deepseek

DeepSeekMath-V2: Towards Self-Verifiable Mathematical Reasoning

Zhihong Shao*, Yuxiang Luo*, Chengda Lu*†, Z.Z. Ren*

Jiewen Hu, Tian Ye, Zhibin Gou, Shirong Ma, Xiaokang Zhang

DeepSeek-AI

zhihongshao@deepseek.com

https://github.com/deepseek-ai/DeepSeek-Math-V2

Abstract

Large language models have made significant progress in mathematical reasoning, which serves as an important testbed for AI and could impact scientific research if further advanced. By scaling reasoning with reinforcement learning that rewards correct final answers, LLMs have improved from poor performance to saturating quantitative reasoning competitions like AIME and HMMT in one year. However, this approach faces fundamental limitations. Pursuing higher final answer accuracy doesn't address a key issue: correct answers don't guarantee correct reasoning. Moreover, many mathematical tasks like theorem proving require rigorous step-by-step derivation rather than numerical answers, making final answer rewards inapplicable. To push the limits of deep reasoning, we believe it is necessary to verify the comprehensiveness and rigor of mathematical reasoning. Self-verification is particularly important for scaling test-time compute, especially for open problems without known solutions. Towards self-verifiable mathematical reasoning, we investigate how to train an accurate and faithful LLM-based verifier for theorem proving. We then train a proof generator using the verifier as the reward model, and incentivize the generator to identify and resolve as many issues as possible in their own proofs before finalizing them. To maintain the generation-verification gap as the generator becomes stronger, we propose to scale verification compute to automatically label new hard-to-verify proofs, creating training data to further improve the verifier. Our resulting model, DeepSeekMath-V2, demonstrates strong theorem-proving capabilities, achieving gold-level scores on IMO 2025 and CMO 2024 and a near-perfect 118/120 on Putnam 2024 with scaled test-time compute. While much work remains, these results suggest that self-verifiable mathematical reasoning is a feasible research direction that may help develop more capable mathematical AI systems.

1. Introduction

The conventional approach to reinforcement learning (RL) for mathematical reasoning involves rewarding large language models (LLMs) based on whether their predicted final answers to quantitative reasoning problems match ground-truth answers (Guo et al., 2025). This methodology suffices to allow frontier LLMs to saturate mathematical competitions that primarily evaluate final answers, such as AIME and HMMT. However, this reward mechanism has two fundamental limitations. First, it serves as an unreliable proxy for reasoning correctness – a model can arrive at the correct answer through flawed logic or fortunate errors. Second, it is

*Core contributors †Work done during internship at DeepSeek-AI.

inapplicable to theorem proving tasks, where problems may not require producing numerical final answers and rigorous derivation is the primary objective.

Consequently, LLMs trained on quantitative reasoning problems with such final answer reward still frequently produce mathematically invalid or logically inconsistent natural-language proofs. Moreover, this training approach does not naturally develop the models' ability to verify proof validity – they exhibit high false-positive rates, often claiming incorrect proofs are valid even when they contain obvious logical flaws.

The lack of a generation-verification gap in natural-language theorem proving hinders further improvement. To address this, we propose developing proof verification capabilities in LLMs. Our approach is motivated by several key observations:

- Humans can identify issues in proofs even without reference solutions – a crucial ability when tackling open problems.
- A proof is more likely to be valid when no issues can be identified despite scaled verification efforts.
- The efforts required to identify valid issues can serve as a proxy for proof quality, which can be exploited to optimize proof generation.

We believe that LLMs can be trained to identify proof issues without reference solutions. Such a verifier would enable an iterative improvement cycle: (1) using verification feedback to optimize proof generation, (2) scaling verification compute to auto-label hard-to-verify new proofs, thereby creating the training data to improve the verifier itself, and (3) using this enhanced verifier to further optimize proof generation. Moreover, a reliable proof verifier enables us to teach proof generators to evaluate proofs as the verifier does. This allows a proof generator to iteratively refine its proofs until it can no longer identify or resolve any issues. In essence, we make the model explicitly aware of its reward function and enable it to maximize this reward through deliberate reasoning rather than blind trial-and-error.

Built on DeepSeek-V3.2-Exp-Base (DeepSeek-AI, 2025), we developed **DeepSeekMath-V2**, a large language model optimized for natural-language theorem proving that demonstrates self-verifiable mathematical reasoning. Our model can assess and iteratively improve its own proofs, achieving gold-level performance in premier high-school mathematics competitions including IMO 2025 and CMO 2024. On the Putnam 2024 undergraduate competition, it scored 118/120, exceeding the highest score of 90 [1] obtained by human participants.

2. Method

2.1. Proof Verification

2.1.1. Training a Verifier to Identify Issues and Score Proofs

We developed high-level rubrics $\mathcal{I}_v$ for proof evaluation (see Appendix A.2) with the goal of training a verifier to evaluate proofs according to these rubrics, mirroring mathematical experts' assessment process. Specifically, given a problem X and a proof Y, the verifier $\pi_\varphi(\cdot|X, Y, \mathcal{I}_v)$ is designed to produce a proof analysis that first summarizes identified issues (if any) and then assigns a score based on three levels: 1 for complete and rigorous proofs with all logical steps clearly justified; 0.5 for proofs with sound overall logic but minor errors or omitted details; and 0 for fundamentally flawed proofs containing fatal logical errors or critical gaps.

[1] `https://kskedlaya.org/putnam-archive/putnam2024stats.html`