

# AlphaGeometry: Solving olympiad geometry without human demonstrations

February 8, 2024

# AlphaGeometry: Solving olympiad geometry without human demonstrations

Trieu H. Trinh et al., Google Deepmind

<https://www.nature.com/articles/s41586-023-06747-5>

IMO 2004 P1:

"Let  $ABC$  be an acute-angled triangle with  $AB = AC$ . 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

$\text{mid\_point}(O, B, C)$  [--]

$\text{same\_line}(B, M, A)$  [00]  $OM=OB$  [01]

$\text{same\_line}(N, C, A)$  [02]  $ON=OB$  [03]

$\angle BAR = \angle RAC$  [04]  $\angle MOR = \angle RON$  [05]

$\text{circle}(B, M, R, P)$  [06]  $\text{circle}(C, N, R, P)$  [07]

Goal

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

Solve

Proof

[01] [03]  $\Rightarrow \angle ONM = \angle NMO$  [08]

[01] [03] [05]  $\Rightarrow RN = RM$  [09]

[01] [03] [09]  $\Rightarrow NM \perp OR$  [10]

AUXILIARY POINT  $K$  :  $KM = KN$

[01] [03]  $KM = KN \Rightarrow MN \perp KO$  [12]

AUXILIARY POINT  $L$  :  $KL = KA, OL = OA$

$KL = KA, OL = OA \Rightarrow KO \perp AL$  [15]  $\angle AKO = \angle OKL$  [16]

[15] [12] [10] [16] [13]  $\Rightarrow RA = RL$  [17]

$OL = OA \Rightarrow \angle OAL = \angle ALO$  [18]

angle-chase: [12] [15] [08] [18]  $\Rightarrow \angle NOA = \angle LOM$  [19]

[01] [03]  $OL = OA$  [19]  $\Rightarrow AN = LM$  [21]

[17] [21] [09]  $\Rightarrow \angle NAR = \angle RLM$  [22]

[02] [04] [00] [22]  $\Rightarrow \text{circle}(L, M, A, R)$  [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]

[00] [02] [25] [26] [27]  $\Rightarrow PC \parallel BP$

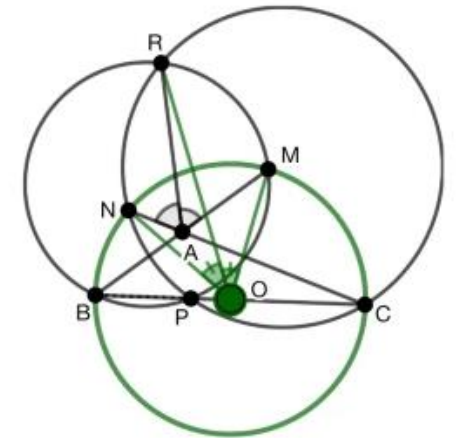
$\Rightarrow \text{same\_line}(B, P, C)$

ABC Unused premise

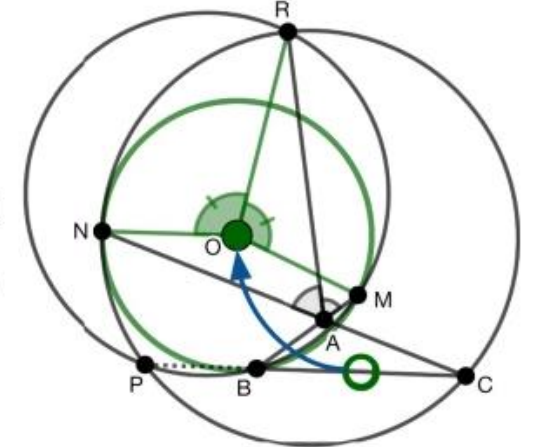
ABC Used premises

ABC Neural net output

ABC Symbolic solver output



Generalize



# AlphaGeometry overview

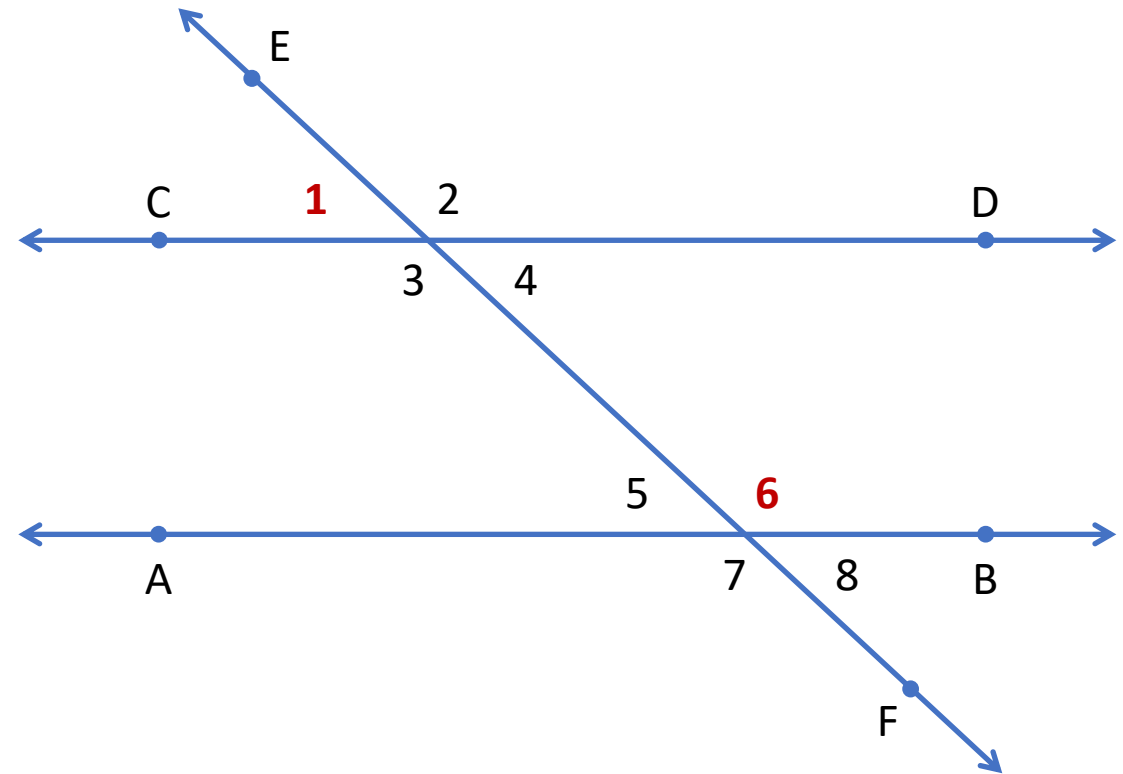
- Solves Euclidean plane geometry proofs from International Mathematical Olympiad contest problems, achieving SOTA
- Combines off the shelf symbolic deduction engine with an LLM
  - Symbolic engine expands list of facts by matching thousands of theorem premises with existing facts, calculating the closure
  - LLM suggests new objects to construct, allowing further deduction
  - The space of auxiliary construction is huge, so LLM must reason well
- They solved the training data problem by generating synthetic proofs
- LLM was pretrained on full proofs, then fine-tuned on subset requiring auxiliary construction

# Symbolic deduction engine

- Symbolic deduction engine has a database of geometric rules
  - They added algebraic rules
- Rules are Horn clauses of the form  $Q(x) \leftarrow P_1(x_1), P_2(x_2), \dots, P_k(x_k)$ 
  - Each  $P_i$  or  $Q$  is a predicate such as “equal segments” or “collinear”
  - Each  $x$  is a set of one or more point objects
- If all of the needed  $P$ ’s are matched, then  $Q(x)$  is added to the list of known facts
- The symbolic engine alone will solve simpler proofs that don’t require constructing additional objects, but it won’t construct new objects
- Existing techniques and code generally ran in seconds

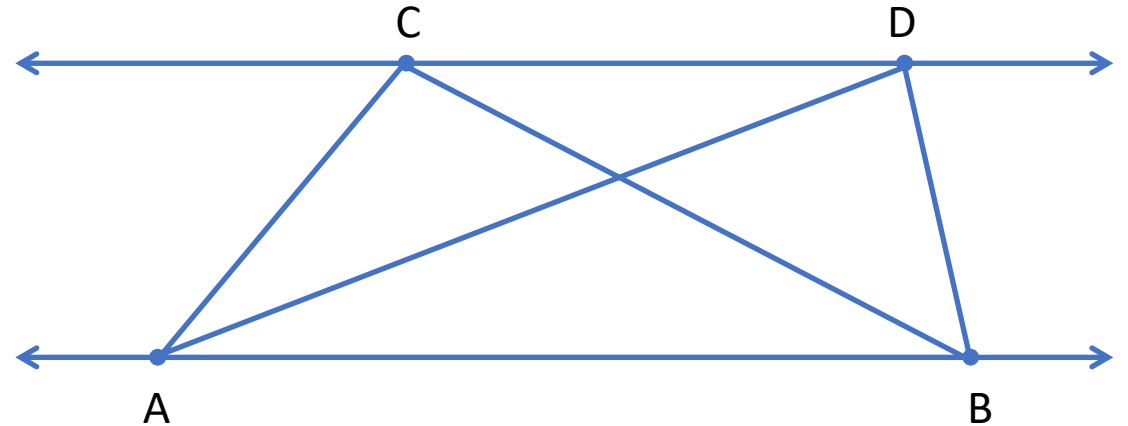
# Deduction example

- We want to prove that angle 1 and angle 6 are supplementary, given parallel lines AB and CD
- First, symbolic engine might prove that angle 1 equals angles 4, 5, and 8
- Then, it might prove that angle 6 and angle 8 are supplementary
- Combining the above two, we have shown that angle 1 and angle 6 are supplementary



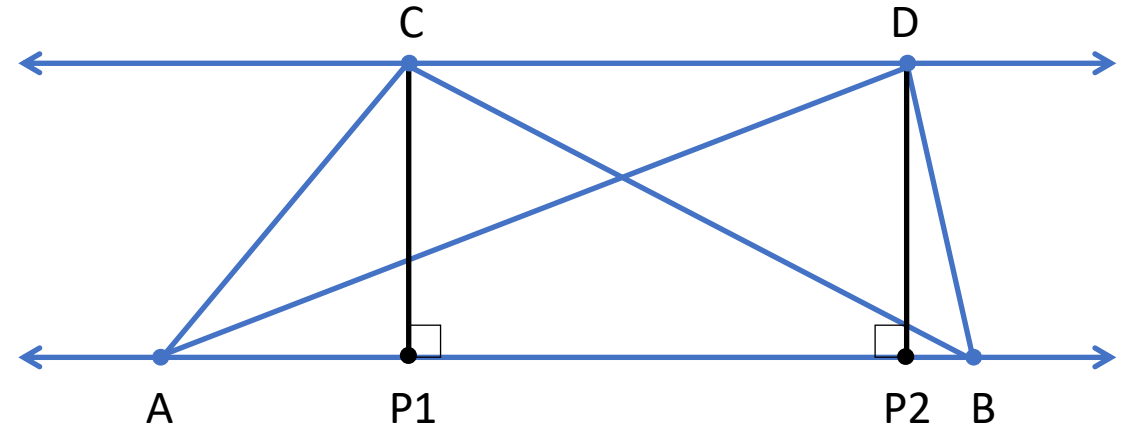
# Deduction insufficiency example [1]

- We want to prove that triangle ABC and triangle ABD have the same area, given parallel lines AB and CD



# Deduction insufficiency example [2]

- We want to prove that triangle ABC and triangle ABD have the same area, given parallel lines AB and CD
- Adding two points P1 and P2 allows us to show the heights of both triangles are the same
- Since both share the same base, the areas are the same



# Training data

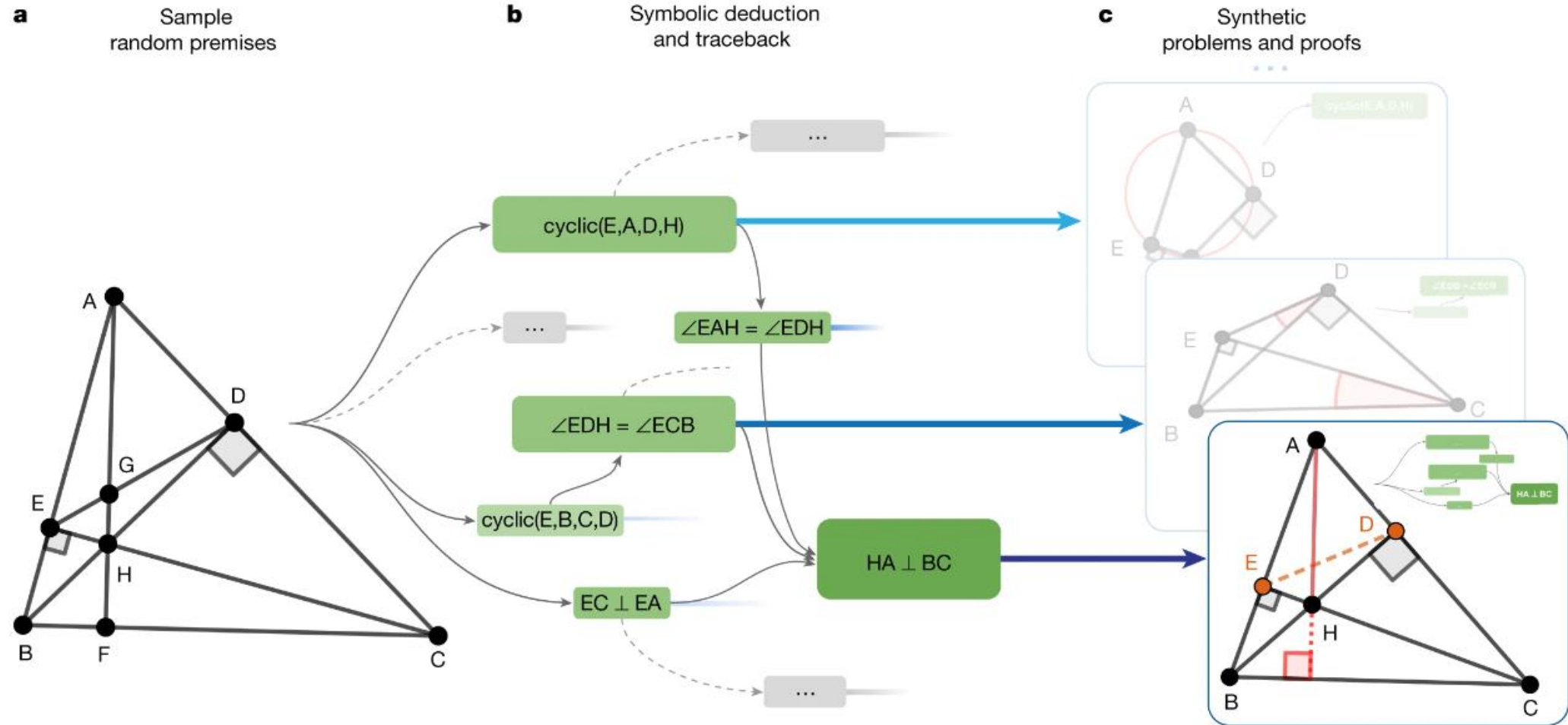
- Arguably the biggest problem was needing a very large set of training data
- Very few geometry proofs available to train on
  - Geometry can be difficult to translate into general purpose mathematical languages such as Lean
  - Geometry-specific languages exist, but each can only express a subset of concepts needed for proofs
- Decided against using any human proofs
- Instead, created their own set of synthetic theorems and proofs



# Synthetic data

- Start by sampling random theorem premises
- Use symbolic deduction engine to expand to deduction closure
  - Track each deduction step, forming a DAG
- For each node  $N$ , trace backward along DAG for minimal set of premises  $P$  and deductions in a subgraph  $G(N)$
- This forms a training example:
  - Premises  $P$
  - Conclusion  $N$
  - Proof  $G(N)$

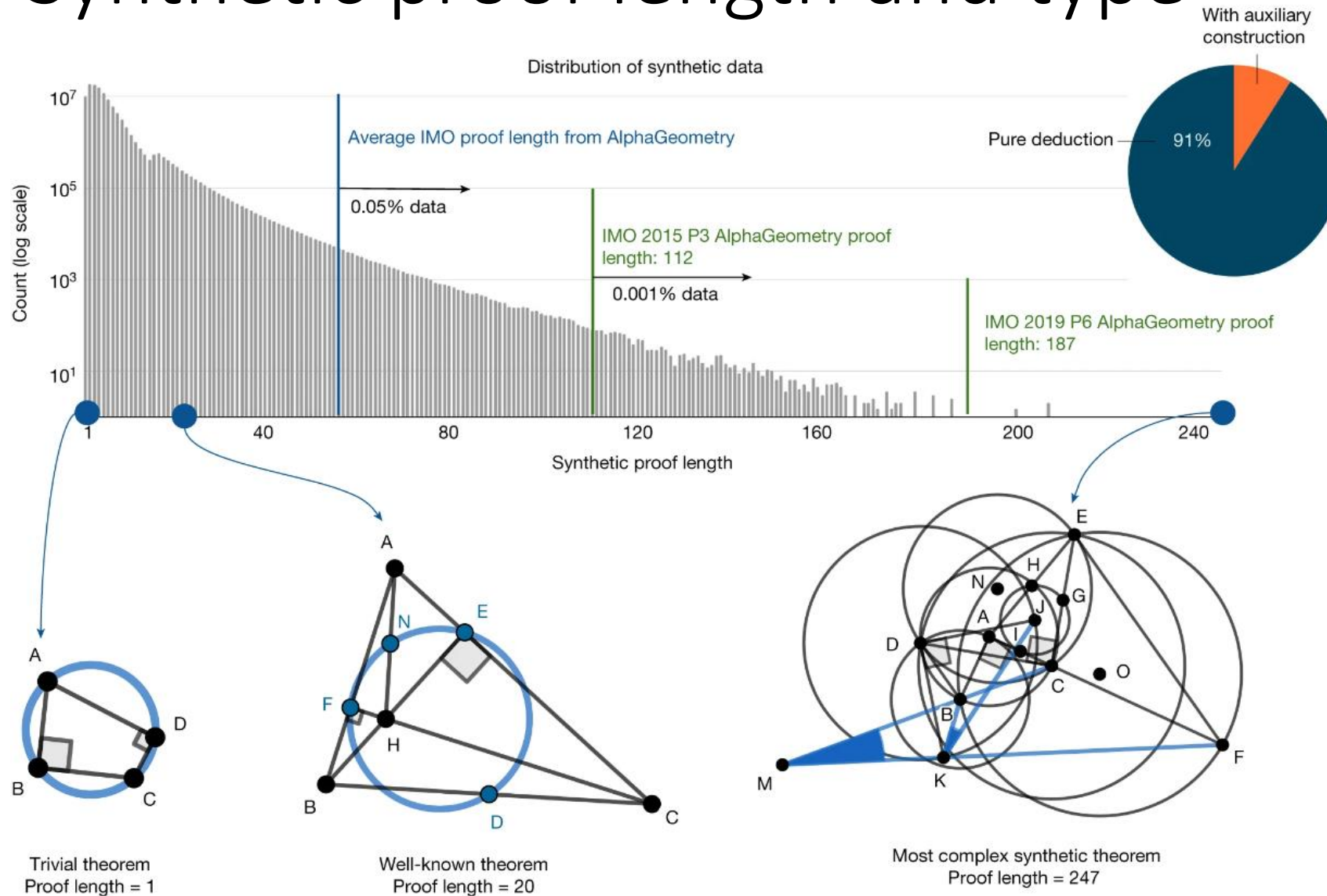
# Synthetic data process



# Synthetic auxiliary construction

- Start with a proof:
  - Premises  $P$
  - Conclusion  $N$
  - Proof  $G(N)$
- The subset of  $P$  that  $N$  is independent of can be removed from  $P$  and added to the proof as auxiliary construction steps
  - Consider our points  $P1$  and  $P2$  from the triangle area example
- We now have harder proof(s) that require auxiliary construction
  - Footnote: they perform exhaustive testing with all subsets of auxiliary points to ensure all of them are really needed

# Synthetic proof length and type



# LLM and Training

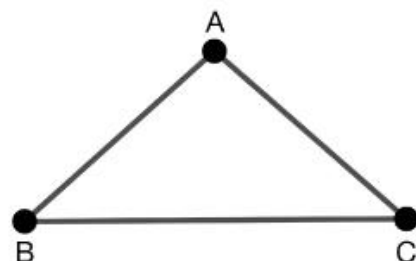
- GPT style decoder only LLM
  - 12 layers, model dimension 1024, 8 attention heads, 4x MLP expansion, ReLU
  - Max sequence length 1024
  - 151 million parameters
  - T5-style relative position embeddings
  - Custom tokenizer, vocab size 757
- Convert each proof into a text string of the form “<premises><conclusion><proof>”
- Pretraining usual next token prediction with cross-entropy loss
  - Over 100 million synthetic proofs in training set
- Fine-tune on just the 9 million proofs requiring auxiliary construction

# Proof Solving with LLM Inference

- Alternate between the symbolic engine and LLM
- Start the LLM with the prompt “<premises><conclusion>” and have it generate one auxiliary construction such as “construct point X so that ABCX is a parallelogram”
  - Use beam search keeping top  $k$  completions (default was  $k=512$ , max depth=16)
- Each symbolic engine turn, expand the deduction closure for all  $k$  new constructions
  - If the conclusion is reached, the proof is complete
- On subsequent LLM turns, append prior constructions to the prompt and have the LLM generate one additional auxiliary construction
  - Note that the LLM doesn't get to see the symbolic engine's deduction closure

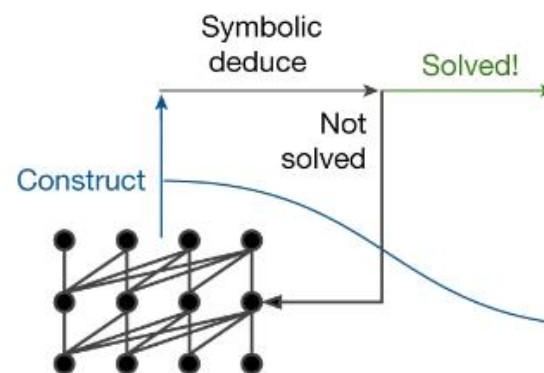
# Proof solving workflow

**a** A simple problem



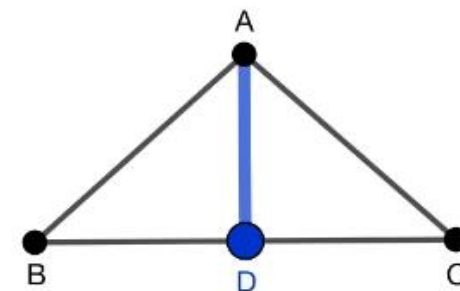
"Let ABC be any triangle with  $AB = AC$ .  
Prove that  $\angle ABC = \angle BCA$ ."

**b** AlphaGeometry



**c** Language model

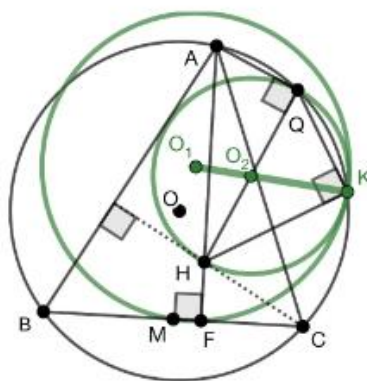
**d** Solution



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

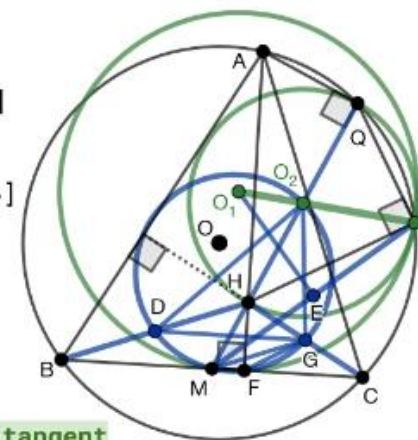
"Let ABC be an acute triangle. Let (O) be its circumcircle, H its orthocenter, and F the foot of the altitude from A. Let M be the midpoint of BC. Let Q be the point on (O) such that  $QH \perp QA$  and let K be the point on (O) such that  $KH \perp KQ$ . Prove that the circumcircles  $(O_1)$  and  $(O_2)$  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]  
 $\angle FK O_1 = \angle FK O_2 \Rightarrow K O_1 \parallel K O_2$  [109]  
 [109]  $\Rightarrow O_1 O_2 K$  collinear  $\Rightarrow (O_1)(O_2)$  tangent



# Results

- 30 IMO proofs since 2000 were this kind of geometry
- AlphaGeometry beat existing computer algebra and search methods, including their attempts to boost DD and GPT-4 with techniques used here

Method		Problems solved (out of 30)
Computer algebra	<a href="#">Wu's method (previous state of the art)</a>	10
	<a href="#">Gröbner basis</a>	4
Search (human-like)	<a href="#">GPT-4</a>	0
	<a href="#">Full-angle method</a>	2
	<a href="#">Deductive database (DD)</a>	7
	<a href="#">DD + human-designed heuristics</a>	9
	DD + AR (ours)	14
	DD + AR + GPT-4 auxiliary constructions	15
	DD + AR + human-designed heuristics	18
	AlphaGeometry	<b>25</b>
	• Without pretraining	21
	• Without fine-tuning	23
	• Only 20% of training data	21
	• Beam search $k=8$	21



# AlphaGeometry conclusion

- AlphaGeometry solves Euclidean plane geometry problems better than any computer baseline, and at the level of top IMO competitors
  - AlphaGeometry solved 25/30 and silver medalist avg. 22.9; gold avg. 25.9
- Data-scarcity was solved by synthetically generating training data
  - Key factor was generating proofs with auxiliary construction
- Final solution uses symbolic engine for deduction and LLM for suggestions for auxiliary construction
  - They call this combination a “neuro-symbolic” system
- These search-style proofs can be made human-readable, unlike the computer algebra techniques which just output True

# References

- AlphaGeometry blog post:  
<https://deepmind.google/discover/blog/alphageometry-an-olympiad-level-ai-system-for-geometry/>
- AlphaGeometry GitHub repo:  
<https://github.com/google-deepmind/alphageometry>
- Generative language modeling for automated theorem proving  
Polu, S. and Sutskever, I. (2020)  
<https://arxiv.org/abs/2009.03393>
- Llemma: An Open Language Model For Mathematics  
Azerbayev, Z. et al. (2023)  
<https://arxiv.org/abs/2310.10631>