Dion Gijswijt
Delft Institute of Applied Mathematics
TU Delft
d.c.gijswijt@tudelft.nl

De oplossing

Wiskundige ontdekkingen door Large Language Models: het Cap Set-probleem

Large Language Models, zoals GPT-3, hebben recent een enorme vlucht genomen. Ze kunnen helpen bij diverse taken zoals het schrijven van teksten of het samenvatten van documenten. Hoewel ze berucht zijn om hun gebrekkige logica en rekenkundige vermogens, lijkt het soms alsof ze echte intelligentie tentoonspreiden en creatieve ideeën hebben. Een natuurlijke vraag is: kunnen deze taalmodellen ook nieuwe wiskundige ontdekkingen doen? Een team van onderzoekers van Google sloeg de handen ineen met wiskundigen en denkt dat het antwoord 'ja' is. Ze combineerden 'Codey', een model voor het genereren van computercode, met een genetisch algoritme om een drietal problemen uit de combinatoriek aan te pakken. Voor het zogenaamde *Cap Set-probleem* slaagden zij erin om daadwerkelijk nieuwe resultaten te behalen [12]. In dit artikel legt Dion Gijswijt uit wat het Cap Set-probleem precies inhoudt en wat de nieuwe ontdekkingen zijn.

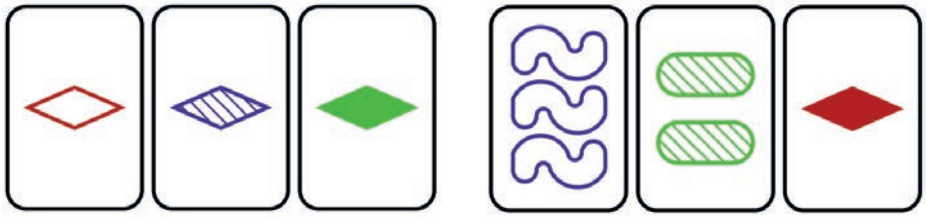
Het idee om AI te gebruiken voor het ontdekken van wiskundige patronen en het bewijzen van nieuwe stellingen, is niet nieuw [2]. Onlangs werden door AlphaTensor nieuwe tensordecomposities gevonden voor snellere matrixvermenigvuldiging [8] en bewees AlphaGeometry dat het meetkundeproblemen van de Internationale Wiskunde Olympiade kan oplossen in voor mensen begrijpelijke uitwerkingen [13]. Dit zijn echter programma's die voor een speciaal type probleem zijn getraind. Voor het hier besproken artikel [12] is dat niet het geval: het gebruikte Large Language Model (LLM) is getraind om goede code te schrijven, maar is niet voor een specifiek probleem als het Cap Set-probleem getraind.

Het Cap Set-probleem

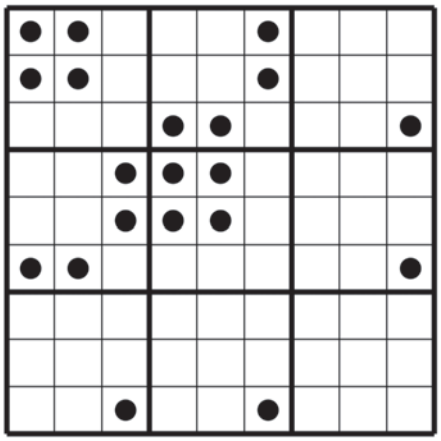
Een cap set is een deelverzameling van $\{0,1,2\}^n$ die geen drie verschillende vectoren bevat waarvan de som gelijk is aan de nulvector modulo 3. Een cap set in dimensie 4 kan bijvoorbeeld niet alledrie de vectoren $(1,1,1,2)$, $(1,1,2,2)$ en $(1,1,0,2)$ bevatten, omdat $(1,1,1,2)+(1,1,2,2)+(1,1,0,2)=(3,3,3,6)=(0,0,0,0)$ modulo 3. De verzameling van alle zestien 0–1-vectoren van lengte 4 is daarentegen wel een cap set.

cap sets in dimensie 4 kunnen worden beschreven in termen van het populaire kaartspel Set. In dit spel zijn er 81 kaarten, elk uniek bepaald door vier eigenschappen (kleur, vorm, aantal, vulling) die elk in drie varianten voorkomen. Drie kaarten vormen een SET als voor elk van de eigenschappen de drie kaarten gelijk zijn of allemaal verschillend. Door de varianten van elke eigenschap te nummeren als 0, 1, 2, corresponderen de kaarten met vectoren in $\{0,1,2\}^4$ en vormen drie kaarten een SET precies dan wanneer de drie corresponderende vectoren opgeteld de nulvector opleveren modulo 3. Op deze manier correspondeert een cap set in $\{0,1,2\}^4$ met een verzameling kaarten die géén SET bevat. We verwijzen naar het NAW-artikel [3] van De Bruijn voor meer details over de wiskunde achter het spel Set.

Het blijkt dat cap sets een combinatorisch heel interessante structuur zijn met relaties tot verschillende deelgebieden van de wiskunde. Zo worden cap sets in de eindige meetkunde bestudeerd onder de naam 'affine caps' en spelen ze in de getaltheorie een belangrijke rol bij het bestuderen van verzamelingen zonder rekenkundige rijen. In de extremale combinatoriek waren cap sets belangrijk in de ontwikkeling van de 'polynomial method' en in de theoretische informatica zijn ze nauw verbonden met de complexiteit van matrix vermenigvuldiging. We verwijzen naar Grochow [9] voor meer informatie over deze en andere verbanden.



Figuur 1 Twee voorbeelden van een SET. In het eerste voorbeeld zijn de kaarten voor twee eigenschappen gelijk en voor twee verschillend. In het tweede voorbeeld zijn de kaarten voor alle eigenschappen verschillend.



Figuur 2 Een optimale cap set in dimensie 4. De eerste twee coordinaten worden gegeven door de horizontale en verticale positie in het grove raster; de laatste twee door de positie binnen een 3×3-blok. De cap set wordt impliciet gegeven door de prioriteitsfunctie `(1+el[0]+el[1]+(el[2]==2)+(el[3]==2))%2`.

Het *Cap Set-probleem* gaat om het bepalen van de maximale grootte van een cap set in een gegeven dimensie n (of hier in ieder geval goede schattingen van te geven). Laten we deze maximale grootte aanduiden met a_n. Het is duidelijk dat $a_1 = 2$ en makkelijk na te gaan dat $a_2 = 4$. Aantonen dat $a_3 = 9$ is al een lastige puzzel en het feit dat $a_4 = 20$ (het maximale aantal kaarten zonder een SET) werd in 1971 bewezen door Pellegrino [10]. In Figuur 2 zie je een optimale cap set in dimensie 4.

Voor dimensies 5 en 6 zijn de optimale waarden $a_5 = 45$ en $a_6 = 112$ pas vrij recent bepaald in [6] en [11]. Al voor dimensie 7 is onbekend wat de maximale grootte van een cap set is: de grootste bekende cap set heeft 236 elementen, maar we weten niet of dat optimaal is. We weten alleen dat er geen cap set met 292 elementen of meer kan bestaan. De 'records' voor kleine dimensies staan in Tabel 1.

Het vinden van cap sets met een computer

Een rechttoe rechtaan brute-force-benadering om a_7 te bepalen, moet alle $\binom{2187}{237} \approx 2\times 10^{324}$ selecties van 237 vectoren uit $\{0,1,2\}^7$ controleren. Dit is zelfs met moderne computers een onhaalbare kaart.

Een bescheidener doel is om 'grote' cap sets te vinden die wellicht niet optimaal zijn, maar wel vorige records verbreken. Een eenvoudige heuristiek is de volgende 'Greedy' aanpak. Eerst zetten we de vectoren van $\{0,1,2\}^n$ in een bepaalde volgorde. Vervolgens, beginnend met een lege verzameling, gaan we de vectoren van de lijst één voor één af. Als de vector met twee andere vectoren in onze verzameling tot de nulvector optelt modulo 3, dan slaan we die vector over. Zo niet, dan voegen we de vector toe aan onze verzameling. Na het verwerken van de hele lijst hebben we een complete cap set: een cap set die niet kan worden uitgebreid door meer vectoren toe te voegen. Het hoeft echter niet van maximale grootte te zijn. De verzameling van zestien 0–1-vectoren in $\{0,1,2\}^4$ vormt bijvoorbeeld een complete cap set. Als deze vectoren aan het begin van de lijst staan, vinden we een cap set van grootte 16 in plaats van de optimale grootte van 20. Met deze aanpak hangt de cap set die we uiteindelijk krijgen dus af van de gekozen ordening en elke complete cap set kan worden verkregen door met een geschikte ordening te beginnen. Het probleem is dus teruggebracht tot het bedenken van goede ordeningen.

De gebruikte aanpak

In plaats van een expliciete lineaire ordening van $\{0,1,2\}^n$ te geven, gebruikt het onderzoeksteam van Google een prioriteitsfunctie $\{0,1,2\}^n \to \mathbb{R}$. De vectoren worden geordend op volgorde van dalende prioriteit (bij gelijke prioriteit gaat het lexicografisch kleinste element voor). Deze prioriteitsfuncties hebben de vorm van een kort stukje Python-code. De invoer van de functie is een vector `el` van lengte `n`. De uitvoer is een getal `score`: de prioriteit die aan de input vector wordt gegeven. De optimale cap set in dimensie 4 uit Figuur 2 kan bijvoorbeeld worden beschreven met de simpele prioriteitsfunctie `score=(1+el[0]+el[1]+(el[2]==2)+(el[3]==2))%2`.

Het Large Language Model wordt gebruikt om diverse geschikte prioriteitsfuncties te 'bedenken'. De benadering is om een pool van prioriteitsfuncties bij te houden. Door twee goed scorende functies te selecteren en deze te combineren tot een prompt (invoer) voor de LLM, wordt een nieuwe prioriteitsfunctie gegenereerd en (indien het een correcte Python-functie is) toegevoegd aan de pool. Een genetisch algoritme wordt gebruikt om goed scorende functies te repliceren en slecht scorende functies uit de pool te verwijderen. We verwijzen naar het artikel [12] voor meer details. Het is interessant om te vermelden dat het gebruikte model, genaamd Codey, een algemeen beschikbaar model is dat was getraind voor het genereren van code en niet specifiek was getraind voor het Cap Set-probleem.

Een nieuw record

Voor dimensie $n = 8$ lukte het de onderzoekers om een cap set van grootte 512 te vinden. Dit is een flinke verbetering ten opzichte van het vorige record van 496 (uit de verdubbeling van de projectieve cap gevonden in [5]). De gevonden prioriteitsfunctie staat weergegeven in Figuur 3.

We zien dat de functie rekening houdt met het aantal `el_count` nullen in de vector, en hoge prioriteit geeft aan vectoren zonder nullen. Ook controleert het of de getallen op de eerste en laatste positie gelijk zijn, en op vergelijkbare wijze voor de getallen op posities 2 en $n-1$ en 3 en $n-2$. Het geeft hogere prioriteit aan vectoren die meer van deze 'reflecties' hebben. We zullen de code hier niet volledig analyseren. Een belangrijk punt is echter dat de code *kan* worden geïnterpreteerd en geanalyseerd door een wiskundige om zo nieuwe inzichten te verkrijgen. Dit staat in schril contrast met een computeraanpak waarbij de uitvoer simpelweg een lijst van 512 vectoren van lengte 8 zou zijn. In feite slaagden de auteurs erin om een eenvoudige, expliciete beschrijving te maken voor deze cap set door de gevonden prioriteitsfunctie te analyseren. Het Large Language Model wordt dus gebruikt om 'ideeën' te genereren voor prioriteitsfuncties die goed werken en deze zijn interpreteerbaar door mensen als een startpunt voor een beter begrip van cap sets. Een ander kenmerk van de methode is dat de prioriteitsfunctie de dimensie als invoer heeft en dus kan worden toegepast op een reeks verschillende dimensies (hoewel de betreffende functie voor dimensies 9 tot 12 in dit geval geen goede cap sets oplevert.)

n	1	2	3	4	5	6	7	8	9	10	11	12
lb	**2**	**4**	**9**	**20**	**45**	**112**	236	512	1082	2432	5488	12928

Tabel 1 De beste ondergrenzen lb voor de maximale grootte van een cap set in dimensie n. Voor $n = 1,\ldots,6$ zijn de grenzen exact. Voor $n = 8$ is de ondergrens verbeterd van 496 naar 512. Voor $n = 9$, is de beste bekende constructie geëvenaard. De grenzen voor $n = 9,\ldots,12$ komen van constructions voor projectieve caps [1].

```
def priority(el: tuple[int, ...], n: int) -> float:
  score = n
  in_el = 0
  el_count = el.count(0)

  if el_count == 0:
    score += n ** 2
    if el[1] == el[-1]:
      score *= 1.5
    if el[2] == el[-2]:
      score *= 1.5
    if el[3] == el[-3]:
      score *= 1.5
  else:
    if el[1] == el[-1]:
      score *= 0.5
    if el[2] == el[-2]:
      score *= 0.5

  for e in el:
    if e == 0:
      if in_el == 0:
        score *= n * 0.5
      elif in_el == el_count - 1:
        score *= 0.5
      else:
        score *= n * 0.5 ** in_el
      in_el += 1
    else:
      score += 1

  if el[1] == el[-1]:
    score *= 1.5
  if el[2] == el[-2]:
    score *= 1.5

  return score
```

Figuur 3 De gevonden prioriteitsfunctie voor dimensie $n = 8$.

Capaciteit en gegeneraliseerde producten

In plaats van te kijken naar de maximale grootte van cap sets in een specifieke dimensie n, is het interessant om het asymptotische gedrag te bepalen. De *cap set-capaciteit* is gedefinieerd als

$$c = \limsup_{n\to\infty} a_n^{1/n}.$$

Omdat $\{0,1\}^n$ een cap set is in dimensie n volgt direct dat $c \in [2,3]$. Met behulp van de polynomiale slice-rank-methode werd in [7] bewezen dat $c < 2{,}756$. Hier zullen we juist kijken naar betere *ondergrenzen* op c en het nieuwe record dat de groep onderzoekers van [12] heeft gevonden met behulp van het Large Language Model.

Als eerste merken we op dat gegeven cap sets $A \subseteq \{0,1,2\}^m$ en $B \subseteq \{0,1,2\}^n$, het product $A \times B = \{(a,b) : a \in A, b \in B\}$ een cap set is van grootte $|A| \cdot |B|$ in dimensie $m+n$. Zodoende geeft een cap set $A \subseteq \{0,1,2\}^m$ voor elke n een cap set A^n van grootte $|A|^n$ in dimensie mn. Hieruit volgt dat $c \geq a_m^{1/m}$ voor elke m. Zo impliceert $a_4 = 20$ bijvoorbeeld dat $c \geq 20^{1/4} \approx 2{,}11$.

Om betere cap sets in hoge dimensies te verkrijgen dan met de eenvoudige productconstructie, introduceerde Edel [4] de volgende meer algemene constructie. Het eerste ingrediënt is een *uitbreidbare collectie van cap sets*. Dit is een drietal (A_0, A_1, A_2) van cap sets in $\{0,1,2\}^n$ met de volgende aanvullende eigenschappen:

(i) Als $x, y \in A_0$ en $z \in A_1 \cup A_2$, dan is $x + y + z \neq 0 \pmod 3$.
(ii) Als $x \in A_0$, $y \in A_1$ en $z \in A_2$, dan is $x + y + z \neq 0 \pmod 3$.

Voorbeeld. In dimensie 6 is er (op symmetriën na) een unieke grootste cap set. Het heeft een grootte van 112 en vormt de basis voor vele constructies van cap sets in hogere dimensies. De cap set kan als volgt worden beschreven.

Laat X de verzameling zijn van alle vectoren in $\{0,1,2\}^6$ die geen 0 hebben en een *even* aantal 1-en. Dus $|X| = \frac{1}{2} \cdot 2^6 = 32$.

De verzameling $Y \subseteq \{0,1,2\}^6$ bestaat uit die vectoren die precies drie 0-en hebben en waarvan de drie posities een van de volgende 10 mogelijkheden zijn:

$$123, 124, 135, 236, 165, 164, 254, 256, 345, 346.$$

Dus $|Y| = 10 \cdot 8 = 80$. Deze tien drietallen kunnen eenvoudig worden onthouden met behulp van een normale dobbelsteen. Een drietal vlakken zit in de lijst als het ofwel een tegengesteld paar vlakken heeft en een hoog vlak $(4,5,6)$, of als het geen tegengesteld paar heeft en minstens twee lage vlakken $(1,2,3)$.

Het is duidelijk dat X en Y disjunct zijn, dus $|X \cup Y| = 112$. We laten het aan de lezer om te verifieren dat $A_1 = X \cup Y$ inderdaad een cap set is. Als we in de definitie van Y juist de andere 10 mogelijke drietallen posities voor de 0-en nemen, vinden we een verzameling Z en is $A_2 = X \cup Z$ wederom een cap set van grootte 112.

Nemen we nu voor A_0 de verzameling van 12 vectoren in $\{0,1,2\}^6$ met precies één positie ongelijk 0, dan is (A_0, A_1, A_2) een uitbreidbare collectie cap sets.

Als (A_0, A_1, A_2) een uitbreidbare collectie cap sets is, dan kunnen verschillende producten van de cap sets A_0, A_1, A_2 combineerd worden tot een grote cap set. Om dit te beschrijven, hebben we het begrip *toelaatbare verzameling* nodig.

Definitie. Een deelverzameling $S \subseteq \{0,1,2\}^m$ is *toelaatbaar* (van gewicht w) als alle $s \in S$ in precies w van de m posities ongelijk aan nul zijn en bovendien het volgende geldt.

(i) Voor elk paar verschillende $s, s' \in S$ is er een positie i zodanig dat $s_i = 0 \neq s'_i$.
(ii) Voor elk drietal verschillende $s, s', s'' \in S$ is er een positie i zodanig dat $\{s_i, s'_i, s''_i\}$ gelijk is aan $\{0,1,2\}$, $\{0,0,1\}$, of $\{0,0,2\}$ (als multiset).

Hoewel de definitie misschien technisch lijkt, heeft deze het volgende gewenste effect.

Stelling 1. *Laat (A_0,A_1,A_2) een uitbreidbare collectie zijn in dimensie n en laat $S\subseteq\{0,1,2\}^m$ een toelaatbare verzameling van gewicht w zijn. Dan is*

$$S(A_0,A_1,A_2)=\bigcup_{s\in S}A_{s_1}\times\cdots\times A_{s_m}$$

een cap set in dimensie mn van grootte

$$|S|\cdot|A_0|^{m-w}|A_1|^w.$$

Merk op dat een toelaatbare deelverzameling $S\subseteq\{0,1,2\}^m$ van gewicht w hoogstens grootte $\binom{m}{w}$ kan hebben omdat geen twee vectoren op dezelfde posities hun nullen hebben staan. Een vermoeden van Tyrrell [14] is dat dit aantal altijd bereikt kan worden.

Vermoeden. *Voor alle gehele getallen $0<w<m$ is er een toelaatbare $S\subseteq\{0,1,2\}^m$ van gewicht w en grootte $\binom{m}{w}$.*

Wanneer een dergelijke toelaatbare verzameling bestaat wordt deze wel een $I(m,w)$ genoemd. Voor het speciale geval $w=m-1$ is het vermoeden waar en heeft de toegestane verzameling extra nuttige eigenschappen.

Stelling 2. *Laat $m\geq 2$. Voor $i=1,\ldots,m$ definieren we de vector*

$$s^{(i)}=(1,\ldots,1,0,2,\ldots,2)\in\{0,1,2\}^m,$$

waarbij de 0 op positie i staat. Nu is

$$S_m=\{s^{(1)},\ldots,s^{(m)}\}$$

een $I(m,m-1)$. Bovendien geldt dat als (A_0,A_1,A_2) een uitbreidbare collectie is, dan is $(S_m(A_0,A_1,A_2),A_1^m,A_2^m)$ dat ook.

Resultaten voor de cap set-capaciteit

Veel van de beste ondergrenzen voor cap sets in grote dimensies (en voor de capaciteit c) zijn verkregen door de constructies uit de vorige sectie toe te passen op de uitbreidbare collectie (A_0,A_1,A_2) in dimensie 6 uit het voorbeeld. In de eerste stap wordt deze collectie uitgebreid tot de uitbreidbare collectie $(S_k(A_0),A_1^k,A_2^k)$ in dimensie $6k$, voor een geschikte k. In de tweede stap wordt deze uitbreidbare collectie met behulp van een toelaatbare verzameling $S\subseteq\{0,1,2\}^m$ uitgebreid tot een cap set $S(S_k(A_0),A_1^k,A_2^k)$ in dimensie $6km$. De crux is om voor gegeven dimensie m en gewicht w een grote toelaatbare verzameling S te vinden, bij voorkeur een $I(m,w)$.

In [4] gebruikte Edel in de eerste stap S_8 en in de tweede een $I(10,5)$ om zo een cap set A in dimensie 480 te construeren van grootte $\binom{10}{5}\cdot 8^5\cdot 12^5\cdot 112^{75}$. Dit levert een ondergrens van $|A|^{1/480}>2{,}2173$ op voor de cap set-capaciteit. Lang was dit de beste ondergrens totdat Tyrrell [14] deze in 2023 verbeterde tot $2{,}2180$ door recursief een geschikte toelaatbare set te construeren in dimensie 1562.

Dit record is nu verpletterd door het team onderzoekers van Google. Zij slaagden erin meerdere grote toelaatbare sets te vinden. Allereerst vonden zij een $I(12,7)$ wat een verbeterde ondergrens van $2{,}2184$ oplevert. De methode is hetzelfde als voor het vinden van cap sets. De vectoren van $\{0,1,2\}^m$ worden in een lijst geordend volgens een priorteitsfunctie. Daarna wordt de lijst doorlopen en wordt van de vectoren een voor een bepaald of die nog kan worden toegevoegd aan de groeiende toelaatbare verzameling.

Na analyse van de gevonden prioriteitsfunctie, bleek de $I(12,7)$ bepaalde symmetrieën te bevatten. De verzameling vectoren bleek invariant onder cyclische rotatie van de eerste drie coördinaten. Hetzelfde gold voor de drie coördinaten daarna en zo verder voor elk volgend setje van drie opeenvolgende coördinaten. Door deze symmetrie vervolgens als extra eis in te bakken, lukte het om voor grotere dimensies goede toegestane verzamelingen te vinden, zoals een $I(15,10)$ en een toelaatbare verzameling S van grootte 237984 in dimensie 24 van gewicht 17. De geconstrueerde cap set $S(S_4(A_0,A_1,A_2),A_1^4,A_2^4)$ in dimensie 576 levert een ondergrens op van $2{,}2202$ voor de cap set-capaciteit, een nieuw record!

Conclusie

De onderzoekers hebben laten zien dat Large Language Models zoals GPT-3 kunnen worden gebruikt om nieuwe wiskundige resultaten te vinden. In dit geval werden voor het Cap Set-probleem, een klassiek probleem in de combinatoriek, verbeterde constructies gevonden. De kracht van de aanpak lijkt vooral te liggen in de unieke mogelijkheid van de taalmodellen om snel diverse heuristieken te genereren in de vorm van (computer)taal. Deze kunnen daardoor relatief eenvoudig door wiskundigen worden geanalyseerd om zo tot nieuwe ideeën te komen.

Het Cap Set-probleem zelf is nog verre van opgelost gezien het grote gat tussen de ondergrens $2{,}220$ en de bovengrens $2{,}756$ op de cap set-capaciteit. Mocht het vermoeden van Tyrrell correct zijn, dan zou dit direct eenondergrens geven van $2{,}233$, een eerste kleine stap.

Referenties

1 J. Bierbrauer en Y. Edel, Large caps in projective galois spaces, *Current research topics in Galois geometry*, Jan de Beule en Leo Storme, eds., Nova Science Publishers, 2012, pp. 87–104.

2 A. Davies, P. Veličković, L. Buesing, S. Blackwell, D. Zheng, N. Tomašev, R. Tanburn, P. Battaglia, C. Blundell, A. Juhász et al., Advancing mathematics by guiding human intuition with AI, *Nature* 600 (2021), 70–74.

3 N. de Bruijn, Set!, *Nieuw archief voor wiskunde* 5/3(4) (2002), 320–325.

4 Y. Edel, Extensions of generalized product caps, *Designs, Codes and Cryptography* 31(1) (2004), 5–14.

5 Y. Edel and J. Bierbrauer, Large caps in small spaces, *Designs, Codes and Cryptography* 23 (2001), 197–212.

6 Y. Edel, S. Ferret, I. Landjev en L. Storme, The classification of the largest caps in $\mathrm{AG}(5,3)$, *Journal of Combinatorial Theory, Series A* 99(1) (2002), 95–110.

7 J.S. Ellenberg en D. Gijswijt, On large subsets of $\mathbb{F}_q^n$ with no three-term arithmetic progression, *Annals of Mathematics* 185(1) (2017), 339–343.

8 A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatain, A. Novikov, F.J. Ruiz, J. Schrittwieser, G. Swirszcz, et al., Discovering faster matrix multiplication algorithms with reinforcement learning, *Nature* 610 (2022), 47–53.

9 J. Grochow, New applications of the polynomial method: the cap set conjecture and beyond, *Bulletin of the American Mathematical Society* 56(1) (2019), 29–64.

10 G. Pellegrino, Sul massimo ordine delle calotte in $S(4,3)$, *Matematiche (Catania)* 25(1–9) (1970), 330.

11 A. Potechin, Maximal caps in $\mathrm{AG}(6,3)$, *Designs, Codes and Cryptography* 46 (2008), 243–259.

12 B. Romera-Paredes, M. Barekatain, A. Novikov, M. Balog, M. P. Kumar, E. Dupont, F.J. Ruiz, J.S. Ellenberg, P. Wang, O. Fawzi, et al., Mathematical discoveries from program search with large language models, *Nature* 625 (2024), 468–475.

13 T.H. Trinh, Y. Wu, Q. V. Le, H. He en T. Luong, Solving olympiad geometry without human demonstrations, *Nature* 625 (2024), 476–482.

14 F. Tyrrell, New lower bounds for cap sets, *Discrete Analysis* 2023(20) (2023), 18 pp,.