authors measured parasite growth rates using blood from several volunteers who carried either two copies of both rs112233623-T and rs9349205-A, two copies of just rs9349205-A, or neither of the variants. It is worth noting that the authors used parasite strains that have undergone adaptation to laboratory culture, and they differ genetically and epigenetically from natural strains. Nevertheless, these experiments clearly indicate that parasite growth is lower across several rounds of invasion, growth and egress in blood from individuals who carry two copies of rs112233623-T compared with those who do not.

The degree of growth inhibition in variant carriers seems to be similar to that seen in a condition called G6PD deficiency, which is caused by a different variant and is also thought to protect against malaria. This variant makes red blood cells susceptible to a cellular state called oxidative stress, which is characterized by a build-up of molecules known as reactive oxygen species (ROS). The authors observe a build-up of ROS in cultured red blood cells carrying rs112233623-T, leading them to propose that a similar mechanism underlies its malaria-protective effect.

Together, these results indicate that lowered expression of cyclin D3, a consequence of the rs112233623-T variant, naturally protects against malaria (Fig. 1b). However, this does raise a difficult problem. The gold-standard approach to confirm the relationship would be to observe consistent evidence for protection across both experimental and epidemiological settings – that is, to look at associations between genetics and observable traits in an endemic population. This type of evidence has been achieved, for example, for a genetic variant that underpins a blood type known as Dantu NE, which is rare globally but common in certain parts of East Africa. This variant impedes the malaria parasite's ability to invade red blood cells, and has been associated with a reduction of around 40% in the risk of severe malaria in East-African populations[6,7]. However, because the rs112233623-T variant is common only in Sardinia, where *P. falciparum* is no longer endemic, it is impossible to test it in an epidemiological study. Without this, the real-world impact of this variant is hard to determine.

Nevertheless, Marini and colleagues' findings point to a new pathway for protection against malaria. They raise the possibility that cyclin D-pathway inhibitors, some of which are already in use to slow the growth of some cancers[8,9], could be effective against malaria. A substantial challenge, however, would be to target erythroid progenitors without adversely affecting other processes.

Gavin Band is at the Centre for Human Genetics, University of Oxford, Oxford OX3 7BN, UK.
e-mail: gavin.band@well.ox.ac.uk

1. Marini, M. G. *et al. Nature* **651**, 698–706 (2026).
2. Majori, G. *Mediterr. J. Hematol. Infect. Dis.* **4**, e2012016 (2012).
3. Weatherall, J. D. in *Infectious Disease and Host–Pathogen Evolution* (ed. Dronamraju, K. R.) 18–36 (Cambridge Univ. Press, 2004).
4. Allison, A. C. *Br. Med. J.* **1**, 290–294 (1954).
5. Sankaran, V. G. *et al. Genes Dev.* **26**, 2075–2087 (2012).
6. Kariuki, S. N. *et al. Nature* **585**, 579–583 (2020).
7. Leffler, E. M. *et al. Science* **356**, eaam6393 (2017).
8. Morrison, L., Loibl, S. & Turner, N. C. *Nature Rev. Clin. Oncol.* **21**, 89–105 (2024).
9. Musgrove, E. A., Caldon, C. E., Barraclough, J., Stone, A. & Sutherland, R. L. *Nature Rev. Cancer* **11**, 558–572 (2011).

The author declares no competing interests.
This article was published online on 18 February 2025.

Artificial intelligence

Mathematicians put AlphaProof to the test

Talia Ringer

Mathematicians use computational tools to prove theorems. An AI model that is trained to use these tools might accelerate mathematical discovery. **See p.607**

Last year, Google DeepMind made headlines for building an artificial-intelligence system that achieved a silver-medal standard on the problems from that year's International Mathematics Olympiad (IMO), a prestigious competition for young mathematicians held annually. It was the first time such a system had performed at a medal-worthy level in solving IMO challenges. Now, on page 607, Hubert *et al.*[1] report the details of the core technology behind that achievement: a tool called AlphaProof, which is designed to prove mathematical statements.

In a talk this September, one of the authors likened AlphaProof's participation in the IMO to having "landed on the Moon". And indeed, it did kick off a space race of sorts, with tools from OpenAI and Google achieving scores worthy of a gold medal[2] on the problems from this year's IMO. In retrospect, last year's silver-medal performance might seem like a footnote. But science moves more slowly and carefully than the news cycle, and so it is not until now, more than a year after the announcement was made, that the details of AlphaProof have been made public. The wait was worth it.

AlphaProof follows previous work by DeepMind on AI tools that teach themselves to play games, such as AlphaZero for Go and chess. This means it treats mathematical proof like a game, using specialized tools called proof assistants, which mathematicians use to write proofs that a computer can check automatically.

Using a proof assistant is a lot like playing a game: a mathematician states a theorem, which they then try to prove interactively. At each step in the proof, the mathematician tries to get closer to their goal of proving the theorem – although they might introduce new goals along the way, in the form of other statements that must also be proved. To do this, the mathematician sends the proof assistant a high-level strategy called a tactic, and the assistant responds by updating the current goal or introducing new ones. If all goes well, the mathematician will eventually progress to a state in which no goals are left: this can be thought of as winning the game. To the proof assistant, this means that the mathematician has crafted a logical argument that a simple logic checker in the proof assistant can check; to the mathematician, it means that the theorem is proved.

Proof assistants provide a high level of certainty. This is useful for mathematicians, but it is even more useful for AI tools such as AlphaProof. AlphaProof can learn quickly by trying to prove a large number of theorems on its own, with immediate, trustworthy feedback from the proof assistant at each step. Furthermore, mathematicians can trust the proofs that AlphaProof produces (as long as the logic checker in the proof assistant is implemented correctly). By contrast, natural-language proofs produced by AI tools can contain subtle mistakes that are hard to spot (see, for example, go.nature.com/43uyhhf). Such errors are not a problem for AlphaProof.

AlphaProof was trained (see Fig. 1) to produce proofs that could be checked by a proof assistant called Lean (see https://lean-lang.org), which is popular among mathematicians, and includes a large library of mathematical theorems and proofs called mathlib (see go.nature.com/41shihf). The first training stage gave the algorithm familiarity with logic, code and mathematics. AlphaProof was then taught, using thousands of human-generated examples, to write proofs in Lean.

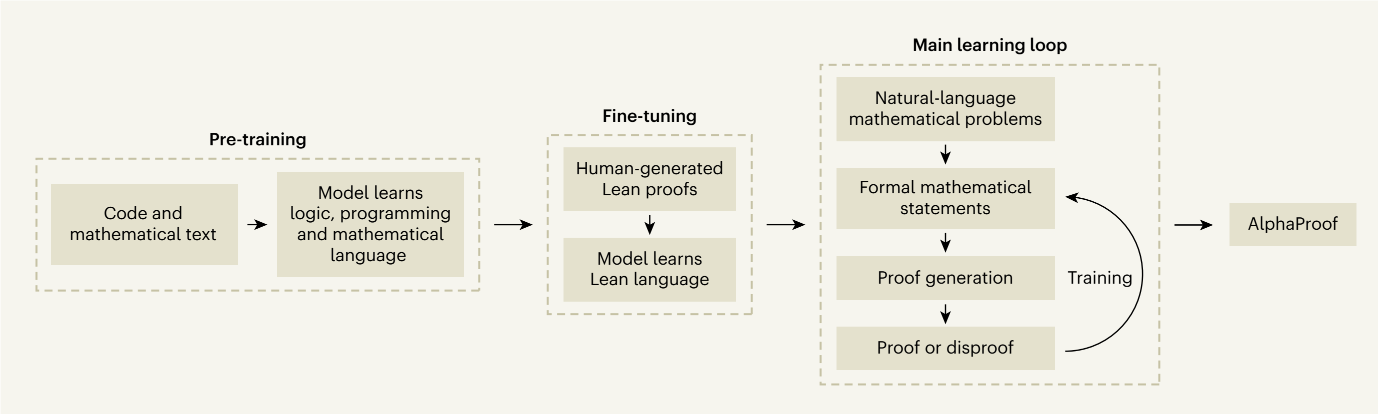

Figure 1 | Training an AI model to solve complex mathematical problems. Hubert *et al.*[1] report an AI model called AlphaProof that was trained, in three stages, to generate mathematical proofs. The first stage was pre-training, in which the model learnt logic, programming and mathematical language from a data set of code and mathematical text. Next, the fine-tuning stage specialized the model using a data set of statements written in the language of the Lean proof assistant, a mathematical tool used to generate proofs. In the main learning loop, the model was presented with a 'curriculum' of problems that had been translated into formal mathematical syntax from natural language. The model attempted to generate proofs using Lean, splitting each into a sequence of logical steps that could be computationally verified. As the model solved more problems, it adapted its approach to proof generation so that its performance improved.

But even with Lean in hand, one problem remains that makes mathematical proof particularly difficult. When you play Go or chess, you always play using the same board. But in the game of mathematical proof, the board is defined by the theorem that is being proved. Even if we consider just the theorems that humanity already knows, only a small portion have been stated in Lean. In other words, many of those boards are not immediately accessible for AlphaProof to play.

To address this difficulty, AlphaProof includes a language model that is based on Google's Gemini and is customized to translate statements from mathematical literature into a format that can be proved (or disproved) inside Lean — a task that is known as autoformalization[3]. Notably, it is OK if the autoformalization step is not quite perfect. AlphaProof will learn from trying to prove or disprove a statement even if it does not quite match the one defined in the mathematical literature. In fact, the system uses the language model to generate multiple translations of each mathematical statement.

The resulting formal statements — all 80 million of them — form the basis of the 'curriculum' that AlphaProof uses to teach itself to write proofs. Furthermore, when the system encounters a particularly hard problem, it can build a problem-specific curriculum by generating variants of that problem and trying to solve those first, although this is computationally expensive.

The computational power needed to build AlphaProof is on a scale that academic groups do not have access to without industrial partners. But it does seem to have paid off. Although AlphaProof is far from the first tool to make it easier to write proofs in a proof assistant[4], it stands out for its effectiveness. The silver-medal performance on the IMO questions, achieved in combination with a complementary tool called AlphaGeometry, is just one mark of this progress. AlphaProof also blew previous work out of the water on popular benchmarks based on competition mathematics.

I am one of a number of researchers who have, at some time, been granted temporary access to AlphaProof by DeepMind. To use it, I would state a theorem in Lean, write "sorry" in place of the proof (or a part of the proof) and ask AlphaProof to fill in the details. Either the tool would produce a machine-checkable proof or disproof, or it would fail to prove or disprove the theorem in the allotted time. In the case of failure, it sometimes helped to sketch out a rough idea of what I thought the proof should look like, guiding AlphaProof by providing the overall structure of the argument but writing "sorry" in place of the details. In the end, provided my theorem statement was correct, I could be certain that if AlphaProof proved the theorem, the proof was correct.

The paper focused on well-established mathematics competition benchmarks. But my experiences ultimately convinced me that it is useful beyond competition mathematics. Two PhD students affiliated with my laboratory each sent me one lemma — a small theorem that is used in the proof of a larger one — that they found tricky. AlphaProof proved one of the lemmas in under a minute, even though the student and their collaborator had been stuck on it for some time. It then disproved the other one, exposing a bug in a definition that the student had written, which they then fixed.

AlphaProof's propensity to find disproofs is perhaps its most surprisingly useful feature. At first, I had thought that it would be useful only when definitions and theorems were stated correctly, but I instead found that it could help me work out where I had gone wrong in formulating a theorem. Alex Kontorovich, a mathematician at Rutgers University in Piscataway, New Jersey, also found AlphaProof useful for exposing such mistakes. "Each time, I could usually quickly identify what assumption I had omitted, adjust the statement, and iterate," he wrote[5]. "This back-and-forth between proposing statements and checking whether they can be established (or refuted) was essential to getting the formalization right."

The proofs themselves are also useful. Floris van Doorn, a mathematician at the University of Bonn in Germany, tested AlphaProof on proofs from an active project he was working on in Lean. He found it helpful, especially for shorter proofs, but also for some longer ones. "If this becomes publicly available," he told me, "I can imagine a workflow where you just call AlphaProof on newly stated lemmas while you're continuing your work on something else. Then you only spend time on proving the lemma if AlphaProof cannot find it."

By contrast, Kevin Buzzard, a mathematician at Imperial College London, did not find AlphaProof useful for translating the proof of Fermat's Last Theorem into Lean. The difference between his experience and my own probably lies in the number of custom definitions in our proofs. I had deliberately asked the PhD students for lemmas that mostly used standard definitions that were already in mathlib; Buzzard's proof development, however, was full of what he described as "bespoke definitions". "In my experience," he wrote, "no AI system is anywhere near useful to me right now."

This difficulty in dealing with "bespoke definitions" aligns with the results of the paper. When tackling the IMO questions, AlphaProof excelled at problems that drew on concepts that were already defined in mathlib. But on problems that relied on concepts that, at the time of the IMO 2024 competition, had not been defined in mathlib, AlphaProof struggled. To solve these problems, the authors relied on a specialized tool called AlphaGeometry. A natural future avenue, then, is to target proofs that use custom definitions. This might require updates to the benchmarks used to evaluate and compare models (because current benchmarks typically assume fixed definitions that already exist), as well as further innovations in AI.

Despite its limitations, AlphaProof is the first AI tool I have used that I found concretely useful for writing proofs, a sentiment shared by some, but not all, of the mathematicians I have corresponded with. The lead author noted in his talk that there are plans to make the tool publicly available, although it is unclear to me how that would work given the financial and business constraints under which companies such as Google operate. In any case, it is clear to me that something is changing in this field; perhaps AlphaProof is a harbinger of what is to come.

Talia Ringer is in the Siebel School of Computing and Data Science, University of Illinois at Urbana–Champaign, Urbana, Illinois, 61801, USA.
e-mail: tringer@illinois.edu

1. Hubert, T. *et al. Nature* **651**, 607–613 (2026).
2. *Nature* **644**, 20 (2025).
3. Wu, Y. *et al.* Preprint at arXiv https://doi.org/10.48550/arXiv.2205.12615 (2022).
4. Ringer, T., Palmskog, K., Sergey, I., Gligoric, M. & Tatlock, Z. *Found. Trends Program. Lang.* **5**, 102–281 (2019).
5. Kontorovich, A. Preprint at arXiv https://doi.org/10.48550/arXiv.2510.15924 (2025).

The author declares no competing interests.
This article was published online on 12 November 2025.

Medical research

AI succeeds in diagnosing rare diseases

Timo Lassmann

An artificial-intelligence system uses clinical data, genetic information and literature searches to suggest diagnoses and provides the underlying reasoning. **See p.775**

For the estimated 300 million people[1] living with a rare disease worldwide, obtaining an accurate diagnosis remains an arduous journey. The 'diagnostic odyssey' often takes five years or more[2], and is marked by repeated specialist consultations, misdiagnoses and unnecessary treatments. With more than 7,000 rare diseases recognized[3], most of which affect fewer than one in 2,000 individuals, even experienced clinicians struggle to connect unfamiliar patterns of symptoms with their underlying causes. On page 775, Zhao *et al.*[4] present an artificial-intelligence system called DeepRare, which uses a range of specialized tools and knowledge sources to generate ranked diagnostic hypotheses for rare diseases, each accompanied by reasoning that links the conclusions to verifiable medical evidence.

The architecture of DeepRare reflects a broader shift in how AI systems tackle complex medical problems. Rather than relying on a single model to produce answers, the system uses a coordinated approach. There is a central reasoning engine, powered by an AI system similar to those underlying chatbots such as ChatGPT, which coordinates multiple specialized steps, each responsible for distinct analytical tasks. One module extracts standardized symptom descriptions from clinical notes. Another searches the medical literature. A third retrieves similar cases from curated databases. Yet another analyses genetic variants when DNA-sequencing data are available. This division of labour mirrors the multidisciplinary team consultations that characterize assessments by specialists at rare-disease centres, but the AI tool operates at a scale and speed that is impossible for human teams.

The results are striking (Fig. 1). In a real-world clinical validation, DeepRare's top-ranked prediction was correct 64% of the time, outperforming experienced rare-disease physicians who had access to search engines. When the correct diagnosis was among the system's five ranked suggestions, accuracy reached 78.5%, compared with 65.6% for physicians. Perhaps most notably for clinical adoption, medical experts validated the system's reasoning chains at a level of 95% agreement. This indicates that the explanations linking clinical data to diagnostic conclusions are clinically coherent and verifiable.

The system's traceable reasoning might prove to be its most notable contribution to possible future use in the clinic. Rare-disease diagnosis is not merely about producing the correct answer; clinicians also need to understand why a diagnosis is plausible and what evidence supports it. DeepRare's citation of specific sources, case reports, database entries and literature references provides an audit trail that supports, rather than supplants, clinical judgement. This transparency distinguishes the system from opaque AI approaches that might perform well but offer clinicians no basis for evaluating the recommendations made by AI. In high-stakes medical decisions, such interpretability is crucial.

What mechanisms underpin this high-performance AI? The authors systematically tested which components contribute most to accuracy, revealing two key factors. One is the retrieval of up-to-date medical knowledge from the web, and the other is a feature called self-reflection loops, which prompt the system to re-evaluate its preliminary conclusions. These capabilities prove particularly valuable given the rapidly evolving understanding of rare diseases, with hundreds of new genetic conditions described annually[5]. More generally, retrieval of current knowledge and iterative self-reflection are broadly applicable techniques that work across many different fields. Any field that requires reasoning using

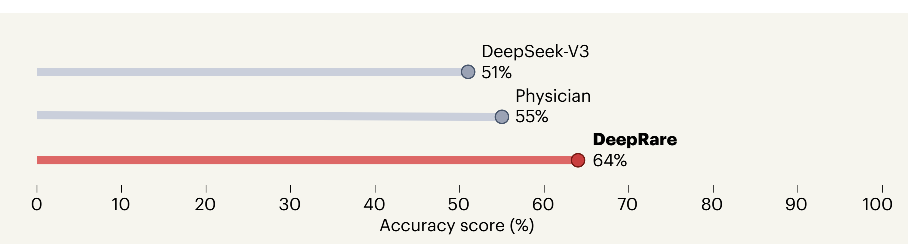

Figure 1 | Using artificial intelligence to diagnose rare diseases. Zhao *et al.*[4] present a tool called DeepRare that can suggest five ranked diagnoses for rare diseases. The system uses a variety of information sources, including clinical data, DNA-sequencing results and searches of published papers. The results it provided are more accurate than those obtained by an early AI tool called DeepSeek-V3 or predictions made by physicians with access to a search engine. The score shown represents how often the system's top prediction was accurate. (Adapted from Fig. 3c of ref. 4.)