Alphaproof and the IMO

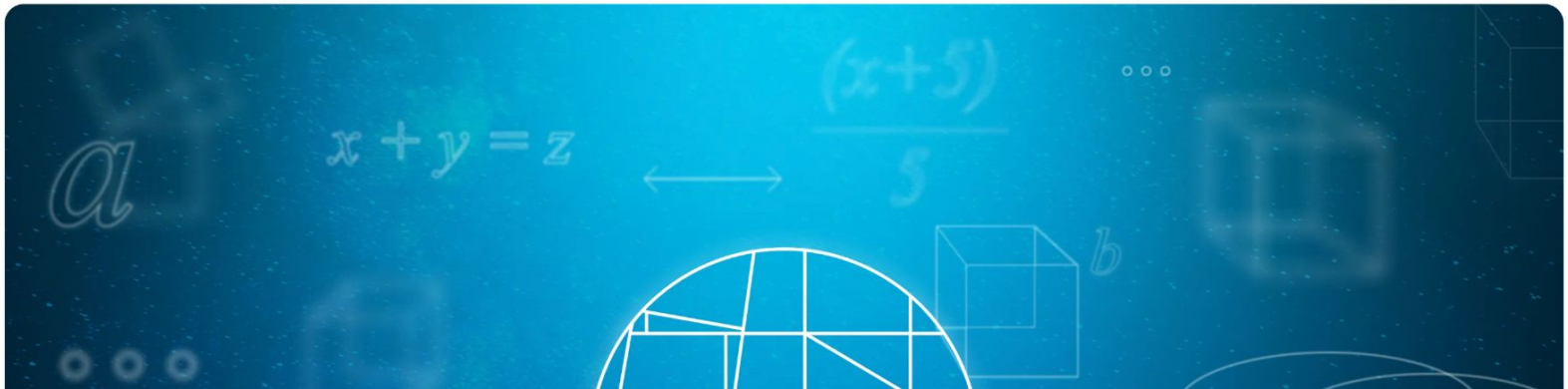
Celio Boulay, Thomas Moulin, Alex Chai

July 25, 2024 Science

AI achieves silver-medal standard solving International Mathematical Olympiad problems

AlphaProof and AlphaGeometry teams

Share



Olympiad-level formal mathematical reasoning with reinforcement learning

Thomas Hubert[*1], Rishi Mehta[1], Laurent Sartran[1], Miklós Z. Horváth[1], Goran Žužić[1], Eric Wieser[†1], Aja Huang[1], Julian Schrittwieser[1], Yannick Schroecker[1], Hussain Masoom[1], Ottavia Bertolli[1], Tom Zahavy[1], Amol Mandhane[1], Jessica Yung[1], Iuliya Beloshapka[1], Borja Ibarz[1], Vivek Veeriah[1], Lei Yu[1], Oliver Nash[1], Paul Lezeau[1], Salvatore Mercuri[1], Calle Sönne[1], Bhavik Mehta[1], Alex Davies[1], Daniel Zheng[1], Fabian Pedregosa[1], Yin Li[1], Ingrid von Glehn[1], Mark Rowland[1], Samuel Albanie[1], Ameya Velingker[1], Simon Schmitt[1], Edward Lockhart[1], Edward Hughes[1], Henryk Michalewski[1], Nicolas Sonnerat[1], Demis Hassabis[1], Pushmeet Kohli[1], and David Silver[1]

[1]*Google DeepMind, London, UK*

July 2025

AI PEER REVIEW ARTICLE PREVIEW

Abstract

A long-standing goal of artificial intelligence is to build systems capable of complex reasoning in vast domains, a task epitomized by mathematics with its boundless concepts and demand for rigorous proof. Recent AI systems, often reliant on human data, typically lack the formal verification necessary to guarantee correctness. By contrast, formal languages such as Lean [1] offer an interactive environment that grounds reasoning, and reinforcement learning (RL) provides a mechanism for learning in such environments. We present AlphaProof, an AlphaZero-inspired [2] agent that learns to find formal proofs through RL by training on millions of auto-formalized problems. For the most difficult problems, it uses Test-Time RL, a method of generating and learning from millions of related problem variants at inference time to enable deep, problem-specific adaptation. AlphaProof substantially improves state-of-the-art results on historical mathematics competition problems. At the 2024 IMO competition, our AI system, with AlphaProof as its core reasoning engine, solved three out of the five non-geometry problems, including the competition's most difficult problem. Combined with AlphaGeometry 2 [3], this performance, achieved with multi-day computation, resulted in reaching a score equivalent to that of a silver medallist, marking the first time an AI system achieved any medal-level performance. Our work demonstrates that learning at scale from grounded experience produces agents with complex mathematical reasoning strategies, paving the way for a reliable AI tool in complex mathematical problem-solving.

Auto Formalization

IMO 2021 Shortlist, Problem A5

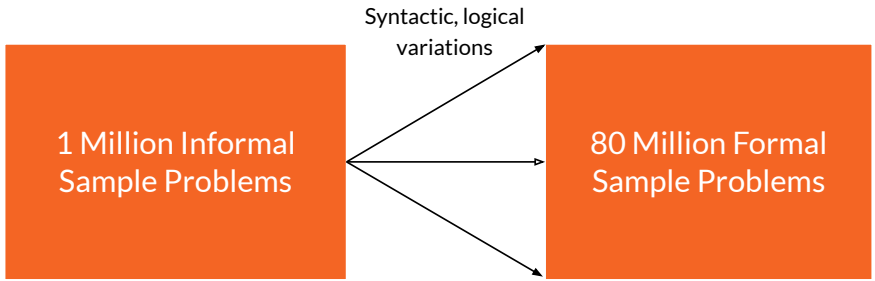
Let $n \geq 2$ be an integer and let $a_1, a_2, \ldots, a_n$ be positive real numbers with sum 1. Prove that

$$\sum_{k=1}^{n} \frac{a_k}{1-a_k}(a_1 + a_2 + \cdots + a_{k-1})^2 < \frac{1}{3}.$$

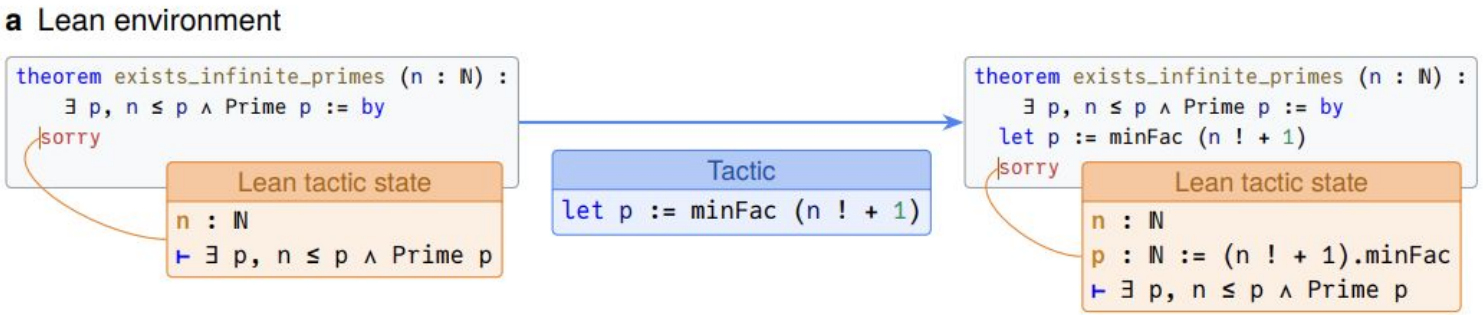
Formalization system

```
theorem imo_shortlist_2021_a5
    (n : ℕ) (h₀ : 2 ≤ n) (a : ℕ → ℝ) (hapos : ∀ i, 0 < a i)
    (hasum : ∑ i in Finset.Icc 1 n, a i = 1) :
    ∑ k in Finset.Icc 1 n, a k / (1 - a k) * (∑ i in Finset.Icc 1 (k-1), a i) ^ 2 < 1 / 3
```

(The actual 2024 IMO problems were formalized by hand though)

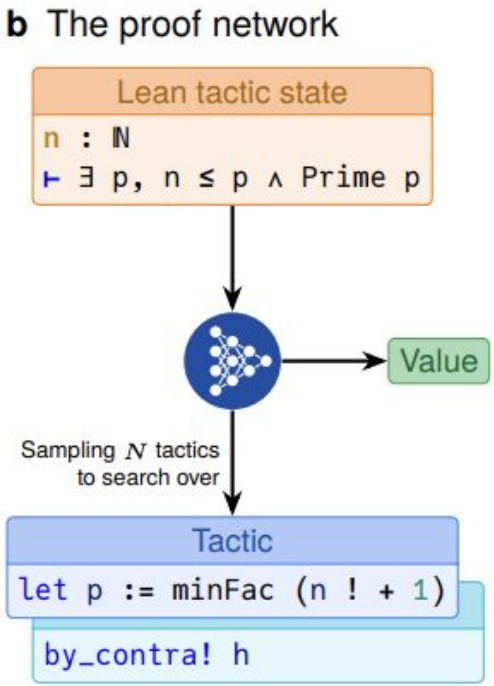


Core : RL environment & Proof Tree



r = -1

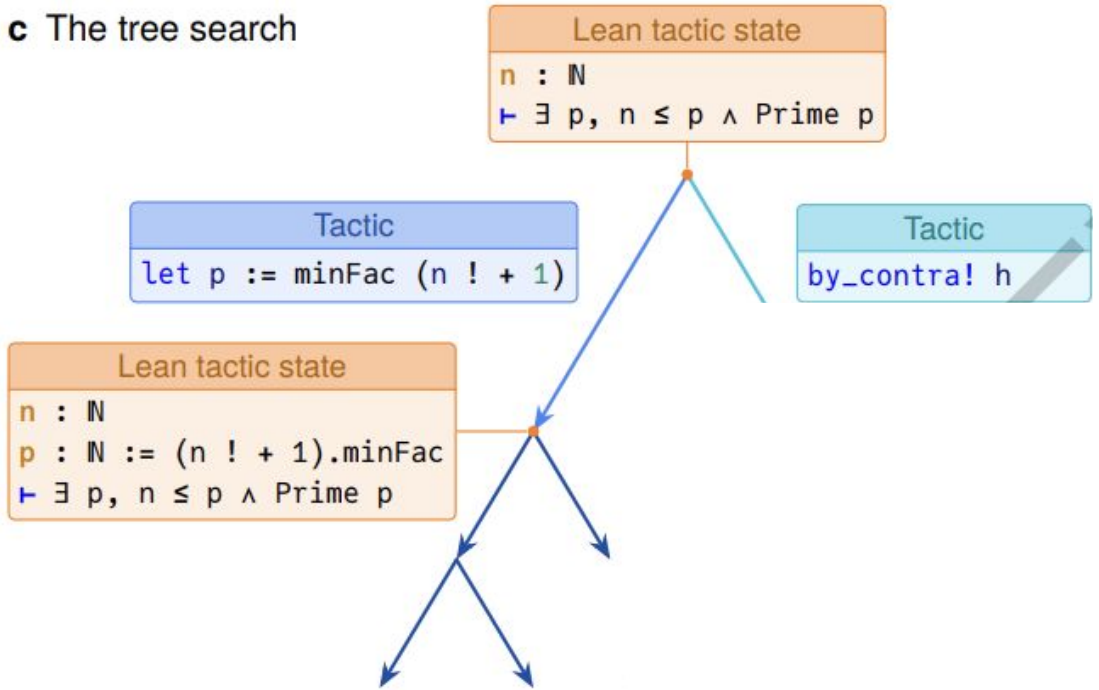
Core : RL environment & Proof Tree

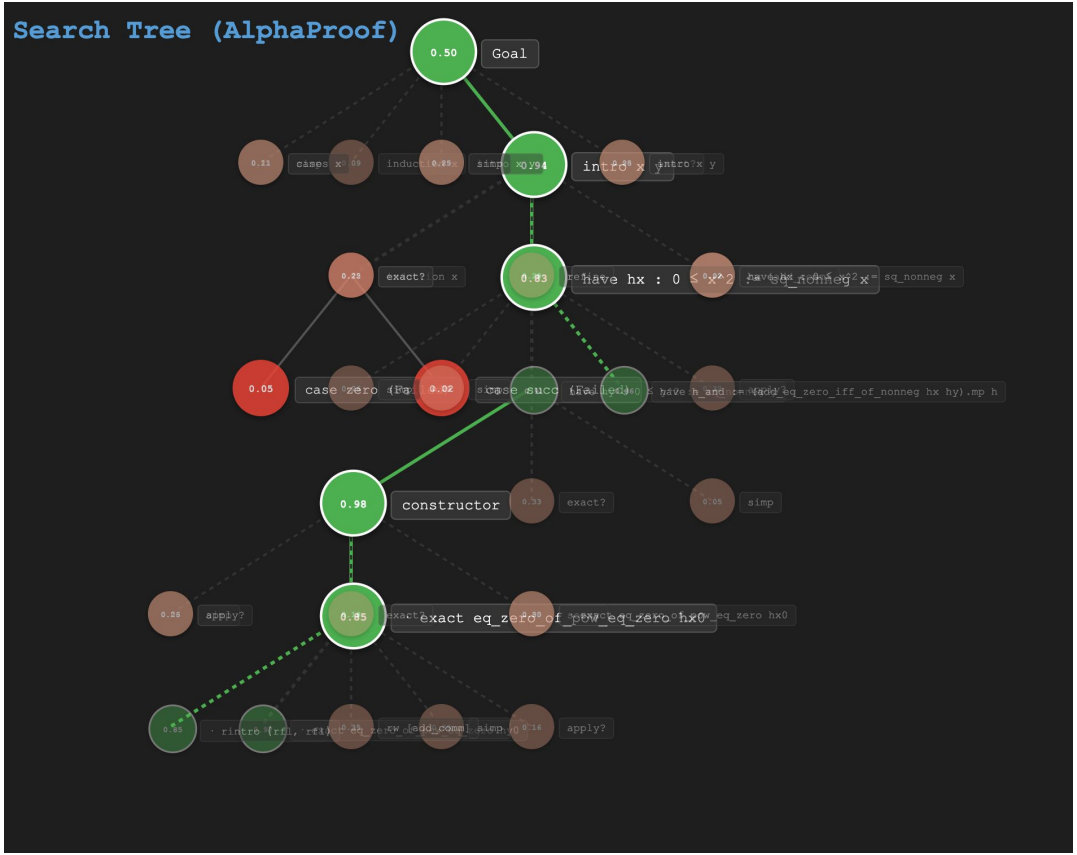


Core : RL environment & Proof Tree

Explore the space of possible proof constructions.

1. Selection
2. Expansion
3. Backpropagation

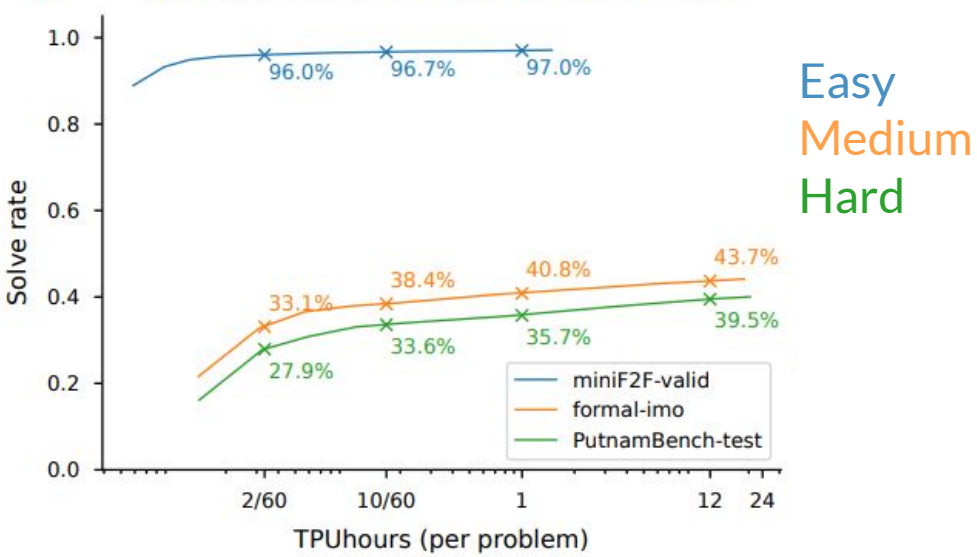




https://thomasmoulin-hub.github.io/animationprez/

Test Time Reinforcement Learning

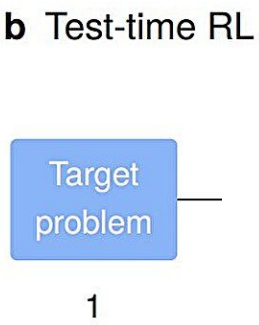
Classic search run



Easy
Medium
Hard

When the standard search tree hits a computational wall on a problem

Test Time Reinforcement Learning

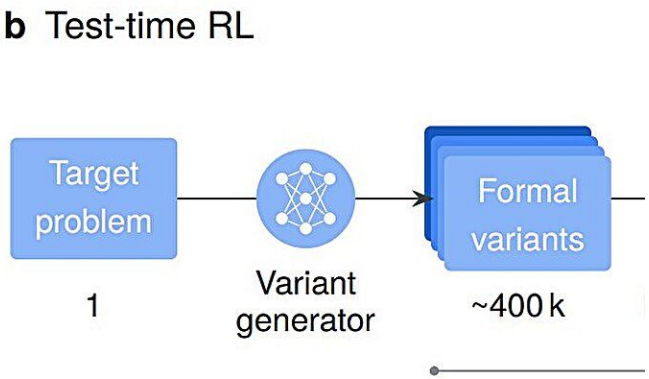


1. Generate hundreds of thousands of synthetic variants of the target problem: simplifications, generalizations, or related lemmas.

2. Run a focused RL loop on the variants, the agent finetune a specialist model on the target problem.

3. Mimics the human strategy of solving a simpler version of a problem to gain the logic needed for the final proof.

Test Time Reinforcement Learning

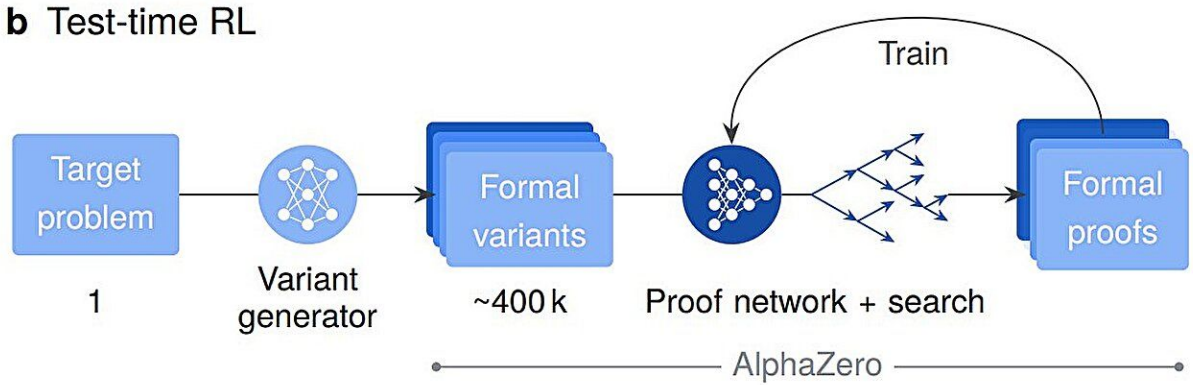


1. Generate hundreds of thousands of synthetic variants of the target problem: simplifications, generalizations, or related lemmas.

2. Run a focused RL loop on the variants, the agent finetune a specialist model on the target problem.

3. Mimics the human strategy of solving a simpler version of a problem to gain the logic needed for the final proof.

Test Time Reinforcement Learning

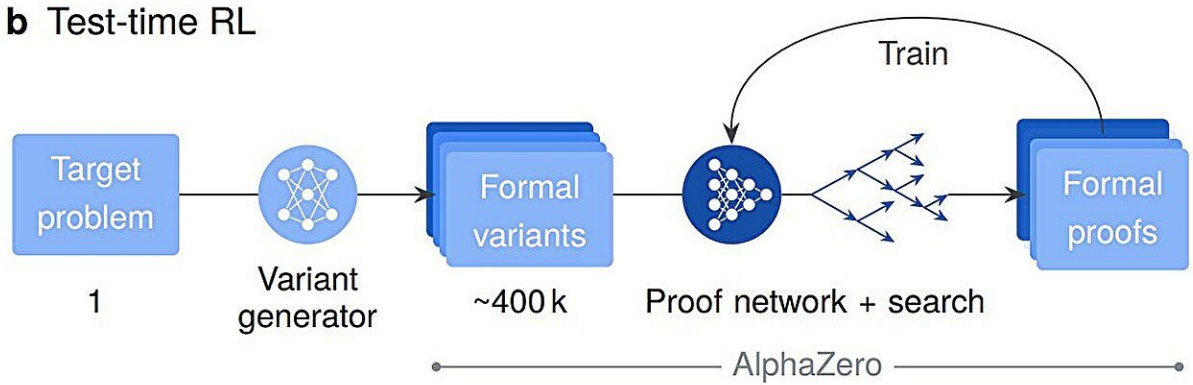


1. Generate hundreds of thousands of synthetic variants of the target problem: simplifications, generalizations, or related lemmas.

2. Run a focused RL loop on the variants, the agent finetune a specialist model on the target problem.

3. Mimics the human strategy of solving a simpler version of a problem to gain the logic needed for the final proof.

Test Time Reinforcement Learning

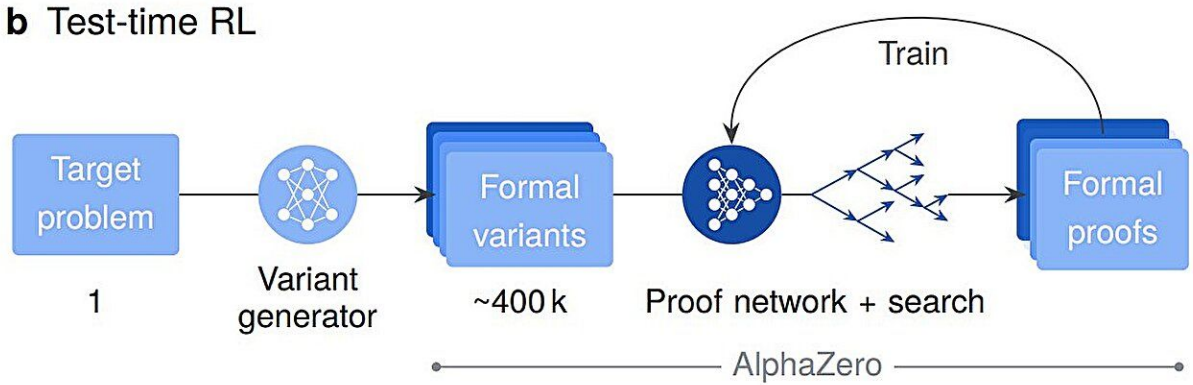


1. Generate hundreds of thousands of synthetic variants of the target problem: simplifications, generalizations, or related lemmas.

2. Run a focused RL loop on the variants, the agent finetune a specialist model on the target problem.

3. Mimics the human strategy of solving a simpler version of a problem to gain the logic needed for the final proof.

Test Time Reinforcement Learning



Solving variants allows the agent to:

- fine-tune its policy for the problem
- discover specific logical structures of the problem
- unlock proofs that were previously unreachable.

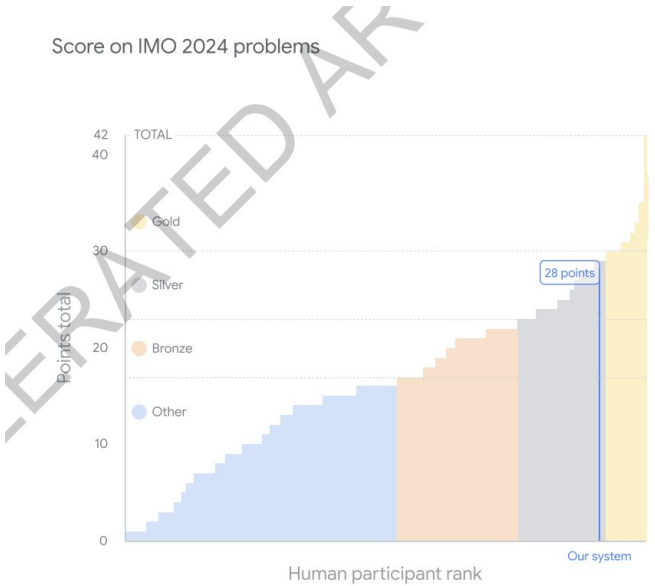
IMO 2024

- Successfully solved 4/6 of the IMO problems 😃
- Ranked as a silver medalist, and pretty close to gold 😃

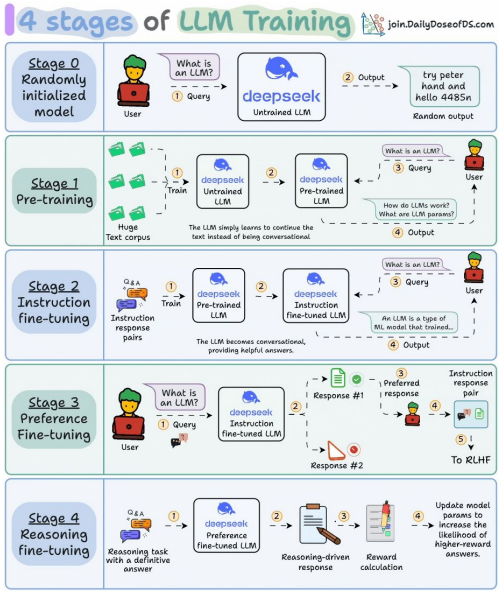
But:

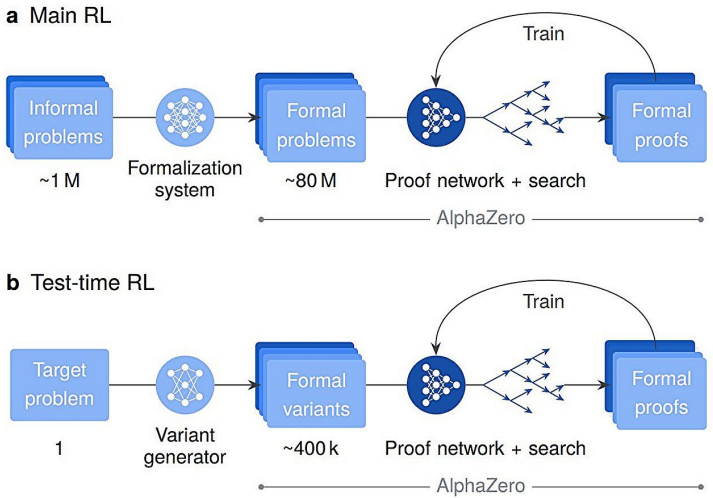
- Required multiple days to complete 😐
- Had to use AlphaGeometry 2 for geometry problems and failed with Combinatorics 😐
- Problems were hand formalized 😐

Problem ID	Domain	System Used	Outcome
P1	Algebra	AlphaProof	Solved
P2	Number Theory	AlphaProof	Solved
P3	Combinatorics	AlphaProof	Unsolved
P4	Geometry	AlphaGeometry 2	Solved
P5	Combinatorics	AlphaProof	Unsolved
P6	Algebra	AlphaProof	Solved (5 human solvers)



Alphaproof vs LLMs



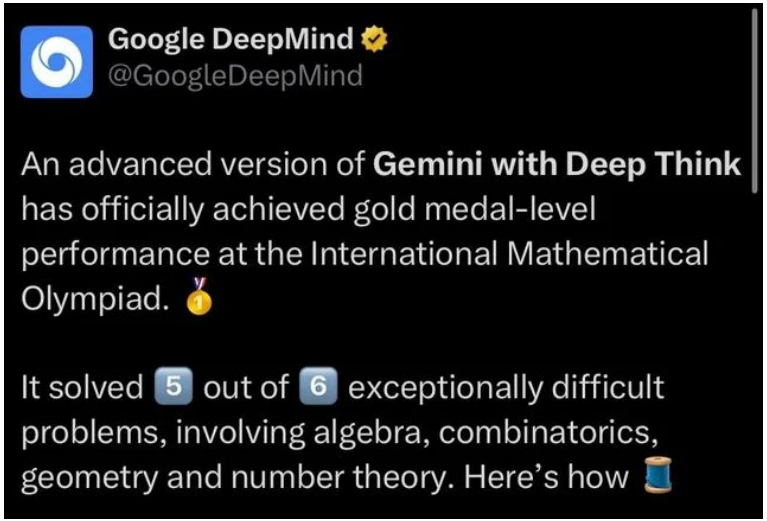


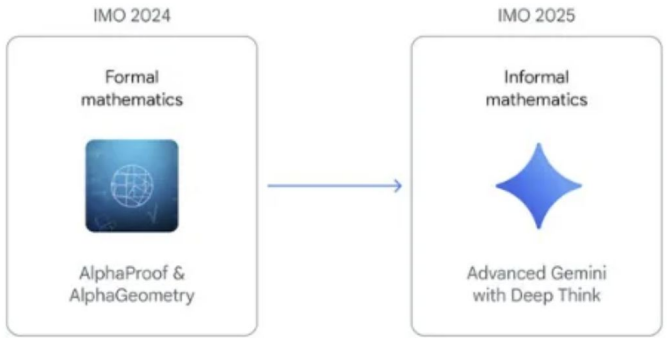
IMO 2025 - Alphaproof 2?

Google later used a version of Gemini with deep think to achieve gold level performance in the 2025 IMO

AND was able to do it faster

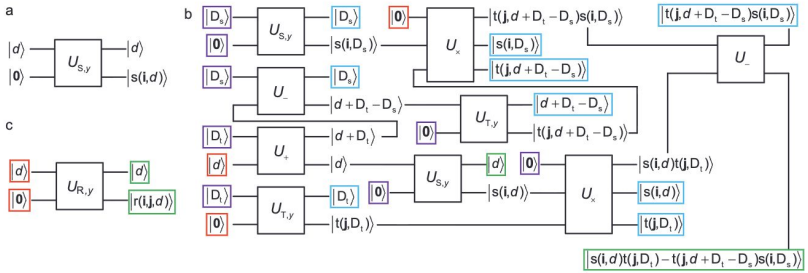
But: more emphasis on end to end English language, so we'll have to see…





Means of Improvement

- Geometry
  - AlphaGeometry
  - Wu's Method
  - Quantum Techniques
- Expansion of Lean libraries
- Optimization - sometimes leveraging LL in tandem
  - SeedProver
- Multiple techniques at once
  - Lemmanade



Name	Compute Budget	Putnam	Fate-H	Fate-X	Combibench
Seed-Prover 1.0 (medium)	18 H20 days / problem	50.4%	35%	9%	39%
AlphaProof	500 TPU days / problem	56.1%	-	-	-
Hilbert	avg pass@1840	70.0%	-	-	-
Aleph Prover	avg 1834 tool calls	75.8%	-	-	-
Seed-Prover 1.5	10 H20 days / problem	**580/87.9%**	**80%**	**33%**	**48%**

Table 2 Performance comparison of Seed-Prover 1.5 against other methods.

Future Speculation - The Bad News

But IMO problems are different from cutting edge research

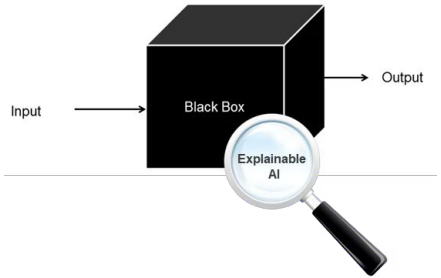
Can't create "new rules"

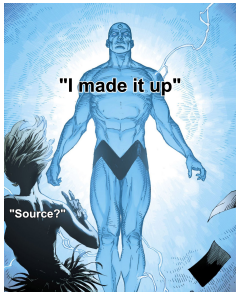
"LLM's Can't Jump"

But could still be useful for Lemmas

https://www.nature.com/articles/d41586-025-03585-5

Future Speculation - The Better News





The TTRL system customizes the algorithm to individual problems

Using Lean allows for direct checking

Interesting that it brings up AI generated results that can be self verifiable, since they're in Lean, which makes hallucinations harder

Is AI better suited to specific roles rather than general ones?

Does this make expert based systems more relevant again?

Thanks!