Reasoning with AI - facts and myths

Bartosz Naskręcki

Faculty of Mathematics and Computer Science, UAM

Mathematics AD 2025

What is a mathematical proof ?

- $Ax: p, q \vdash p$
- $p \vdash q \rightarrow p\ [\rightarrow R]$
- $\vdash p \rightarrow (q \rightarrow p)[\rightarrow R]$

- The formula p or q proves p.
- The formula p proves
- The claim is true

Theorem: There are infinitely many prime numbers.

Proof: suppose there are a finite number of them.
Let be prime numbers.
Then is different from the previous ones and has divisor one.
Qed.

Updated list of "100 Theorems"

- The initial state of the list and what has been realized from it.
- Only Fermat remains unformed.
- Formalization in different languages
- Will computers soon be proving theorems on their own?

Formalizing 100 Theorems

There used to exist a "top 100" of mathematical theorems on the web, which is a rather arbitrary list (and most of the theorems seem rather elementary), but still is nice to look at. On the current page I will keep track of which theorems from this list have been formalized. Currently the fraction that already has been formalized seems to be

99%

The page does not keep track of *all* formalizations of these theorems. It just shows formalizations in systems that have formalized a significant number of theorems, or that have formalized a theorem that none of the others have done. The systems that this page refers to are (in order of the number of theorems that have been formalized, so the more interesting systems for mathematics are near the top):

Isabelle	88
HOL Light	87
Coq	79
Lean	76
Metamath	74
Mizar	69
ProofPower	43
nqthm/ACL2	37
PVS	26
NuPRL/MetaPRL	8

Theorems in the list which have not been formalized yet are in italics. Formalizations of constructive proofs are in italics too. The difficult proofs in the list (according to John all the others are not a serious challenge "given a week or two") have been underlined. The formalizations under a theorem are in the order of the list of systems, and *not* in chronological order.

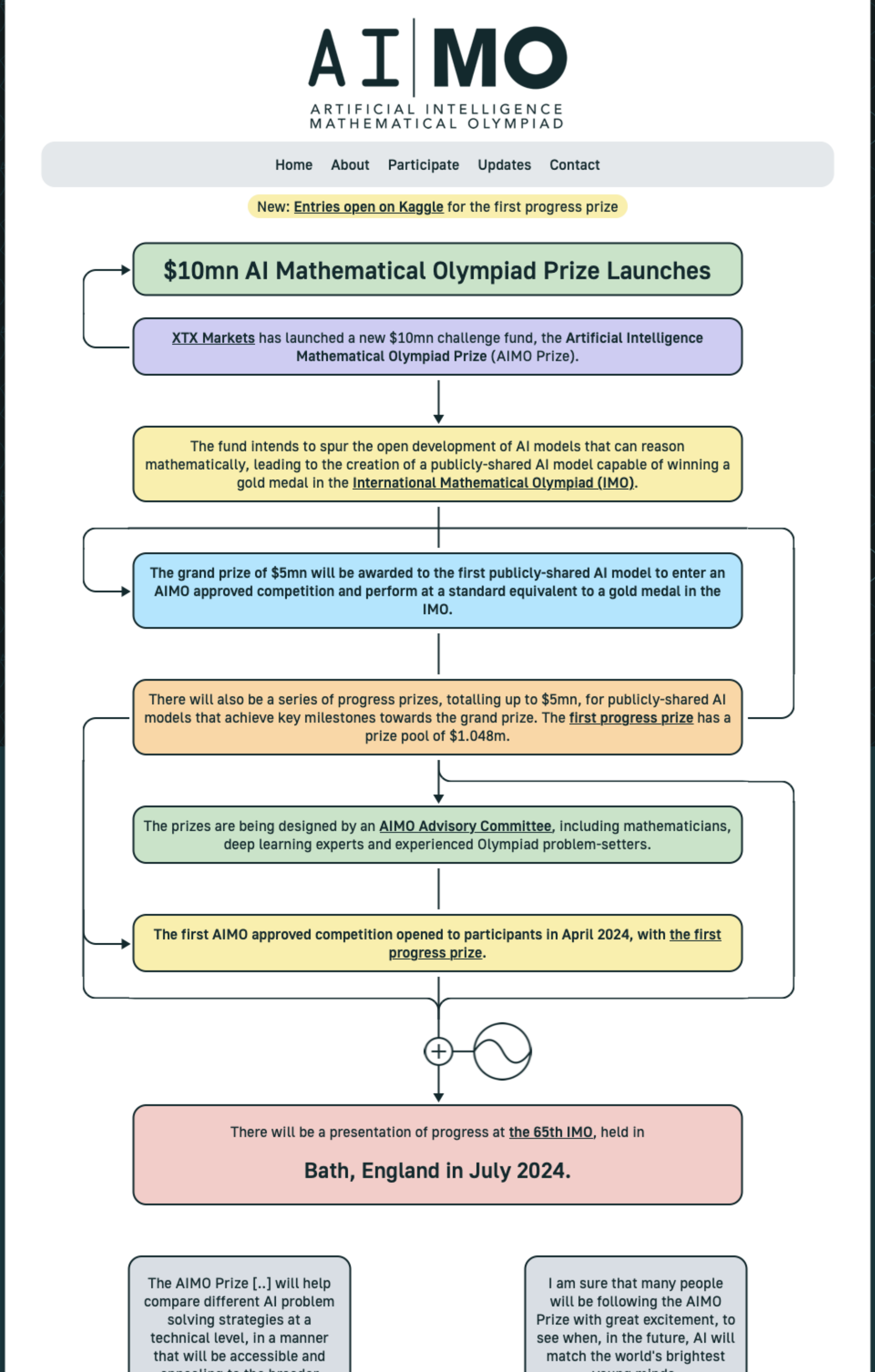
AIMO Award

AIMO competition in a nutshell:

- Goal: Develop AI capable of winning a gold medal at IMO.
- Prize: $10 million.
- Organizers: XTX Markets and AIMO Advisory Committee.

First stage: Progress awards from April 2024.

- Target objective: Solve problems with a level of difficulty IMO.
- Benefits: Financial rewards, prestige, collaboration with experts.
- Read more: https://aimoprize.com/



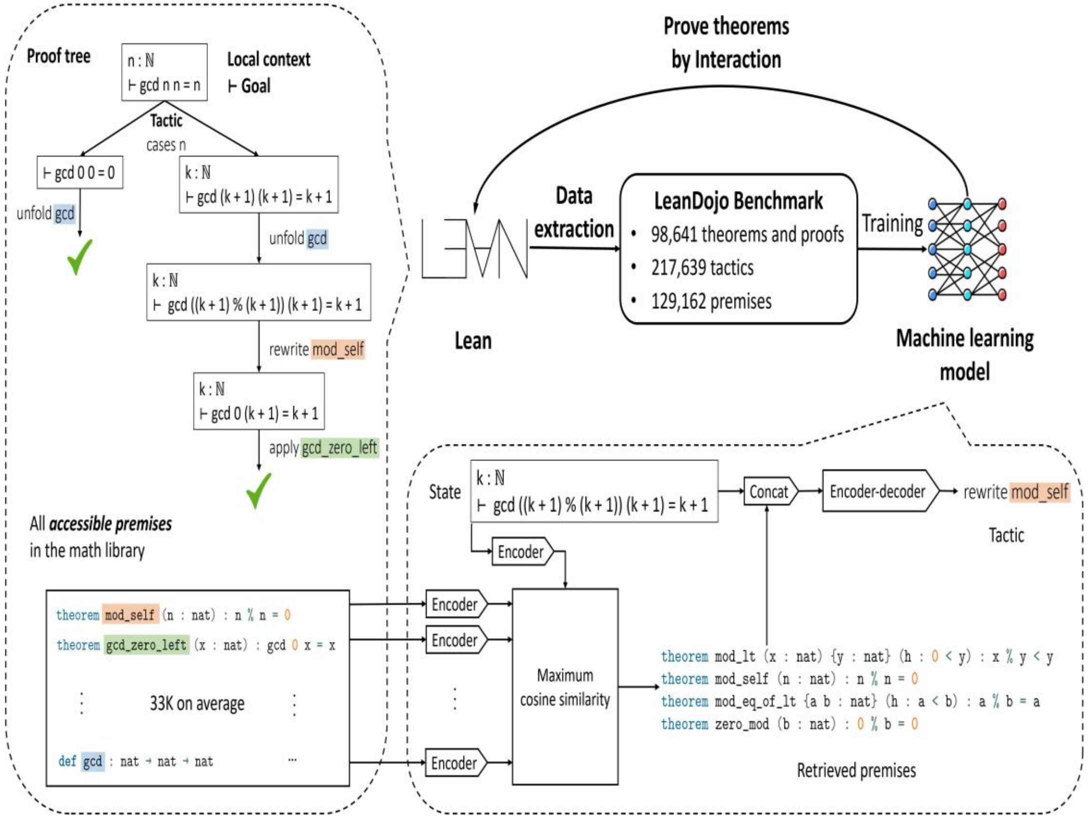
```lean
/-- Let $H$ be a subgroup of $G \times G'$. Then there exists a subgroup $H_0$ of $G$, a
subgroup $H_1$ of $G'$, and a homomorphism $\phi: G \to G'$ such that
$$ H := \{ (x, \phi(x) + y): x \in H_0, y \in H_1 \}.$$
In particular, $|H| = |H_0| |H_1|$. -/
lemma goursat (H : Submodule (ZMod 2) (G × G')) :
    ∃ (H₀ : Submodule (ZMod 2) G) (H₁ : Submodule (ZMod 2) G') (φ : G →+ G'),
      (∀ x : G × G', x ∈ H ↔ (x.1 ∈ H₀ ∧ x.2 - φ x.1 ∈ H₁)) ∧
       Nat.card H = Nat.card H₀ * Nat.card H₁ := by
  obtain ⟨S₁, S₂, f, φ, hf, hf_inv⟩ := H.exists_equiv_fst_sndModFst
  use S₁, S₂, φ
  constructor ; swap
  · show Nat.card H = _
    exact Eq.trans (Nat.card_eq_of_bijective f f.bijective) (Nat.card_prod S₁ S₂)
  · intro x
    · constructor
      · intro hx
        let x : H := { val := x, property := hx }
        · constructor
          · exact Set.mem_of_eq_of_mem (hf x).1.symm (f x).1.property
          · exact Set.mem_of_eq_of_mem (hf x).2.symm (f x).2.property
      · intro hx
        · let x₁ : S₁ := { val := x.1, property := hx.1 }
          let x₂ : S₂ := { val := x.2 - φ x.1, property := hx.2 }
          exact Set.mem_of_eq_of_mem (by rw [hf_inv, sub_add_cancel]) (f.symm (x₁, x₂)).property
```

The Lean Community and its mathematical library

- Lean command assistant, mainly developed by Leonardo de Moura.
- Lean math library (mathlib) is an active community project.
- The goal is to create a unified library of Lean formalized mathematics.

Interactive theorem proving with AI

- LLM language models are becoming powerful enough to explore hypotheses and automatically prove claims when combined with mathematics formalization techniques (e.g., Lean)
- LeanDojo
- MAmmoTH
- AlphaGeometry/AlphaProof
- FunSearch/AlphaEvolve



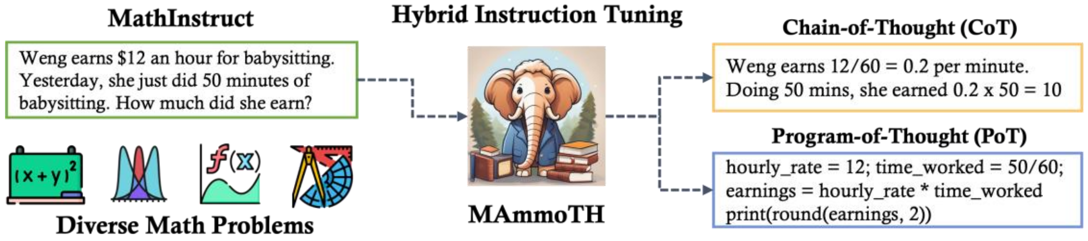
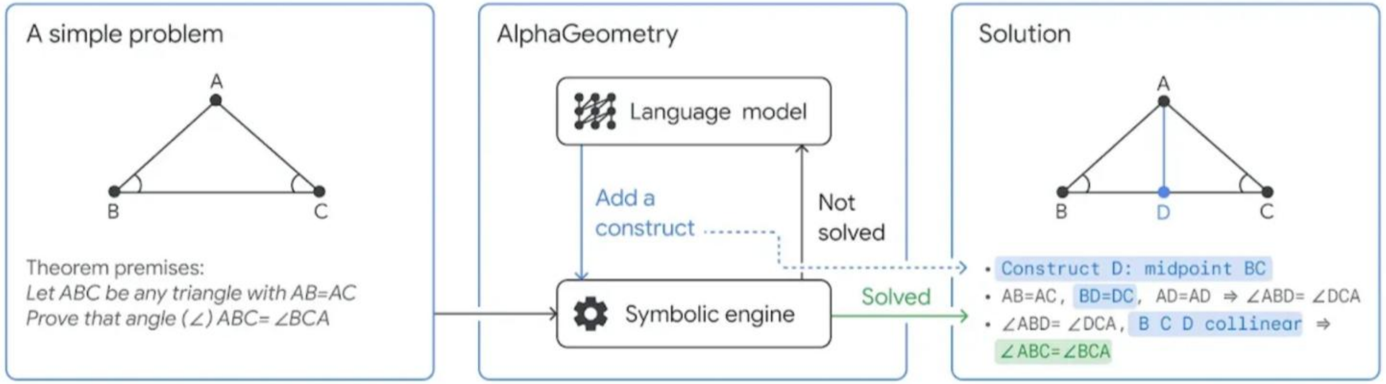


Table 1 | Main results on our IMO-AG-30 test benchmark

Method		Problems solved (out of 30)
Computer algebra	Wu's method[21] (previous state of the art)	10
	Gröbner basis[20]	4
Search (human-like)	GPT-4 (ref. 25)	0
	Full-angle method[30]	2
	Deductive database (DD)[10]	7
	DD+human-designed heuristics[17]	9
	DD+AR (ours)	14
	DD+AR+GPT-4 auxiliary constructions	15
	DD+AR+human-designed heuristics	18
	AlphaGeometry	25
	• Without pretraining	21
	• Without fine-tuning	23

We compare AlphaGeometry to other state-of-the-art methods (computer algebra and search approaches), most notably Wu's method. We also show the results of DD+AR (our contribution) and its variants, resulting in the strongest baseline DD+AR+human-designed heuristics. Finally, we include ablation settings for AlphaGeometry without pretraining and fine-tuning.



AlphaGeometry

- Two-stage system based on a deductive mechanism supported by a large language model
- Language model is responsible for heuristic construction of proof steps (learns patterns)
- Deductive mechanism verifies proof steps in formal programming language

Training on synthetic data

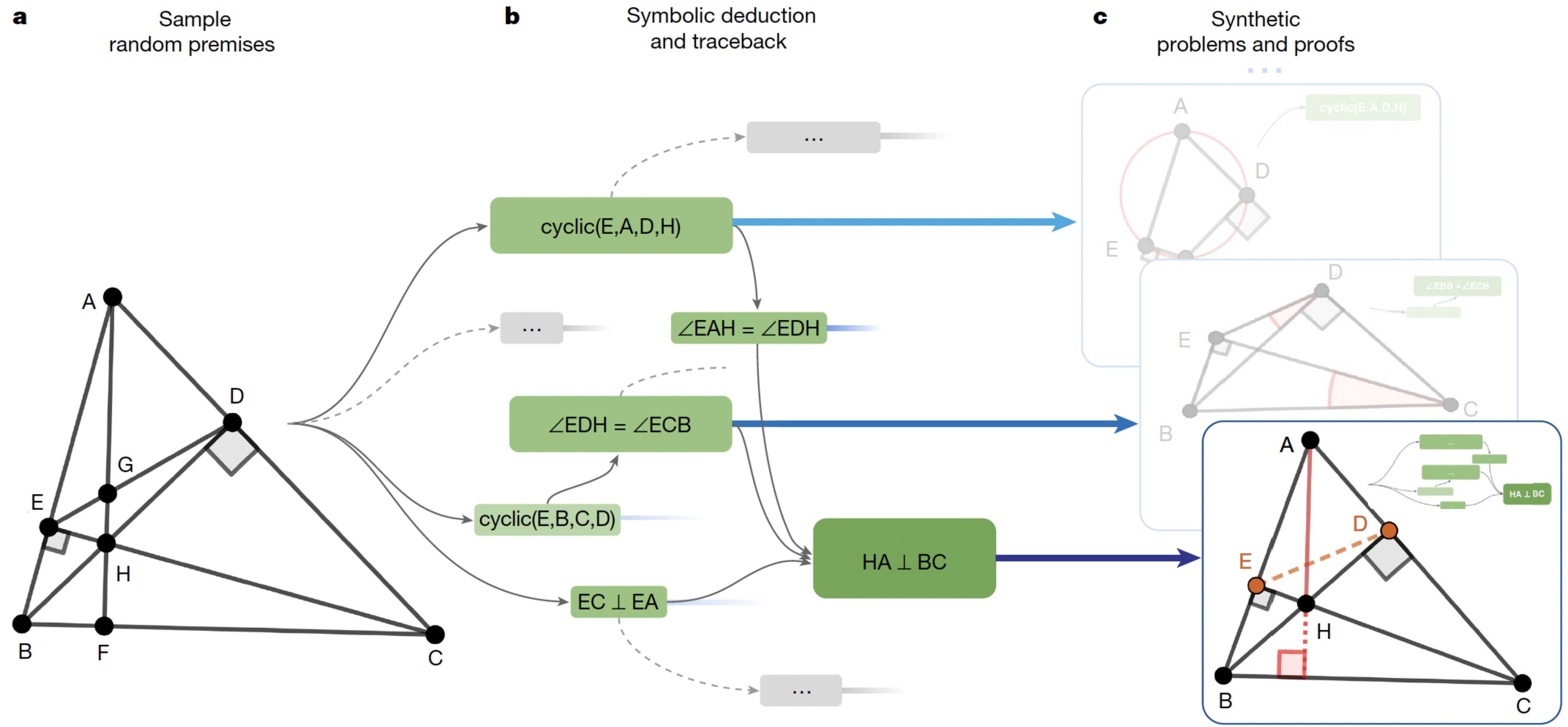
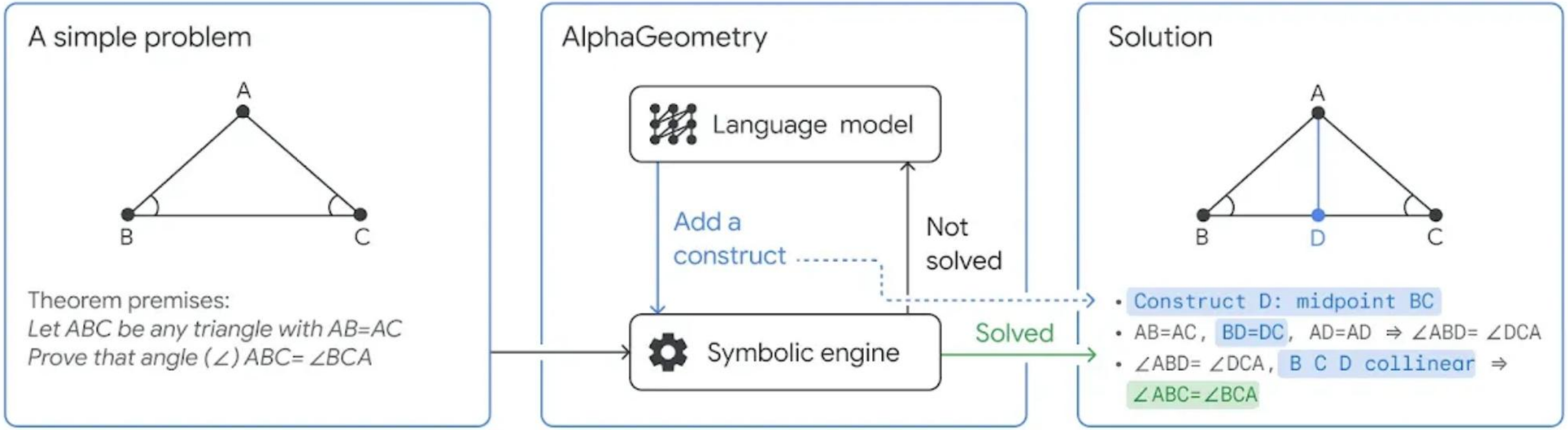


Fig. 3 | AlphaGeometry synthetic-data-generation process. a, We first sample a large set of random theorem premises. **b**, We use the symbolic deduction engine to obtain a deduction closure. This returns a directed acyclic graph of statements. For each node in the graph, we perform traceback to find its minimal set of necessary premise and dependency deductions. For example, for the rightmost node 'HA ⊥ BC', traceback returns the green subgraph. **c**, The minimal premise and the corresponding subgraph constitute a synthetic problem and its solution. In the bottom example, points E and D took part in the proof despite being irrelevant to the construction of HA and BC; therefore, they are learned by the language model as auxiliary constructions.



How does it work?

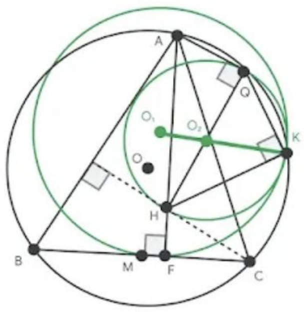
- AlphaGeometry solves a simple problem: Based on the problem diagram and its theoretical assumptions (left), AlphaGeometry (center) first uses its symbol engine to derive new statements about the diagram until it finds a solution or exhausts new statements.
- If a solution is not found, the AlphaGeometry language model adds one potentially useful construct (blue), opening new deduction paths for the symbolic engine.
- This loop continues until a solution is found (on the right). In this example, only one construction is required.

IMO difficult problems

- AlphaGeometry solves the Olympic problem:
- Problem 3 from the 2015 International Mathematical Olympiad (left) and a condensed version of the AlphaGeometry solution (right).
- Blue elements are added constructions. The AlphaGeometry solution includes 109 logical steps.

IMO 2015 P3

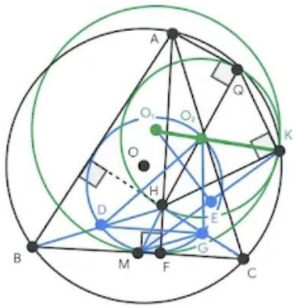
Let ABC be an acute triangle. Let (O) be its circumcircle, H its orthocenter, and F the foot of the altitude from A. Let M be the midpoint of BC. Let Q be the point on (O) such that QH ⊥ QA and let K be the point on (O) such that KH ⊥ KQ. Prove that the circumcircles (O_1) and (O_2) of triangles FKM and KQH are tangent to each other.



AlphaGeometry

Solution

[...]
Construct D: midpoint BH [a]
[a], O_2 midpoint HQ ⇒ BQ ∥ O_2D [20]
[...]
Construct G: midpoint HC [b]
$\angle GMD = \angle GO_2D$ ⇒ M O_2 G D cyclic [26]
[...]
[a],[b] ⇒ BC ∥ DG [30]
[...]
Construct E: midpoint MK [c]
[c] ⇒ $\angle KFC = \angle KO_1E$ [104]
[...]
$\angle FKO_1 = \angle FKO_2$ ⇒ KO_1 ∥ KO_2 [109]
[109] ⇒ O_1O_2K collinear ⇒ (O_1)(O_2) tangent



```
I0715 18:22:18.185354 140195178881472 training_loop.py:492] Training loop: creating task for mode beam_search
I0715 18:22:18.185726 140195178881472 training_loop.py:685] Creating logging writer (train) for mode beam_search
I0715 18:22:18.186243 140195178881472 training_loop.py:652] Compiling mode beam_search with jit.
I0715 18:22:18.186725 140195178881472 training_loop.py:89] registering functions: dict_keys([])
I0715 18:22:18.193519 140195178881472 graph.py:498] translated_imo_2000_p6
I0715 18:22:18.193661 140195178881472 graph.py:499] a b c = triangle a b c; d = orthocenter d a b c; e f g h = incenter2 e f g
e f; n = reflect n j f g; o = reflect o k f g; p = on_line p l m, on_line p n o ? cong h p h e
I0715 18:22:23.730203 140195178881472 ddar.py:60] Depth 1/1000 time = 5.354817628860474
I0715 18:22:33.802905 140195178881472 ddar.py:60] Depth 2/1000 time = 10.072417736053467
I0715 18:23:06.330376 140195178881472 ddar.py:60] Depth 3/1000 time = 32.52716135978699
I0715 18:23:33.040738 140195178881472 ddar.py:60] Depth 4/1000 time = 26.71002507209778
I0715 18:24:06.242242 140195178881472 ddar.py:60] Depth 5/1000 time = 33.20120716094971
I0715 18:24:45.153569 140195178881472 ddar.py:60] Depth 6/1000 time = 38.91098976135254
I0715 18:25:26.074163 140195178881472 ddar.py:60] Depth 7/1000 time = 40.92014789581299
I0715 18:26:06.920439 140195178881472 ddar.py:60] Depth 8/1000 time = 40.845935344696045
I0715 18:26:47.938530 140195178881472 ddar.py:60] Depth 9/1000 time = 40.92165970802307
I0715 18:27:29.389933 140195178881472 ddar.py:60] Depth 10/1000 time = 41.27131628990173
I0715 18:28:11.641505 140195178881472 ddar.py:60] Depth 11/1000 time = 42.08259844779968
I0715 18:28:53.577872 140195178881472 ddar.py:60] Depth 12/1000 time = 41.90423107147217
I0715 18:28:53.578418 140195178881472 alphageometry.py:221] DD+AR failed to solve the problem.
I0715 18:28:53.578677 140195178881472 alphageometry.py:539] Depth 0. There are 1 nodes to expand:
I0715 18:28:53.578778 140195178881472 alphageometry.py:543] {S} a : ; b : ; c : ; d : T a c b d 00 T a d b c 01 ; e : C b c e 0
8 ^ c a c h c h c b 09 ; i : C b c i 10 T a i b c 11 ; j : C a c j 12 T a c b j 13 ; k : C a b k 14 T a b c k 15 ; l : D e i e
 D g k g o 23 ; p : C l m p 24 C n o p 25 ? D h p h e {F1} x00
I0715 18:28:53.578863 140195178881472 alphageometry.py:548] Decoding from {S} a : ; b : ; c : ; d : T a c b d 00 T a d b c 01 
 a h a h a c 08 ^ c a c h c h c b 09 ; i : C b c i 10 T a i b c 11 ; j : C a c j 12 T a c b j 13 ; k : C a b k 14 T a b c k 15
: D f k f o 22 D g k g o 23 ; p : C l m p 24 C n o p 25 ? D h p h e {F1} x00
I0715 18:28:53.730479 140195178881472 decoder_stack.py:275] dstack: embeddings = Traced<ShapedArray(bfloat16[2,1,1024])>with<Dy
I0715 18:28:53.730726 140195178881472 decoder_stack.py:316] dstack: scanning over 1 windows.
I0715 18:28:53.730822 140195178881472 transformer_layer.py:657] tlayer: Skipping XL cache for mode beam_search.
I0715 18:28:53.730890 140195178881472 transformer_layer.py:657] tlayer: Skipping XL cache for mode beam_search.
```

Large Reasoning Models

Models trained on basic LLMs

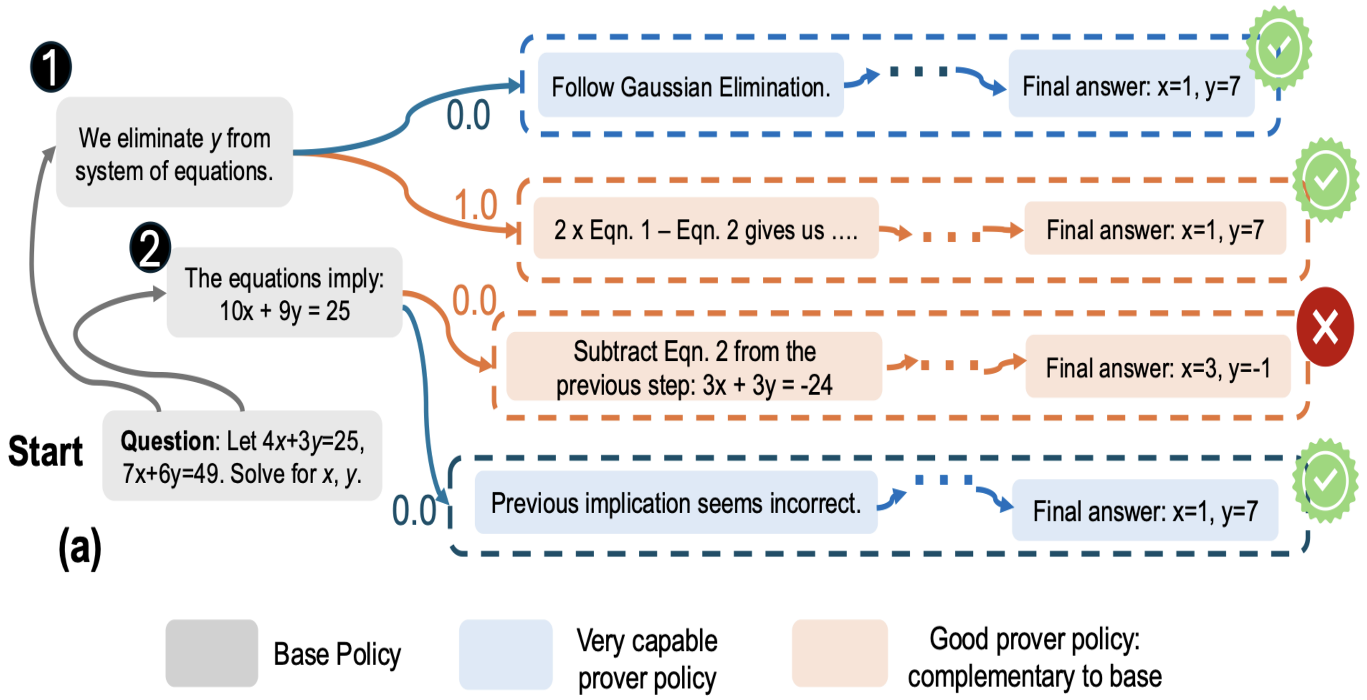
Characterized by a high proportion of RL-supported post-training in different variants

Trained on human and synthetic reasoning

The jump in quality of these models in the last few months is very clear

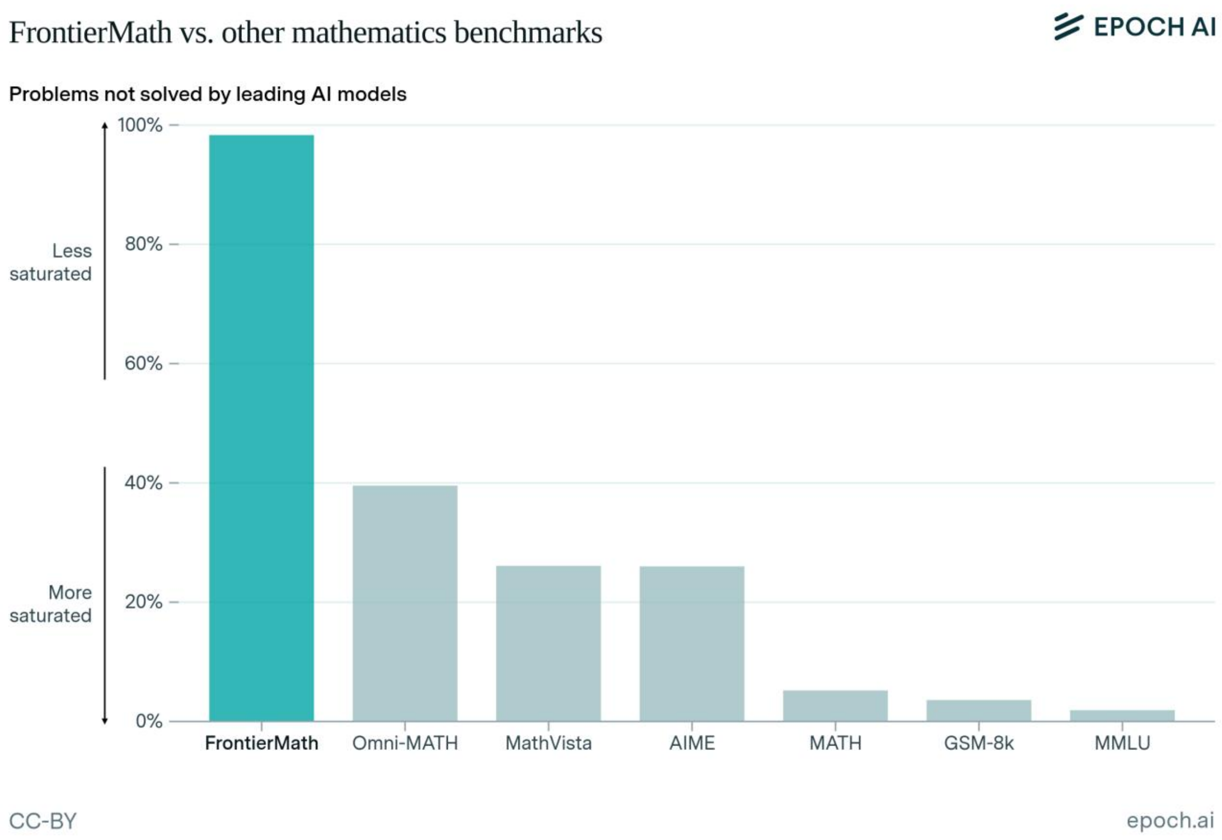
What is PRM (Process Reward Model)?

- **Main idea:** It evaluates **every step** in the AI thought process, not just the final result.
- **How it works.**
  Rewards the model for correct, logical intermediate steps.
- **Objective:**
  Teaches AI **how to think** and solve complex problems, not just guess answers.
- **Effect:**
  Significantly improved reasoning in math and logic tasks.



Arxiv: 2410.08146

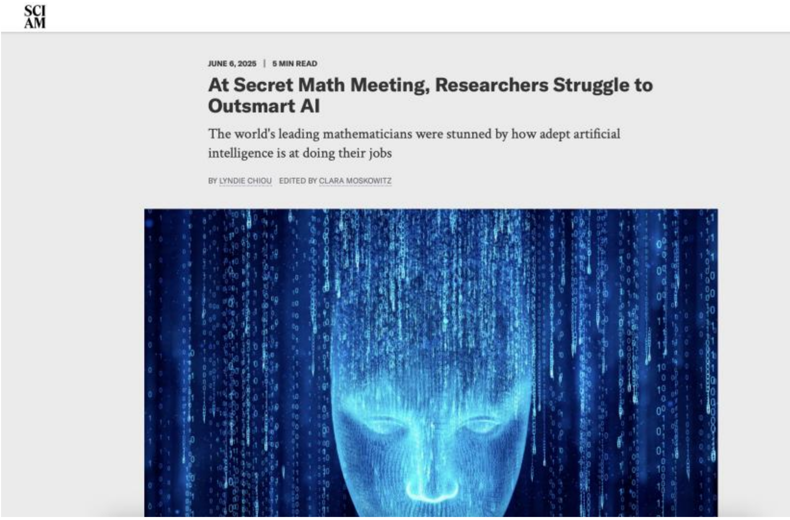
Classic benchmarks



- Classic math benchmarks are starting to become completely saturated
- Need to create benchmarks with a higher standard of tasks
- Real chance to measure useful features of models
- Attempts to construct such benchmarks are not always successful

FrontierMath Tier 4

EPOCH AI


SCI AM

JUNE 6, 2025 | 5 MIN READ

At Secret Math Meeting, Researchers Struggle to Outsmart AI

The world's leading mathematicians were stunned by how adept artificial intelligence is at doing their jobs

BY LYNDIE CHIOU EDITED BY CLARA MOSKOWITZ

- **Goal:** Create research-level math problems that challenge even leading experts.
- **Challenge:** Testing whether artificial intelligence will match the creativity and genius of human mathematicians.
- **Requirements:** Problems must have an unambiguous, computationally verifiable and "guess-proof" answer.
- **Process:** Experts create a problem and then provide a complete solution with verification code.
- **Collaboration:** project supported by OpenAI, with public testing conducted by Epoch AI.

Beginning new era

- The new models practically become a tool for building mathematics, comparing articles, and provide an opportunity for real discussion with the model.
- In my opinion, the convenience of using the new reasoning models is comparable to an assistant with very specialized qualifications

3. When it \emph{does} split

1. **Over a separably (or algebraically) closed residue field.** If $k = \overline{k}$ is separably closed, then:

- Every torsor under a connected smooth k-group (in particular under E_0) is trivial, because

$$H^1(\overline{k},\, E_0) \;=\; 0.$$

- Hence each component E_s^{ϕ} has a $\overline{k}$-point, and one can choose one point in each component so as to define a section
$\Phi_{\overline{k}} \to (E_s)_{\overline{k}}$.
- As a result, over $\overline{k}$ (or any strict henselization of R) the extension

$$0 \longrightarrow (E_0)_{\overline{k}} \longrightarrow (E_s)_{\overline{k}} \longrightarrow \Phi_{\overline{k}} \longrightarrow 0$$

\emph{does} split. In particular,

$$(E_s)_{\overline{k}} \;\simeq\; (E_0)_{\overline{k}} \times_{\overline{k}} \Phi_{\overline{k}},$$

as group-schemes.

2. **Over the strict Henselization or completion.** If R^{sh} is the strict Henselization of R, then its residue field is separably closed, so the same argument applies. Equivalently, after passing to $\backslash\mathrm{Spec} R^{\mathrm{sh}}$, the component group becomes a constant finite group, and each E_0-torsor in the special fibre is trivial.

Compare whether the methods to prove lower bounds on heights from the paper by Petsche are comparable or even similar to the ones from the paper of Naskręcki and Streng. I want to see how the generalizations from Hindry-Silverman to both other papers are comparable or not.

Is the approach of N-S redundant compared to Petsche?

Do you think the number field case of Petsche could be reproved and generalized with the techniques of N-S?

Thoughts (experimental) Auto

Expand to view model thoughts

This is an excellent and insightful question that gets to the heart of the evolution of methods for proving lower bounds on the canonical height. Let's break down the comparison between the papers by Petsche, Naskręcki & Streng (N-S), and their common ancestor, Hindry-Silverman (H-S).

Executive Summary

- **Petsche's method** is a significant and distinct refine[…] of one of Hindry & Silverman's approaches ([7] in his bibliography). Its main achievement is obtaining boun[…]ith **polynomial dependence on the field degree d**, a

Forecasts

- AlphaEvolve proved in May some new mathematical results (incremental growth)
- I expect that by 2028 some serious mathematical hypothesis will be completely proven by LRM
- Models will be created to make hypotheses independently and solve them (formally)
- Support in the revision of emerging texts
- Mathematics education will become very interactive

Generating math as code

Google DeepMind 2025-5-16

AlphaEvolve: A coding agent for scientific and algorithmic discovery

Alexander Novikov*, Ngân Vũ*, Marvin Eisenberger*, Emilien Dupont*, Po-Sen Huang*, Adam Zsolt Wagner*, Sergey Shirobokov*, Borislav Kozlovskii*, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli and Matej Balog*
Google DeepMind[1]

In this white paper, we present *AlphaEvolve*, an evolutionary coding agent that substantially enhances capabilities of state-of-the-art LLMs on highly challenging tasks such as tackling open scientific problems or optimizing critical pieces of computational infrastructure. *AlphaEvolve* orchestrates an autonomous pipeline of LLMs, whose task is to improve an algorithm by making direct changes to the code. Using an evolutionary approach, continuously receiving feedback from one or more evaluators, *AlphaEvolve* iteratively improves the algorithm, potentially leading to new scientific and practical discoveries. We demonstrate the broad applicability of this approach by applying it to a number of important computational problems. When applied to optimizing critical components of large-scale computational

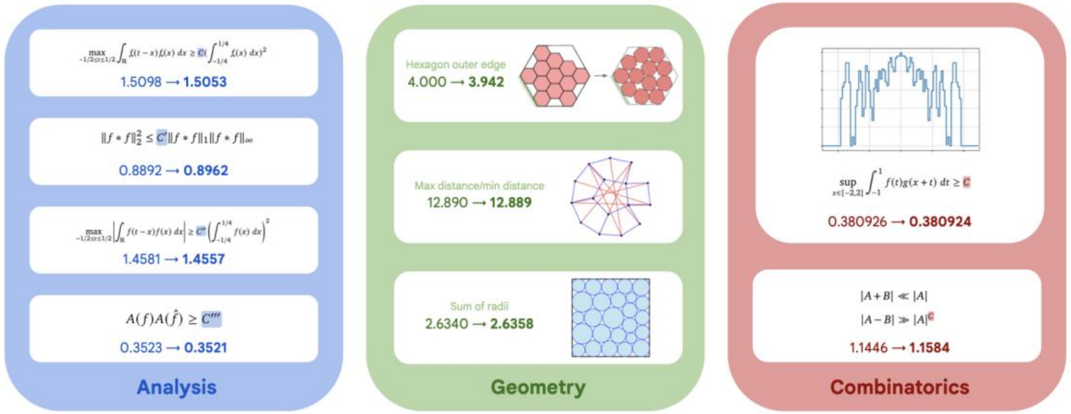


Figure 5 | Examples of SOTA-breaking mathematical constructions discovered with *AlphaEvolve*. The versatility of *AlphaEvolve* allows us to tackle problems in analysis (autocorrelation and uncertainty inequalities), geometry (packing and minimum/maximum distance problems) and combinatorics (Erdős's minimum overlap problem and sums and differences of finite sets).

Conclusion

$$\boxed{E' \;:\; (2-t^2)\,y^2 \;=\; x(x-1)(x-t^2) \quad \text{has} \quad \backslash\mathrm{rk}\,E'(\backslash\mathrm{Qbar}(t)) \;\geq 1,}$$

with the explicit infinite-order section

$$P \;=\; (2, \sqrt{2}).$$

You can of course do the same construction for \emph{any} constant $c \notin \{0, 1\}$: twist by $c - t^2$, then $x = c$ gives the section

$$P_c = (c, \sqrt{c(c-1)}) \in E'_c(\backslash\mathrm{Qbar}(t)),$$

and for generic c this is again infinite.

Generalization and hypotheses

- Models so far can't combine two completely separate areas of mathematics
- They can't find a meaningful generalization
- Most of the hypotheses they make are quite limited, but they are able to attempt to

The o4-mini model is already doing very well in generating solutions from the latest math

Problem. Let $F(\tau)$ be the unique weakly holomorphic modular function on $\mathrm{SL}_2(\mathbb{Z})$ satisfying:

(1) We have that
$$F\left(\frac{1+\sqrt{-3}}{2}\right) = F(i) = F\left(\frac{1+\sqrt{-7}}{2}\right) = F\left(\frac{-27+\sqrt{-2663}}{16}\right) = 0.$$

(2) If $q := e^{2\pi i\tau}$, then the Fourier expansion of $F(\tau)$ begins with the terms
$$F(\tau) = q^{-4} + \sum_{n\geq -3} a(n)q^n.$$

If K is the Galois closure of the smallest number field containing the coefficients of $F(\tau)$, then determine the height of the rational number
$$\alpha := \sum_{m=1}^{10}\left(\frac{\mathrm{Tr}_{K/\mathbb{Q}}(a(3))}{m!} - \frac{p_m+1}{p_m-1}\right),$$
where $p_1 = 2, p_2 = 3, \ldots$ are the prime numbers ordered by size.

- I'm working out the formula for α, which involves computing a sum of terms with specific coefficients. First, I need to compute the trace of a(m) for m = 1 to 10, using the equation a(m) = α_m J + β_m. The trace involves terms like α_m * Tr(J) and β_m multiplied by 86. With this, α depends on unknowns like Tr(J). It seems I should check if the sum of α_n/m! is zero. After that, I'll also compute the numeric sum to see if it's close to zero.
- It seems like the problem involves a potential misprint with the value √(-2663), where they likely meant √(-163), based on the context of Heegner numbers and rational j-values. The key connection is the j-value for discriminant -163, which produces a large rational j-value. The incorrect print-out could be an accidental error with the numbers and their formatting. If they meant to include √(-163), the adjusted calculation would yield the proper context, which makes sense in terms of class numbers and rational coefficients.
- I think the problem centers around calculating the height of a rational number α. It involves j-expansions and coefficients, and while the values involved are large, we can compute things like the minimal polynomial of J, traces, and potentially α itself. The process gives us a way to simplify the calculations and determine the height. It seems like the result approaches a value of 404669, and that viewing this calculation as a rational number could lead to the correct solution, possibly with a final height of 404669.

Analyzed

```
6235301*27720 == 404669*3628800
```

```
False
```

- I'm considering the F coefficients a(n), and I'm wondering if they're too large. I know that z expansions of j^4 give integer coefficients, so it seems like a(n) might turn out to be a rational integer. But, J must be a rational integer since based on disc -163, and that

First, one shows by the usual "elimination-in-j" argument that

$$F(\tau) \;=\; (j(\tau))^4 + (1647 - J)\,j(\tau)^3 + (-5\,832\,000 - 1647J)\,j(\tau)^2 + (5\,832\,000\,J)\,j(\tau)$$

where

$$J \;=\; j\!\left(\tfrac{-27+\sqrt{-2663}}{16}\right)$$

is the single algebraic coefficient. From this one computes—by convolving the known Fourier expansion of j up to q^{10}—that the coefficient of q^3 in F is

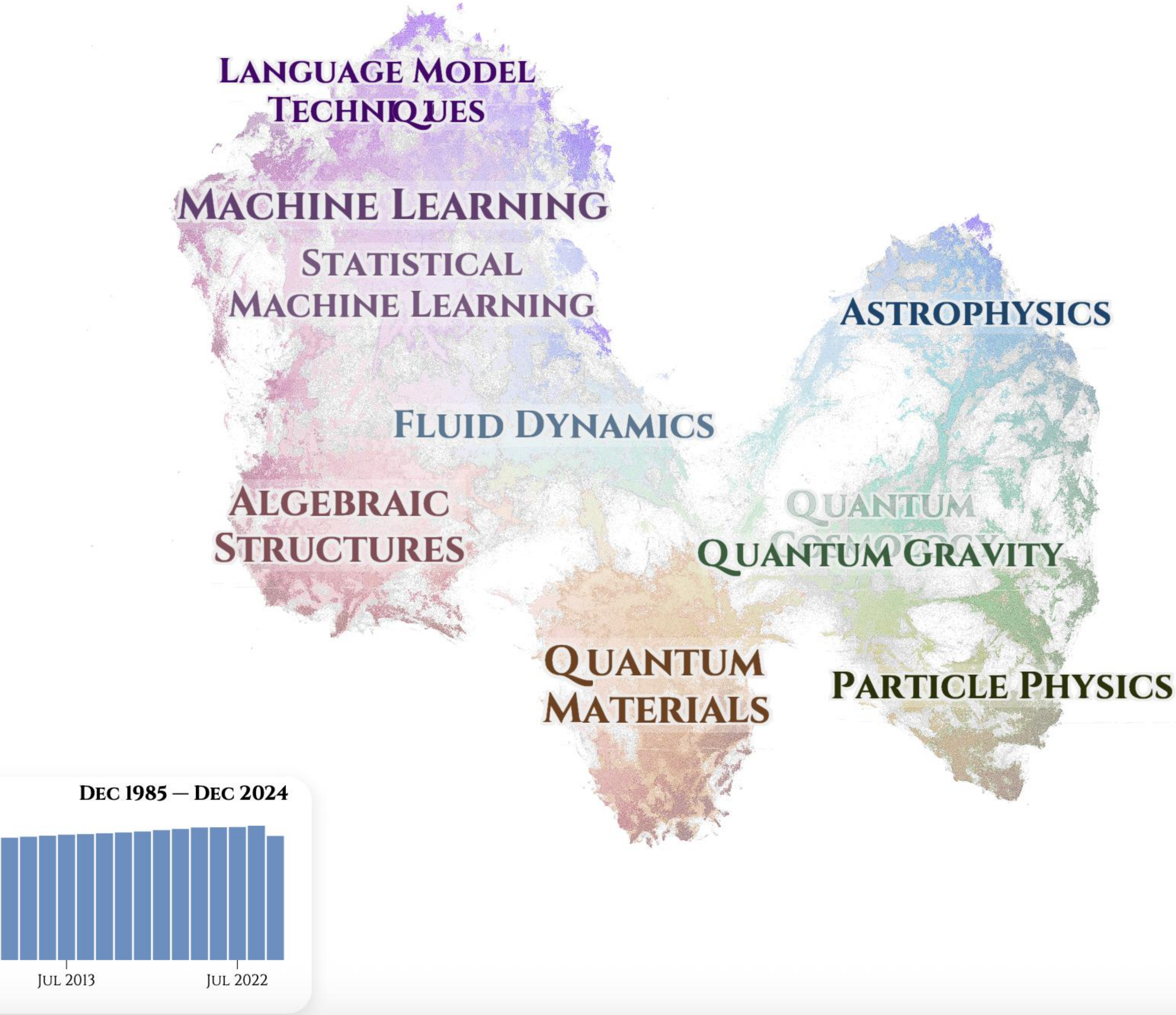
$$a(3) \;=\; \alpha_3\,J + \beta_3$$

with

$$\alpha_3 = -41\,538\,855\,420\,473\,232, \qquad \beta_3 = 62\,875\,447\,983\,452\,051\,280.$$

Since K is the Galois-closure of $\backslash\mathrm{Q}(J)$ one finds

$$\backslash\mathrm{Tr}_{K/\backslash\mathrm{Q}}(a(3)) \;=\; \alpha_3\,\backslash\mathrm{Tr}_{K/\backslash\mathrm{Q}}(J) \;+\; \beta_3\,[K : \backslash\mathrm{Q}].$$



Post-AI mathematics

- It's the idea that counts!
- Searching for patterns is easy
- Supplementing with found knowledge is easy
- Quite powerful resources are needed
- Testing different solutions at scale

Thank you for your attention