

# Naukowe odkrycia dokonywane przez algorytmy sztucznej inteligencji

czyli czy sztuczna inteligencja  
otrzyma Nagrodę Nobla

Adam Żychowski



# Nobel Turing Challenge

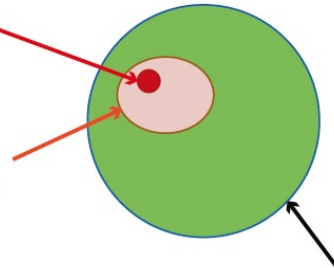
*“The Nobel Turing Challenge is a grand challenge aiming at developing a highly autonomous AI and robotics system that can make major scientific discoveries, some which may be worthy of the Nobel Prize and even beyond.”*

[www.nobelturingchallenge.org](http://www.nobelturingchallenge.org)

## a Game of GO

Game of GO recorded in the past

Game of GO played and learned by AlphaGo



AlphaGo Zero generated possible moves out of an entire state space

An entire Game of GO (Approximately  $10^{170}$  state space complexity and  $10^{360}$  game tree complexity)

## b Scientific Discovery

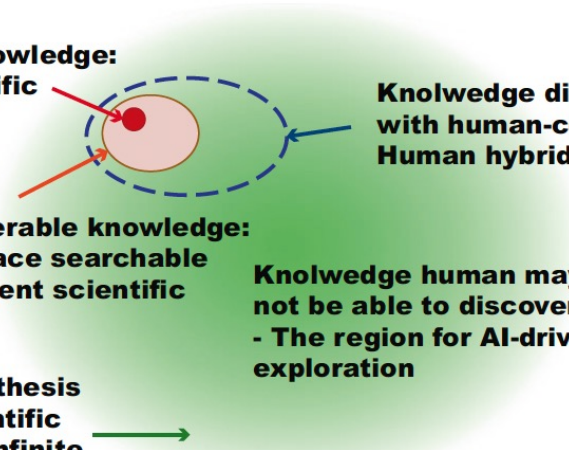
Discovered knowledge: Current scientific knowledge

Human discoverable knowledge: Hypothesis space searchable extending current scientific knowledge

An entire hypothesis space for scientific knowledge is infinite or undefinable (a boundary is not clear)

Knowledge discoverable with human-centric AI-Human hybrid system

Knowledge human may not be able to discover - The region for AI-driven exploration



Kitano, H. (2016). *Artificial intelligence to win the Nobel prize and beyond: Creating the engine for scientific discovery*. *AI magazine*, 37(1), 39-49.

Kitano, H. (2021). *Nobel Turing Challenge: creating the engine for scientific discovery*. *NPJ systems biology and applications (Nature Portfolio)*, 7(1), 29.

# Twierdzenie o czterech barwach

1850 - sformułowanie hipotezy

1879 - pierwszy "dowód", Alfred Kempe

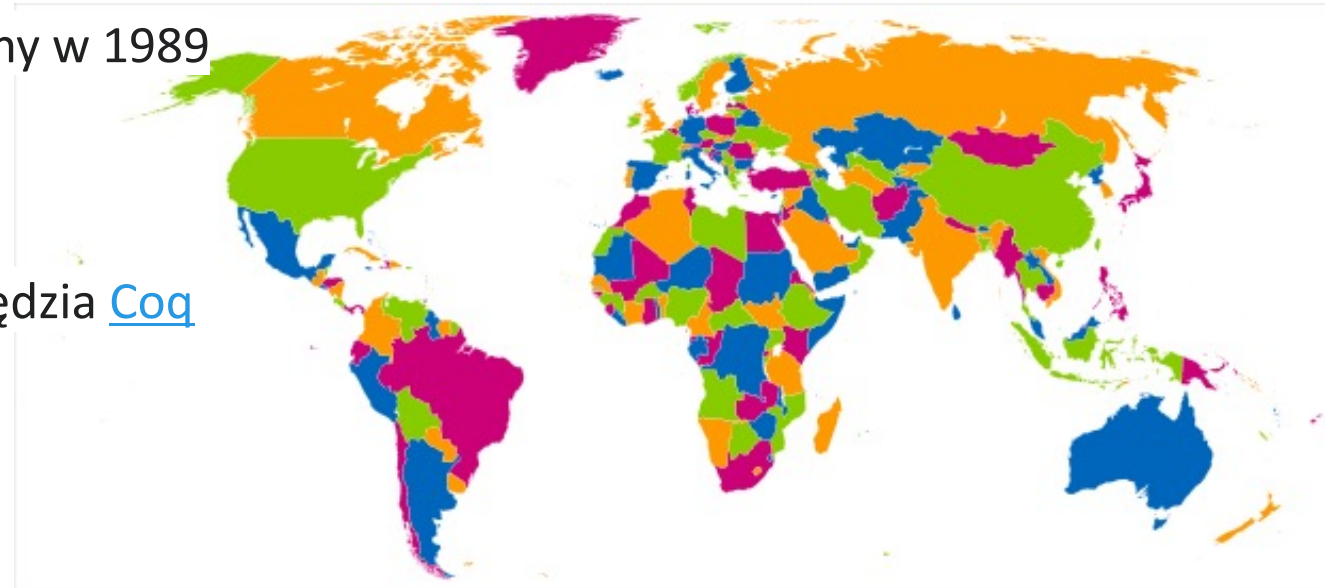
1890 - odkrycie błędu w dowodzie (1896 niezależnie odkrywa błąd Vallée Poussin)

1976 - Kenneth Appel i Wolfgang Haken ogłosili dowód przy użyciu komputera, który rozpatrzył 1482 przypadki – ponad 400 stron dowodu, praca opublikowana w 1981

1986 - znaleziono błąd w obliczeniach, poprawiony w 1989

1996 - redukcja do 633 przypadków

2005 - sformalizowany dowód przy pomocy narzędzia [Coq](#)

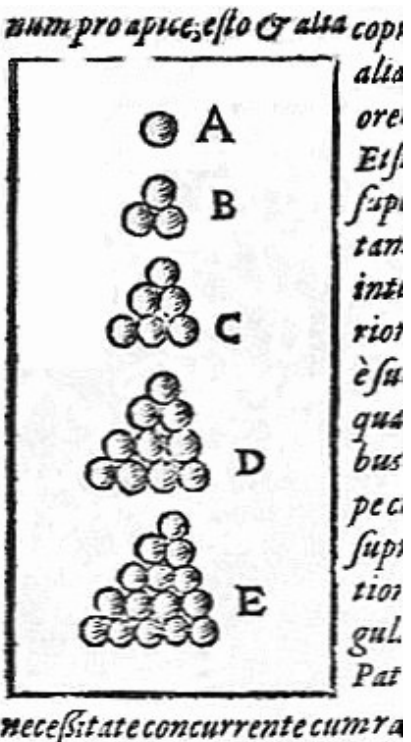
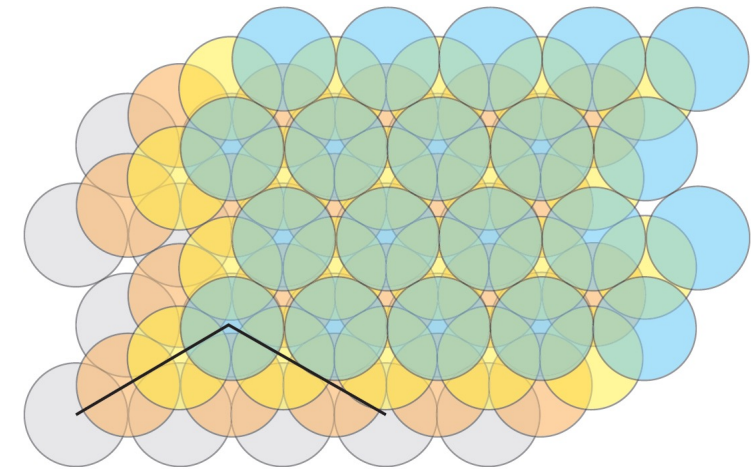


# Postulat Keplera (1611)

Trójwymiarowe kule w trójwymiarowej przestrzeni najciaśniej da się umieścić, gdy ich środki tworzą na płaszczyznach przekroju sześciokąty. Gęstość:  $\frac{\pi}{2\sqrt{3}} \approx 0.7404\dots$



- Tomas Hales – dowód w kilku pracach (usprawianiany) w latach 1994 - 1998
- ponad 250 stron i kilka GB danych
- programowanie liniowe i całkowitoliczbowe –  $10^5$  zagadnień, każde 100-200 zmiennych
- po 4 latach recenzji (12 osób) opublikowany w 2005 roku w *Annals of Mathematics* z dopiskiem, że recenzenci są w 99% pewni jego poprawności
- ostateczna praca: 120 stron, rezultat 20 lat badań, Tomasa Halesa i 21 jego współpracowników



# Boolean Pythagorean triples

Zbiór  $\{1, \dots, 7824\}$  może być podzielony na dwa podzbiory, takie że żaden z nich nie zawiera trójki pitagorejskiej ( $a^2 + b^2 = c^2$ ), lecz nie jest to możliwe dla zbiorów  $\{1, \dots, 7825\}$ .

- *Udowodnione* w 2016.
- SAT solver
- 4 CPU years, 200 terabajtów wygenerowanych danych. Najdłuższy zatwierdzony dowód.

Heule, M. J., Kullmann, O., & Marek, V. W. (2016). ***Solving and verifying the boolean pythagorean triples problem via cube-and-conquer.*** In International Conference on Theory and Applications of Satisfiability Testing (pp. 228-245).

# AlphaTensor

$$\begin{bmatrix} 1 & 0 & 2 \\ 3 & 1 & 0 \\ 5 & -1 & 2 \end{bmatrix} \times \begin{bmatrix} 2 & -1 & 0 \\ 5 & 1 & -1 \\ -2 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -2 & -1 & 0 \\ 11 & -2 & -1 \\ 1 & -6 & 1 \end{bmatrix}$$

$$3 \times 2 + 1 \times 5 + 0 \times -2 = 11$$

$$\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \times \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{bmatrix}$$

Standard algorithm

$$\begin{aligned} h_1 &= a_{1,1} b_{1,1} \\ h_2 &= a_{1,1} b_{1,2} \\ h_3 &= a_{1,2} b_{2,1} \\ h_4 &= a_{1,2} b_{2,2} \\ h_5 &= a_{2,1} b_{1,1} \\ h_6 &= a_{2,1} b_{1,2} \\ h_7 &= a_{2,2} b_{2,1} \\ h_8 &= a_{2,2} b_{2,2} \end{aligned}$$

$$\begin{aligned} c_{1,1} &= h_1 + h_3 \\ c_{1,2} &= h_2 + h_4 \\ c_{2,1} &= h_5 + h_7 \\ c_{2,2} &= h_6 + h_8 \end{aligned}$$

Strassen's algorithm

$$\begin{aligned} h_1 &= (a_{1,1} + a_{2,2})(b_{1,1} + b_{2,2}) \\ h_2 &= (a_{2,1} + a_{2,2})b_{1,1} \\ h_3 &= a_{1,1}(b_{1,2} - b_{2,2}) \\ h_4 &= a_{2,2}(-b_{1,1} + b_{2,1}) \\ h_5 &= (a_{1,1} + a_{1,2})b_{2,2} \\ h_6 &= (-a_{1,1} + a_{2,1})(b_{1,1} + b_{1,2}) \\ h_7 &= (a_{1,2} - a_{2,2})(b_{2,1} + b_{2,2}) \end{aligned}$$

$$\begin{aligned} c_{1,1} &= h_1 + h_4 - h_5 + h_7 \\ c_{1,2} &= h_3 + h_5 \\ c_{2,1} &= h_2 + h_4 \\ c_{2,2} &= h_1 - h_2 + h_3 + h_6 \end{aligned}$$

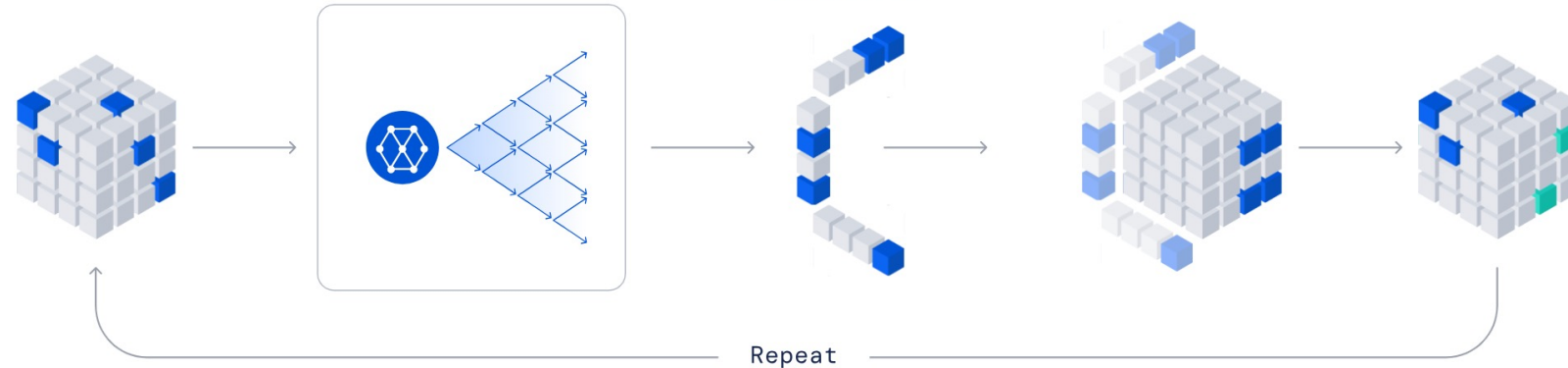
Current state

AlphaTensor

Algorithmic instruction

State update

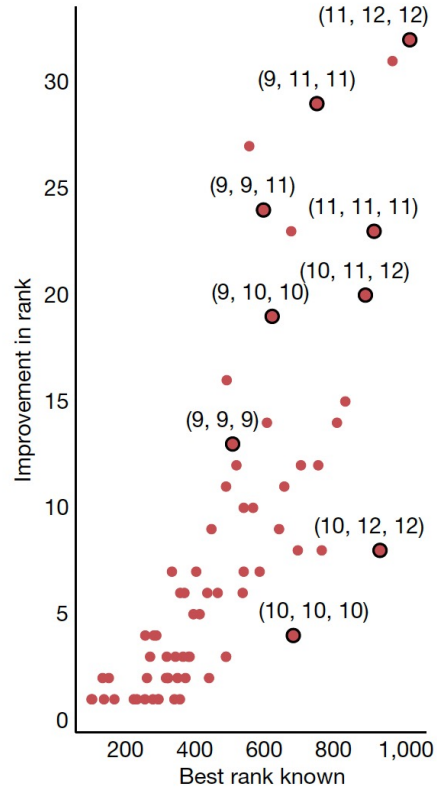
New state



# AlphaTensor

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & a_{1,5} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} & a_{2,5} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} & a_{3,5} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} & a_{4,5} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} & b_{1,5} \\ b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} & b_{2,5} \\ b_{3,1} & b_{3,2} & b_{3,3} & b_{3,4} & b_{3,5} \\ b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} & b_{4,5} \\ b_{5,1} & b_{5,2} & b_{5,3} & b_{5,4} & b_{5,5} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} & c_{1,5} \\ c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} & c_{2,5} \\ c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} & c_{3,5} \\ c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} & c_{4,5} \end{pmatrix}$$

Size (n, m, p)	Best method known	Best rank known	AlphaTensor rank Modular Standard
(2, 2, 2)	(Strassen, 1969) <sup>2</sup>	7	7 7 7
(3, 3, 3)	(Laderman, 1976) <sup>15</sup>	23	23 23 23
(4, 4, 4)	(Strassen, 1969) <sup>2</sup> (2, 2, 2) ⊗ (2, 2, 2)	49	47 49
(5, 5, 5)	(3, 5, 5) + (2, 5, 5)	98	96 98
(2, 2, 3)	(2, 2, 2) + (2, 2, 1)	11	11 11 11
(2, 2, 4)	(2, 2, 2) + (2, 2, 2)	14	14 14 14
(2, 2, 5)	(2, 2, 2) + (2, 2, 3)	18	18 18 18
(2, 3, 3)	(Hopcroft and Kerr, 1971) <sup>16</sup>	15	15 15 15
(2, 3, 4)	(Hopcroft and Kerr, 1971) <sup>16</sup>	20	20 20 20
(2, 3, 5)	(Hopcroft and Kerr, 1971) <sup>16</sup>	25	25 25 25
(2, 4, 4)	(Hopcroft and Kerr, 1971) <sup>16</sup>	26	26 26 26
(2, 4, 5)	(Hopcroft and Kerr, 1971) <sup>16</sup>	33	33 33 33
(2, 5, 5)	(Hopcroft and Kerr, 1971) <sup>16</sup>	40	40 40 40
(3, 3, 4)	(Smirnov, 2013) <sup>18</sup>	29	29 29 29
(3, 3, 5)	(Smirnov, 2013) <sup>18</sup>	36	36 36 36
(3, 4, 4)	(Smirnov, 2013) <sup>18</sup>	38	38 38 38
(3, 4, 5)	(Smirnov, 2013) <sup>18</sup>	48	47 47 47
(3, 5, 5)	(Sedoglavic and Smirnov, 2021) <sup>19</sup>	58	58 58 58
(4, 4, 5)	(4, 4, 2) + (4, 4, 3)	64	63 63 63
(4, 5, 5)	(2, 5, 5) ⊗ (2, 1, 1)	80	76 76 76



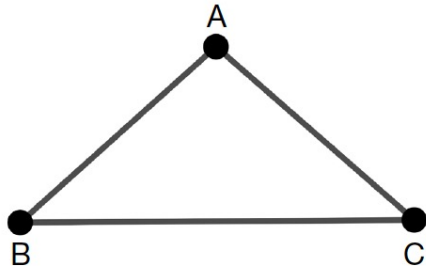
$$\begin{aligned} h_1 &= a_{3,2}(-b_{2,1} - b_{2,5} - b_{3,1}) \\ h_2 &= (a_{2,2} + a_{2,5} - a_{3,5})(-b_{2,5} - b_{5,1}) \\ h_3 &= (-a_{3,1} - a_{4,1} + a_{4,2})(-b_{1,1} + b_{2,5}) \\ h_4 &= (a_{1,2} + a_{1,4} + a_{3,4})(-b_{2,5} - b_{4,1}) \\ h_5 &= (a_{1,5} + a_{2,2} + a_{2,5})(-b_{2,4} + b_{5,1}) \\ h_6 &= (-a_{2,2} - a_{2,5} - a_{4,5})(b_{2,3} + b_{5,1}) \\ h_7 &= (-a_{1,1} + a_{4,1} - a_{4,2})(b_{1,1} + b_{2,4}) \\ h_8 &= (a_{3,2} - a_{3,3} - a_{4,3})(-b_{2,3} + b_{3,1}) \\ h_9 &= (-a_{1,2} - a_{1,4} + a_{4,4})(b_{2,3} + b_{4,1}) \\ h_{10} &= (a_{2,2} + a_{2,5})b_{5,1} \\ h_{11} &= (-a_{2,1} - a_{4,1} + a_{4,2})(-b_{1,1} + b_{2,2}) \\ h_{12} &= (a_{4,1} - a_{4,2})b_{1,1} \\ h_{13} &= (a_{1,2} + a_{1,4} + a_{2,4})(b_{2,2} + b_{4,1}) \\ h_{14} &= (a_{1,3} - a_{3,2} + a_{3,3})(b_{2,4} + b_{3,1}) \\ h_{15} &= (-a_{1,2} - a_{1,4})b_{4,1} \\ h_{16} &= (-a_{3,2} + a_{3,3})b_{3,1} \\ h_{17} &= (a_{1,2} + a_{1,4} - a_{2,1} + a_{2,2} - a_{2,3} + a_{2,4} - a_{3,2} + a_{3,3} - a_{4,1} + a_{4,2})b_{2,2} \\ h_{18} &= a_{2,1}(b_{1,1} + b_{1,2} + b_{5,2}) \\ h_{19} &= -a_{2,3}(b_{3,1} + b_{3,2} + b_{5,2}) \\ h_{20} &= (-a_{1,5} + a_{2,1} + a_{2,3} - a_{2,5})(-b_{1,1} - b_{1,2} + b_{1,4} - b_{5,2}) \\ h_{21} &= (a_{2,1} + a_{2,3} - a_{2,5})b_{5,2} \\ h_{22} &= (a_{1,3} - a_{1,4} - a_{2,4})(b_{1,1} + b_{1,2} - b_{1,4} - b_{3,1} - b_{3,2} + b_{3,4} + b_{4,4}) \\ h_{23} &= a_{1,3}(-b_{3,1} + b_{3,4} + b_{4,4}) \\ h_{24} &= a_{1,5}(-b_{4,4} - b_{5,1} + b_{5,4}) \\ h_{25} &= -a_{1,1}(b_{1,1} - b_{1,4}) \\ h_{26} &= (-a_{1,3} + a_{1,4} + a_{1,5})b_{4,4} \\ h_{27} &= (a_{1,3} - a_{3,1} + a_{3,3})(b_{1,1} - b_{1,4} + b_{1,5} + b_{3,5}) \\ h_{28} &= -a_{3,4}(-b_{3,5} - b_{4,1} - b_{4,5}) \\ h_{29} &= a_{3,1}(b_{1,1} + b_{1,5} + b_{3,5}) \\ h_{30} &= (a_{3,1} - a_{3,3} + a_{3,4})b_{3,5} \\ h_{31} &= (-a_{1,4} - a_{1,5} - a_{3,4})(-b_{4,4} - b_{5,1} + b_{5,4} - b_{5,5}) \\ h_{32} &= (a_{2,1} + a_{4,1} + a_{4,4})(b_{1,3} - b_{4,1} - b_{4,2} - b_{4,3}) \\ h_{33} &= a_{4,3}(-b_{3,1} - b_{3,3}) \\ h_{34} &= a_{4,4}(-b_{1,3} + b_{4,1} + b_{4,3}) \\ h_{35} &= -a_{4,5}(b_{1,3} + b_{5,1} + b_{5,3}) \\ h_{36} &= (a_{2,3} - a_{2,5} - a_{4,5})(b_{3,1} + b_{3,2} + b_{3,3} + b_{5,2}) \\ h_{37} &= (-a_{4,1} - a_{4,4} + a_{4,5})b_{1,3} \\ h_{38} &= (-a_{2,3} - a_{3,1} + a_{3,3} - a_{3,4})(b_{3,5} + b_{4,1} + b_{4,2} + b_{4,5}) \\ h_{39} &= (-a_{3,1} - a_{4,1} - a_{4,4} + a_{4,5})(b_{1,3} + b_{5,1} + b_{5,3} + b_{5,5}) \\ h_{40} &= (-a_{1,3} + a_{1,4} + a_{1,5} - a_{4,4})(-b_{3,1} - b_{3,3} + b_{3,4} + b_{4,4}) \\ h_{41} &= (-a_{1,1} + a_{4,1} - a_{4,5})(b_{1,3} + b_{3,1} + b_{3,3} - b_{3,4} + b_{5,1} + b_{5,3} - b_{5,4}) \\ h_{42} &= (-a_{2,1} + a_{2,5} - a_{3,5})(-b_{1,1} - b_{1,2} - b_{1,5} + b_{4,1} + b_{4,2} + b_{4,5} - b_{5,2}) \\ h_{43} &= a_{2,4}(b_{4,1} + b_{4,2}) \\ h_{44} &= (a_{2,3} + a_{3,2} - a_{3,3})(b_{2,2} - b_{3,1}) \\ h_{45} &= (-a_{3,3} + a_{3,4} - a_{4,3})(b_{3,5} + b_{4,1} + b_{4,3} + b_{4,5} + b_{5,1} + b_{5,3} + b_{5,5}) \\ h_{46} &= -a_{3,5}(-b_{5,1} - b_{5,5}) \\ h_{47} &= (a_{2,1} - a_{2,5} - a_{3,1} + a_{3,5})(b_{1,1} + b_{1,2} + b_{1,5} - b_{4,1} - b_{4,2} - b_{4,5}) \\ h_{48} &= (-a_{2,3} + a_{3,3})(b_{2,2} + b_{3,2} + b_{3,5} + b_{4,1} + b_{4,2} + b_{4,5}) \\ h_{49} &= (-a_{1,1} - a_{1,3} + a_{1,4} + a_{1,5} - a_{2,1} - a_{2,3} + a_{2,4} + a_{2,5})(-b_{1,1} - b_{1,2} + b_{1,4}) \\ h_{50} &= (-a_{1,4} - a_{2,4})(b_{2,2} - b_{3,1} - b_{3,2} + b_{3,4} - b_{4,2} + b_{4,4}) \end{aligned}$$

$$\begin{aligned} h_{51} &= a_{2,2}(b_{2,1} + b_{2,2} - b_{5,1}) \\ h_{52} &= a_{4,2}(b_{1,1} + b_{2,1} + b_{2,3}) \\ h_{53} &= -a_{1,2}(-b_{2,1} + b_{2,4} + b_{4,1}) \\ h_{54} &= (a_{1,2} + a_{1,4} - a_{2,2} - a_{2,5} - a_{3,2} + a_{3,3} - a_{4,2} + a_{4,3} - a_{4,4} - a_{4,5})b_{2,3} \\ h_{55} &= (a_{1,4} - a_{4,4})(-b_{2,3} + b_{3,1} + b_{3,3} - b_{3,4} + b_{4,3} - b_{4,4}) \\ h_{56} &= (a_{1,1} - a_{1,5} - a_{4,1} + a_{4,5})(b_{3,1} + b_{3,3} - b_{3,4} + b_{5,1} + b_{5,3} - b_{5,4}) \\ h_{57} &= (-a_{3,1} - a_{4,1})(-b_{1,3} - b_{1,5} - b_{2,5} - b_{5,1} - b_{5,3} - b_{5,5}) \\ h_{58} &= (-a_{1,4} - a_{1,5} - a_{3,4} - a_{3,5})(-b_{5,1} + b_{5,4} - b_{5,5}) \\ h_{59} &= (-a_{3,3} + a_{3,4} - a_{4,3} + a_{4,4})(b_{4,1} + b_{4,3} + b_{4,5} + b_{5,1} + b_{5,3} + b_{5,5}) \\ h_{60} &= (a_{2,5} + a_{4,5})(b_{2,3} - b_{3,1} - b_{3,2} - b_{3,3} - b_{5,2} - b_{5,3}) \\ h_{61} &= (a_{1,4} + a_{3,4})(b_{1,1} - b_{1,4} + b_{1,5} - b_{2,5} - b_{4,4} + b_{4,5} - b_{5,1} + b_{5,4} - b_{5,5}) \\ h_{62} &= (a_{2,1} + a_{4,1})(b_{1,2} + b_{1,3} + b_{2,2} - b_{4,1} - b_{4,2} - b_{4,3}) \\ h_{63} &= (-a_{3,3} - a_{4,3})(-b_{2,3} - b_{3,3} - b_{3,5} - b_{4,1} - b_{4,3} - b_{4,5}) \\ h_{64} &= (a_{1,1} - a_{1,3} - a_{1,4} + a_{3,1} - a_{3,3} - a_{3,4})(b_{1,1} - b_{1,4} + b_{1,5}) \\ h_{65} &= (-a_{1,1} + a_{4,1})(-b_{1,3} + b_{1,4} + b_{2,4} - b_{5,1} - b_{5,3} + b_{5,4}) \\ h_{66} &= (a_{1,1} - a_{1,2} + a_{1,3} - a_{1,5} - a_{2,2} - a_{2,5} - a_{3,2} + a_{3,3} - a_{4,1} + a_{4,2})b_{2,4} \\ h_{67} &= (a_{2,5} - a_{3,5})(b_{1,1} + b_{1,2} + b_{1,5} - b_{2,5} - b_{4,1} - b_{4,2} - b_{4,5} + b_{5,2} + b_{5,5}) \\ h_{68} &= (a_{1,1} + a_{1,3} - a_{1,4} - a_{1,5} - a_{4,1} - a_{4,3} + a_{4,4} + a_{4,5})(-b_{3,1} - b_{3,3} + b_{3,4}) \\ h_{69} &= (-a_{1,3} + a_{1,4} - a_{2,3} + a_{2,4})(-b_{2,4} - b_{3,1} - b_{3,2} + b_{3,4} - b_{5,2} + b_{5,4}) \\ h_{70} &= (a_{2,3} - a_{2,5} + a_{4,3} - a_{4,5})(-b_{3,1} - b_{3,2} - b_{3,3}) \\ h_{71} &= (-a_{3,1} + a_{3,3} - a_{3,4} + a_{3,5} - a_{4,1} + a_{4,3} - a_{4,4} + a_{4,5})(-b_{5,1} - b_{5,3} - b_{5,5}) \\ h_{72} &= (-a_{2,1} - a_{2,4} - a_{4,1} - a_{4,4})(b_{4,1} + b_{4,2} + b_{4,3}) \\ h_{73} &= (a_{1,3} - a_{1,4} - a_{1,5} + a_{2,3} - a_{2,4} - a_{2,5})(b_{1,1} + b_{1,2} - b_{1,4} + b_{2,4} + b_{5,2} - b_{5,4}) \\ h_{74} &= (a_{2,1} - a_{2,3} + a_{2,4} - a_{3,1} + a_{3,3} - a_{3,4})(b_{4,1} + b_{4,2} + b_{4,5}) \\ h_{75} &= (-a_{1,2} + a_{1,4} - a_{2,2} - a_{2,5} - a_{3,1} + a_{3,2} + a_{3,4} + a_{3,5} - a_{4,1} + a_{4,2})b_{2,5} \\ h_{76} &= (a_{1,3} + a_{3,3})(-b_{1,1} + b_{1,4} - b_{1,5} + b_{2,4} + b_{3,4} - b_{3,5}) \\ c_{1,1} &= -h_{10} + h_{12} + h_{14} - h_{15} - h_{16} + h_{53} + h_{5} - h_{66} - h_{7} \\ c_{2,1} &= h_{10} + h_{11} - h_{12} + h_{13} + h_{15} + h_{16} - h_{17} - h_{44} + h_{51} \\ c_{3,1} &= h_{10} - h_{12} + h_{15} + h_{16} - h_{1} + h_{2} + h_{3} - h_{4} + h_{75} \\ c_{4,1} &= -h_{10} + h_{12} - h_{15} - h_{16} + h_{52} + h_{54} - h_{6} - h_{8} + h_{9} \\ c_{1,2} &= h_{13} + h_{15} + h_{20} + h_{21} - h_{22} + h_{23} + h_{25} - h_{43} + h_{49} + h_{50} \\ c_{2,2} &= -h_{11} + h_{12} - h_{13} - h_{15} - h_{16} + h_{17} + h_{18} - h_{19} - h_{21} + h_{43} + h_{44} \\ c_{3,2} &= -h_{16} - h_{19} - h_{21} - h_{28} - h_{29} - h_{38} + h_{42} + h_{44} - h_{47} + h_{48} \\ c_{4,2} &= h_{11} - h_{12} - h_{18} + h_{21} - h_{32} + h_{33} - h_{34} - h_{36} + h_{62} - h_{70} \\ c_{1,3} &= h_{15} + h_{23} + h_{24} + h_{34} - h_{37} + h_{40} - h_{41} + h_{55} - h_{56} - h_{9} \\ c_{2,3} &= -h_{10} + h_{19} + h_{32} + h_{35} + h_{36} + h_{37} - h_{43} - h_{60} - h_{6} - h_{72} \\ c_{3,3} &= -h_{16} - h_{28} + h_{33} + h_{37} - h_{39} + h_{45} - h_{46} + h_{63} - h_{71} - h_{8} \\ c_{4,3} &= h_{10} + h_{15} + h_{16} - h_{33} + h_{34} - h_{35} - h_{37} - h_{54} + h_{6} + h_{8} - h_{9} \\ c_{1,4} &= -h_{10} + h_{12} + h_{14} - h_{16} + h_{23} + h_{24} + h_{25} + h_{26} + h_{5} - h_{66} - h_{7} \\ c_{2,4} &= h_{10} + h_{18} - h_{19} + h_{20} - h_{22} - h_{24} - h_{26} - h_{5} - h_{69} + h_{73} \\ c_{3,4} &= -h_{14} + h_{16} - h_{23} - h_{26} + h_{27} + h_{29} + h_{31} + h_{46} - h_{58} + h_{76} \\ c_{4,4} &= h_{12} + h_{25} + h_{26} - h_{33} - h_{35} - h_{40} + h_{41} + h_{65} - h_{68} - h_{7} \\ c_{1,5} &= h_{15} + h_{24} + h_{25} + h_{27} - h_{28} + h_{30} + h_{31} - h_{4} + h_{61} + h_{64} \\ c_{2,5} &= -h_{10} - h_{18} - h_{2} - h_{30} - h_{38} + h_{42} - h_{43} + h_{46} + h_{67} + h_{74} \\ c_{3,5} &= -h_{10} + h_{12} - h_{15} + h_{28} + h_{29} - h_{2} - h_{30} - h_{3} + h_{46} + h_{4} - h_{75} \\ c_{4,5} &= -h_{12} - h_{29} + h_{30} - h_{34} + h_{35} + h_{39} + h_{3} - h_{45} + h_{57} + h_{59} \end{aligned}$$

Extended Data Fig. 2 | Algorithm for multiplying 4 × 5 by 5 × 5 matrices in standard arithmetic with 76 multiplications. This outperforms the previously best known algorithm, which involves 80 multiplications.

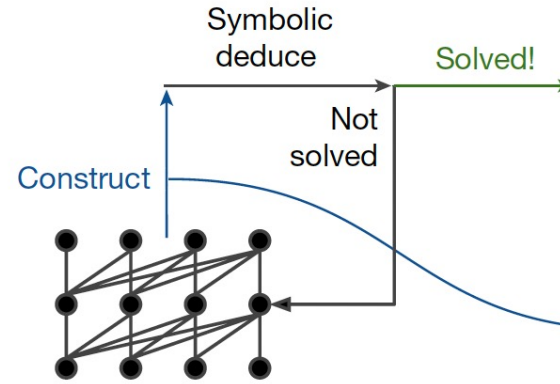
# AlphaGeometry

**a** A simple problem



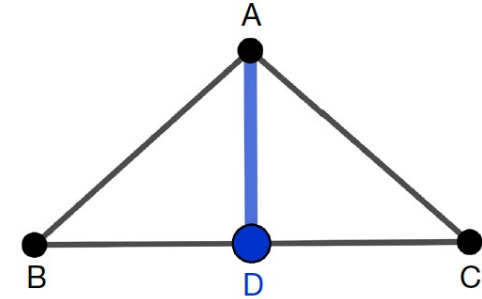
“Let ABC be any triangle with  $AB = AC$ . Prove that  $\angle ABC = \angle BCA$ .”

**b** AlphaGeometry



**c** Language model

**d** Solution



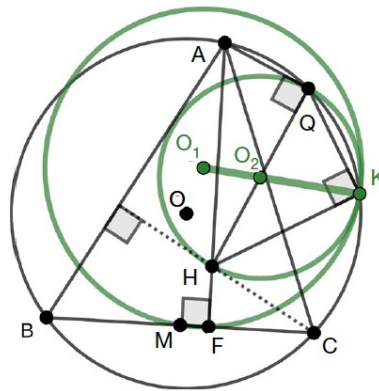
**Construct D: midpoint BC,**

$AB=AC, BD = DC, AD=AD \Rightarrow \angle ABD=\angle DCA$  [1]

[1], **B C D collinear**  $\Rightarrow \angle ABC=\angle BCA$

**e** 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  $QH \perp QA$  and let K be the point on (O) such that  $KH \perp KQ$ . Prove that the circumcircles ( $O_1$ ) and ( $O_2$ ) of triangles FKM and KQH are tangent to each other.”



Alpha-Geometry

**f** Solution

**Construct D: midpoint BH [a]**

[a],  $O_2$  midpoint HQ  $\Rightarrow BQ \parallel O_2D$  [20]

**Construct G: midpoint HC [b]** ...

$\angle GMD = \angle GO_2D \Rightarrow M O_2 G D$  cyclic [26]

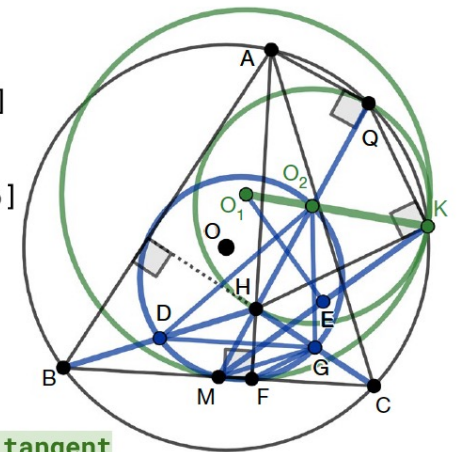
[a], [b]  $\Rightarrow BC \parallel DG$  [30]

**Construct E: midpoint MK [c]**

..., [c]  $\Rightarrow \angle KFC = \angle KO_1E$  [104]

$\angle FKO_1 = \angle FKO_2 \Rightarrow KO_1 \parallel KO_2$  [109]

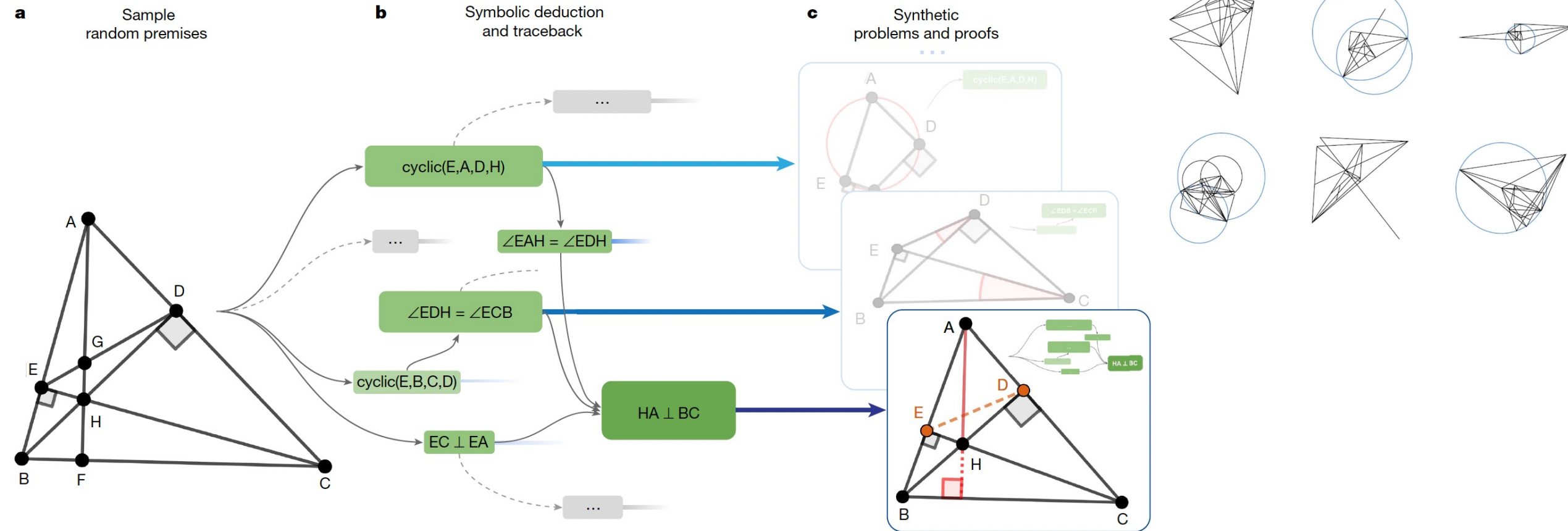
[109]  $\Rightarrow O_1, O_2, K$  collinear  $\Rightarrow (O_1), (O_2)$  tangent



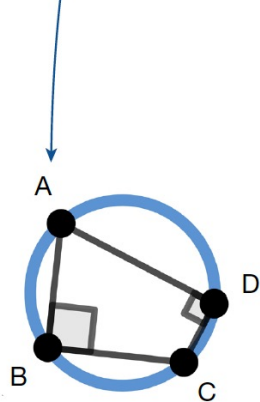
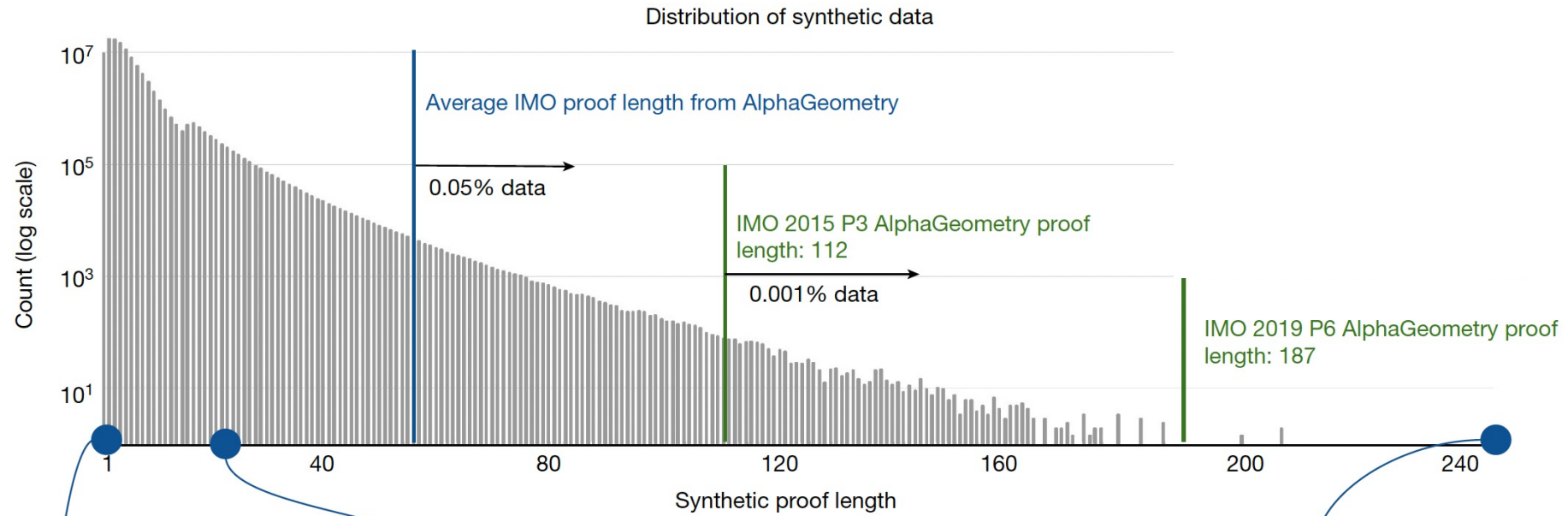


# AlphaGeometry

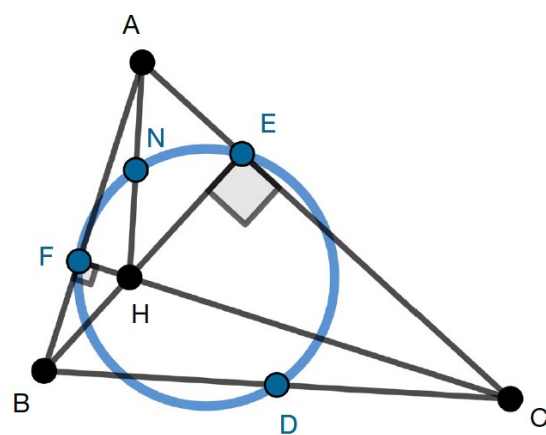
1 miliard sztucznie wygenerowanych konstrukcji  
z każdej konstrukcji wygenerowanie możliwych zależności  
ostatecznie 100 milionów unikalnych twierdzeń



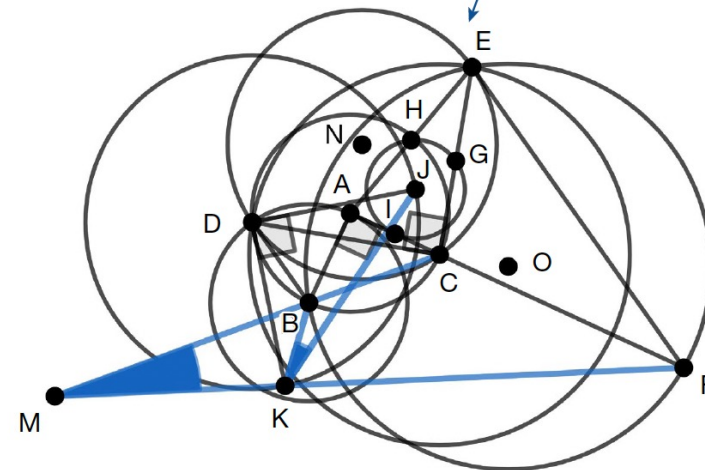
# AlphaGeometry



Trivial theorem  
Proof length = 1



Well-known theorem  
Proof length = 20

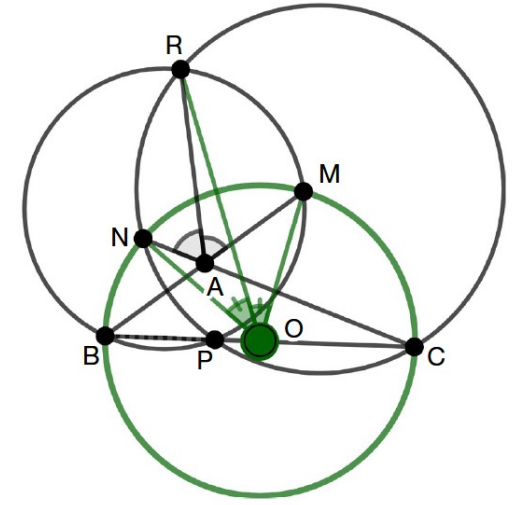


Most complex synthetic theorem  
Proof length = 247

# AlphaGeometry

IMO 2004 P1:

“Let  $ABC$  be an acute-angled triangle with  $AB \neq AC$ . The circle with diameter  $BC$  intersects the sides  $AB$  and  $AC$  at  $M$  and  $N$  respectively. Denote by  $O$  the midpoint of the side  $BC$ . The bisectors of the angles  $\angle BAC$  and  $\angle MON$  intersect at  $R$ . Prove that the circumcircles of the triangles  $BMR$  and  $CNR$  have a common point lying on the side  $BC$ .”



Translate

Premise

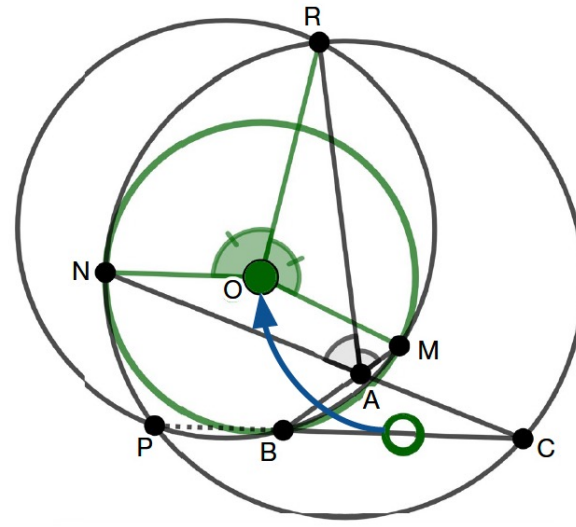
```
A B C O M N R P : Points
mid_point(O, B, C) [--]
same_line(B, M, A) [00] OM=OB [01]
same_line(N, C, A) [02] ON=OB [03]
∠BAR=∠RAC [04] ∠MOR=∠RON [05]
circle(B, M, R, P) [06] circle(C, N, R, P) [07]
Goal
same_line(P, B, C)
```

Solve

```
Proof
[01][03] ⇒ ∠ONM=∠NMO [08]
[01][03][05] ⇒ RN=RM [09]
[01][03][09] ⇒ NM ⊥ OR [10]
AUXILIARY POINT K : KM = KN
[01][03] KM = KN ⇒ MN ⊥ KO [12]
AUXILIARY POINT L : KL = KA, OL = OA
KL = KA, OL = OA ⇒ KO ⊥ AL [15] ∠AKO=∠OKL [16]
[15][12][10][16][13] ⇒ RA=RL [17]
OL = OA ⇒ ∠OAL=∠ALO [18]
angle-chase:[12][15][08][18] ⇒ ∠NOA=∠LOM [19]
[01][03] OL = OA [19] ⇒ AN=LM [21]
[17][21][09] ⇒ ∠NAR=∠RLM [22]
[02][04][00][22] ⇒ circle(L, M, A, R) [23]
similar ⇒ circle(R, L, N, A) [24]
[23][24] ⇒ ∠RMA=∠RNA [25]
[06] ⇒ ∠BPR=∠BMR [26]
[07] ⇒ ∠NCP=∠NRP [27]
[00][02][25][26][27] ⇒ PC // BP
⇒ same_line(B, P, C)
```

Traceback

Generalize



- ABC Unused premise
- ABC Used premises
- ABC Neural net output
- ABC Symbolic solver output

# AlphaGeometry

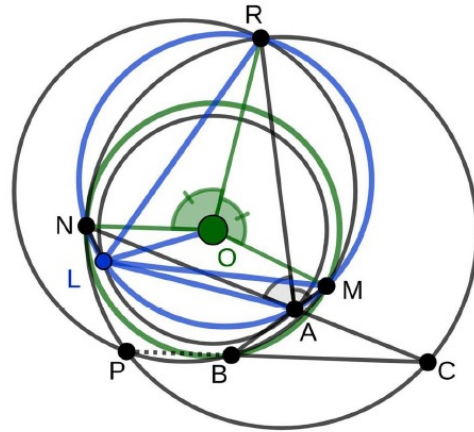
**Table 1 | Main results on our IMO-AG-30 test benchmark**

Method		Problems solved (out of 30)
Computer algebra	Wu's method <sup>21</sup> (previous state of the art)	10
	Gröbner basis <sup>20</sup>	4
Search (human-like)	GPT-4 (ref. 25)	0
	Full-angle method <sup>30</sup>	2
	Deductive database (DD) <sup>10</sup>	7
	DD+human-designed heuristics <sup>17</sup>	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.

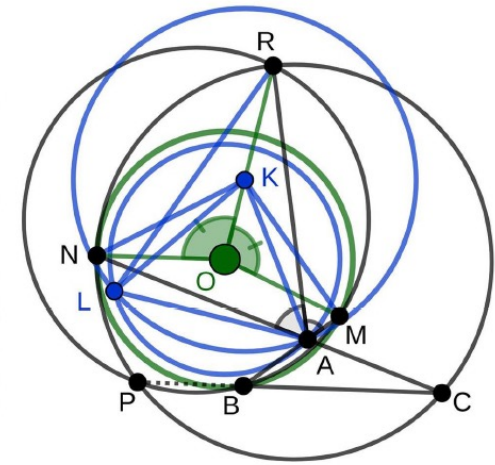
**Original problem statement:**

Let ABC be an acute-angled triangle with  $AB \neq AC$ . Let O be any point. The circle with diameter BC intersects the sides AB and AC at M and N respectively. Denote by O the midpoint of the side BC. The bisectors of the angles  $\angle BAC$  and  $\angle MON$  intersect at R. Prove that the circumcircles of the triangles BMR and CNR have a common point lying on the side BC.



**Adapted problem statement:**

Let ABC be a triangle. Let O be any point. Define point M as the intersection of circle (O, B) and line AB. Define point N as the intersection of circle (O, B) and line AC. Define point R such that AR is the bisector of  $\angle BAC$  and OR is the bisector of  $\angle MON$ . Define point O1 as the circumcenter of triangle BRM. Define point O2 as the circumcenter of triangle NRC. Define point P as the intersection of circles (O1, R) and (O2, R). Prove that B, C, P are collinear.



**Human proof:**

Let L be the reflection of A about OR

$\angle RLM = \angle NAR$  (LN is the reflection of AM about OR)  
 $= \angle RAM$  (AR is bisector of  $\angle NAM$ )  
 $\Rightarrow L, M, A, R$  is cyclic

Similarly, ANLR is cyclic

$\Rightarrow RMNA$  is cyclic  
 So  $BPR = BMR$  (BMRP is cyclic)  
 $= AMR$   
 $= ANR$  (RMNA is cyclic)  
 $= CNR$   
 $= CPR$  (CNRP is cyclic)  
 $\Rightarrow BP \parallel CP$

$\Rightarrow B, P,$  and  $C$  is collinear.

Minimal construction

Short, high-level deductions

Readable algebraic steps

**AlphaGeometry proof:**

AUXILIARY POINT K :  $KM=KN$   
 AUXILIARY POINT L :  $KL=KA, OL=OA$

$[01][03] \Rightarrow \angle ONM = \angle NMO$  [08]  
 $[01][03][05] \Rightarrow RN = RM$  [09]  
 $[09][01][03] \Rightarrow NM \perp OR$  [10]  
 $[01][03][KM=KN] \Rightarrow MN \perp KO$  [12]  
 $[KL=KA][OL=OA] \Rightarrow KO \perp AL$  [15]  $\angle AKO = \angle OKL$  [16]  
 $[15][12][10][16][KL=KA] \Rightarrow RA = RL$  [17]  
 $[OL=OA] \Rightarrow \angle OAL = \angle ALO$  [18]  
 angle-chase:  $[12][15][08][18] \Rightarrow \angle NOA = \angle LOM$  [19]  
 $[19][01][03][OL=OA] \Rightarrow AN = LM$  [21]  
 $[17][21][09] \Rightarrow \angle NAR = \angle RLM$  [22]  
 $[22][02][04][00] \Rightarrow \text{circle}(L, M, A, R)$  [23]

similar  $\Rightarrow \text{circle}(R, L, N, A)$  [24]

$[23][24] \Rightarrow \angle RMA = \angle RNA$  [25]  
 $[06] \Rightarrow \angle BPR = \angle BMR$  [26]  
 $[07] \Rightarrow \angle NCP = \angle NRP$  [27]  
 angle-chase:  $[25][00][02][26][27] \Rightarrow PC \parallel BP$

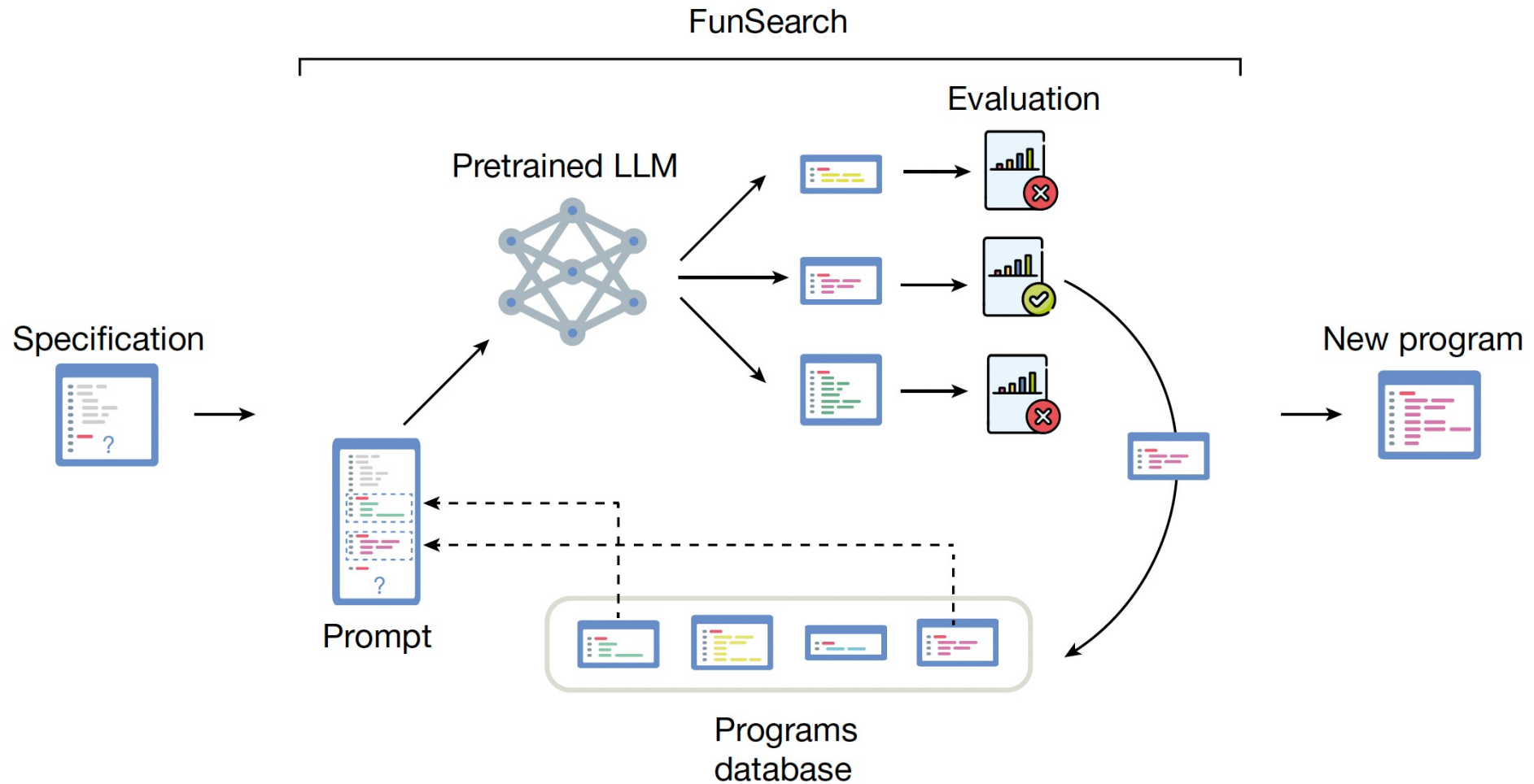
$\Rightarrow \text{same\_line}(B, P, C)$

Redundant

Verbose, low-level steps

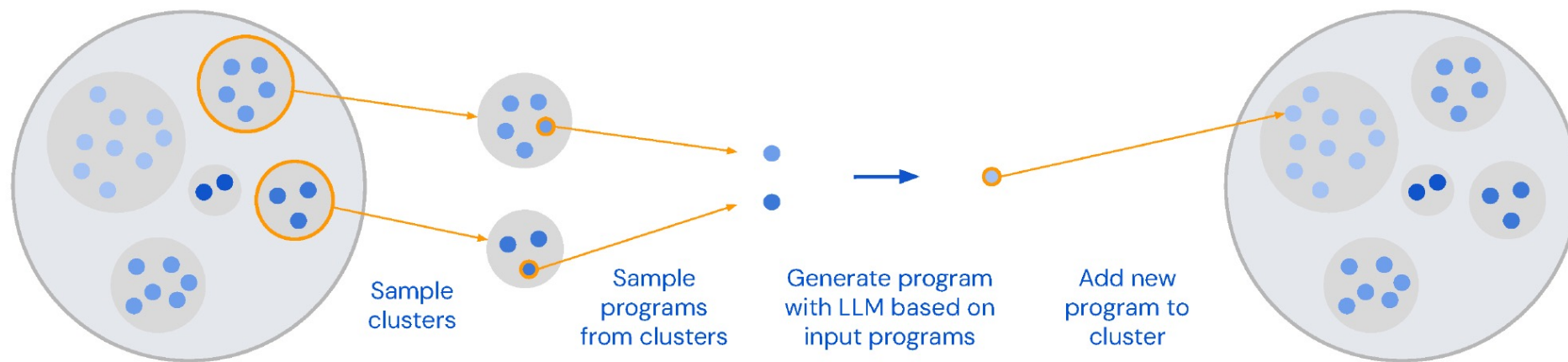
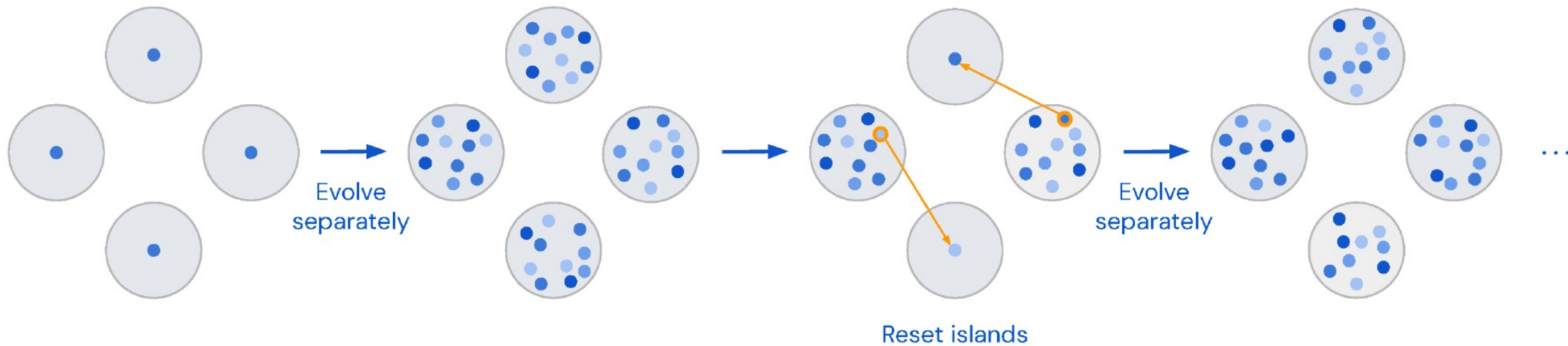
Low readability

# FunSearch



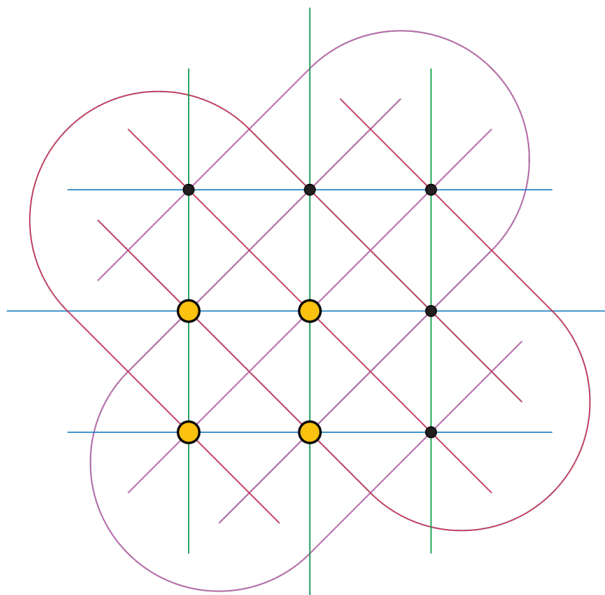
Romera-Paredes, B., Barekatin, M., Novikov, A., Balog, M., Kumar, M. P., Dupont, E., ... & Fawzi, A. (2024). ***Mathematical discoveries from program search with large language models***. *Nature*, 625(7995), 468-475.

# FunSearch



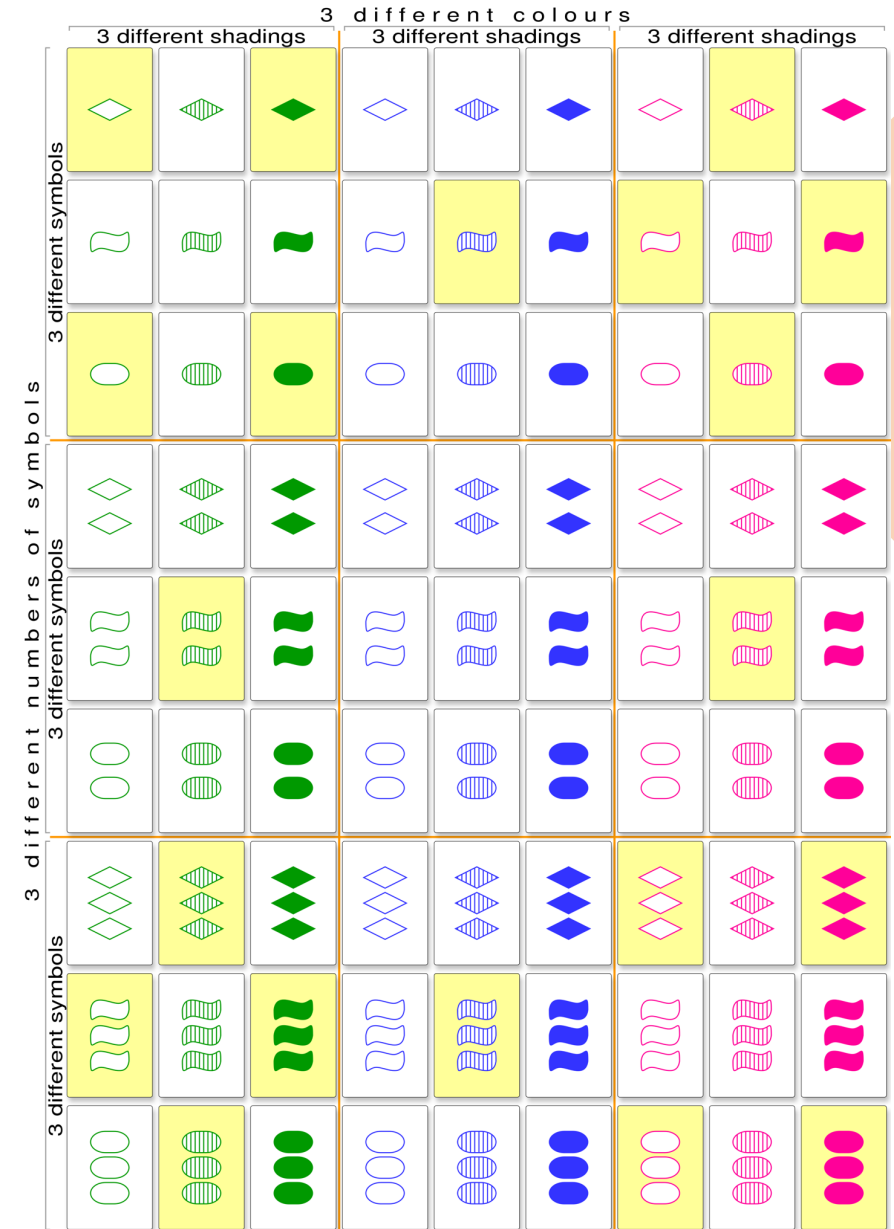
# FunSearch – cap set problem

znaleźć jak największy zbiór punktów w  $\mathbb{Z}_3^n$  \*,  
z których żadne 3 nie są współliniowe



Rozwiązanie dla  $n=2$

$n$	3	4	5	6	7	8
Best known	9	20	45	112	236	496
FunSearch	9	20	45	112	236	512



Przykład dla  $n=4$

\* an  $n$ -dimensional affine space over a three-element field



# FunSearch – online bin packing



[https://deepmind.google/api/blob/website/media/binpack\\_animation\\_v3\\_DcLQBh5.mp4](https://deepmind.google/api/blob/website/media/binpack_animation_v3_DcLQBh5.mp4)

	<b>OR1</b>	<b>OR2</b>	<b>OR3</b>	<b>OR4</b>	<b>Weibull 5k</b>	<b>Weibull 10k</b>	<b>Weibull 100k</b>
First fit	6.42%	6.45%	5.74%	5.23%	4.23%	4.20%	4.00%
Best fit	5.81%	6.06%	5.37%	4.94%	3.98%	3.90%	3.79%
<b>FunSearch</b>	<b>5.30%</b>	<b>4.19%</b>	<b>3.11%</b>	<b>2.47%</b>	<b>0.68%</b>	<b>0.32%</b>	<b>0.03%</b>

Fraction of excess bins (lower is better) for various bin packing heuristics on the OR and Weibull datasets. FunSearch outperforms first fit and best fit across problems and instance sizes.

# FunSearch

**a**

```
"""Finds large cap sets."""
import numpy as np
import utils_capset

# Function to be executed by FunSearch.
def main(n):
    """Runs `solve` on `n`-dimensional cap set and
    ↪ evaluates the output."""
    solution = solve(n)
    return evaluate(solution, n)

def evaluate(candidate_set, n):
    """Returns size of candidate_set if it is a cap
    ↪ set, None otherwise."""
    if utils_capset.is_capset(candidate_set, n):
        return len(candidate_set)
    else:
        return None

def solve(n):
    """Builds a cap set of dimension `n` using
    ↪ `priority` function."""
    # Precompute all priority scores.
    elements = utils_capset.get_all_elements(n)
    scores = [priority(el, n) for el in elements]
    # Sort elements according to the scores.
    elements = elements[np.argsort(scores,
    ↪ kind='stable')[::-1]]

    # Build `capset` greedily, using scores for
    ↪ prioritization.
    capset = []
    for element in elements:
        if utils_capset.can_be_added(element, capset):
            capset.append(element)
    return capset
```

```
# Function to be evolved by FunSearch.
def priority(element, n):
    """Returns the priority with which we want to add
    ↪ `element` to the cap set."""
    return 0.0
```

**b**

```
"""Finds good assignment for online 1d bin
↪ packing."""
import numpy as np
import utils_packing

# Function to be executed by FunSearch.
def main(problem):
    """Runs `solve` on online 1d bin packing instance,
    ↪ and evaluates the output."""
    bins = problem.bins
    # Packs `problem.items` into `bins` online.
    for item in problem.items:
        # Extract bins that have space to fit item.
        valid_bin_indices =
        ↪ utils_packing.get_valid_bin_indices(item,
        ↪ bins)
        best_index = solve(item,
        ↪ bins[valid_bin_indices])
        # Add item to the selected bin.
        bins[valid_bin_indices[best_index]] -= item
    return evaluate(bins, problem)

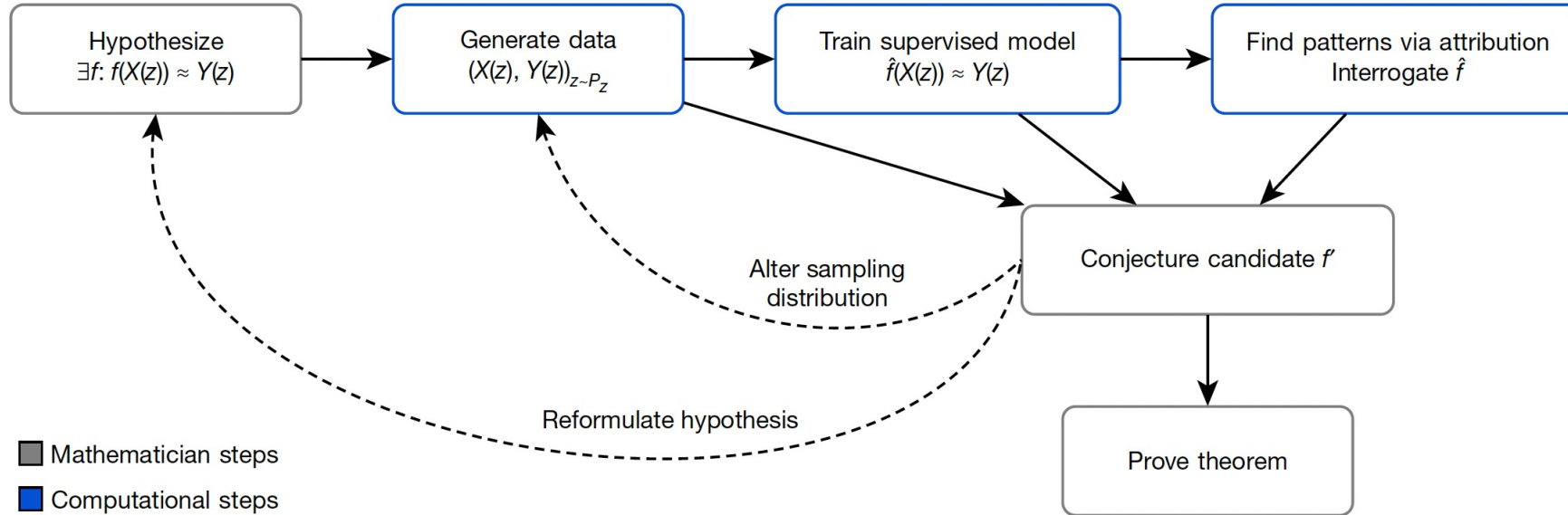
def evaluate(bins, problem):
    """Returns the negative of the number of bins
    ↪ required to pack items in `problem`."""
    if utils_packing.is_valid_packing(bins, problem):
        return -utils_packing.count_used_bins(bins,
        ↪ problem)
    else:
        return None

def solve(item, bins):
    """Selects the bin with the highest value according
    ↪ to `heuristic`."""
    scores = heuristic(item, bins)
    return np.argmax(scores)
```




```
# Function to be evolved by FunSearch.
def heuristic(item, bins):
    """Returns priority with which we want to add
    ↪ `item` to each bin."""
    return -(bins - item)
```



# Advancing mathematics by guiding human intuition with AI



Davies, A., Veličković, P., Buesing, L., Blackwell, S., Zheng, D., Tomašev, N., ... & Kohli, P. (2021). **Advancing mathematics by guiding human intuition with AI**. *Nature*, 600(7887), 70-74.

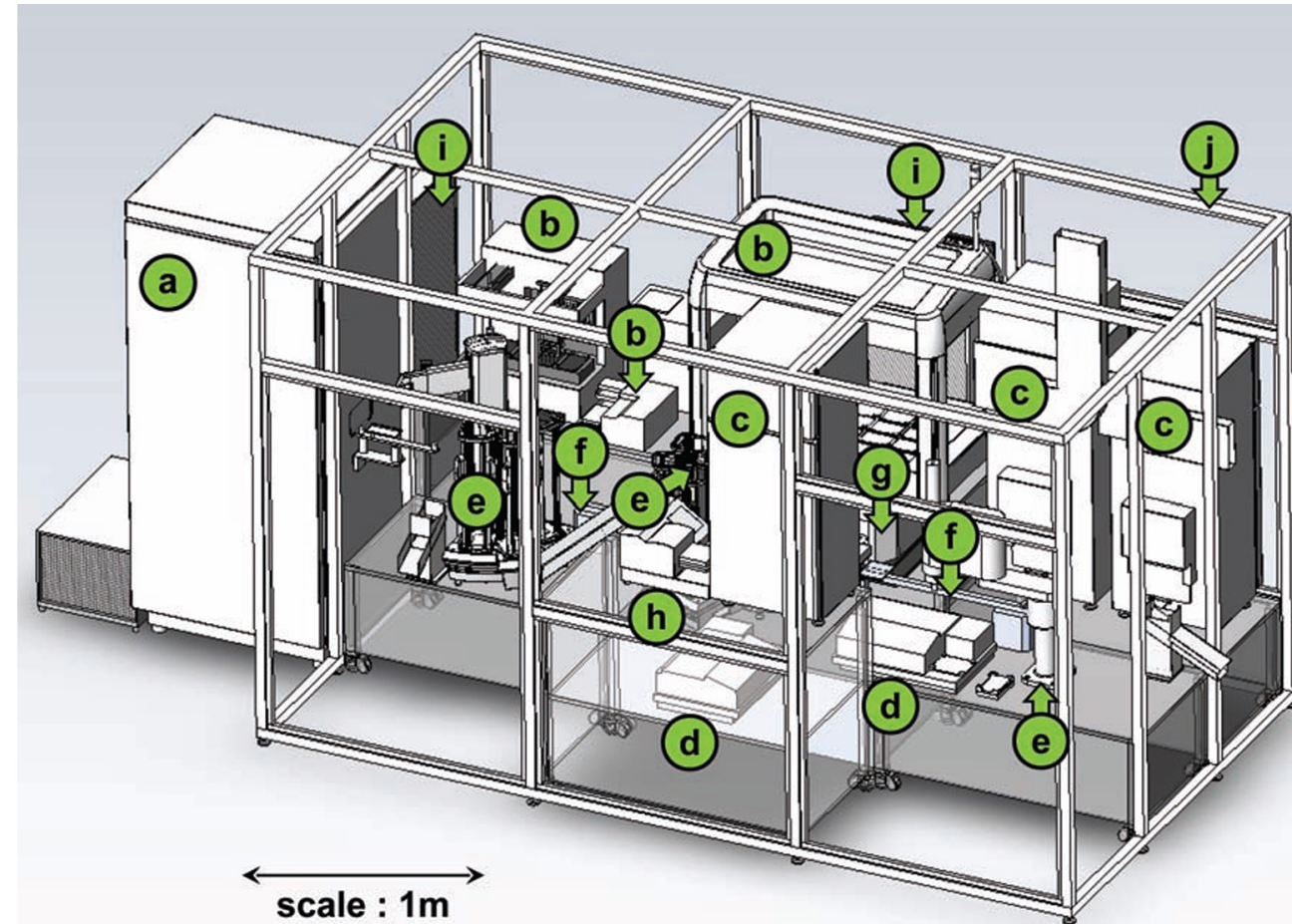
z: Knot	X(z): Geometric invariants				Y(z): Algebraic invariants		
	Volume	Chern-Simons	Meridional translation	...	Signature	Jones polynomial	...
	2.0299	0	$i$	...	0	$t^{-2} - t^{-1} + 1 - t + t^2$	...
	2.8281	-0.1532	$0.7381 + 0.8831i$	...	-2	$t - t^2 + 2t^3 - t^4 + t^5 - t^6$	...
	3.1640	0.1560	$-0.7237 + 1.0160i$	...	0	$t^{-2} - t^{-1} + 2 - 2t + t^2 - t^3 + t^4$	...

Blundell, C., Buesing, L., Davies, A., Veličković, P., & Williamson, G. (2022). **Towards combinatorial invariance for Kazhdan-Lusztig polynomials**. *Representation Theory of the American Mathematical Society*, 26(37), 1145-1191.

# The Automation of Science

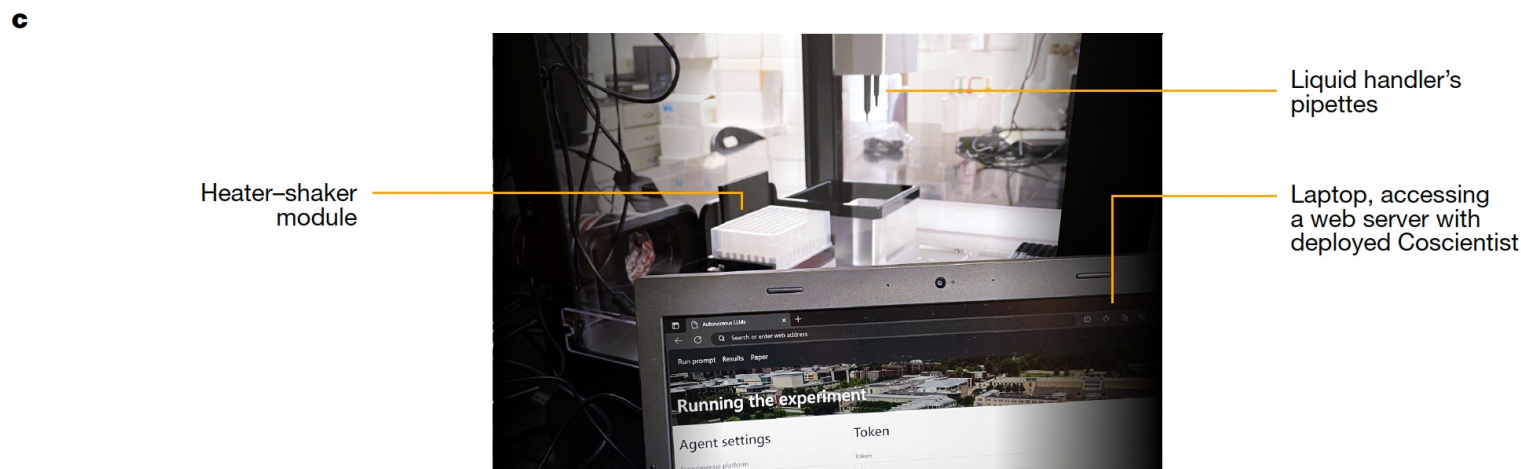
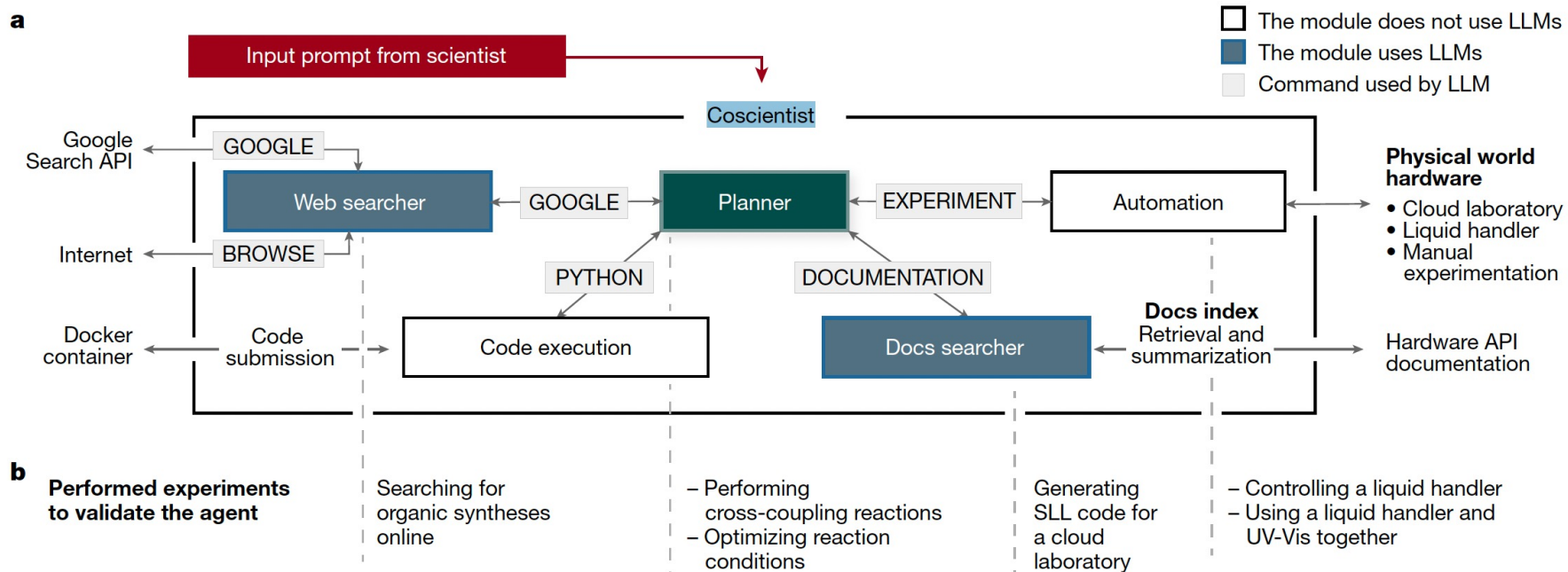
King, R. D., Rowland, J., Oliver, S. G., Young, M., Aubrey, W., Byrne, E., ... & Clare, A. (2009). *The automation of science*. Science, 324(5923), 85-89.

**Fig. 1.** The Robot Scientist Adam. The advances that distinguish Adam from other complex laboratory systems are the individual design of the experiments to test hypotheses and the utilization of complex internal cycles. Adam's basic operations are selection of specified yeast strains from a library held in a freezer, inoculation of these strains into microtiter plate wells containing rich medium, measurement of growth curves on rich medium, harvesting of a defined quantity of cells from each well, inoculation of these cells into wells containing defined media (minimal synthetic dextrose medium plus up to four added metabolites from a choice of six), and measurement of growth curves on the specified media. To achieve this functionality, Adam has the following components: a, an automated  $-20^{\circ}\text{C}$  freezer; b, three liquid handlers (one of which can separately control 96 fluid channels simultaneously); c, three automated  $+30^{\circ}\text{C}$  incubators; d, two automated plate readers; e, three robot arms; f, two automated plate slides; g, an automated plate centrifuge; h, an automated plate washer; i, two high-efficiency particulate air filters; and j, a rigid transparent plastic enclosure. There are also two bar code readers, seven cameras, 20 environment sensors, and four personal computers, as well as the software. Adam is capable of designing and initiating over a thousand new



strain and defined-growth-medium experiments each day (from a selection of thousands of yeast strains), with each experiment lasting up to 5 days. The design enables measurement of  $\text{OD}_{595\text{nm}}$  for each experiment at least once every 30 min (more often if running at less than full capacity), allowing accurate growth curves to be recorded (typically we take over a hundred measurements a day per well), plus associated metadata. See the supporting online material for pictures and a video of Adam in action.

# Coscientist



Boiko, D. A., MacKnight, R., Kline, B., & Gomes, G. (2023). *Autonomous chemical research with large language models*. Nature, 624(7992), 570-578.

# Coscientist

d

You need to perform Suzuki and Sonogashira reactions using available reagents...

User prompt

First, let's find the appropriate conditions for both Suzuki and Sonogashira reactions...

Planner's output

...  
GOOGLE Suzuki reaction conditions optimal

An optimal set of conditions for Suzuki-Miyaura cross-coupling reactions can be obtained...

...  
GOOGLE Sonogashira reaction conditions optimal

The optimal conditions for the Sonogashira reaction, a cross-coupling reaction of a vinyl or aryl halide with a terminal...

... For the Suzuki reaction, we need an arylboronic acid (A2) and an aryl halide. Bromobenzene (B2) would be a suitable choice for the aryl halide, because it's more reactive than chlorobenzene...

...

...  
... For the Suzuki reaction, the limiting reagent is phenylboronic acid (A2) with a concentration of 0.13 mmol ml<sup>-1</sup>. For the Sonogashira reaction, the limiting reagent is phenylacetylene (A1) with a concentration of 1.35 M or 1.35 mmol ml<sup>-1</sup>.

```
PYTHON
def calculate_volumes(concentration,
    desired_mmol=0.025):
    volume = desired_mmol /
    concentration
    return volume

# Suzuki reaction
conc_A2 = 0.13 # mmol/mL
...
```

0.19230769230769232 0.018518518518518517

...

...  
DOCUMENTATION how to use heater-shaker module

...

Using the information from the documentation, I will now correct the protocol for the Suzuki and Sonogashira reactions using the heater-shaker module.

```
EXPERIMENT
from opentrons import protocol_api
...
```

```
2
3 metadata = {
4     'apiLevel': '2.13'
5 }
6
7 def run(protocol: protocol_api.ProtocolContext):
8     # Load labware
9     hs_mod = protocol.load_module('heaterShakerModuleV1', 10)
10    reagents = protocol.load_labware('nest_96_wellplate_2ml_deep', 5)
11    reaction_plate = hs_mod.load_labware('nest_96_wellplate_2ml_deep')
12    tiprack20 = protocol.load_labware('opentrons_96_tiprack_20ul', 1)
13    tiprack300 = protocol.load_labware('opentrons_96_tiprack_300ul', 2)
14
15    # Load pipettes
16    p20 = protocol.load_instrument('p20_single_gen2', 'left', tip_racks=[tiprack20])
17    p300 = protocol.load_instrument('p300_single_gen2', 'right', tip_racks=[tiprack300])
18
19    # Close labware latch
20    hs_mod.close_labware_latch()
21
22    # Transfer reagents for Suzuki reaction (0.025 mmol scale)
23
24    ...
25
26    # Transfer reagents for Sonogashira reaction (0.025 mmol scale)
27    p20.pick_up_tip()
28    p20.aspirate(18.5, reagents['A1'])
29    p20.dispense(18.5, reaction_plate['B1'])
30    p20.drop_tip()
31
32    p20.pick_up_tip()
33    p20.aspirate(13.4, reagents['B1'])
34    p20.dispense(13.4, reaction_plate['B1'])
35    p20.drop_tip()
36
37    p300.pick_up_tip()
38    p300.aspirate(83.3, reagents['C2'])
39    p300.dispense(83.3, reaction_plate['B1'])
40    p300.drop_tip()
41
42    p20.pick_up_tip()
43    p20.aspirate(17.7, reagents['D1'])
44    p20.dispense(17.7, reaction_plate['B1'])
45    p20.drop_tip()
46
47    # Heat and shake the reactions
48    hs_mod.set_target_temperature(75)
49    hs_mod.wait_for_temperature()
50    hs_mod.set_and_wait_for_shake_speed(500)
51    protocol.delay(minutes=60) # Let the reactions run for 1 hour
52
53    # Deactivate heater and shaker
54    hs_mod.deactivate_heater()
55    hs_mod.deactivate_shaker()
56    hs_mod.open_labware_latch()
57
```

Method metadata

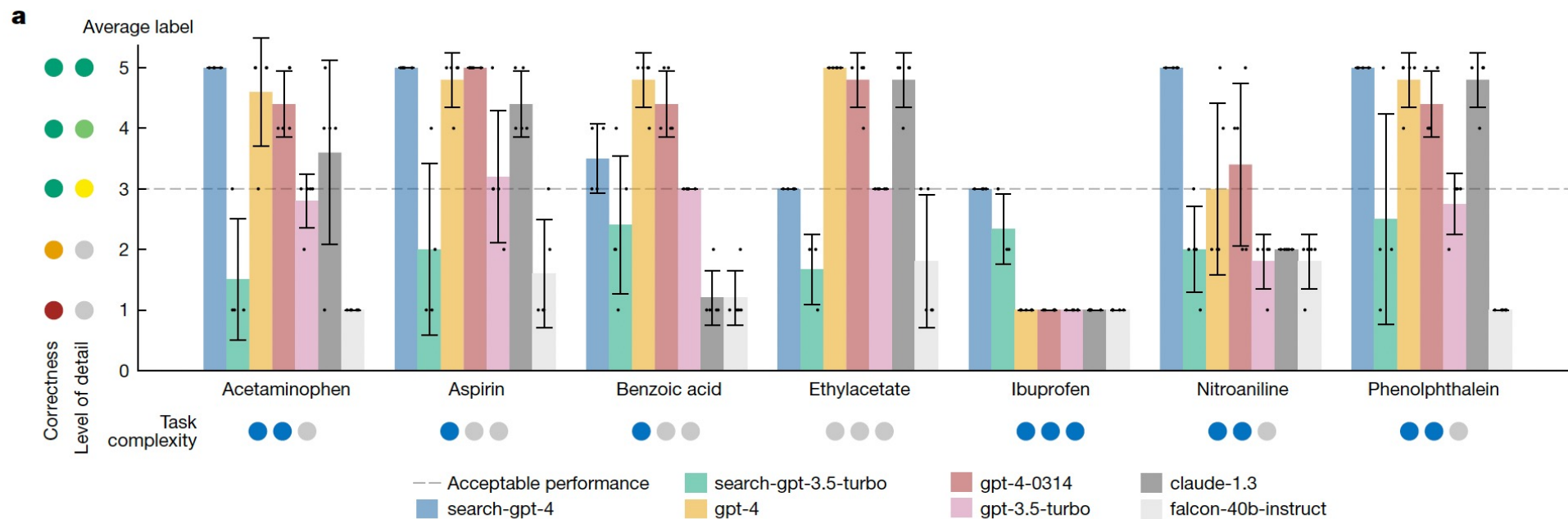
Preparing labware and modules

Reagent transfer for Sonogashira reaction

Setting up heater-shaker module

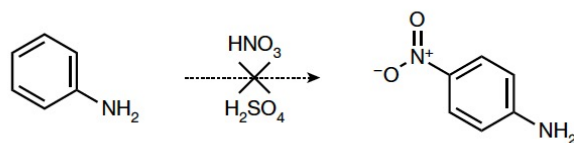
Turning the module off

# Coscientist

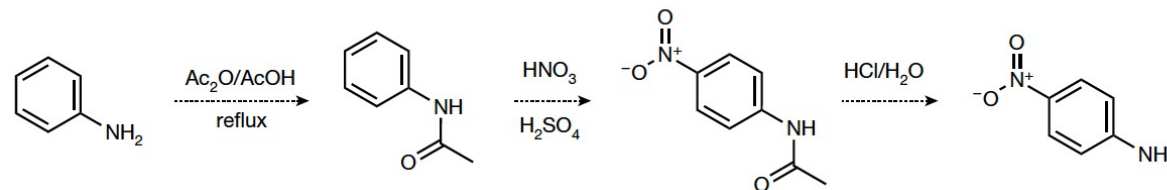


**b**

Incorrect synthesis steps but makes chemical sense (GPT-3.5, no search) ②

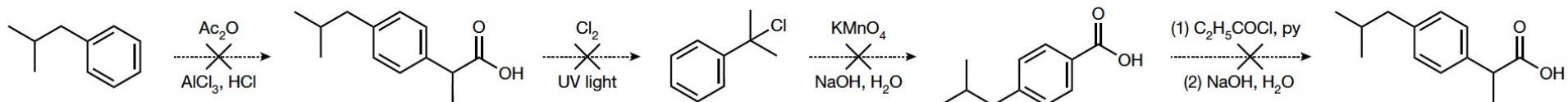


Correct synthesis, including detailed experimental procedure (GPT-4 with search) ⑤



**c**

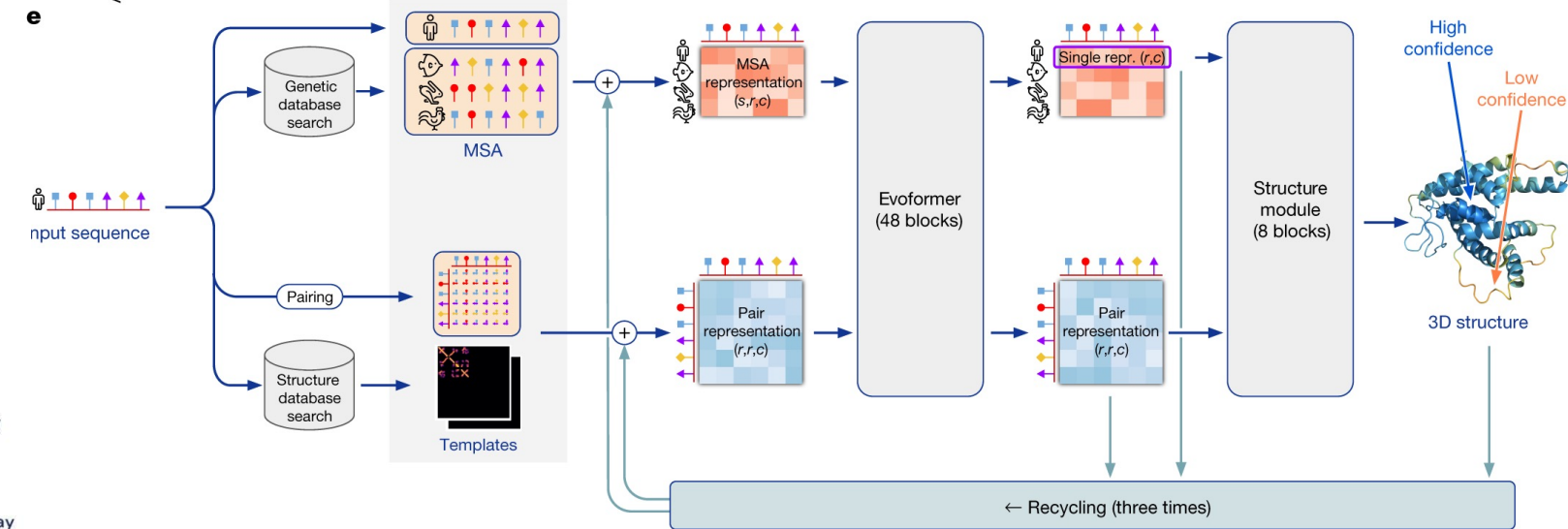
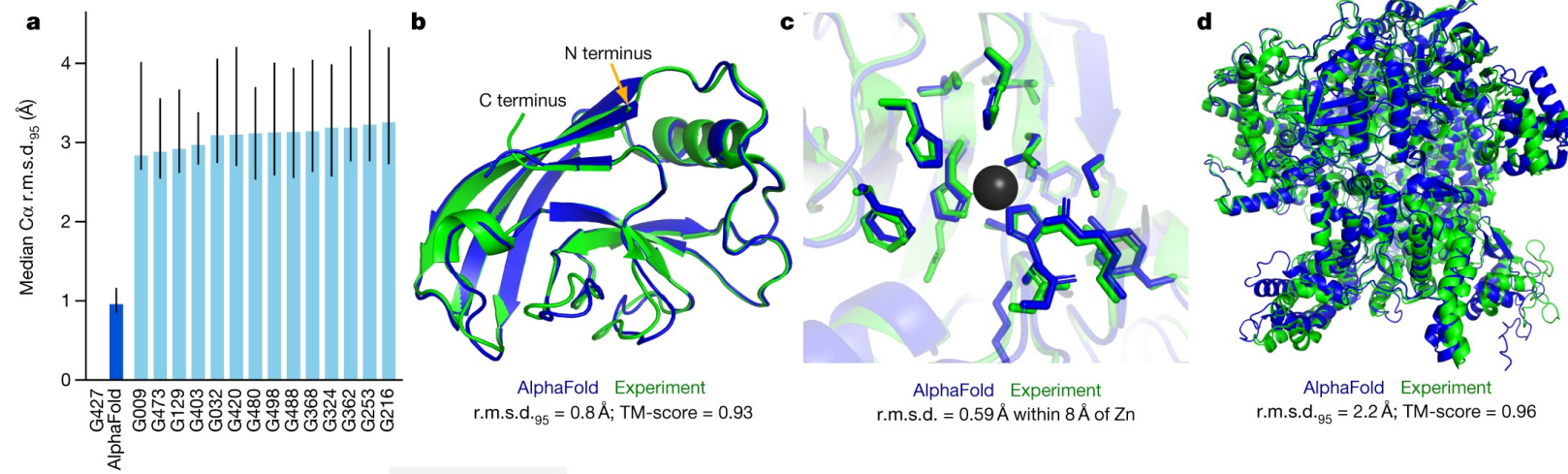
Incorrect synthesis steps, does not make chemical sense (GPT-4, no search) ①



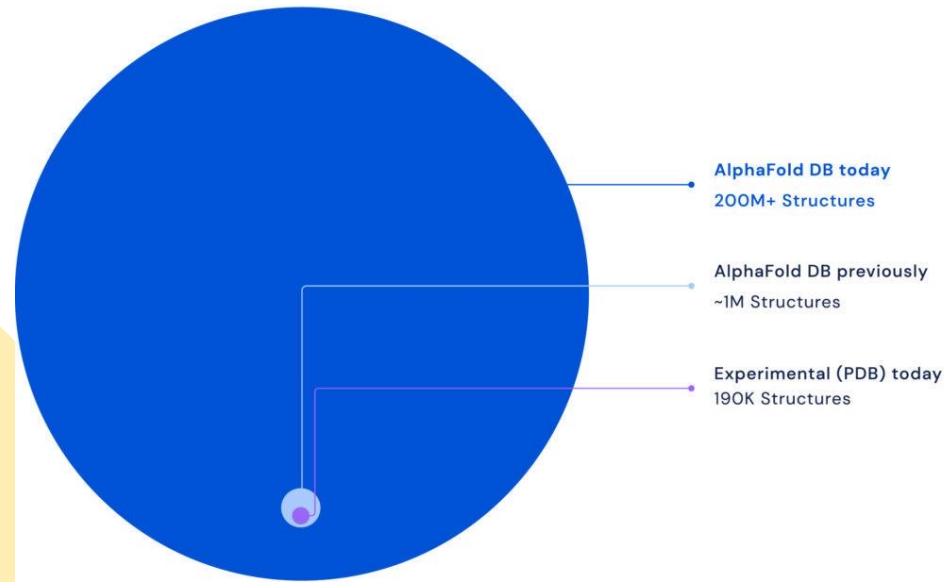


# AlphaFold

- przewidywanie przestrzennej struktury białek na podstawie sekwencji aminokwasów



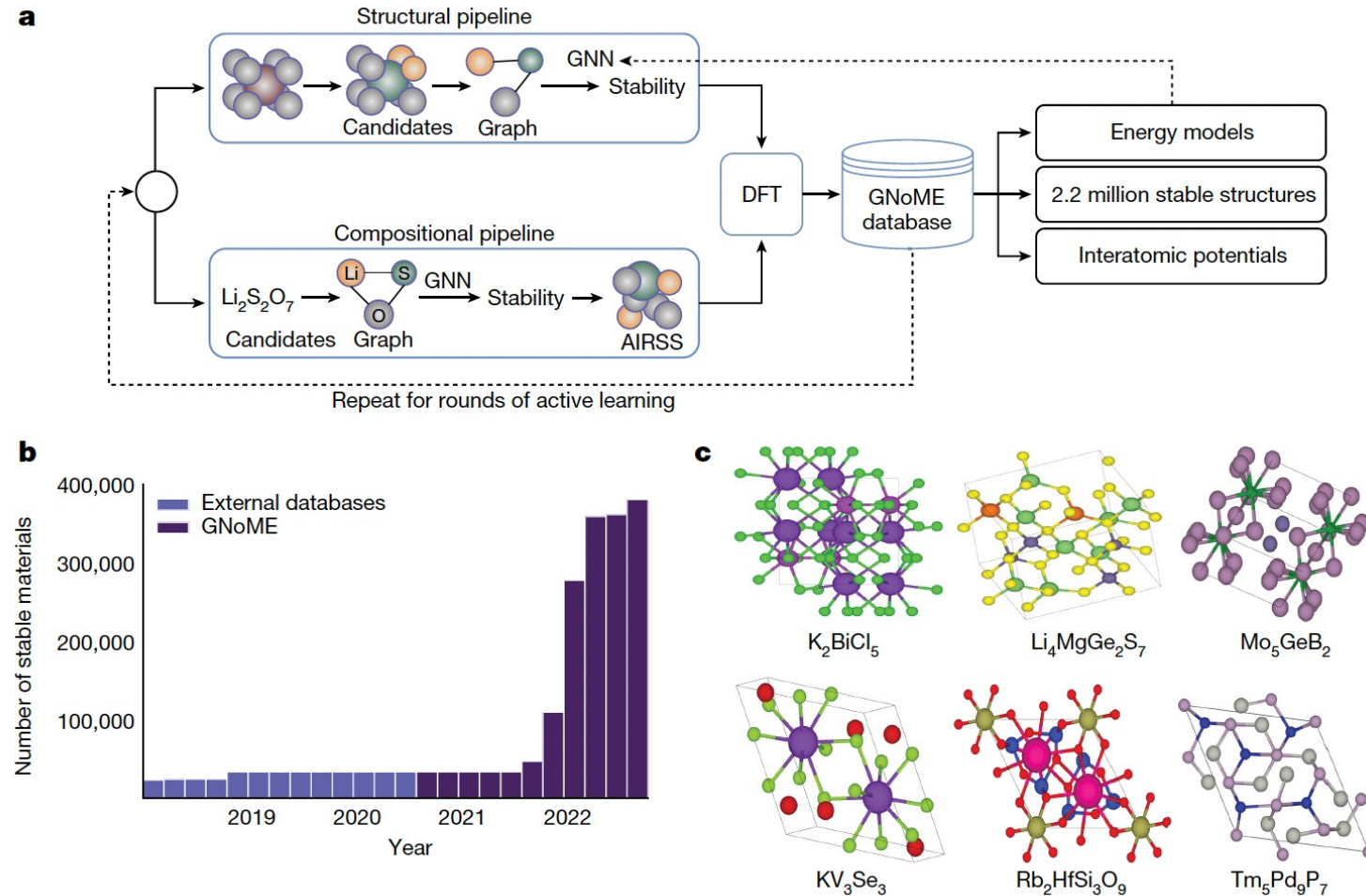
Number of Protein Structures



Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., ... & Hassabis, D. (2021). *Highly accurate protein structure prediction with AlphaFold*. *Nature*, 596(7873), 583-589.

# GNoME

- Graph Networks for Materials Exploration
- 2.2 miliony nowych materiałów krystalicznych (28 tys. znanych poprzednio)
- 380 tys. stabilnych
- 736 z nich zweryfikowano i stworzono fizycznie



Merchant, A., Batzner, S., Schoenholz, S. S., Aykol, M., Cheon, G., & Cubuk, E. D. (2023). **Scaling deep learning for materials discovery**. Nature, 624(7990), 80-85.

# Co nas czeka?

## obecnie

- korekta tekstu
- tworzenie wstępu
- wstępna recenzja

## niedługo

- podsuwanie nowych idei
- opracowanie wyników, przegląd literatury
- inteligentny asystent
- agregowanie wiedzy

## trochę później

- kreatywne rozwiązywanie nowych problemów
- tworzenie całych publikacji (do korekty przez ludzi)
- rola recenzenta

## odległa(?) przyszłość

- tworzenie nauki (być może niezrozumiałej dla człowieka)
- inny protokół komunikacji niż prace naukowe w czasopismach