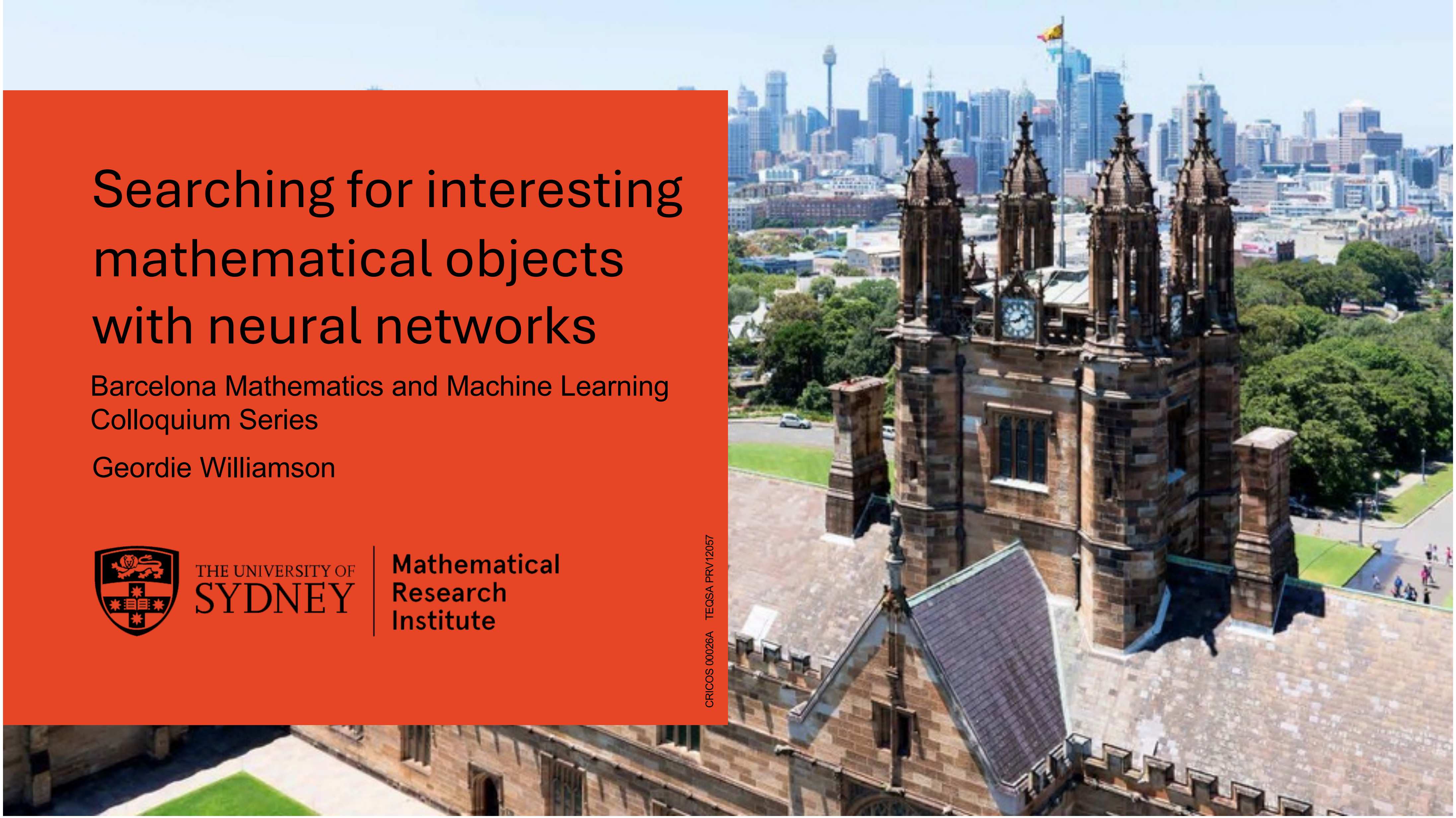


Searching for interesting mathematical objects with neural networks

Barcelona Mathematics and Machine Learning Colloquium Series

Geordie Williamson

THE UNIVERSITY OF SYDNEY | Mathematical Research Institute

CRICOS 00026A TEQSA PRV12057

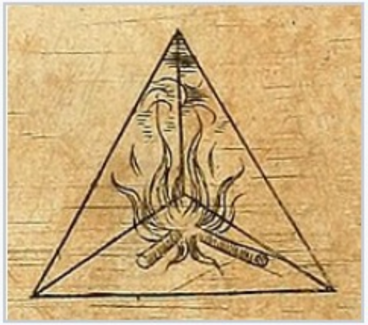
Theorem (Theaetetus?)

There are five regular solids:

cube

icosahedron

octahedron



tetrahedron

dodecahedron

- Unclear if first icosahedron existed physically.

- Enormous influence on the last two millennia of mathematics and physics.

AI and mathematics

AI is a major tool in applied mathematics.

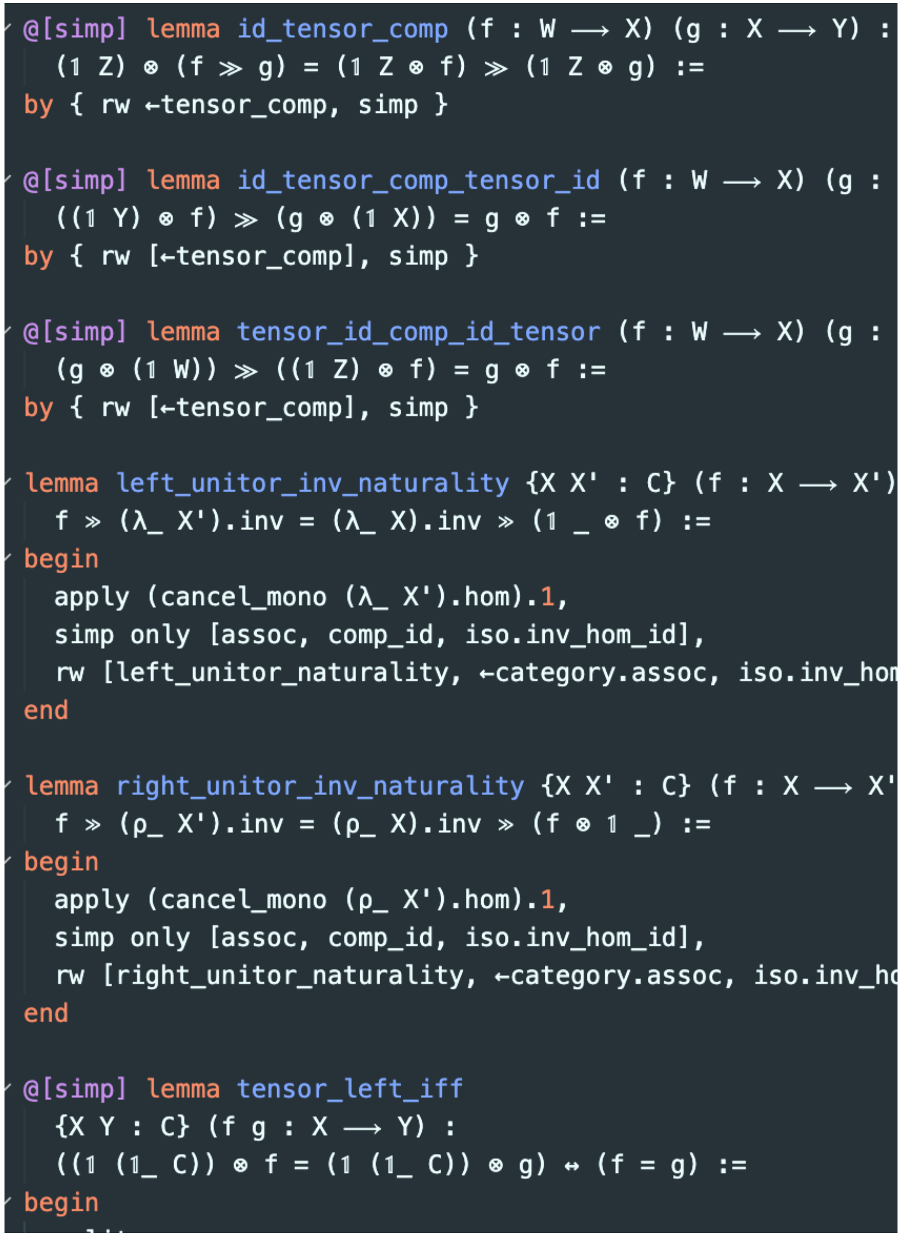
Major recent progress with development and adoption of formal proof assistants (*Lean*, *Isabelle,…*).

Automated proof *remains a major challenge*.

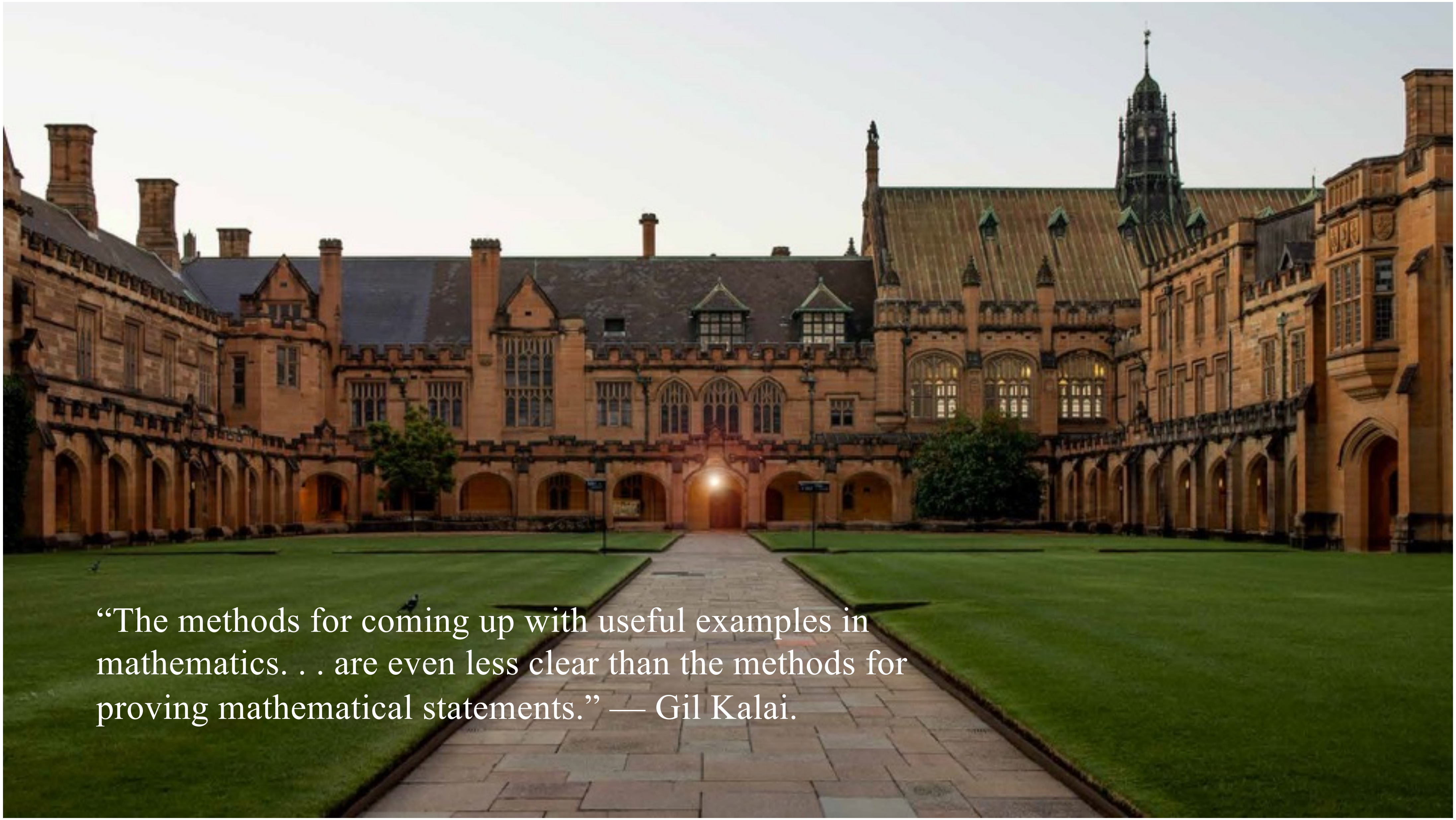
AlphaProof achieved IMO Silver Medal performance in 2024.

Systems that can prove lemmas at graduate level will probably arrive within 1-2 years. (Might need much compute!)

This talk is *not* about Lean, large language models or formal proof assistants.



```
@[simp] lemma id_tensor_comp (f : W ⟶ X) (g : X ⟶ Y) :
  (𝟙 Z) ⊗ (f ≫ g) = (𝟙 Z ⊗ f) ≫ (𝟙 Z ⊗ g) :=
by { rw ←tensor_comp, simp }

@[simp] lemma id_tensor_comp_tensor_id (f : W ⟶ X) (g :
  ((𝟙 Y) ⊗ f) ≫ (g ⊗ (𝟙 X)) = g ⊗ f :=
by { rw [←tensor_comp], simp }

@[simp] lemma tensor_id_comp_id_tensor (f : W ⟶ X) (g :
  (g ⊗ (𝟙 W)) ≫ ((𝟙 Z) ⊗ f) = g ⊗ f :=
by { rw [←tensor_comp], simp }

lemma left_unitor_inv_naturality {X X' : C} (f : X ⟶ X')
  f » (λ_ X').inv = (λ_ X).inv » (𝟙 _ ⊗ f) :=
begin
  apply (cancel_mono (λ_ X').hom).1,
  simp only [assoc, comp_id, iso.inv_hom_id],
  rw [left_unitor_naturality, ←category.assoc, iso.inv_hom
end

lemma right_unitor_inv_naturality {X X' : C} (f : X ⟶ X'
  f » (ρ_ X').inv = (ρ_ X).inv » (f ⊗ 𝟙 _) :=
begin
  apply (cancel_mono (ρ_ X').hom).1,
  simp only [assoc, comp_id, iso.inv_hom_id],
  rw [right_unitor_naturality, ←category.assoc, iso.inv_ho
end

@[simp] lemma tensor_left_iff
  {X Y : C} (f g : X ⟶ Y) :
  ((𝟙 (𝟙_ C)) ⊗ f = (𝟙 (𝟙_ C)) ⊗ g) ↔ (f = g) :=
begin
```



“The methods for coming up with useful examples in mathematics. . . are even less clear than the methods for proving mathematical statements.” — Gil Kalai.

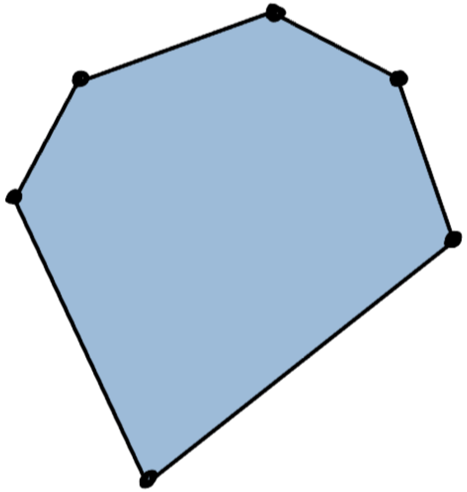
Plan of talk

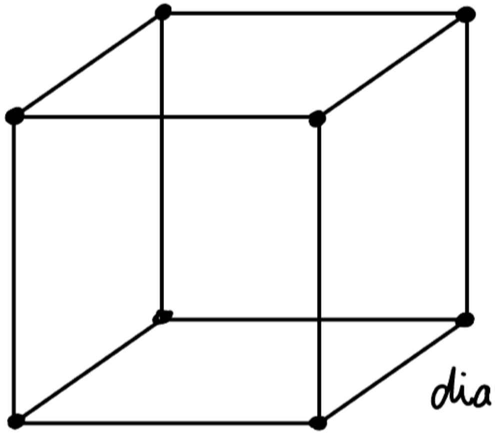
- Some hard problems in mathematics
- A review of neural networks and transformers
- Using neural networks to find interesting mathematical objects

(This talk is *not* meant to be neural network evangelism!)

The width of polytopes

Hirsch conjecture (1957): The vertex-edge graph of any polytope with *n* facets in dimension *d* has diameter at most *n*-*d*.





$d=3, \; n=6,$

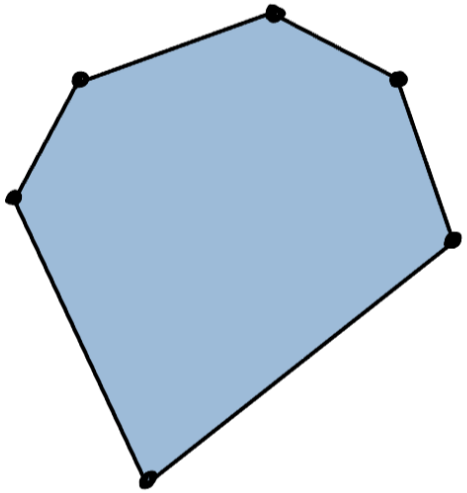
diameler $3 \leqslant 6-3=3$

$d=2, \; n=6,$

diameter $3 \leqslant 6-2=4.$

The width of polytopes

Hirsch conjecture (1957): The vertex-edge graph of any polytope with *n* facets in dimension *d* has diameter at most *n*-*d*.



Smale's 9th Problem: How hard is linear programming?

$d = 2$, $n = 6$,

diameter $3 \leq 6 - 2 = 4$.

Polytopes of exponential width would prove that the simplex method is (worst-case) *exponential time*.

Few people believe these exist, but *I do*!

The width of polytopes

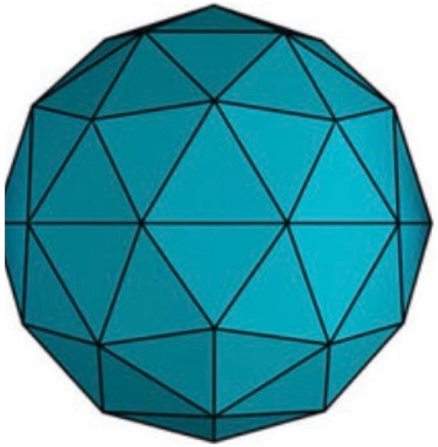
Hirsch conjecture (1957): The vertex-edge graph of any polytope with *n* facets in dimension *d* has diameter at most *n*-*d*.

Santos (2012): There exists a 43-dimensional polytope with 86 facets of diameter 44 > 43 = 86 – 43.
Thus, the Hirsch conjecture is *false!*

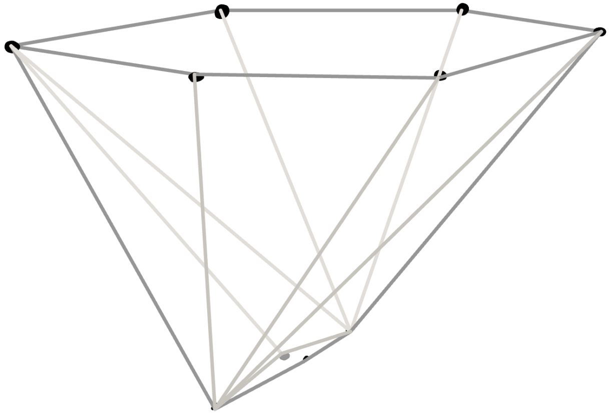
Santos reduced this problem to an extremely difficult problem in 5 dimensions. Very few other examples known. No examples which don't use Santos' 5d trick are known.

The width of polytopes

Hirsch conjecture (1957): The vertex-edge graph of any polytope with *n* facets in dimension *d* has diameter at most *n*-*d*.

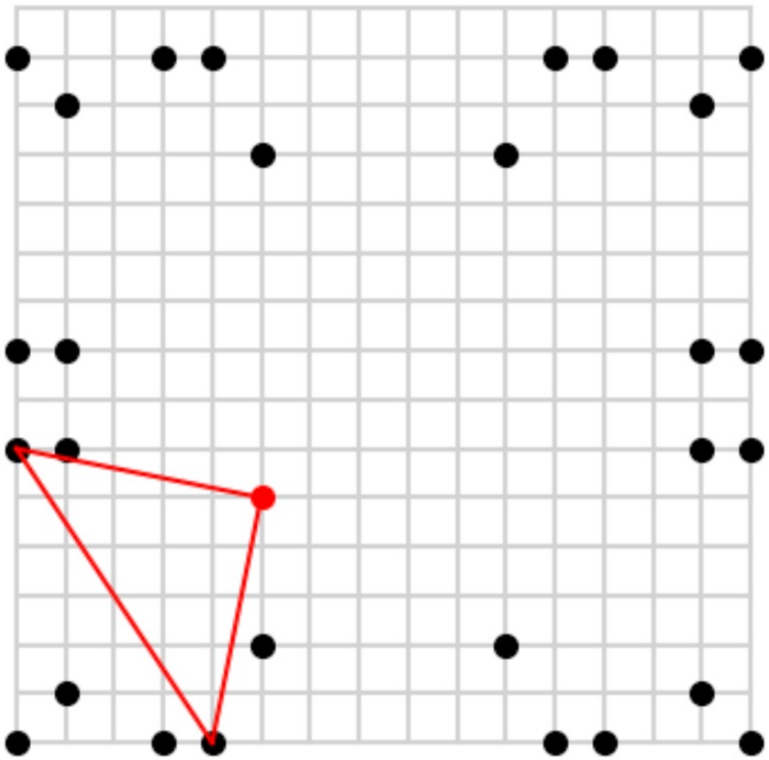


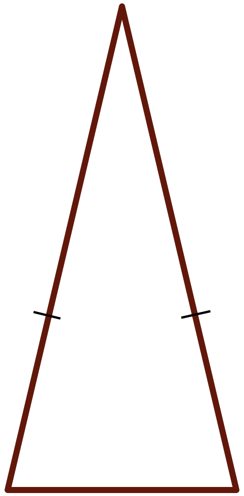
Santos (2012): There exists a 43-dimensional polytope with 86 facets of diameter 44 > 43 = 86 – 43.
Thus, the Hirsch conjecture is *false!*



Davies, Gupta, Racanière, Swirszcz, Wagner, Weber, Williamson:
“Hopper” algorithm using transformer neural network.
Millions of new counter-examples + smallest known counterexample (19 dimensional).

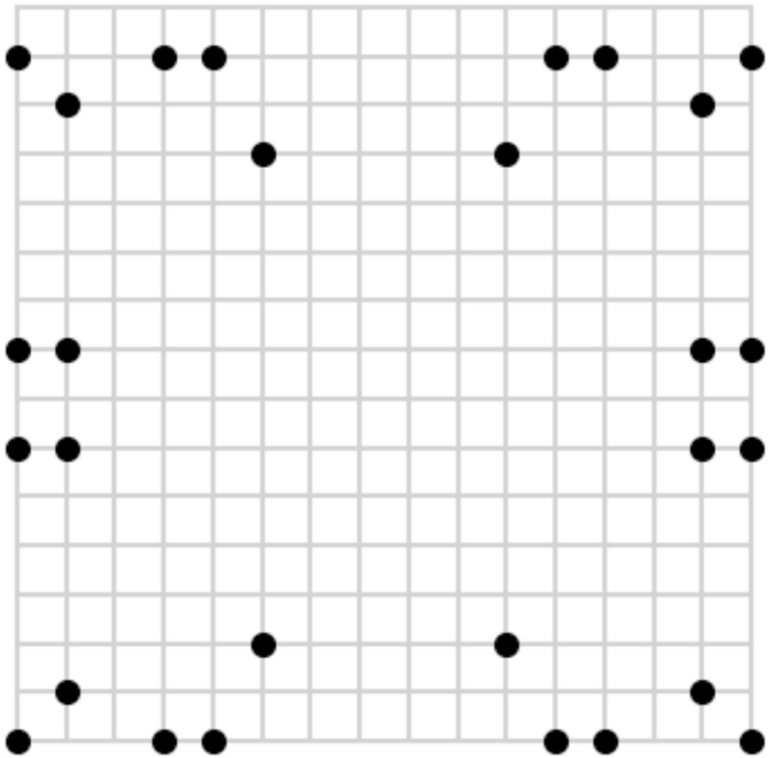
What’s the largest subset of an $n \times n$ grid without isosceles triangles?





an *isosceles* triangle

What’s the largest subset of an $n\times n$ grid without isosceles triangles?



- A $2d$ variant of a classic problem in number theory and additive combinatorics (“progression free subsets”, Green–Tao theorem).

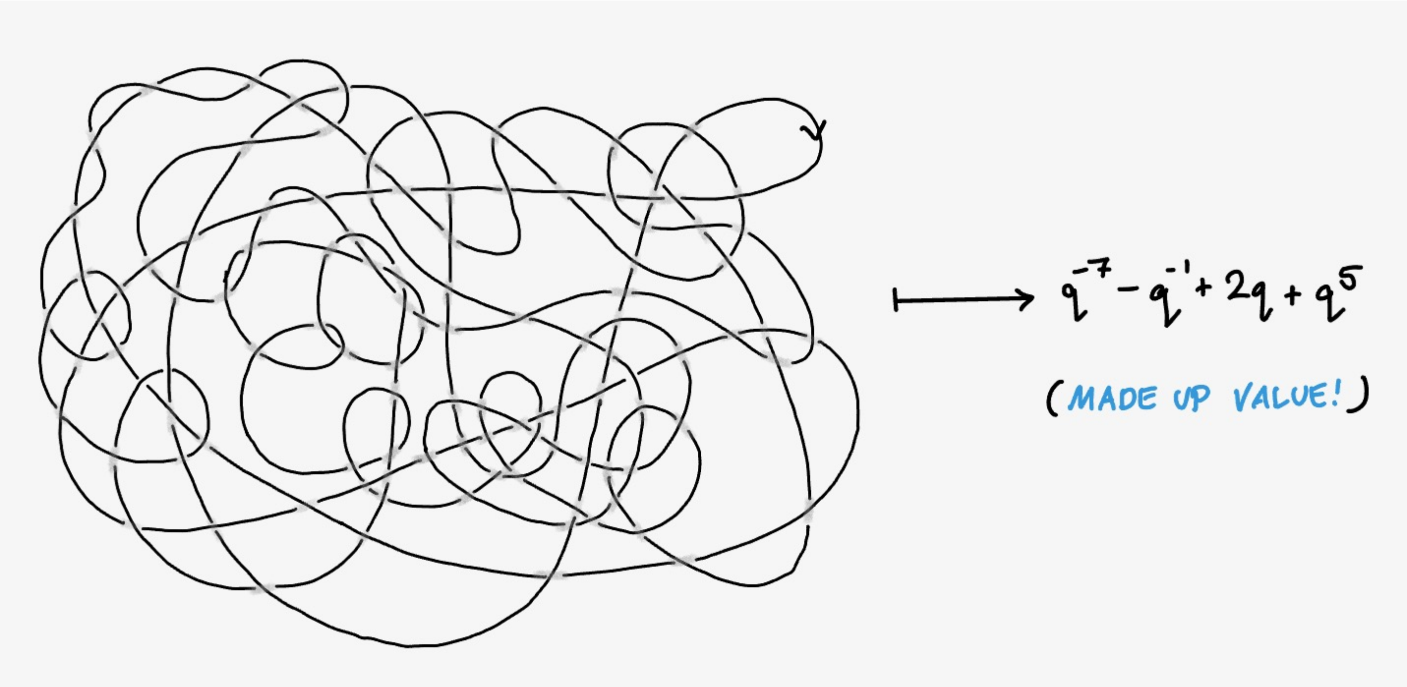
- Little is known about optimal solutions.

- Hard to produce examples which are close to extremal.

- Highly addictive problem!

Does the Jones polynomial detect the unknot?

knot L (oriented) $\longrightarrow$ $V_L \in \mathbb{Z}[q, q^{-1}]$ (Jones polynomial)



Jones unknot problem to matrix problem (Birman, Bigelow, Itô)

$$A = \begin{pmatrix} -q & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & -q^{-1} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1-q & -q^{-1} & q^{-1} \\ 1-q^2 & -q^{-1} & 0 \\ 1 & -q^{-1} & 0 \end{pmatrix}$$

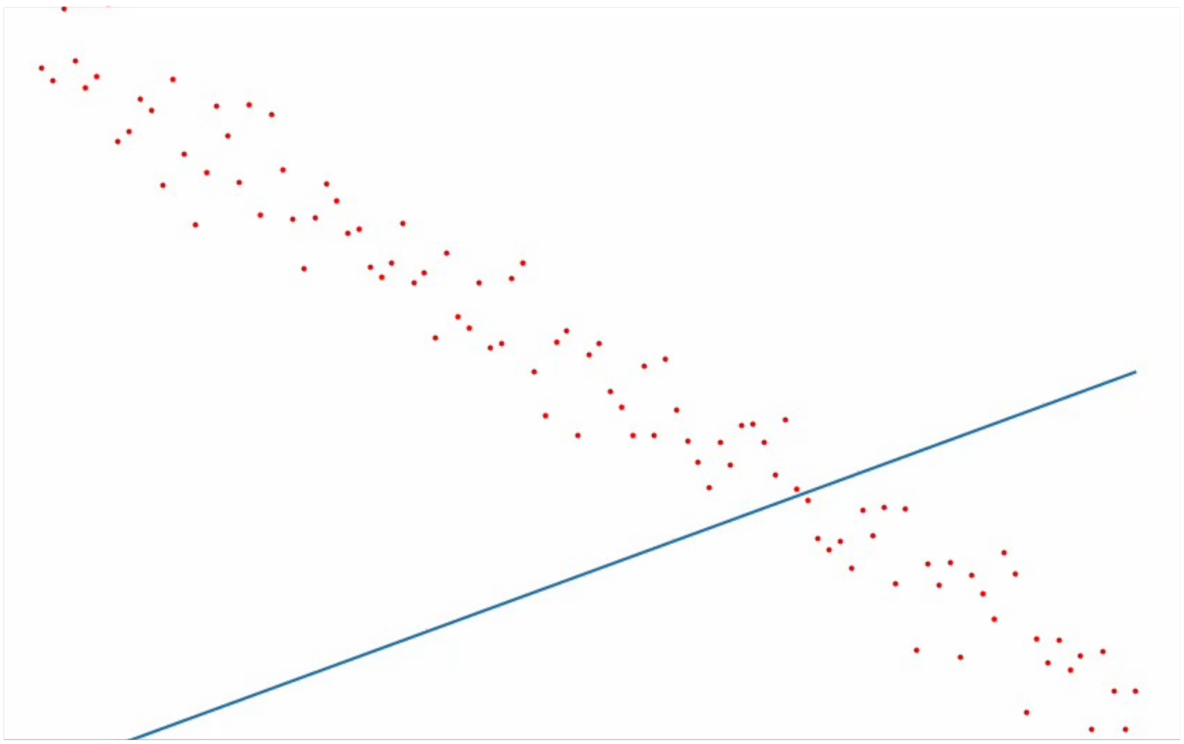
Do *A* and *B* satisfy any non-trivial (multiplicative) relation? Unsolved since 1974.

If yes, then the Jones polynomial *does not* detect the unknot.

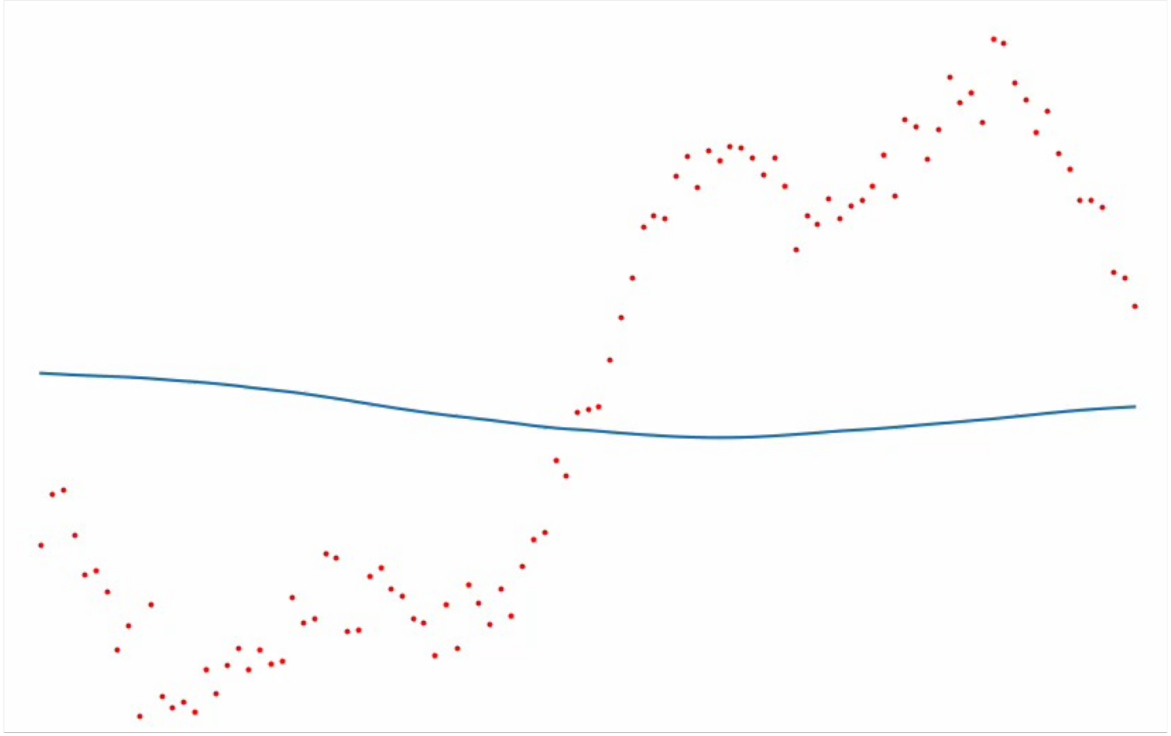
For experts: General case related to faithfulness of certain Jones representations of braid group.

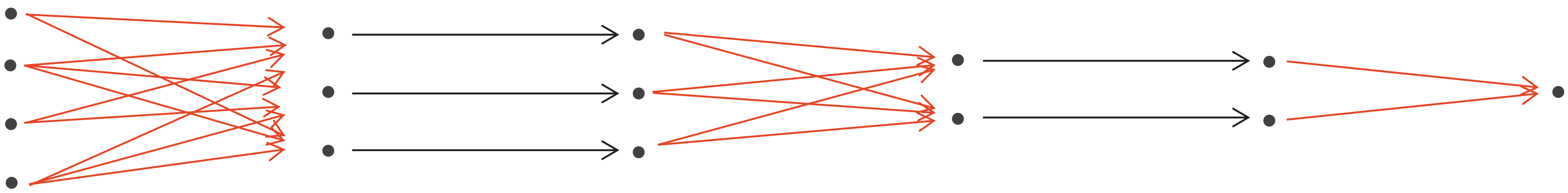
Neural networks and transformers

The line of best fit



Neural network: a powerful approximator





$$\varphi \;:\; \mathbb{R}^{10^4} \xrightarrow[\textit{linear}]{} \mathbb{R}^{10^3} \xrightarrow[\textit{ReLU}]{} \mathbb{R}^{10^3} \xrightarrow[\textit{linear}]{} \mathbb{R}^{10^2} \xrightarrow[\textit{ReLU}]{} \mathbb{R}^{10^2} \xrightarrow[\textit{linear}]{} \mathbb{R}$$

ReLU = coordinatewise max with 0 (“non-linearity”)

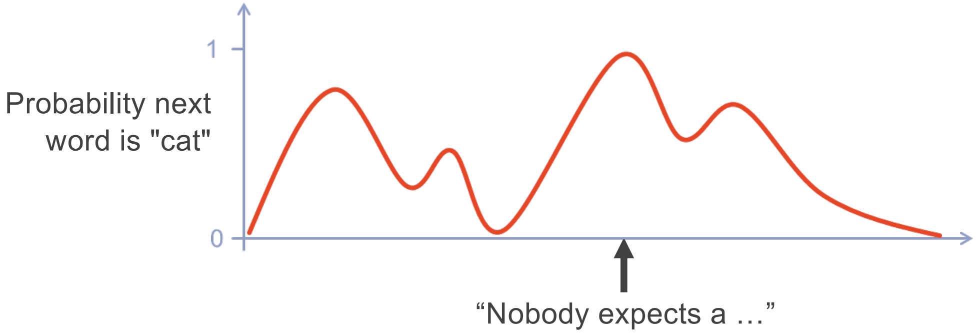
Trained to approximate a target function $\tilde{\varphi}$ via gradient descent on

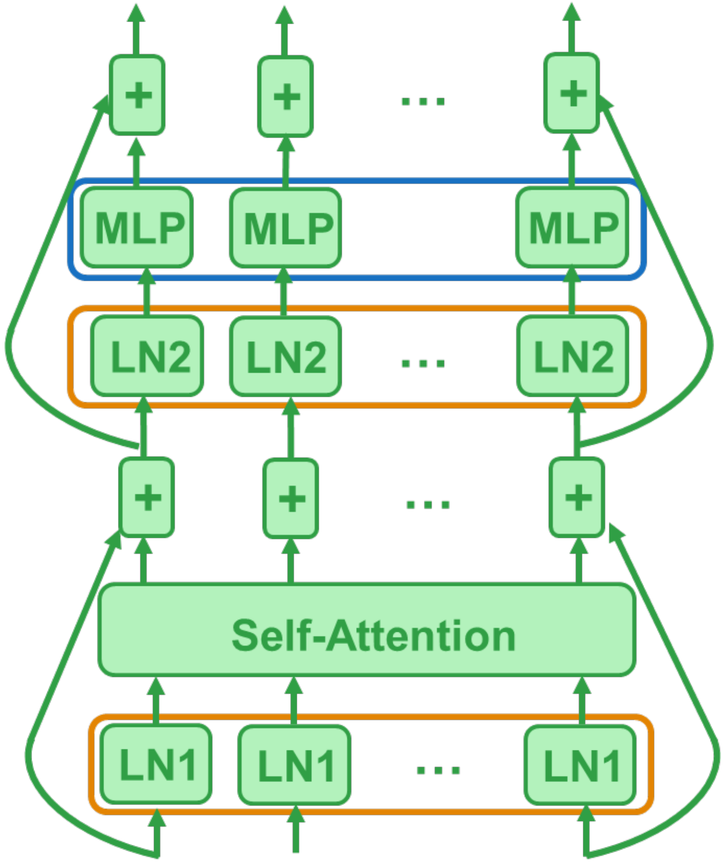
$$Loss \;= \sum\; l\left(\tilde{\varphi}(x), \varphi(x)\right) \longleftarrow$$

e.g. mean squared error
or cross entropy

Transformers

- A new architecture introduced in 2017 for language translation
- The architecture behind Large Language Models like ChatGPT
- Broadly applicable to tasks involving sequential input (e.g. translation, generation, simplification…)





Transformers: basic idea

We want a map

$$f\,:\, sequences\ of\ tokens\ \rightarrow\ \mathbb{R}^T$$

so that the t^{th} coordinate of f is

$$f(t_1 \dots t_{m-1})_t = P(t|t_1 t_2 \dots t_{m-1}).$$

Remark: Given f, we can generate samples from the distribution by first sampling first token t_1 according to $f(\emptyset)$,
then sampling t_2 according to $f(t_1)$,
then sampling t_3 according to $f(t_1 t_2)\dots$

Transformers: some philosophy

How does the brain do symbolic manipulation?

Hypothesis 1 (GOFAI): It is symbols all the way down.

In other words, our brains do something similar to high-school algebra simplifications, e.g.

$$(x-y)(x+y) = (x^2 + xy \ - yx \ - y^2) = x^2 + xy \ - xy \ - y^2 = x^2 - y^2$$

Hypothesis 2 (RNNs): It is just one representation.

As we read *Anna Karenina* we update a *single* state in $\mathbb{R}^{1\,000\,000\,000}$.

Hypothesis 3 (transformers): It is a mix of the two.

As we read *Anna Karenina* we update many states in $\mathbb{R}^{1000} \oplus \mathbb{R}^{1000} \oplus \mathbb{R}^{1000} \oplus \mathbb{R}^{1000} \oplus \mathbb{R}^{1000} \oplus \mathbb{R}^{1000}$ ...

Moreover, these states states “talk to each other” through attention mechanism.

Inspiration for this take was Hinton’s interview at Sana AI summit (see youtube).

Applications in mathematics

Searching for four leaf clover

As mathematicians, we usually have some heuristics as to where to look.

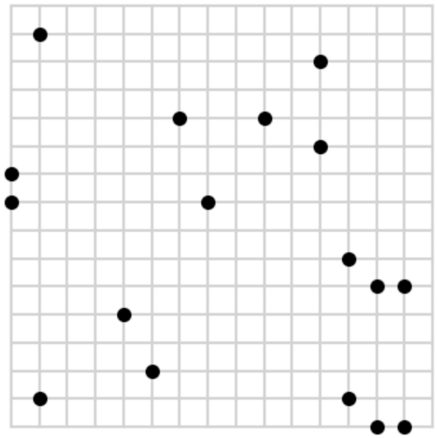
Computers are very good at search,
but often the search space is just too large.

10^{10} might be doable, 10^{50} is typically impossible.

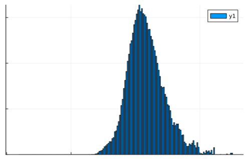
Sometimes just a little help from neural networks can guide us in interesting directions.

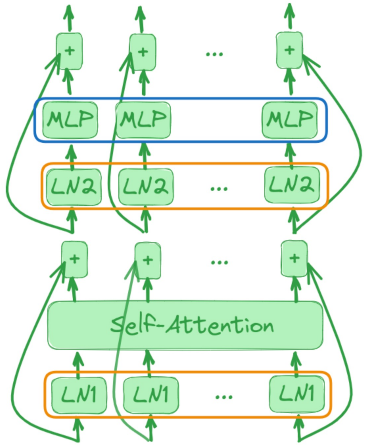
Source: Pickpik

Learning from good solutions



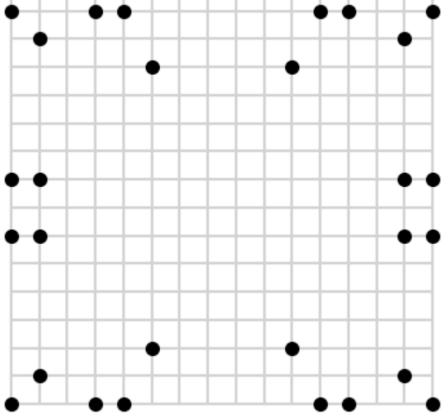
We generate many examples by local search (randomly adding points, without creating isosceles triangles).
We take **top 10%.**





Each construction gives token sequence. These are used to train a GPT-2 style transformer. We then sample from the transformer to produce new examples.

Can Iterate!



A tiny percentage of samples do better than the training set under local search. Amazingly, one often arrives at a near optimal solution.

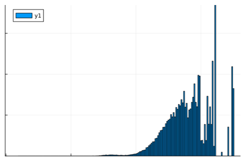


Image ©: dataflowr

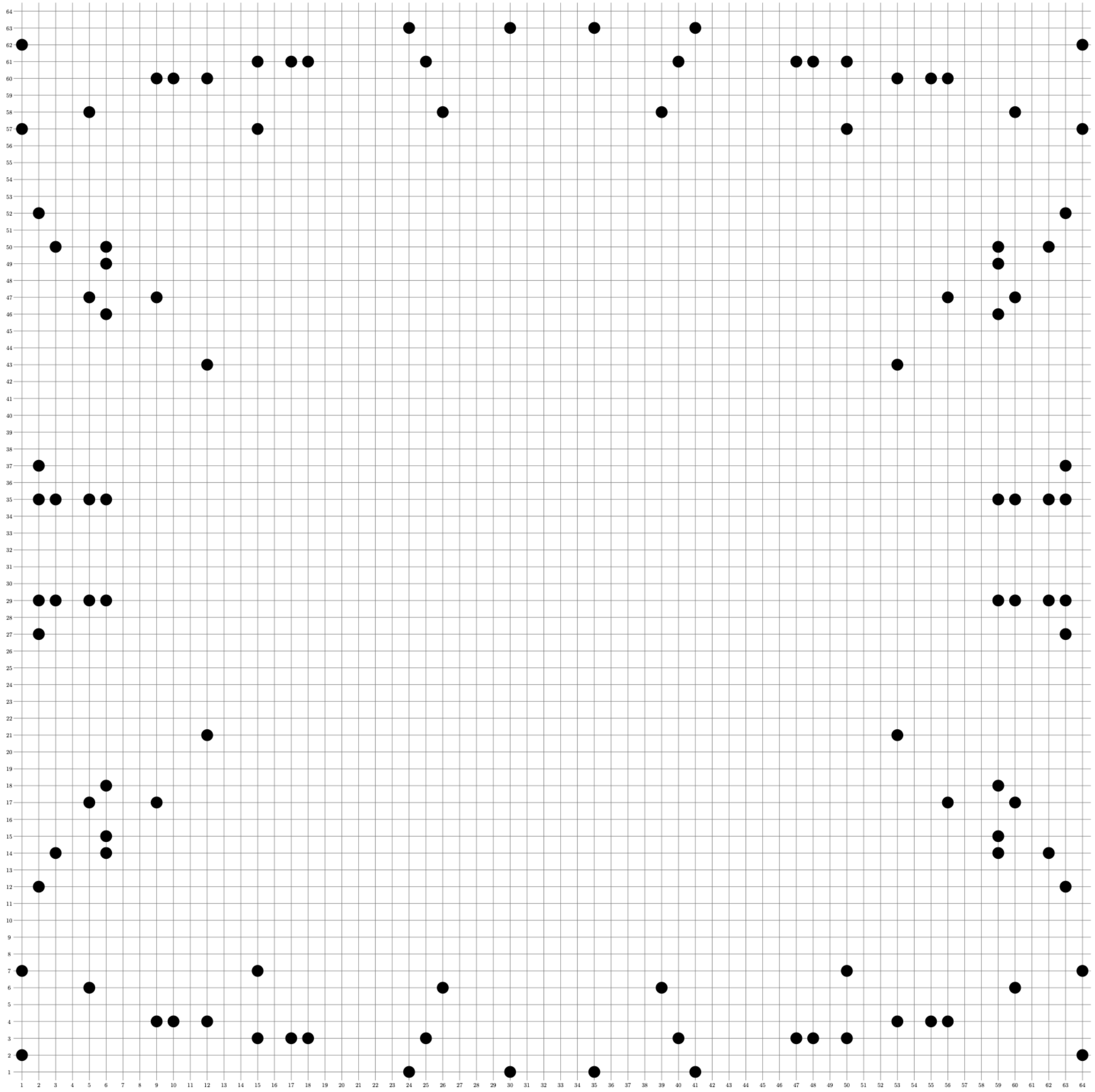
Transformers as generators

Technique works surprisingly well on several problems, but is (of course) not a silver bullet.

This technique generates best known solutions to several problems in extremal combinatorics.

On the right is pictured the best known construction for isosceles free for $n = 64$.

(This example involved months of CPU + a few days of GPU time.)



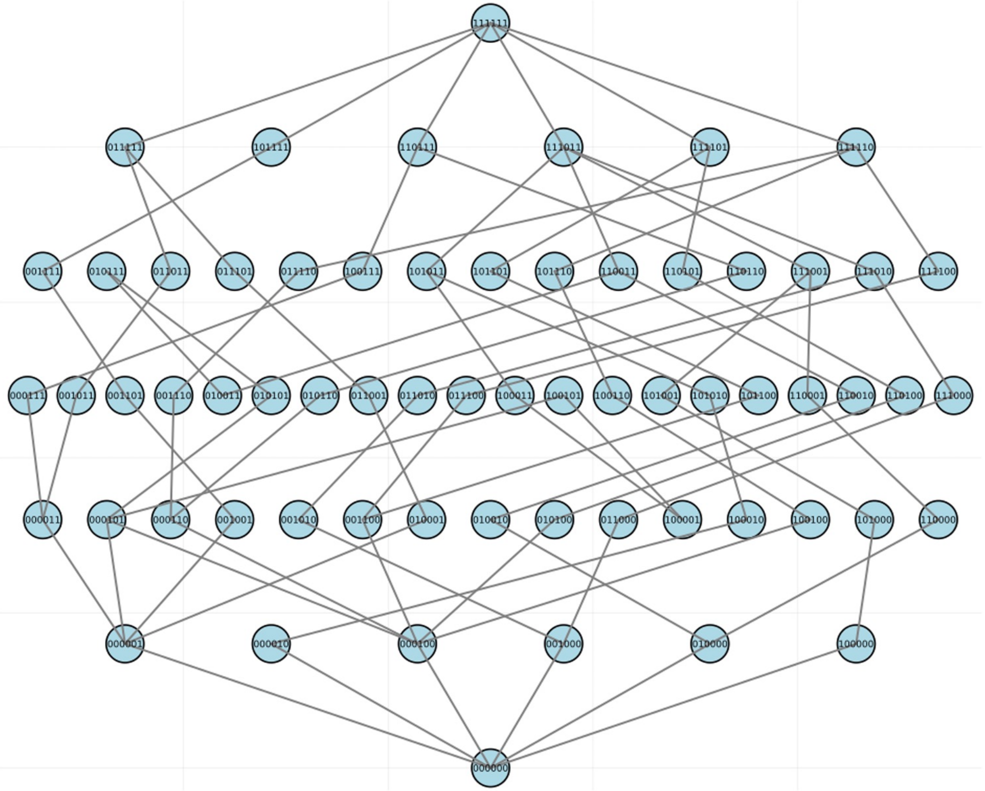
Transformers as generators

Technique works surprisingly well on several problems, but is (of course) not a silver bullet.

This technique generates best known solutions to several problems in extremal combinatorics.

We are also able to disprove a 30-year old conjecture of Niali Graham about the graph of the hypercube:

What is the smallest subgraph of the hypercube which still has diameter n?



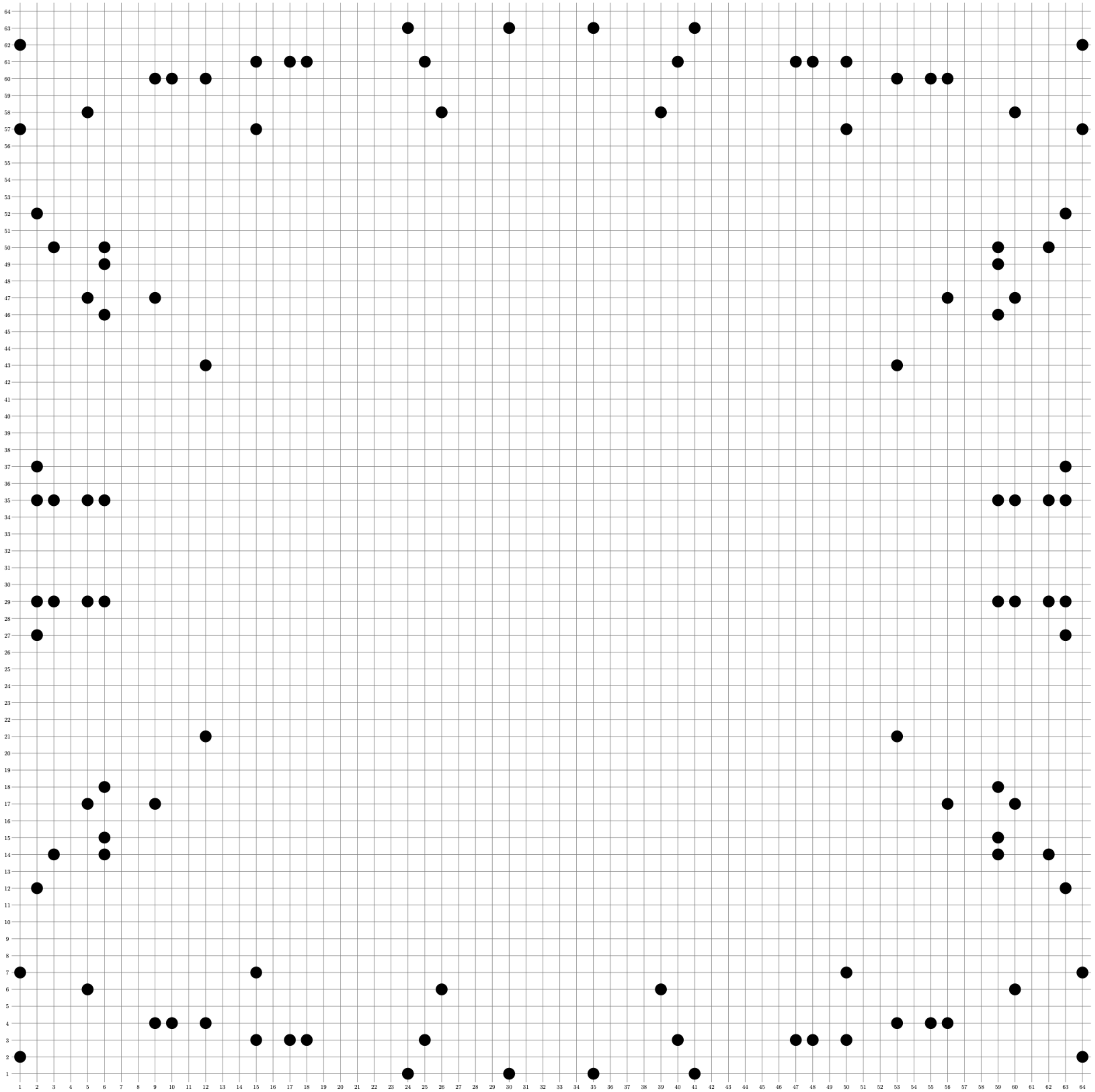
81 edges vs. conjectured 82!

Transformers as generators

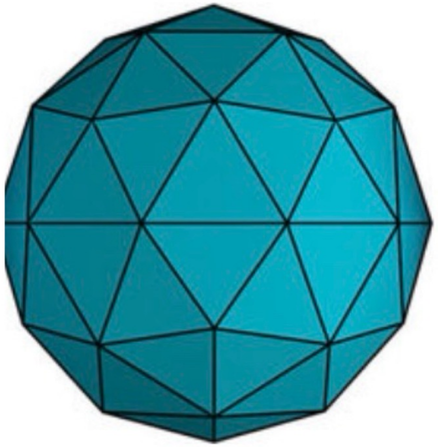
Technique is *broadly applicable*.
Tools from AI are *standard*.

Applications outside of mathematics?

Charton, Ellenberg, Wagner, W.
PatternBoost: Constructions in Mathematics with a Little Help from AI
https://arxiv.org/abs/2411.00566

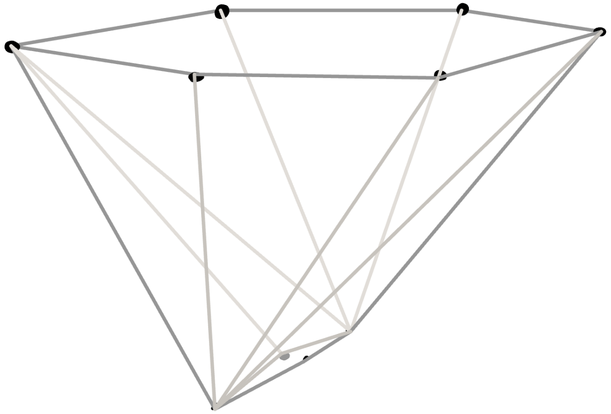


The width of polytopes



Hirsch conjecture (1957): The vertex-edge graph of any polytope with *n* facets in dimension *d* has diameter at most *n*-*d*.

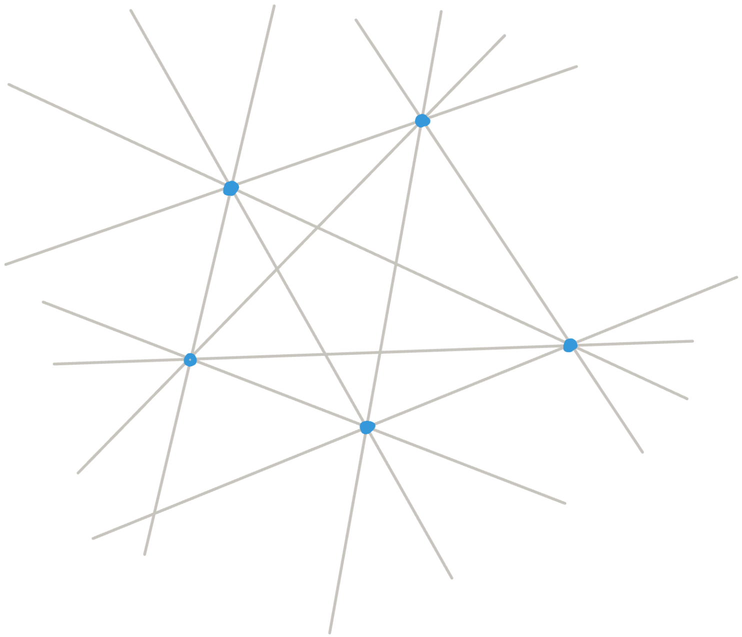
Santos (2012): There exists a 43-dimensional polytope with 86 facets of diameter 44 > 43 = 86 – 43.
Thus, the Hirsch conjecture is *false!*



How to generate interesting examples of polytopes?

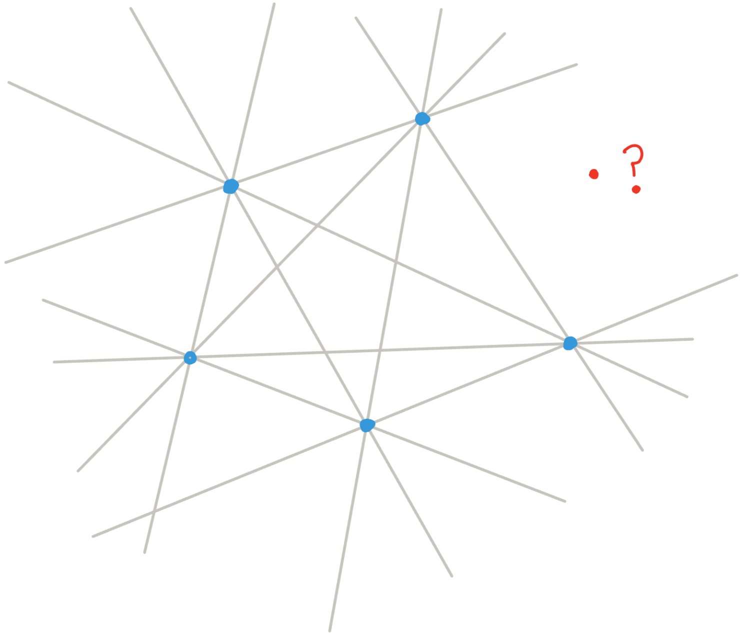
Hopper algorithm

Build up polytope by adding point after point:



Hopper algorithm

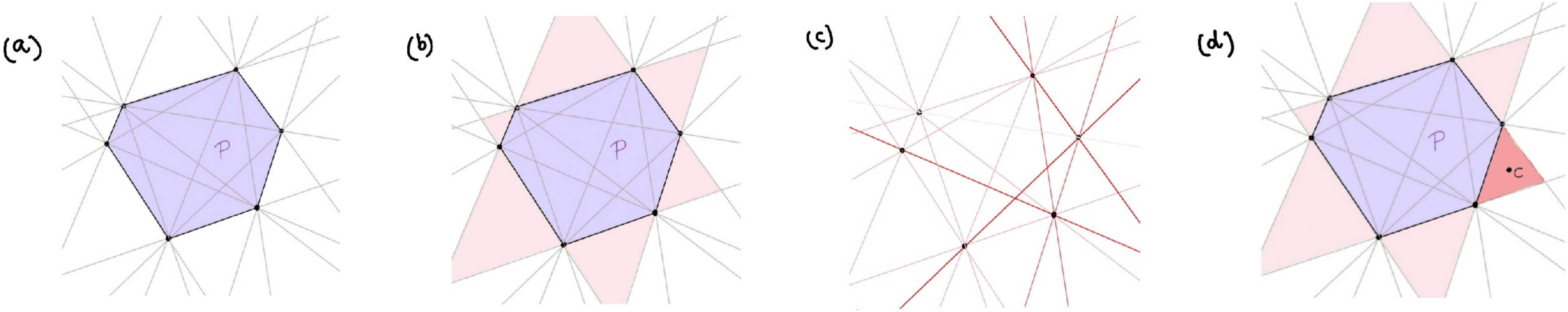
Build up polytope by adding point after point:



Where to add new point?

20 points in 4 dimensions determine
22 950 110 195 021
possible regions!

Hopper algorithm



- Start by random sampling. Remember which hyperplanes were involved.

- Transformer neural network tries to predict hyperplanes which will be involved in promising polytopes (with respect to several heuristics).

- Eventually, roughly 1 in 10^9 searches yields a counter-example.

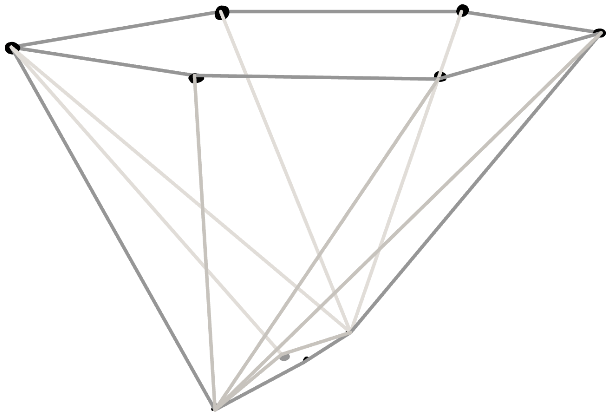
Hopper algorithm: results

Millions of new counter-examples to Hirsch conjecture.
Smallest known counter-example (dimension 19).
(Prior best was dimension 20 by Maschke-Santos-Weibel).

Transformer *appears* to be doing something,
but *difficult to quantify*. (Experiments take many weeks.)

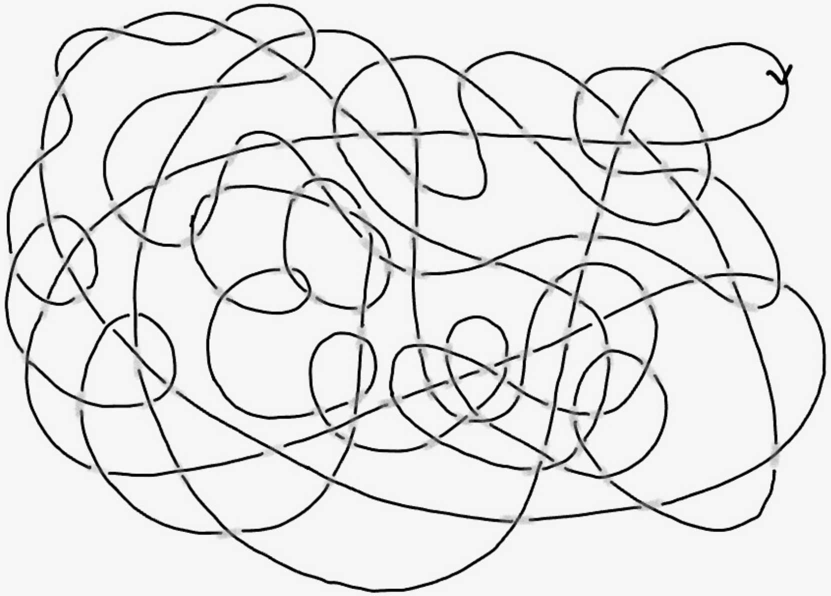
All counter-examples share a striking geometric feature.
amongst *almost degenerate* polytopes.

Davies, Gupta, Racanière, Swirszcz, Wagner, Weber, W.
Advancing geometry with AI. Multi-agent generation of polytopes.
arXiv:2502.05199



Does the Jones polynomial detect the unknot?

knot L (oriented) $\longrightarrow$ $V_L \in \mathbb{Z}[q, q^{-1}]$ (Jones polynomial)



$\longmapsto q^{-7} - q^{-1} + 2q + q^{5}$

(MADE UP VALUE!)

Does the Jones polynomial detect the unknot?

Joint work with Charton, Narayanan and Yacobi (building on earlier work with Gibson-Yacobi).

We concentrate on matrix problems: do there exist unexpected relations between matrices. E.g.

$$A = \begin{pmatrix} -q & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & -q^{-1} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1-q & -q^{-1} & q^{-1} \\ 1-q^2 & -q^{-1} & 0 \\ 1 & -q^{-1} & 0 \end{pmatrix}$$

Appears to be extremely difficult problem. In [CNWY], several new (non-AI) ideas introduced, which allow us to recover *all known relations, and many new ones*.
However, we don’t (yet) find any previously unknown case of non-faithfulness.

The focus here is an interesting part of this work where we use neural networks.

Thought experiment

Imagine you are looking for a rare flower in the bush.

After several days searching you find nothing.

However, you have learnt a lot of other things about the bush.

Can these other things be useful in some way?

Areas of the bush which seem unusual might be more likely to contain rare species, and in particular the flower you are after.

You might want to concentrate your search on areas of “maximum surprise”. That is, areas where your ability to predict your environment is low.

Thought experiment in practice

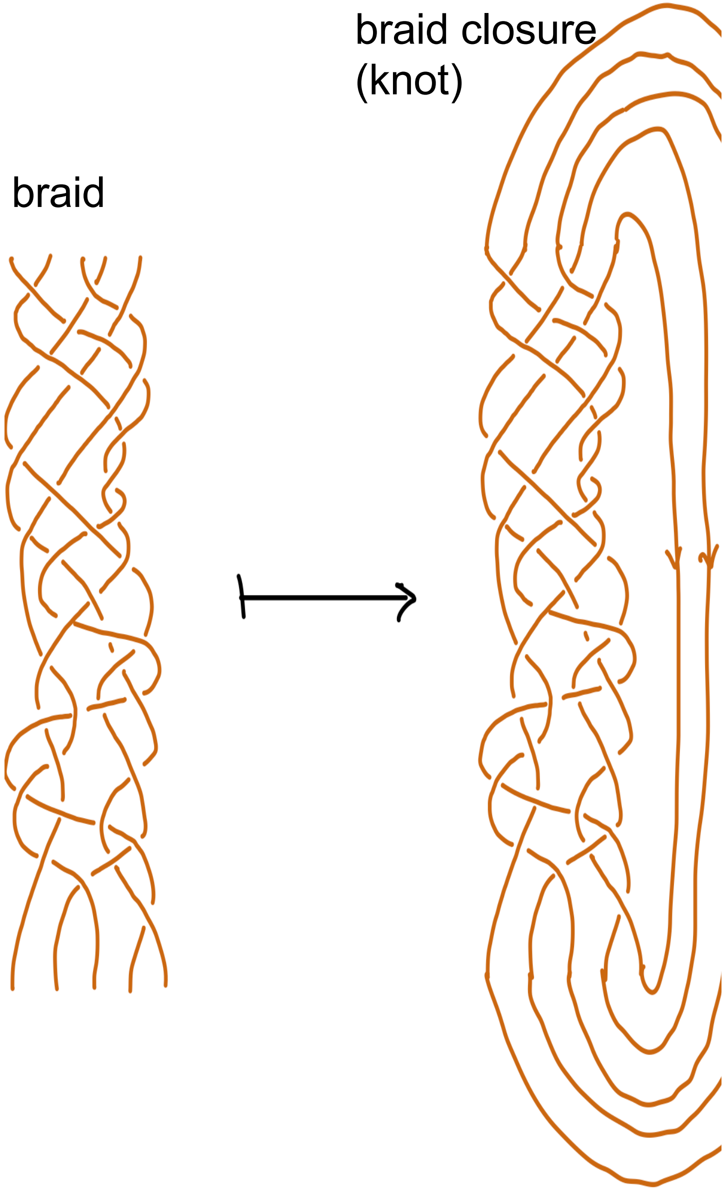
We are looking for a rare knot / braid.

Train a neural network to predict important property of braid (“right descent set”) from matrices alone. Thus, we train the neural network to make an apparently unrelated prediction.

Neural network achieves good accuracy on this problem.

“Descent set confusion”: l^1 norm between prediction and reality.

Use descent set confusion as a score function,
i.e. braids with high descent set confusion
(i.e. unexpected for neural network)
are more likely to be investigated.

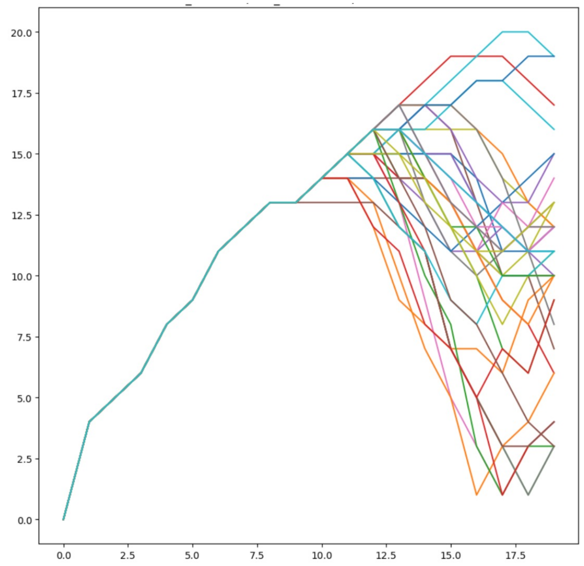


Thought experiment in practice

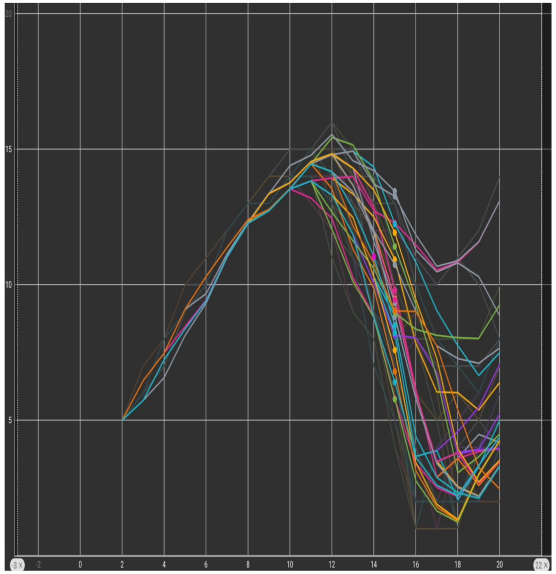
To our delight, this actually works!

On the right we see the results on a toy example. Descent set confusion *dramatically improves* the probability of successful search.

For experts: Here neural network is functioning like a value network in reinforcement learning.



Without neural network, success rate **15%**



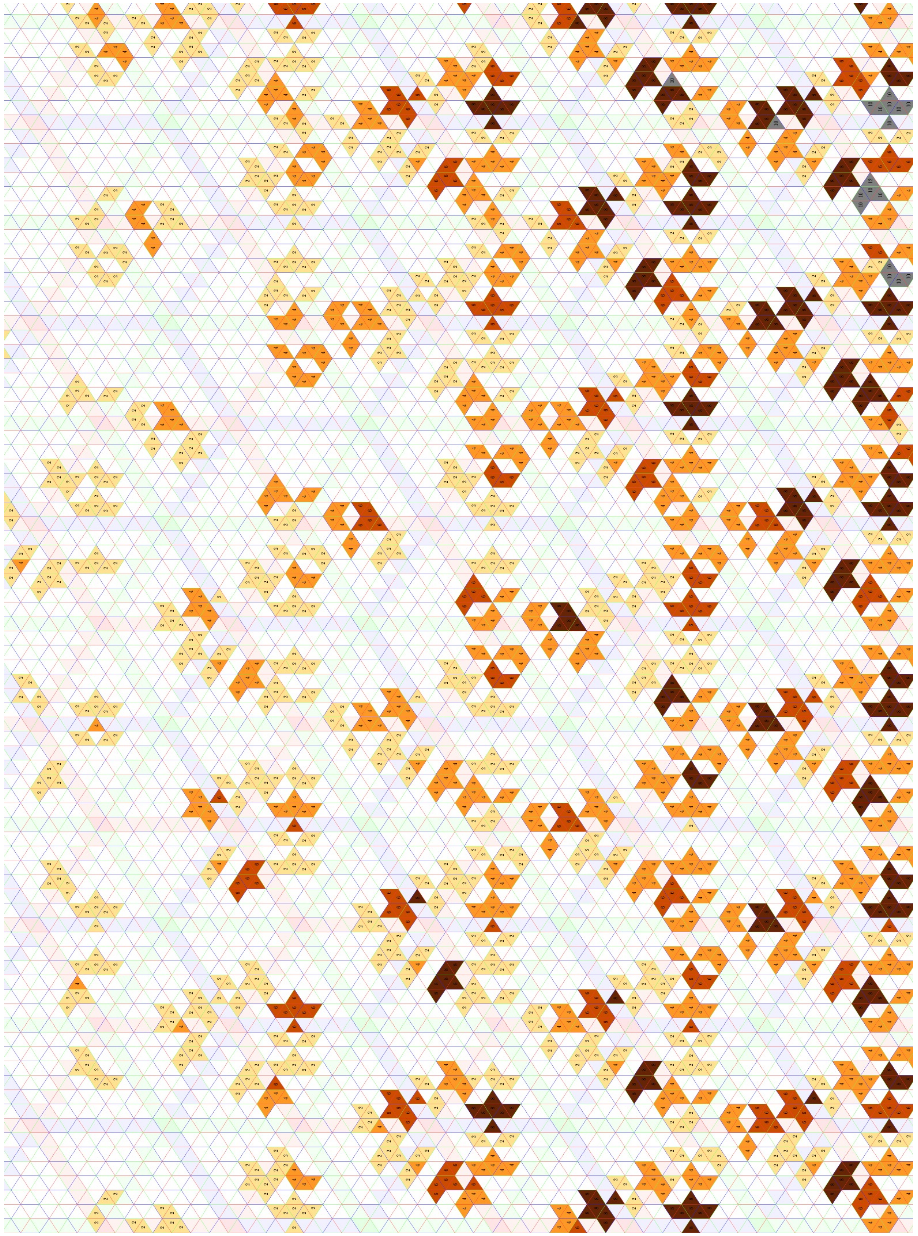
With neural network, success rate 29/39 = **74%**

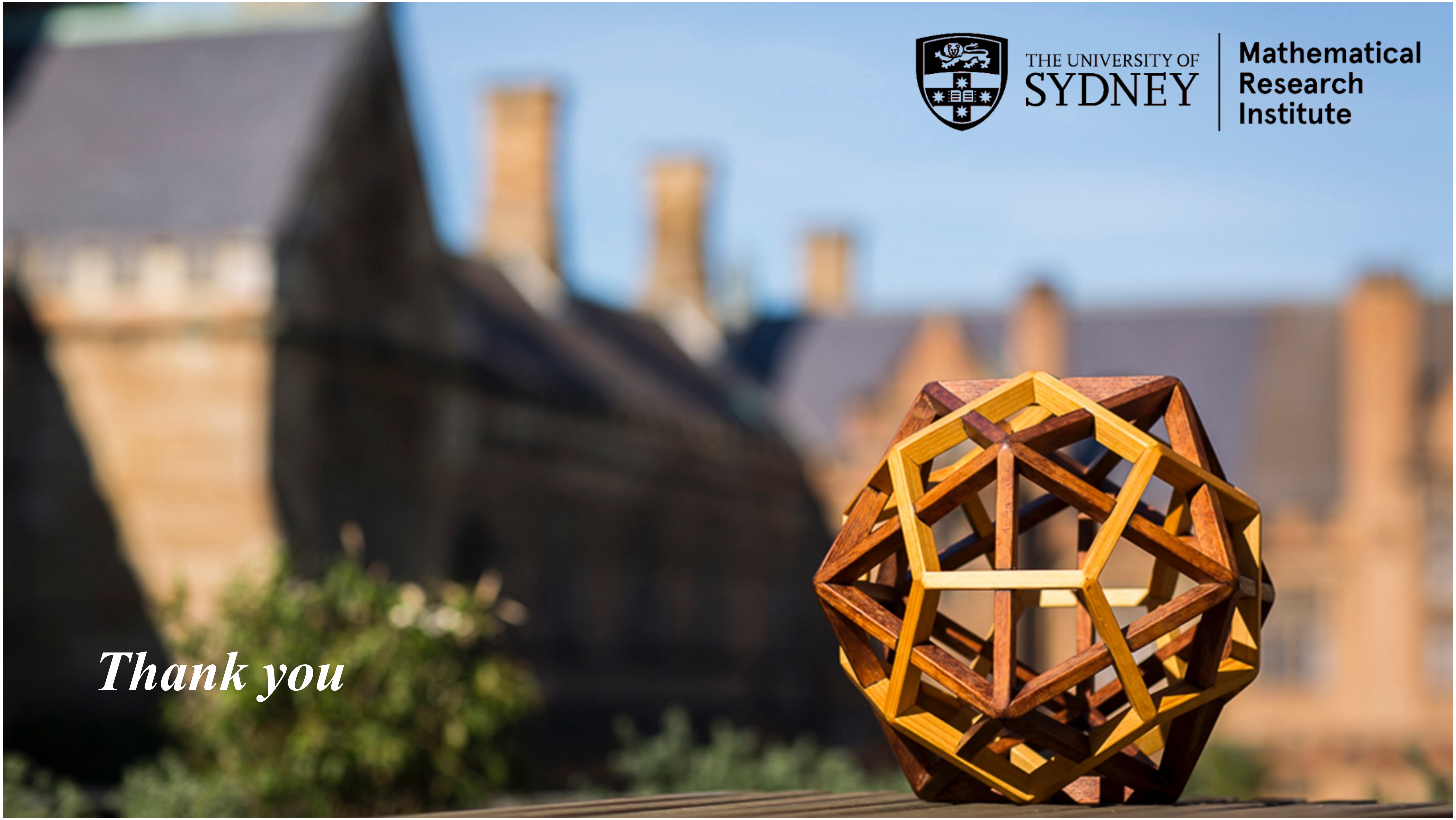
Summary

I have presented three examples where neural networks have helped discover interesting new examples in mathematics.

These are difficult problems, and on no problem do we find a complete solution. However, we do find several new examples and counter-examples.

These techniques are very flexible. Their potential appears almost limitless, once we find the right general approaches.





THE UNIVERSITY OF SYDNEY

Mathematical Research Institute

Thank you