL: https://arxiv.org/abs/2311.12022.