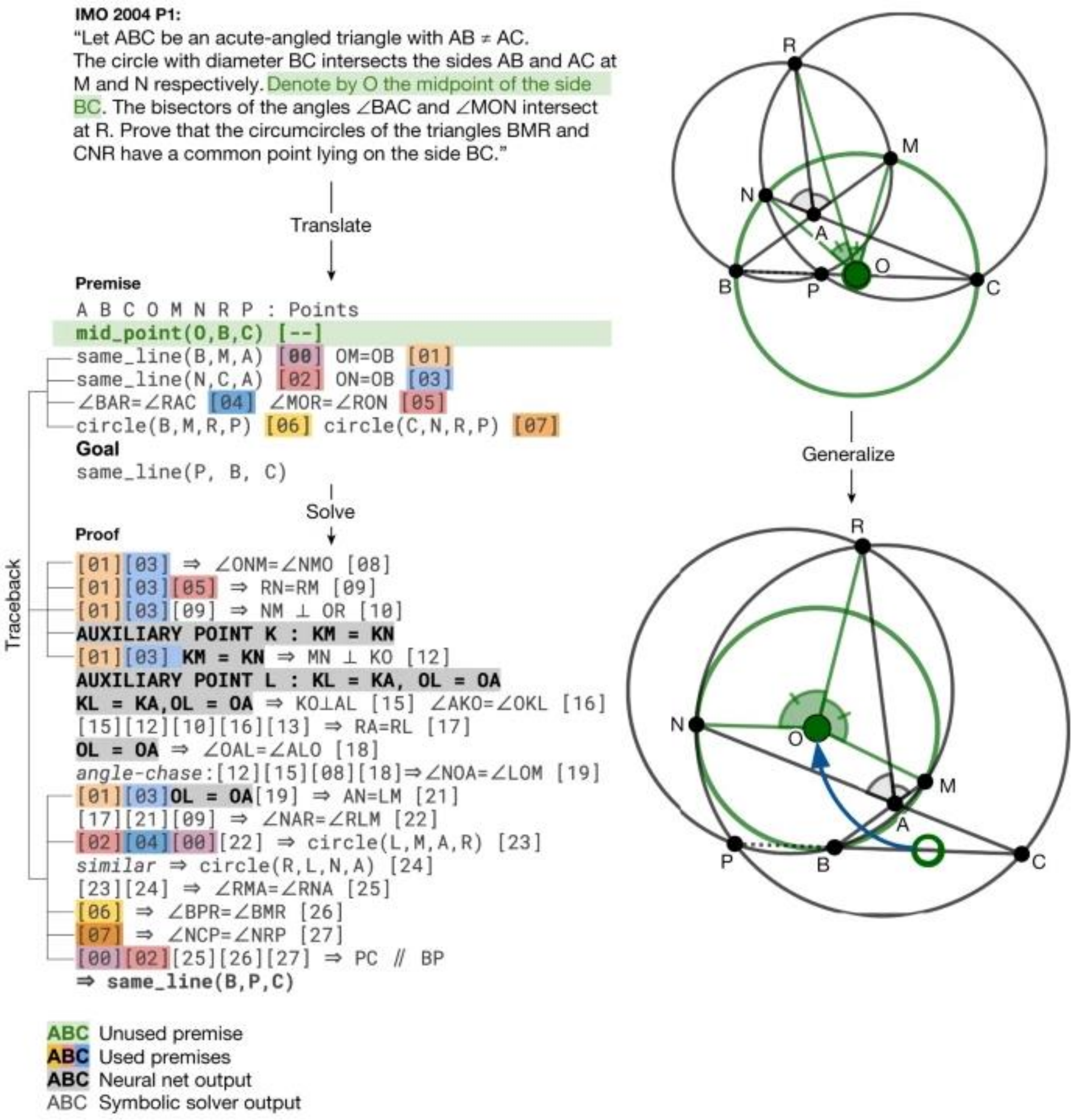
AlphaGeometry: Solving olympiad geometry without human demonstrations

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



AlphaGeometry overview

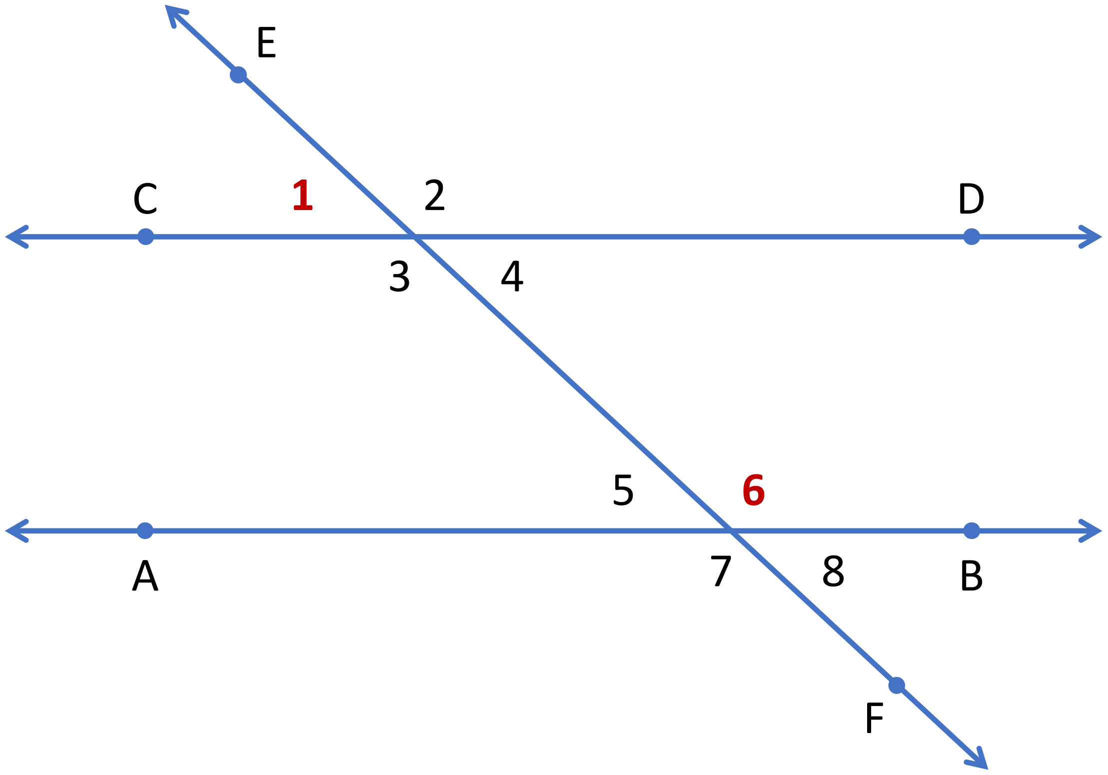
- Solves Euclidean plane geometry proofs from International Mathematical Olympiad contest problems, achieving SOTA
- Combines off the shelf symbolic deduction engine with an LLM
  - Symbolic engine expands list of facts by matching thousands of theorem premises with existing facts, calculating the closure
  - 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 $Q(x) \leftarrow P_1(x_1), P_2(x_2), \ldots, P_k(x_k)$
  - 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

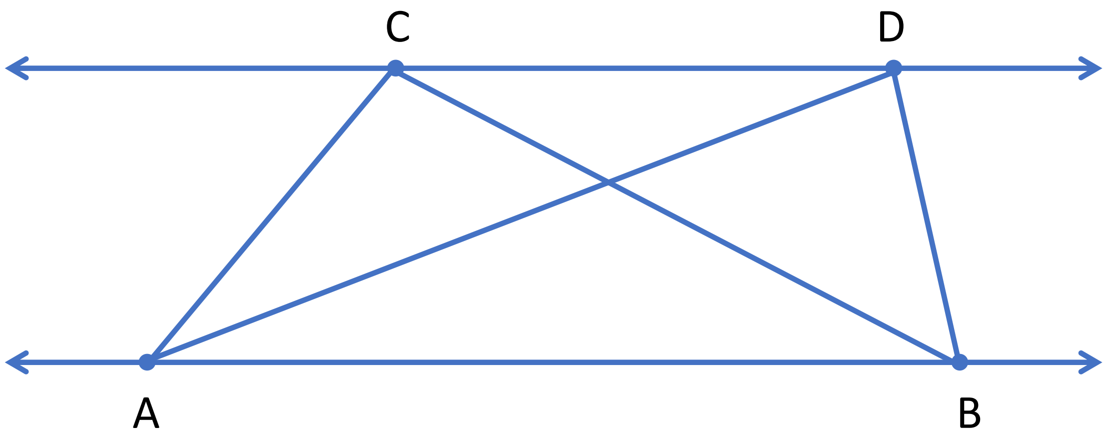
Deduction example

- 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



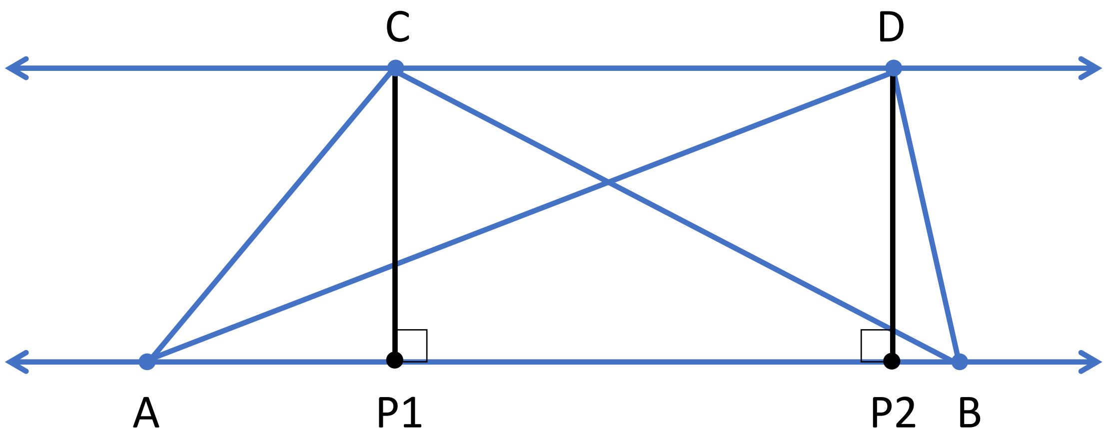
Deduction insufficiency example [1]

- 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



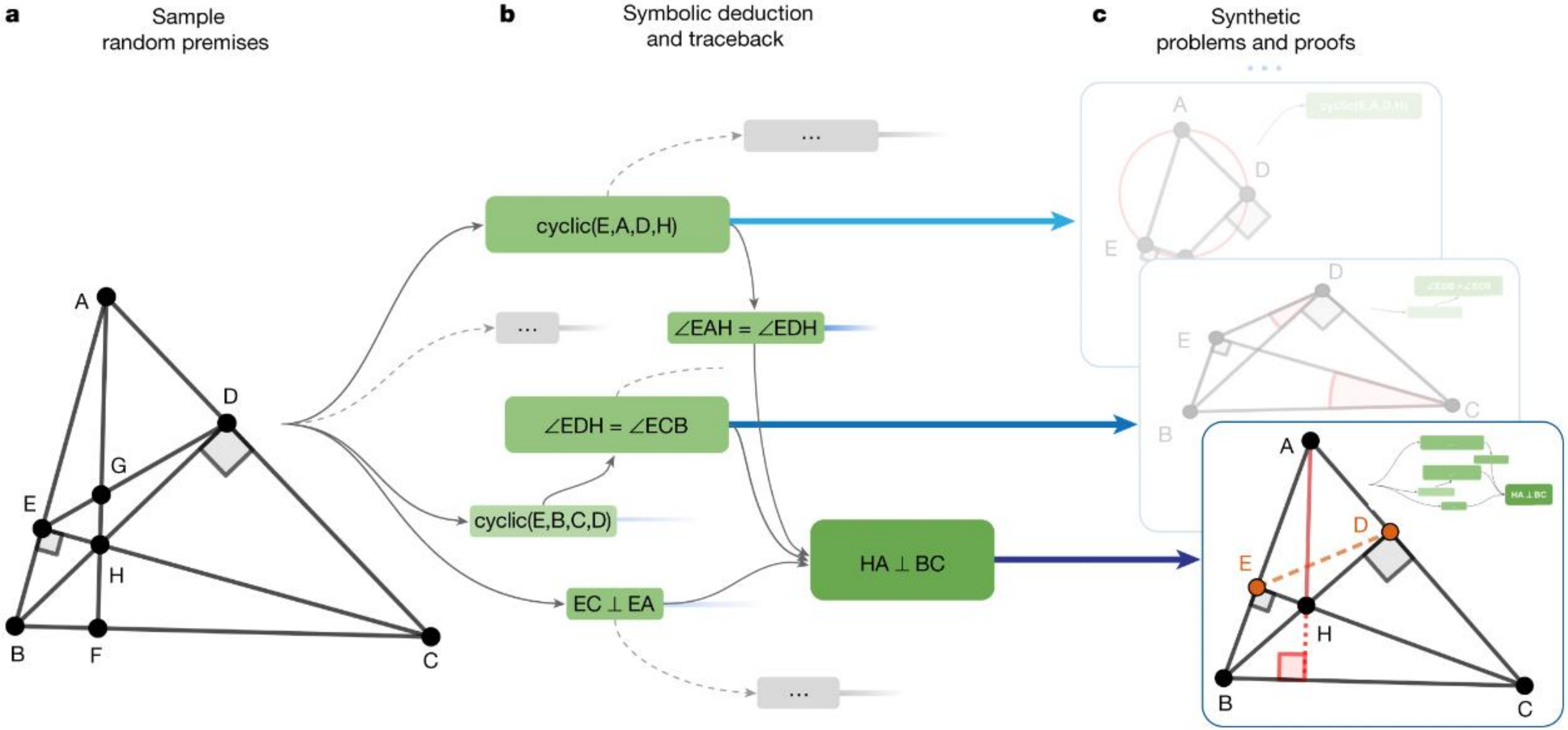
Training data

- 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)

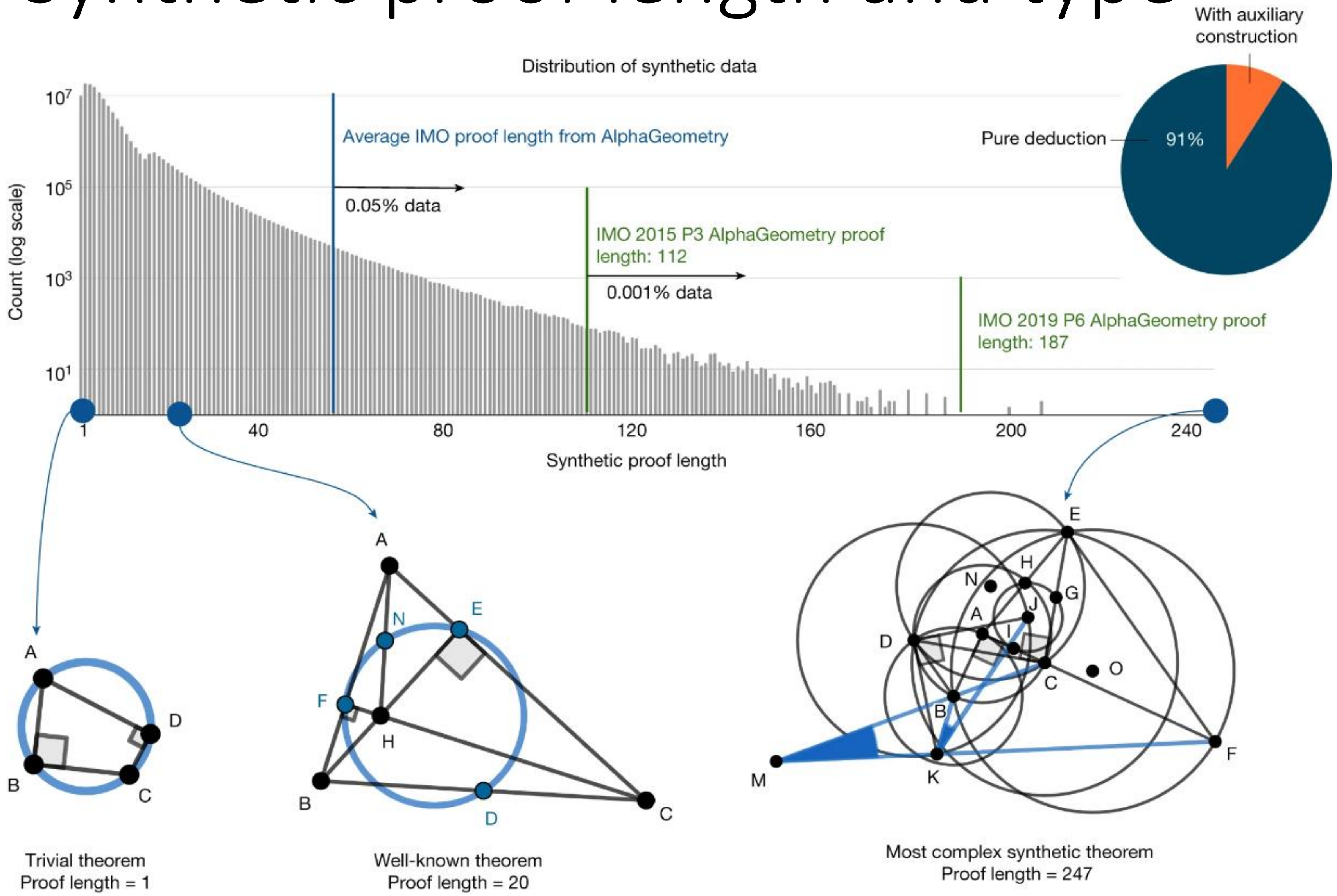
Synthetic data process



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



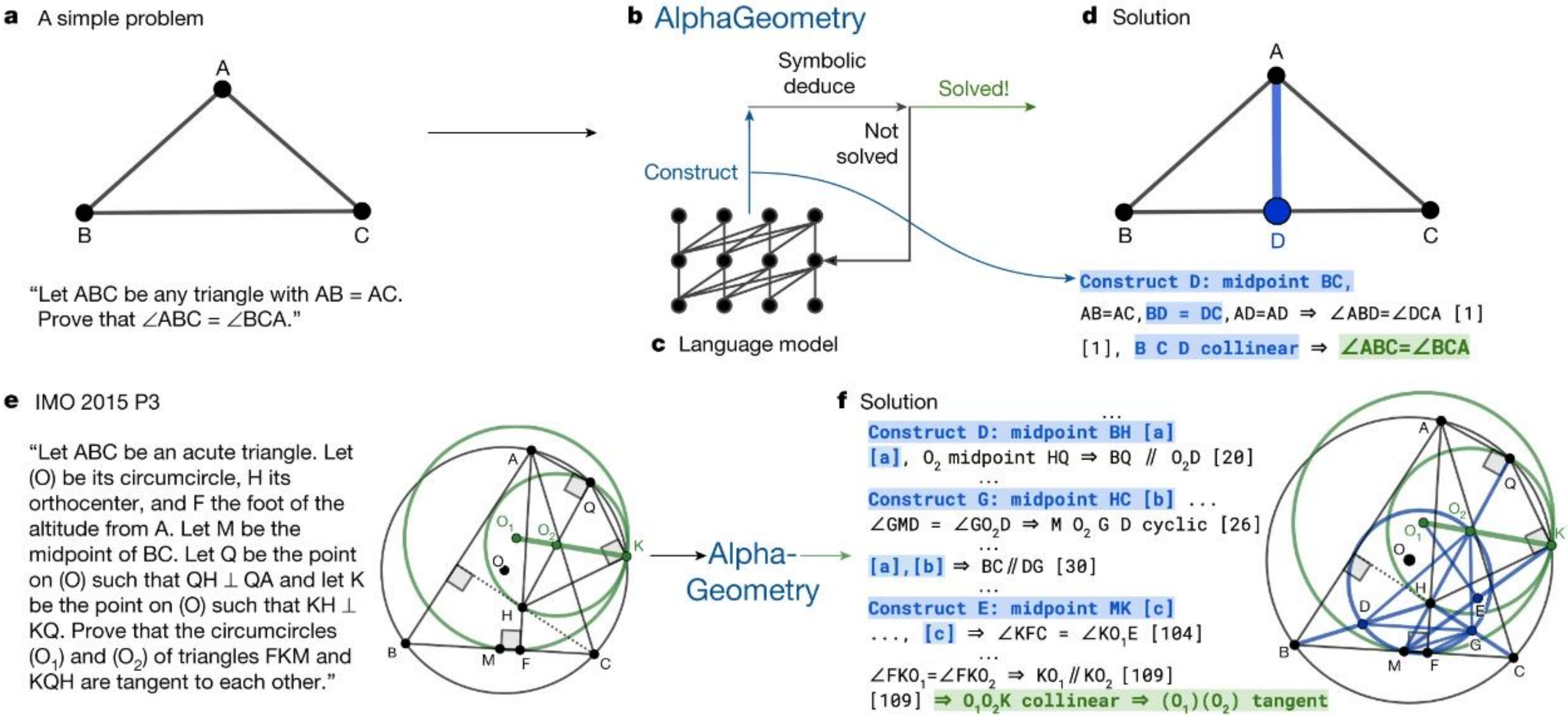
LLM and Training

- 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



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	Wu's method (previous state of the art)	10
	Gröbner basis	4
Search (human-like)	GPT-4	0
	Full-angle method	2
	Deductive database (DD)	7
	DD + human-designed heuristics	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
	• 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