Harvard CS 285: Reinforcement Learning — April 2026

Reinforcement Learning for the Game of Formal Mathematics

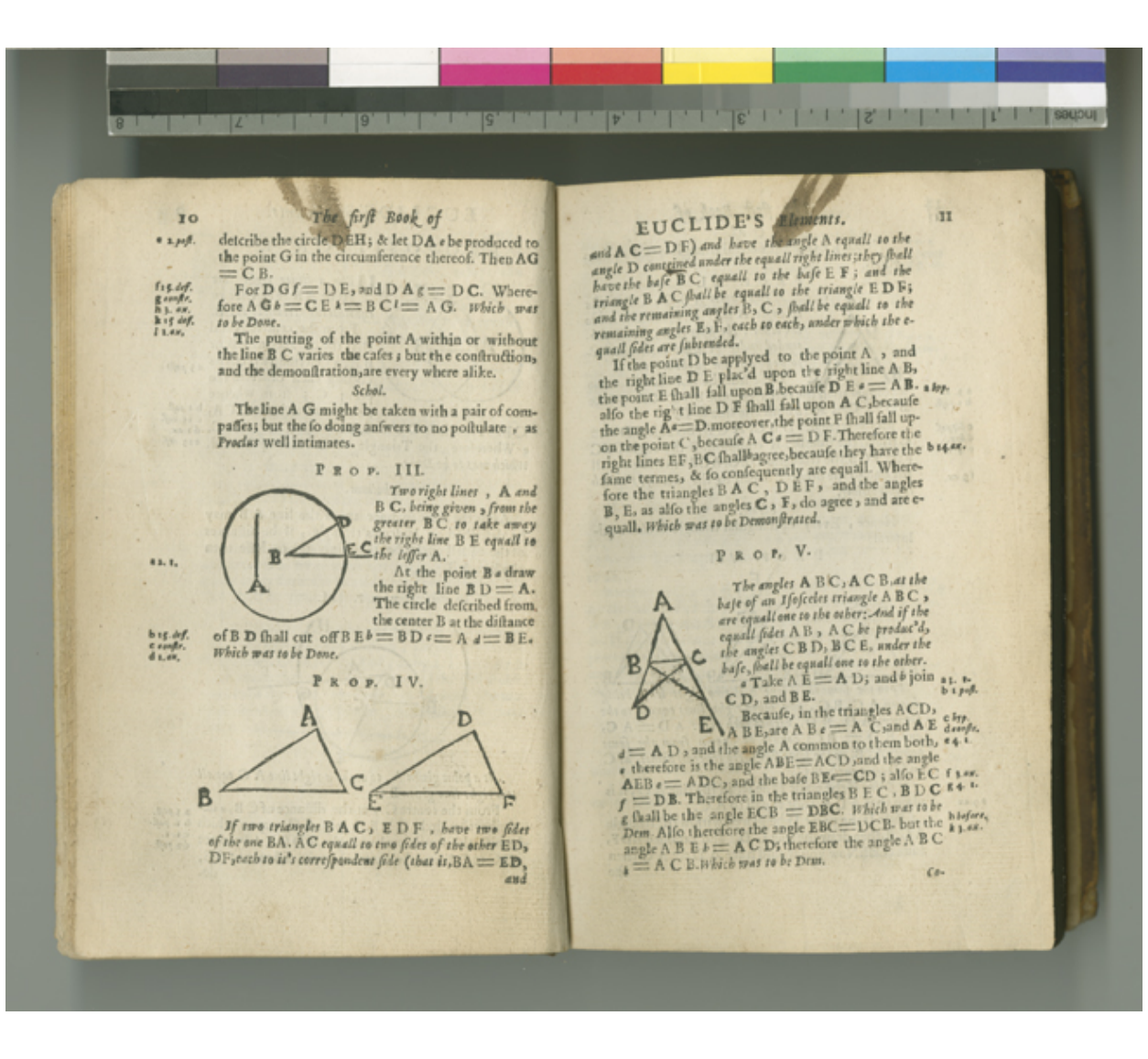
Gabriel Poesia
Research Fellow
Kempner Institute, Harvard University
gabriel_poesia@fas.harvard.edu
https://gpoesia.com

Plan for today

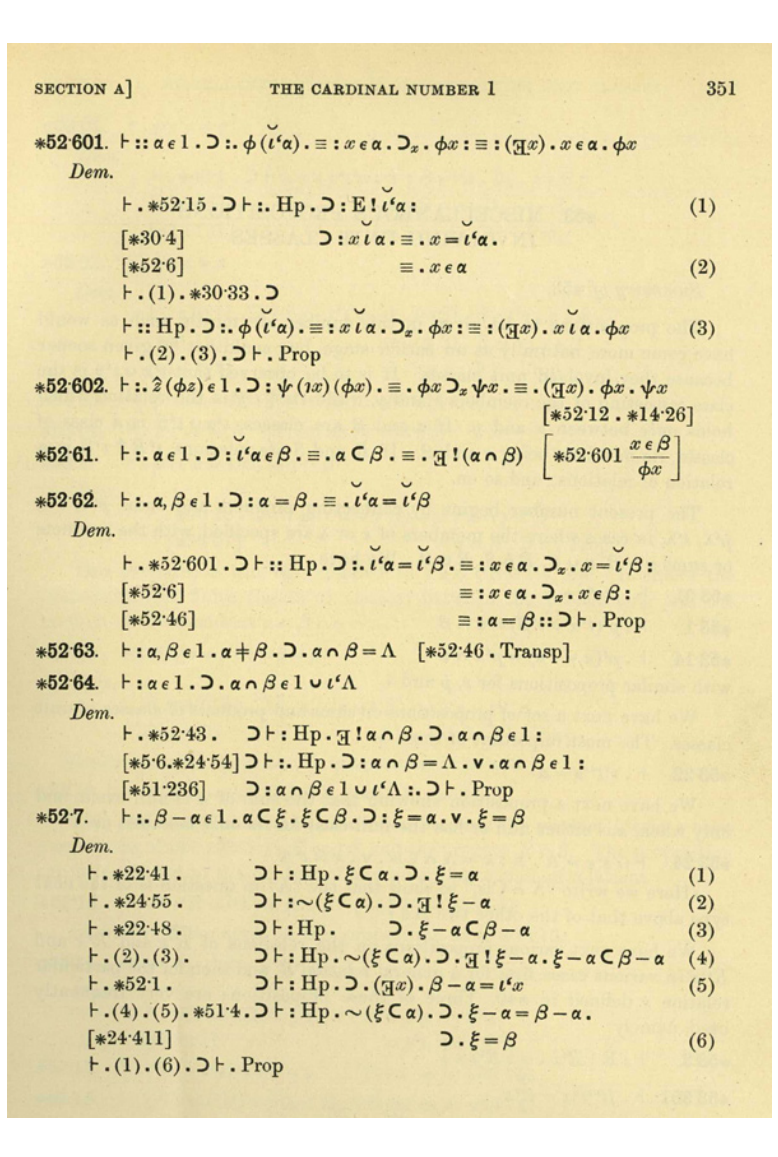
∀x ∃y ∀z P(x,y,z)
A
B
C
3
+x P((Q,y'R:
∃S. P(Q,R,S))
theorem my_theorem
(P Q : Prop), (P ∧ Q) → (P
begin
intros ⟨P h,
exact ⟨ hp hq,
⟨hp hq⟩,
end
∃x P(x)
TODO: Prove Lemma
LEAN

WHAT IS FORMAL MATHEMATICS?

Mathematics is (mostly) informal



The foundational crisis

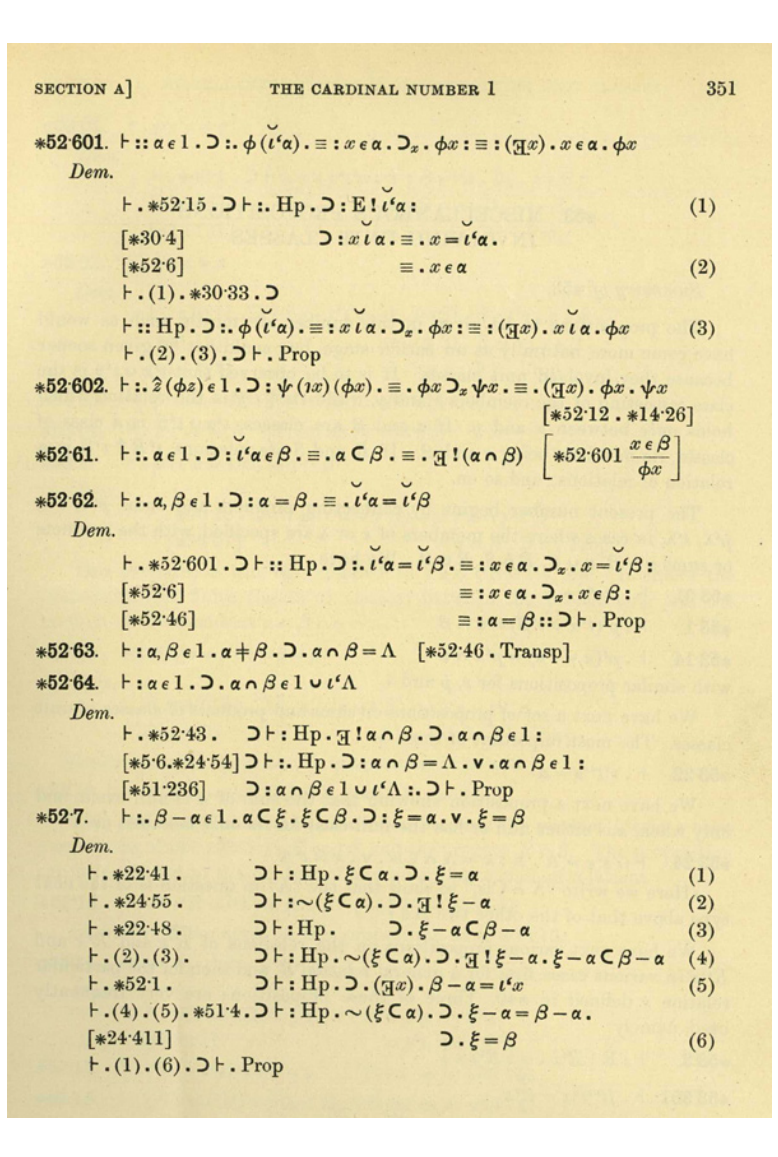


SECTION A] THE CARDINAL NUMBER 1 351

*52·601. ⊢ :: α ∈ 1 . ⊃ :. φ(ῖ‘α) . ≡ : x ∈ α . ⊃ₓ . φx : ≡ : (∃x) . x ∈ α . φx
Dem.
⊢ . *52·15 . ⊃ ⊢ :. Hp . ⊃ : E ! ῖ‘α : (1)
[*30·4] ⊃ : x ι̇ α . ≡ . x = ῖ‘α .
[*52·6] ≡ . x ∈ α (2)
⊢ . (1) . *30·33 . ⊃
⊢ :: Hp . ⊃ :. φ(ῖ‘α) . ≡ : x ι̇ α . ⊃ₓ . φx : ≡ : (∃x) . x ι̇ α . φx (3)
⊢ . (2) . (3) . ⊃ ⊢ . Prop
*52·602. ⊢ :. ẑ(φz) ∈ 1 . ⊃ : ψ(ıx)(φx) . ≡ . φx ⊃ₓ ψx . ≡ . (∃x) . φx . ψx
[*52·12 . *14·26]
*52·61. ⊢ :. α ∈ 1 . ⊃ : ῖ‘α ∈ β . ≡ . α ⊂ β . ≡ . ∃ ! (α ∩ β) $\left[*52\cdot601 \frac{x \in \beta}{\phi x}\right]$
*52·62. ⊢ :. α, β ∈ 1 . ⊃ : α = β . ≡ . ῖ‘α = ῖ‘β
Dem.
⊢ . *52·601 . ⊃ ⊢ :: Hp . ⊃ :. ῖ‘α = ῖ‘β . ≡ : x ∈ α . ⊃ₓ . x = ῖ‘β :
[*52·6] ≡ : x ∈ α . ⊃ₓ . x ∈ β :
[*52·46] ≡ : α = β :: ⊃ ⊢ . Prop
*52·63. ⊢ : α, β ∈ 1 . α ≠ β . ⊃ . α ∩ β = Λ [*52·46 . Transp]
*52·64. ⊢ : α ∈ 1 . ⊃ . α ∩ β ∈ 1 ∪ ι‘Λ
Dem.
⊢ . *52·43 . ⊃ ⊢ : Hp . ∃ ! α ∩ β . ⊃ . α ∩ β ∈ 1 :
[*5·6 . *24·54] ⊃ ⊢ :. Hp . ⊃ : α ∩ β = Λ . v . α ∩ β ∈ 1 :
[*51·236] ⊃ : α ∩ β ∈ 1 ∪ ι‘Λ :. ⊃ ⊢ . Prop
*52·7. ⊢ :. β − α ∈ 1 . α ⊂ ξ . ξ ⊂ β . ⊃ : ξ = α . v . ξ = β
Dem.
⊢ . *22·41 . ⊃ ⊢ : Hp . ξ ⊂ α . ⊃ . ξ = α (1)
⊢ . *24·55 . ⊃ ⊢ : ∼(ξ ⊂ α) . ⊃ . ∃ ! ξ − α (2)
⊢ . *22·48 . ⊃ ⊢ : Hp . ⊃ . ξ − α ⊂ β − α (3)
⊢ . (2) . (3) . ⊃ ⊢ : Hp . ∼(ξ ⊂ α) . ⊃ . ∃ ! ξ − α . ξ − α ⊂ β − α (4)
⊢ . *52·1 . ⊃ ⊢ : Hp . ⊃ . (∃x) . β − α = ι‘x (5)
⊢ . (4) . (5) . *51·4 . ⊃ ⊢ : Hp . ∼(ξ ⊂ α) . ⊃ . ξ − α = β − α .
[*24·411] ⊃ . ξ = β (6)
⊢ . (1) . (6) . ⊃ ⊢ . Prop

The foundational crisis

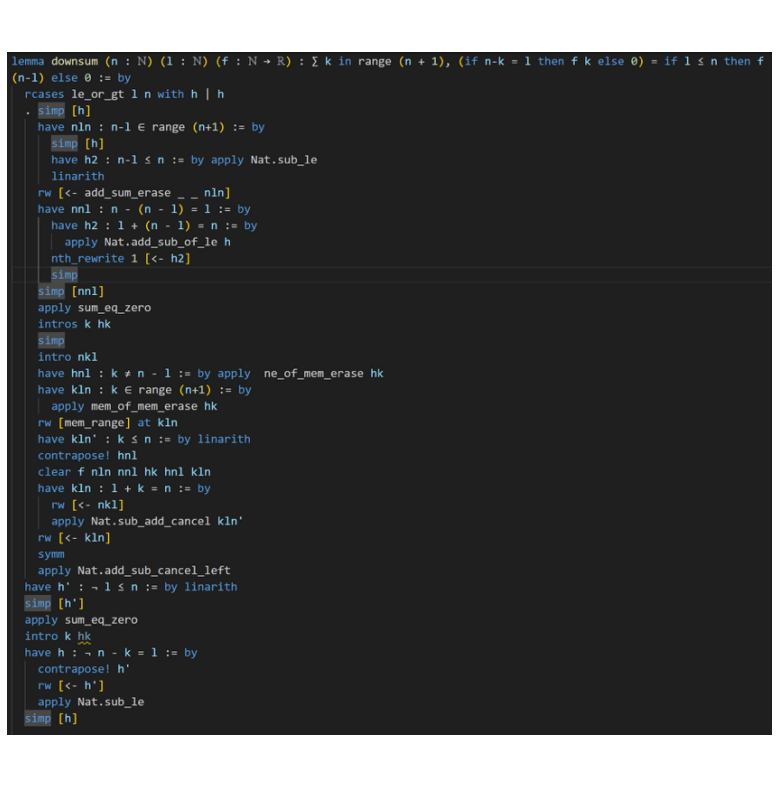
Can we do any mathematics from axioms and inference rules?



SECTION A] THE CARDINAL NUMBER 1 351

*52·601. ⊢ :: α ∈ 1 . ⊃ :. φ(ῖ‘α) . ≡ : x ∈ α . ⊃ₓ . φx : ≡ : (∃x) . x ∈ α . φx
Dem.
⊢ . *52·15 . ⊃ ⊢ :. Hp . ⊃ : E ! ῖ‘α : (1)
[*30·4] ⊃ : x ι̇ α . ≡ . x = ῖ‘α .
[*52·6] ≡ . x ∈ α (2)
⊢ . (1) . *30·33 . ⊃
⊢ :: Hp . ⊃ :. φ(ῖ‘α) . ≡ : x ι̇ α . ⊃ₓ . φx : ≡ : (∃x) . x ι̇ α . φx (3)
⊢ . (2) . (3) . ⊃ ⊢ . Prop
*52·602. ⊢ :. ẑ(φz) ∈ 1 . ⊃ : ψ(ıx)(φx) . ≡ . φx ⊃ₓ ψx . ≡ . (∃x) . φx . ψx
[*52·12 . *14·26]
*52·61. ⊢ :. α ∈ 1 . ⊃ : ῖ‘α ∈ β . ≡ . α ⊂ β . ≡ . ∃ ! (α ∩ β) $\left[*52\cdot601 \frac{x \in \beta}{\phi x}\right]$
*52·62. ⊢ :. α, β ∈ 1 . ⊃ : α = β . ≡ . ῖ‘α = ῖ‘β
Dem.
⊢ . *52·601 . ⊃ ⊢ :: Hp . ⊃ :. ῖ‘α = ῖ‘β . ≡ : x ∈ α . ⊃ₓ . x = ῖ‘β :
[*52·6] ≡ : x ∈ α . ⊃ₓ . x ∈ β :
[*52·46] ≡ : α = β :: ⊃ ⊢ . Prop
*52·63. ⊢ : α, β ∈ 1 . α ≠ β . ⊃ . α ∩ β = Λ [*52·46 . Transp]
*52·64. ⊢ : α ∈ 1 . ⊃ . α ∩ β ∈ 1 ∪ ι‘Λ
Dem.
⊢ . *52·43 . ⊃ ⊢ : Hp . ∃ ! α ∩ β . ⊃ . α ∩ β ∈ 1 :
[*5·6 . *24·54] ⊃ ⊢ :. Hp . ⊃ : α ∩ β = Λ . v . α ∩ β ∈ 1 :
[*51·236] ⊃ : α ∩ β ∈ 1 ∪ ι‘Λ :. ⊃ ⊢ . Prop
*52·7. ⊢ :. β − α ∈ 1 . α ⊂ ξ . ξ ⊂ β . ⊃ : ξ = α . v . ξ = β
Dem.
⊢ . *22·41 . ⊃ ⊢ : Hp . ξ ⊂ α . ⊃ . ξ = α (1)
⊢ . *24·55 . ⊃ ⊢ : ∼(ξ ⊂ α) . ⊃ . ∃ ! ξ − α (2)
⊢ . *22·48 . ⊃ ⊢ : Hp . ⊃ . ξ − α ⊂ β − α (3)
⊢ . (2) . (3) . ⊃ ⊢ : Hp . ∼(ξ ⊂ α) . ⊃ . ∃ ! ξ − α . ξ − α ⊂ β − α (4)
⊢ . *52·1 . ⊃ ⊢ : Hp . ⊃ . (∃x) . β − α = ι‘x (5)
⊢ . (4) . (5) . *51·4 . ⊃ ⊢ : Hp . ∼(ξ ⊂ α) . ⊃ . ξ − α = β − α .
[*24·411] ⊃ . ξ = β (6)
⊢ . (1) . (6) . ⊃ ⊢ . Prop

C is getting smaller



```
lemma downsum (n : ℕ) (l : ℕ) (f : ℕ → ℝ) : ∑ k in range (n + 1), (if n-k = l then f k else 0) = if l ≤ n then f
(n-l) else 0 := by
  rcases le_or_gt l n with h | h
  · simp [h]
    have nln : n-l ∈ range (n+1) := by
      simp [h]
      have h2 : n-l ≤ n := by apply Nat.sub_le
      linarith
    rw [<- add_sum_erase _ _ nln]
    have nnl : n - (n - l) = l := by
      have h2 : l + (n - l) = n := by
        apply Nat.add_sub_of_le h
      nth_rewrite 1 [<- h2]
      simp
    simp [nnl]
    apply sum_eq_zero
    intros k hk
    simp
    intro nkl
    have hnl : k ≠ n - l := by apply  ne_of_mem_erase hk
    have kln : k ∈ range (n+1) := by
      apply mem_of_mem_erase hk
    rw [mem_range] at kln
    have kln' : k ≤ n := by linarith
    contrapose! hnl
    clear f nln nnl hk hnl kln
    have kln : l + k = n := by
      rw [<- nkl]
      apply Nat.sub_add_cancel kln'
    rw [<- kln]
    symm
    apply Nat.add_sub_cancel_left
  have h' : ¬ l ≤ n := by linarith
  simp [h']
  apply sum_eq_zero
  intro k hk
  have h : ¬ n - k = l := by
    contrapose! h'
    rw [<- h']
    apply Nat.sub_le
  simp [h]
```

LEAN

Lean

Example: Natural numbers

A first theorem: `add_zero`

Let's prove step-by-step

```lean
theorem zero_add (n : Nat) : add zero n = n := by
```

Proving `zero_add`, continued

Goal 2 (inductive step):

```lean
n' : Nat
ih : add zero n' = n'
⊢ add zero (succ n') = succ n'
```

Building up: more lemmas

Tactics: the vocabulary

The game of formal theorem proving

The MDP: State and Action

The MDP: Transition and Reward

The game

Chess / Go	Theorem proving

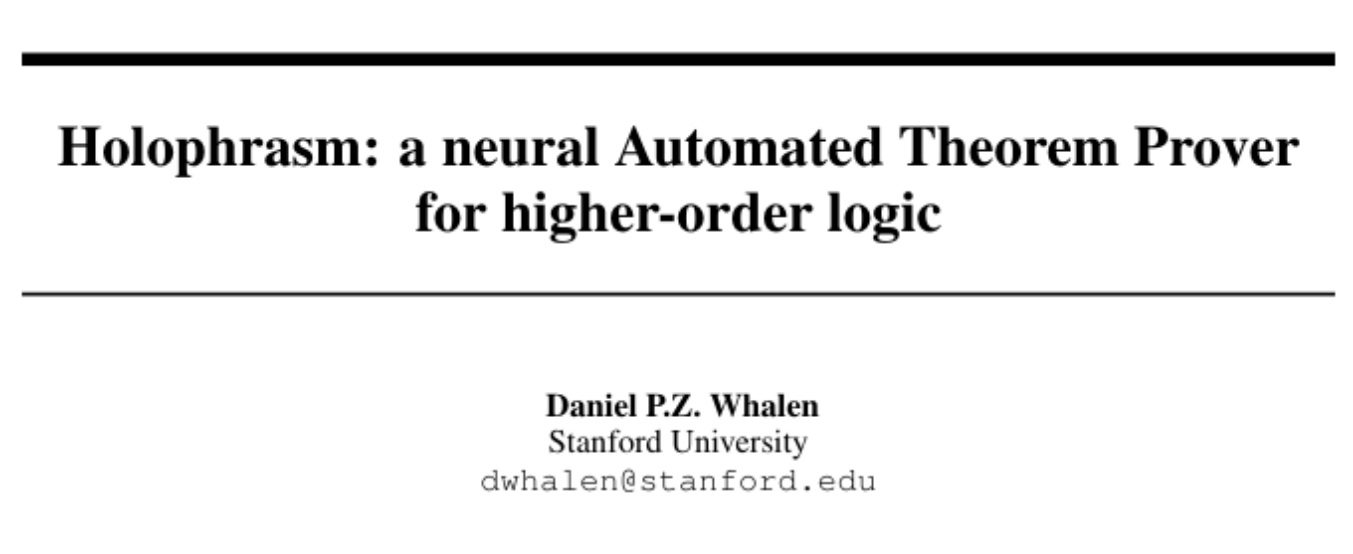
Key challenges

Two initial learning signals

The general recipe (preview)

Part I: The classical era (2016–2020)

The "modern classical" era



Holophrasm: a neural Automated Theorem Prover for higher-order logic

Daniel P.Z. Whalen
Stanford University
dwhalen@stanford.edu

Holophrasm: search and results

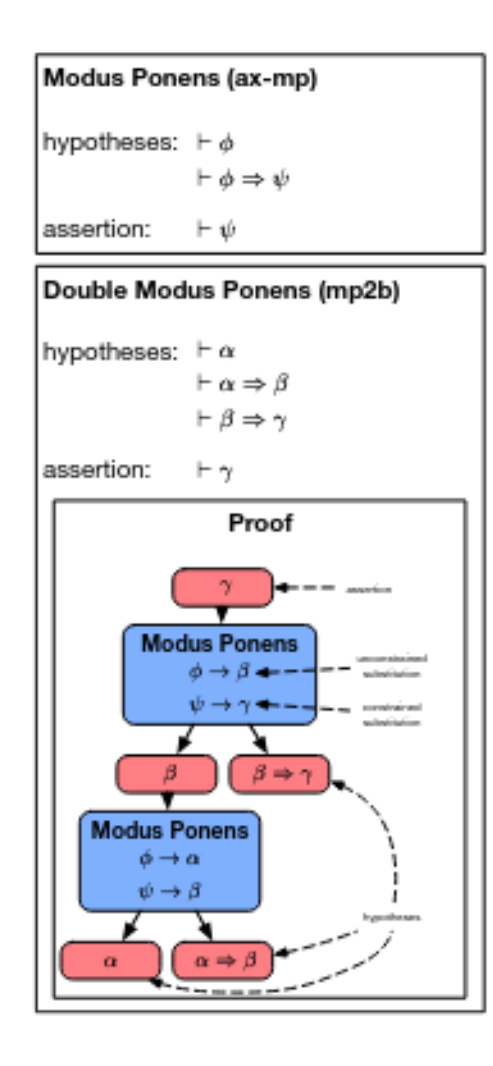
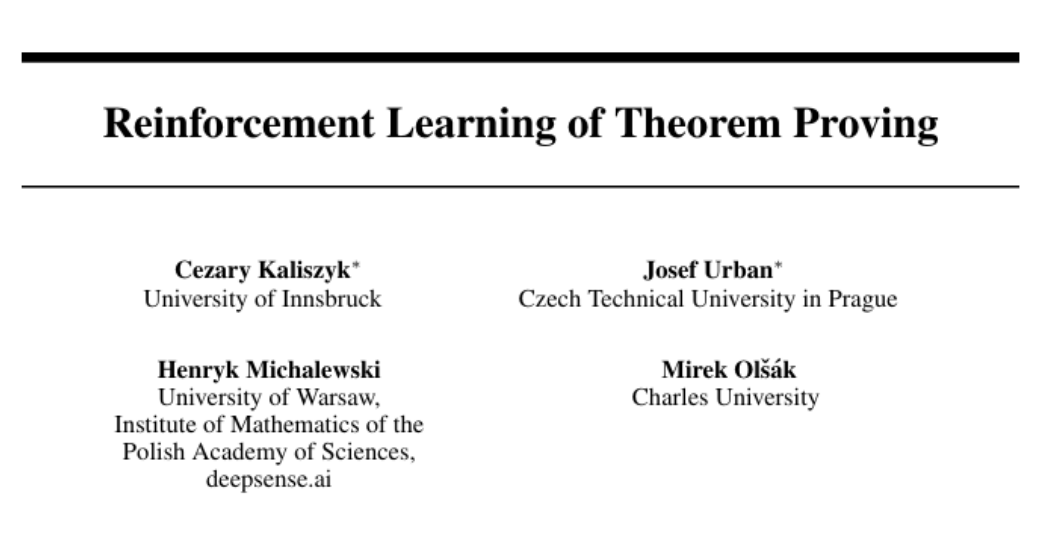

Modus Ponens (ax-mp)
hypotheses: $\vdash \phi$
$\vdash \phi \Rightarrow \psi$
assertion: $\vdash \psi$
Double Modus Ponens (mp2b)
hypotheses: $\vdash \alpha$
$\vdash \alpha \Rightarrow \beta$
$\vdash \beta \Rightarrow \gamma$
assertion: $\vdash \gamma$
Proof
γ
Modus Ponens
$\phi \to \beta$
$\psi \to \gamma$
β
$\beta \Rightarrow \gamma$
Modus Ponens
$\phi \to \alpha$
$\psi \to \beta$
α
$\alpha \Rightarrow \beta$

Figure 1: Examples of an axiom, **ax-mp** and a proposition, **mp2b**. The red nodes are expressions, and the blue nodes each describe a theorem and substitutions that will prove the parent.



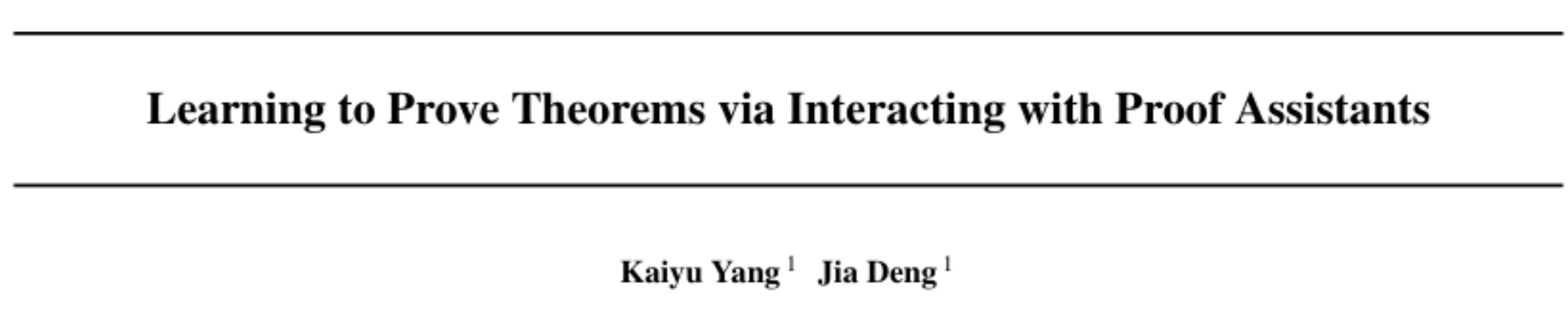
Reinforcement Learning of Theorem Proving

Cezary Kaliszyk*
University of Innsbruck

Josef Urban*
Czech Technical University in Prague

Henryk Michalewski
University of Warsaw,
Institute of Mathematics of the
Polish Academy of Sciences,
deepsense.ai

Mirek Olšák
Charles University

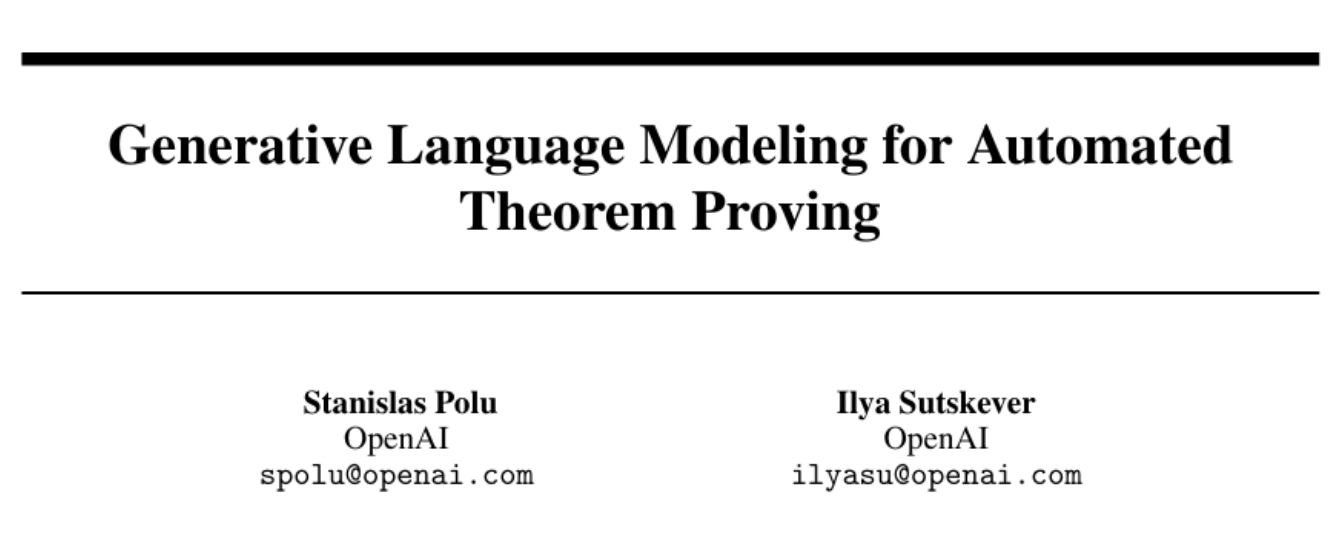


Learning to Prove Theorems via Interacting with Proof Assistants

Kaiyu Yang[1] **Jia Deng**[1]

Learning to Prove Theorems via Interacting with Proof Assistants

Kaiyu Yang[1] **Jia Deng**[1]



Generative Language Modeling for Automated Theorem Proving

Stanislas Polu
OpenAI
spolu@openai.com

Ilya Sutskever
OpenAI
ilyasu@openai.com

Generative Language Modeling for Automated Theorem Proving

Stanislas Polu
OpenAI
`spolu@openai.com`

Ilya Sutskever
OpenAI
`ilyasu@openai.com`

Generative Language Modeling for Automated Theorem Proving

Stanislas Polu
OpenAI
`spolu@openai.com`

Ilya Sutskever
OpenAI
`ilyasu@openai.com`

Generative Language Modeling for Automated Theorem Proving

Stanislas Polu
OpenAI
`spolu@openai.com`

Ilya Sutskever
OpenAI
`ilyasu@openai.com`

Summary of Part I

	Policy	Value	Search	Training signal

PART III: THE ROAD TO IMO

miniF2F: a shared benchmark

Why tactics make this much difference

Formal Mathematics Statement Curriculum Learning

Stanislas Polu[1] Jesse Michael Han[1] Kunhao Zheng[2] Mantas Baksys[3] Igor Babuschkin[1] Ilya Sutskever[1]

Strategy 1: synthetic data

Strategy 2: manual formalization

Expert iteration on all statements

Combine everything: mathlib + synthetic inequalities + manually formalized statements. Run expert iteration:

Results on miniF2F

HyperTree Proof Search for Neural Theorem Proving

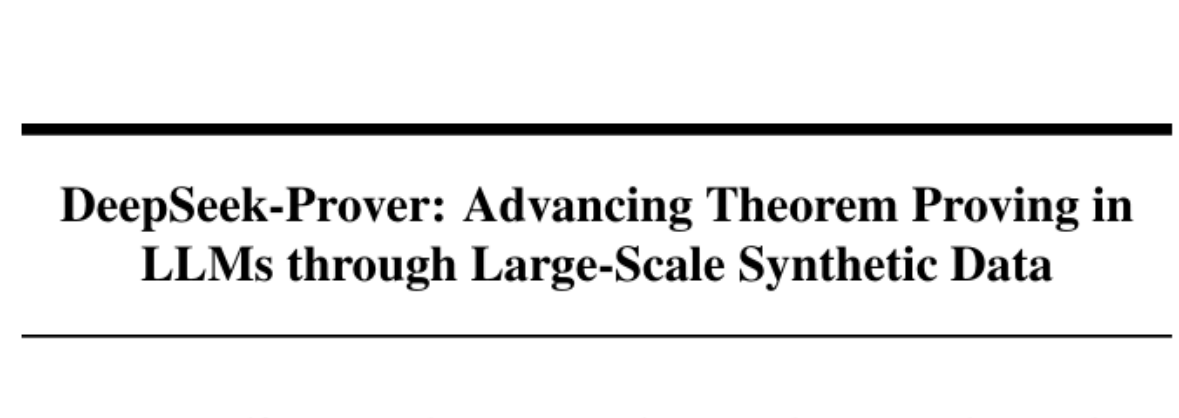
Guillaume Lample*† Marie-Anne Lachaux*† Thibaut Lavril*† Xavier Martinet*†
Amaury Hayat§ Gabriel Ebner‡ Aurélien Rodriguez† Timothée Lacroix*†

HTPS: results

Also introduces **online training**: trainer and prover run concurrently, model weights updated continuously (vs. discrete expert iteration rounds).

BREAKING THE DATA WALL

The data wall



DeepSeek-Prover: Advancing Theorem Proving in LLMs through Large-Scale Synthetic Data

Huajian Xin[1,2] **Daya Guo**[1] **Zhihong Shao**[1] **Z.Z. Ren**[1] **Qihao Zhu**[1] **Bo Liu**[1]
Chong Ruan[1] **Wenda Li**[3] **Xiaodan Liang**[2,4*]
[1]DeepSeek [2]Sun Yat-sen University [3]University of Edinburgh [4]MBZUAI
{xinhj, guoday, zhihongshao, rzz, zhuqh, chong.ruan}@deepseek.com,
benjaminliu.eecs@gmail.com, wli8@ed.ac.uk, xdliang328@gmail.com

Step 1: Autoformalization

Use an LLM to translate natural-language problems into Lean 4 statements:

Hypothesis rejection

DeepSeek-Prover V1: results

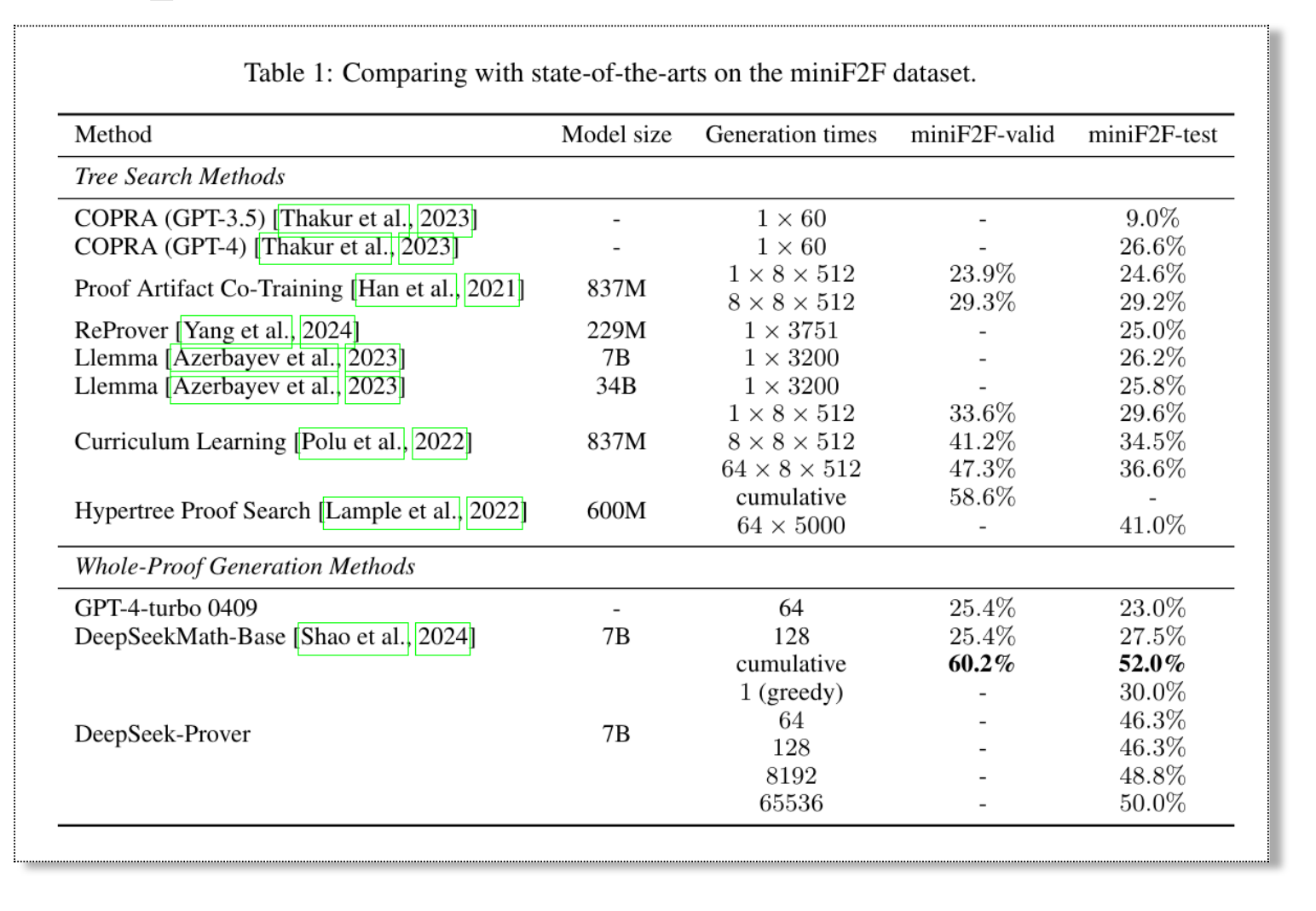
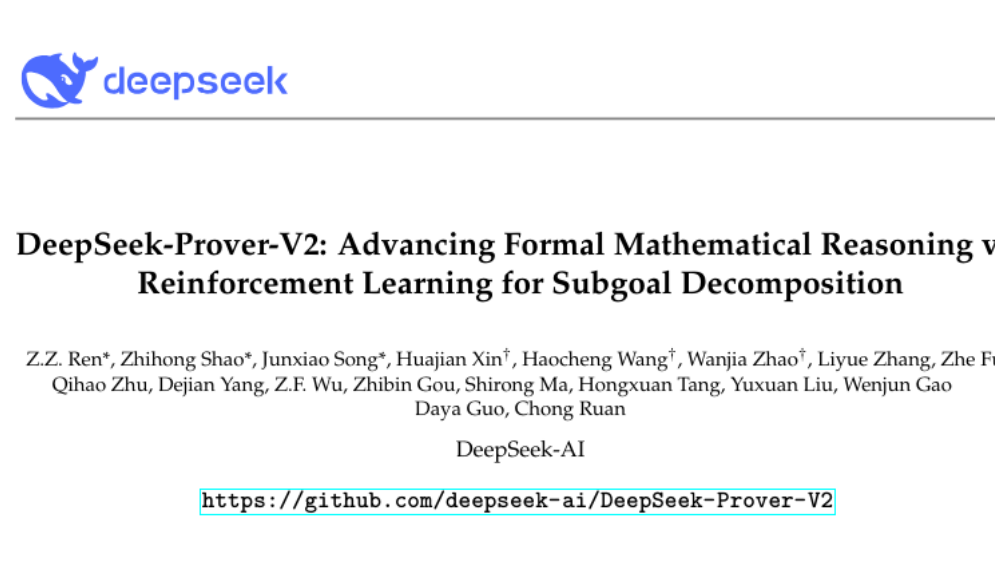


Table 1: Comparing with state-of-the-arts on the miniF2F dataset.

Method	Model size	Generation times	miniF2F-valid	miniF2F-test
Tree Search Methods				
COPRA (GPT-3.5) [Thakur et al., 2023]	-	1×60	-	9.0%
COPRA (GPT-4) [Thakur et al., 2023]	-	1×60	-	26.6%
Proof Artifact Co-Training [Han et al., 2021]	837M	$1 \times 8 \times 512$ $8 \times 8 \times 512$	23.9% 29.3%	24.6% 29.2%
ReProver [Yang et al., 2024]	229M	1×3751	-	25.0%
Llemma [Azerbayev et al., 2023]	7B	1×3200	-	26.2%
Llemma [Azerbayev et al., 2023]	34B	1×3200	-	25.8%
Curriculum Learning [Polu et al., 2022]	837M	$1 \times 8 \times 512$ $8 \times 8 \times 512$ $64 \times 8 \times 512$	33.6% 41.2% 47.3%	29.6% 34.5% 36.6%
Hypertree Proof Search [Lample et al., 2022]	600M	cumulative 64×5000	58.6% -	- 41.0%
Whole-Proof Generation Methods				
GPT-4-turbo 0409	-	64	25.4%	23.0%
DeepSeekMath-Base [Shao et al., 2024]	7B	128	25.4%	27.5%
DeepSeek-Prover	7B	cumulative 1 (greedy) 64 128 8192 65536	**60.2%** - - - - -	**52.0%** 30.0% 46.3% 46.3% 48.8% 50.0%



DeepSeek-Prover-V2: Advancing Formal Mathematical Reasoning via Reinforcement Learning for Subgoal Decomposition

Z.Z. Ren*, Zhihong Shao*, Junxiao Song*, Huajian Xin†, Haocheng Wang†, Wanjia Zhao†, Liyue Zhang, Zhe Fu
Qihao Zhu, Dejian Yang, Z.F. Wu, Zhibin Gou, Shirong Ma, Hongxuan Tang, Yuxuan Liu, Wenjun Gao
Daya Guo, Chong Ruan

DeepSeek-AI

`https://github.com/deepseek-ai/DeepSeek-Prover-V2`

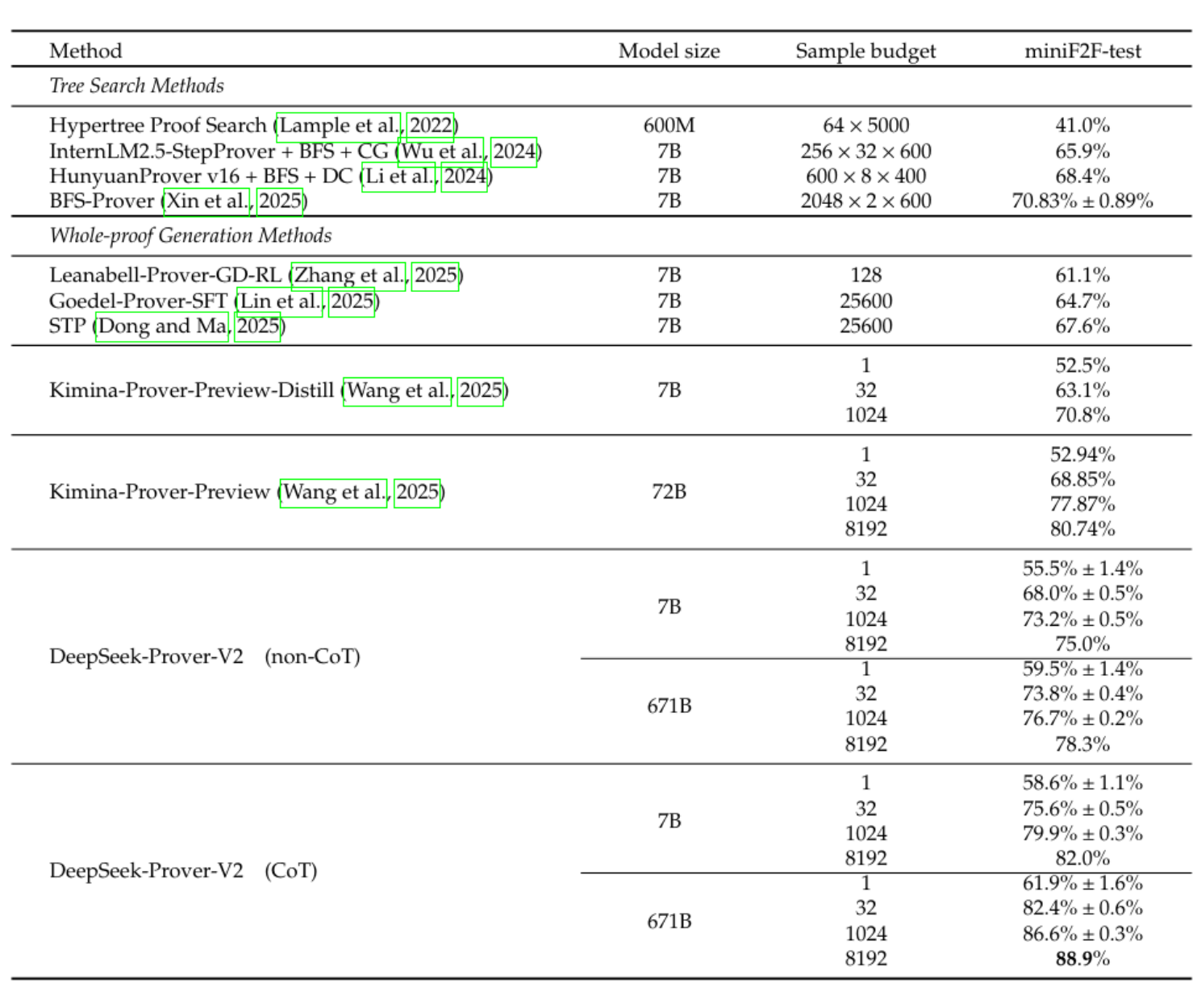
Subgoal decomposition

The reasoning LLM produces a Lean skeleton with `sorry` placeholders. A specialized 7B prover fills each one:

Subgoal decomposition → much longer proofs

IMO 1992 Problem 1 — a ~280-line proof:

DeepSeek-Prover V2: results



Method	Model size	Sample budget	miniF2F-test
Tree Search Methods			
Hypertree Proof Search (Lample et al., 2022)	600M	64×5000	41.0%
InternLM2.5-StepProver + BFS + CG (Wu et al., 2024)	7B	$256 \times 32 \times 600$	65.9%
HunyuanProver v16 + BFS + DC (Li et al., 2024)	7B	$600 \times 8 \times 400$	68.4%
BFS-Prover (Xin et al., 2025)	7B	$2048 \times 2 \times 600$	70.83% ± 0.89%
Whole-proof Generation Methods			
Leanabell-Prover-GD-RL (Zhang et al., 2025)	7B	128	61.1%
Goedel-Prover-SFT (Lin et al., 2025)	7B	25600	64.7%
STP (Dong and Ma, 2025)	7B	25600	67.6%
Kimina-Prover-Preview-Distill (Wang et al., 2025)	7B	1 32 1024	52.5% 63.1% 70.8%
Kimina-Prover-Preview (Wang et al., 2025)	72B	1 32 1024 8192	52.94% 68.85% 77.87% 80.74%
DeepSeek-Prover-V2 (non-CoT)	7B	1 32 1024 8192	55.5% ± 1.4% 68.0% ± 0.5% 73.2% ± 0.5% 75.0%
	671B	1 32 1024 8192	59.5% ± 1.4% 73.8% ± 0.4% 76.7% ± 0.2% 78.3%
DeepSeek-Prover-V2 (CoT)	7B	1 32 1024 8192	58.6% ± 1.1% 75.6% ± 0.5% 79.9% ± 0.3% 82.0%
	671B	1 32 1024 8192	61.9% ± 1.6% 82.4% ± 0.6% 86.6% ± 0.3% **88.9**%

Article

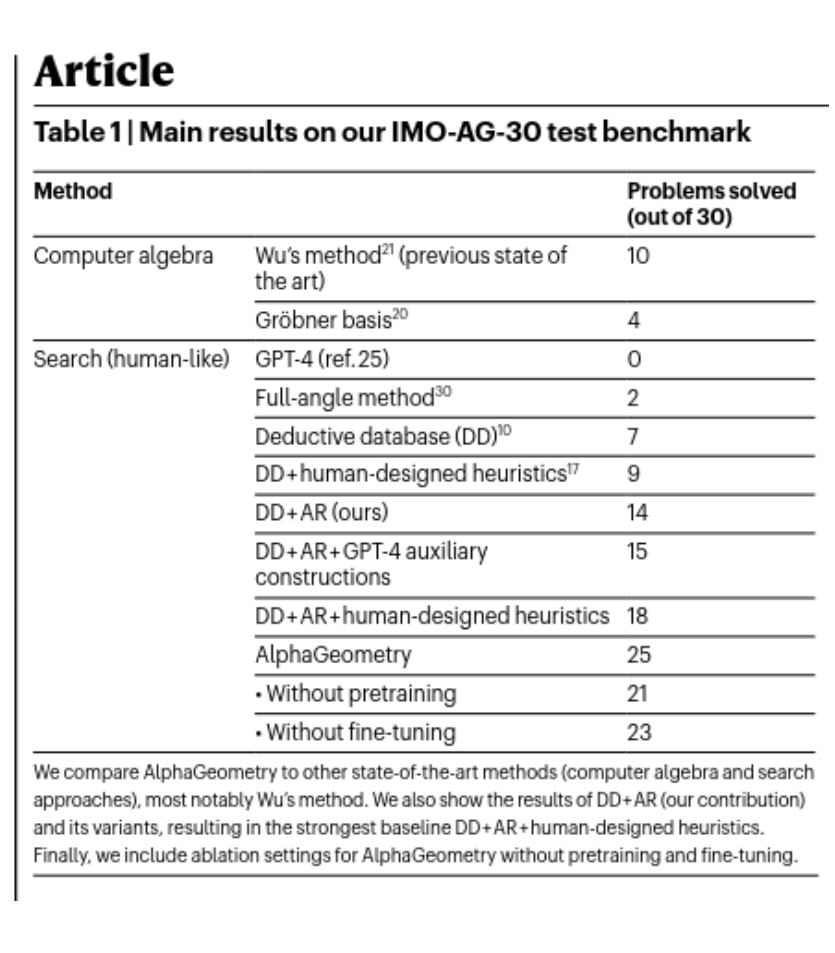
Solving olympiad geometry without human demonstrations

https://doi.org/10.1038/s41586-023-06747-5 **Trieu H. Trinh**[1,2✉], **Yuhuai Wu**[1], **Quoc V. Le**[1], **He He**[2] **& Thang Luong**[1✉]

Synthetic data generation for geometry

Synthetic auxiliary constructions

AlphaGeometry: results



Article

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

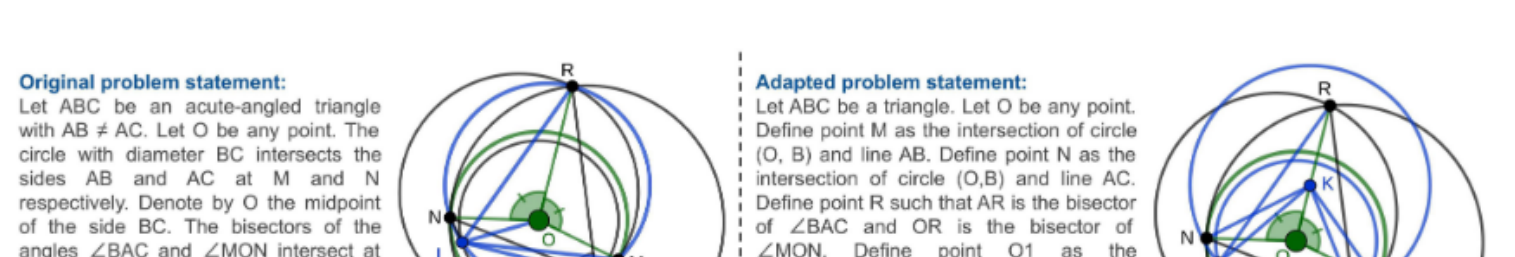
Method		Problems solved (out of 30)
Computer algebra	Wu's method[21] (previous state of the art)	10
	Gröbner basis[20]	4
Search (human-like)	GPT-4 (ref. 25)	0
	Full-angle method[30]	2
	Deductive database (DD)[10]	7
	DD+human-designed heuristics[17]	9
	DD+AR (ours)	14
	DD+AR+GPT-4 auxiliary constructions	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.

A more general proof

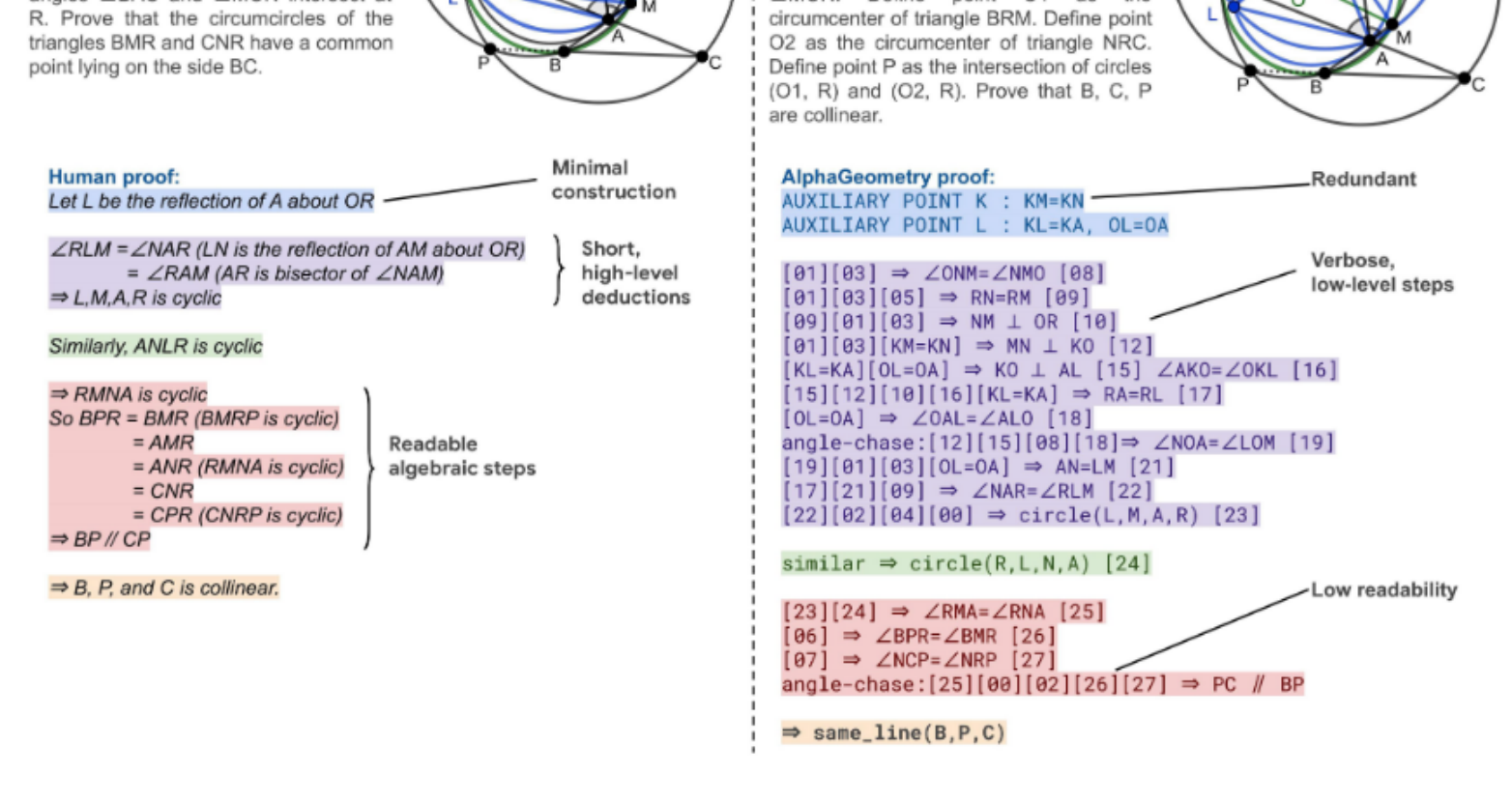
On IMO 2004 P1, AlphaGeometry found a proof that was **more general** than the original problem:

...but sometimes more verbose



Original problem statement:
Let ABC be an acute-angled triangle with AB ≠ 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 ∠BAC and ∠MON intersect at

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 ∠BAC and OR is the bisector of ∠MON. Define point O1 as the


P
B
C
A
M
L
P
B
A
M
C
Minimal construction
Short, high-level deductions
Readable algebraic steps
Redundant
Verbose, low-level steps
Low readability

R. Prove that the circumcircles of the triangles BMR and CNR have a common point lying on the side BC.

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

∠RLM = ∠NAR (LN is the reflection of AM about OR)
= ∠RAM (AR is bisector of ∠NAM)
⇒ L,M,A,R is cyclic

Similarly, ANLR is cyclic

⇒ RMNA is cyclic
So BPR = BMR (BMRP is cyclic)
= AMR
= ANR (RMNA is cyclic)
= CNR
= CPR (CNRP is cyclic)
⇒ BP // CP

⇒ B, P, and C is collinear.

AlphaGeometry proof:

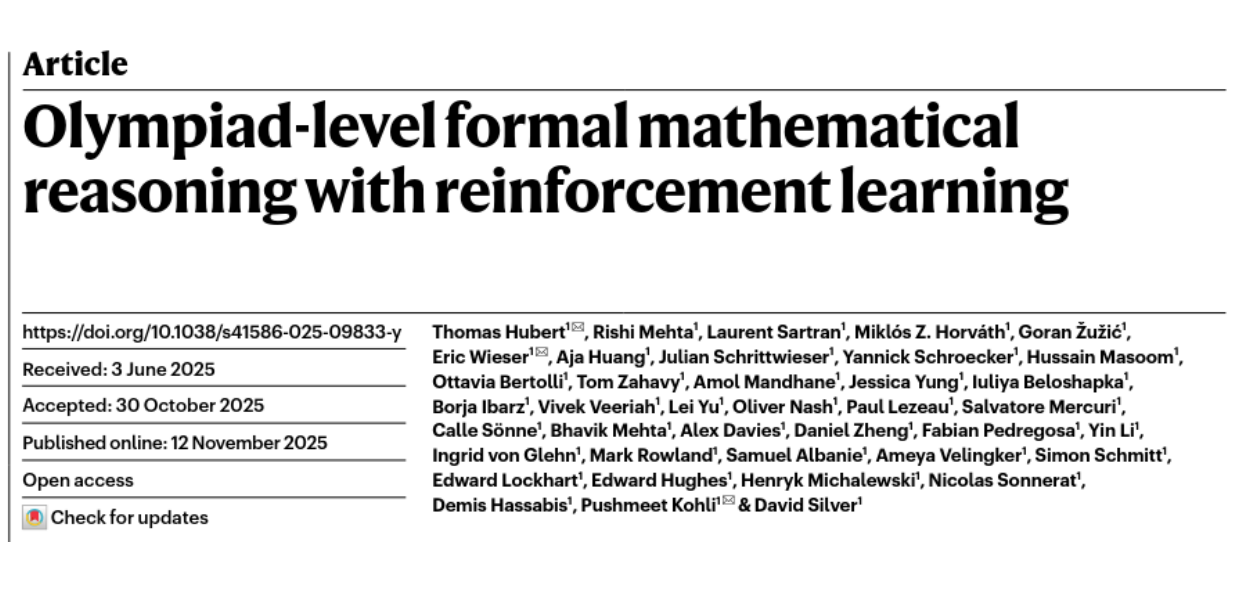
```
AUXILIARY POINT K : KM=KN
AUXILIARY POINT L : KL=KA, OL=OA

[01][03] ⇒ ∠ONM=∠NMO [08]
[01][03][05] ⇒ RN=RM [09]
[09][01][03] ⇒ NM ⊥ OR [10]
[01][03][KM=KN] ⇒ MN ⊥ KO [12]
[KL=KA][OL=OA] ⇒ KO ⊥ AL [15] ∠AKO=∠OKL [16]
[15][12][10][16][KL=KA] ⇒ RA=RL [17]
[OL=OA] ⇒ ∠OAL=∠ALO [18]
angle-chase:[12][15][08][18]⇒ ∠NOA=∠LOM [19]
[19][01][03][OL=OA] ⇒ AN=LM [21]
[17][21][09] ⇒ ∠NAR=∠RLM [22]
[22][02][04][00] ⇒ 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]
angle-chase:[25][00][02][26][27] ⇒ PC // BP

⇒ same_line(B,P,C)
```



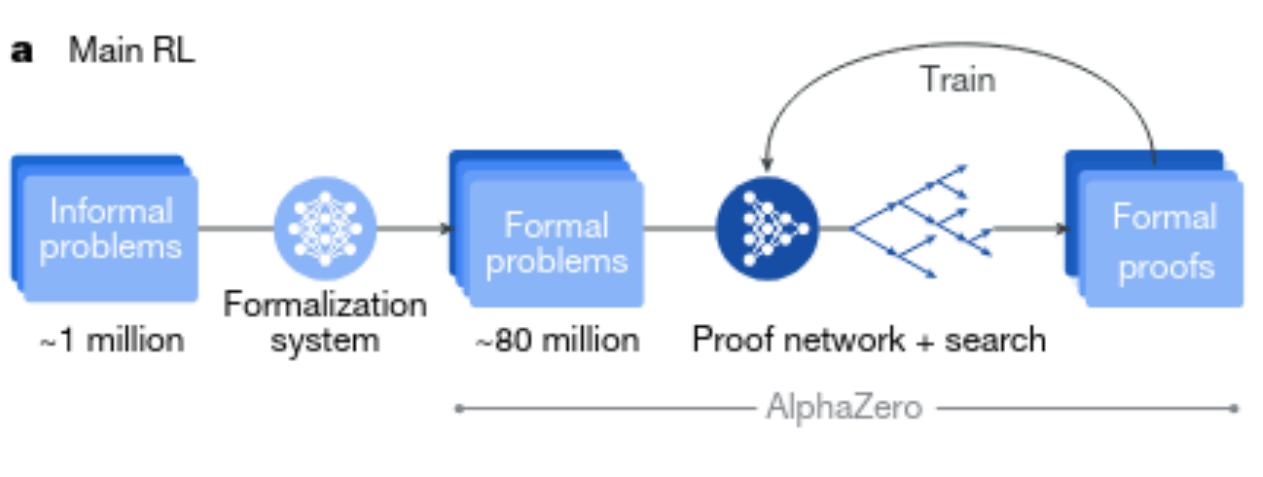
Article

Olympiad-level formal mathematical reasoning with reinforcement learning

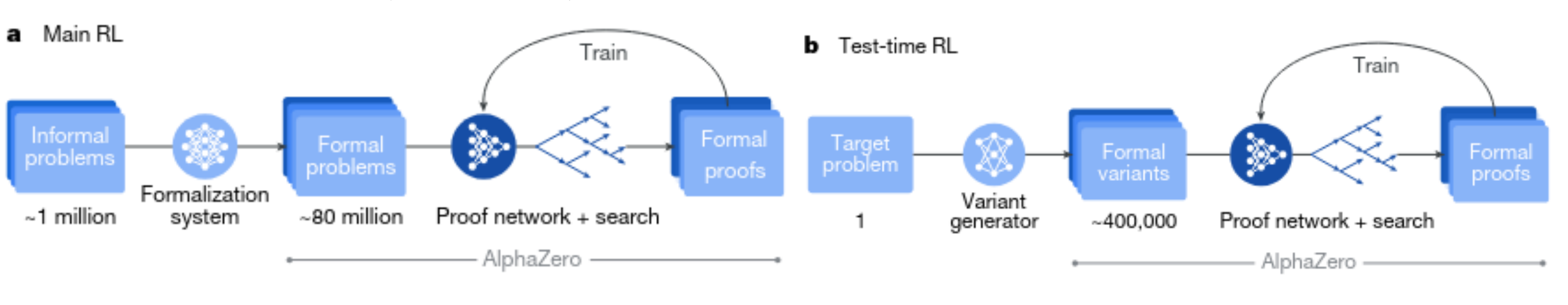
https://doi.org/10.1038/s41586-025-09833-y
Received: 3 June 2025
Accepted: 30 October 2025
Published online: 12 November 2025
Open access
Check for updates

Thomas Hubert[1]✉, Rishi Mehta[1], Laurent Sartran[1], Miklós Z. Horváth[1], Goran Žužić[1], Eric Wieser[1]✉, Aja Huang[1], Julian Schrittwieser[1], Yannick Schroecker[1], Hussain Masoom[1], Ottavia Bertolli[1], Tom Zahavy[1], Amol Mandhane[1], Jessica Yung[1], Iuliya Beloshapka[1], Borja Ibarz[1], Vivek Veeriah[1], Lei Yu[1], Oliver Nash[1], Paul Lezeau[1], Salvatore Mercuri[1], Calle Sönne[1], Bhavik Mehta[1], Alex Davies[1], Daniel Zheng[1], Fabian Pedregosa[1], Yin Li[1], Ingrid von Glehn[1], Mark Rowland[1], Samuel Albanie[1], Ameya Velingker[1], Simon Schmitt[1], Edward Lockhart[1], Edward Hughes[1], Henryk Michalewski[1], Nicolas Sonnerat[1], Demis Hassabis[1], Pushmeet Kohli[1]✉ & David Silver[1]

AlphaProof: architecture

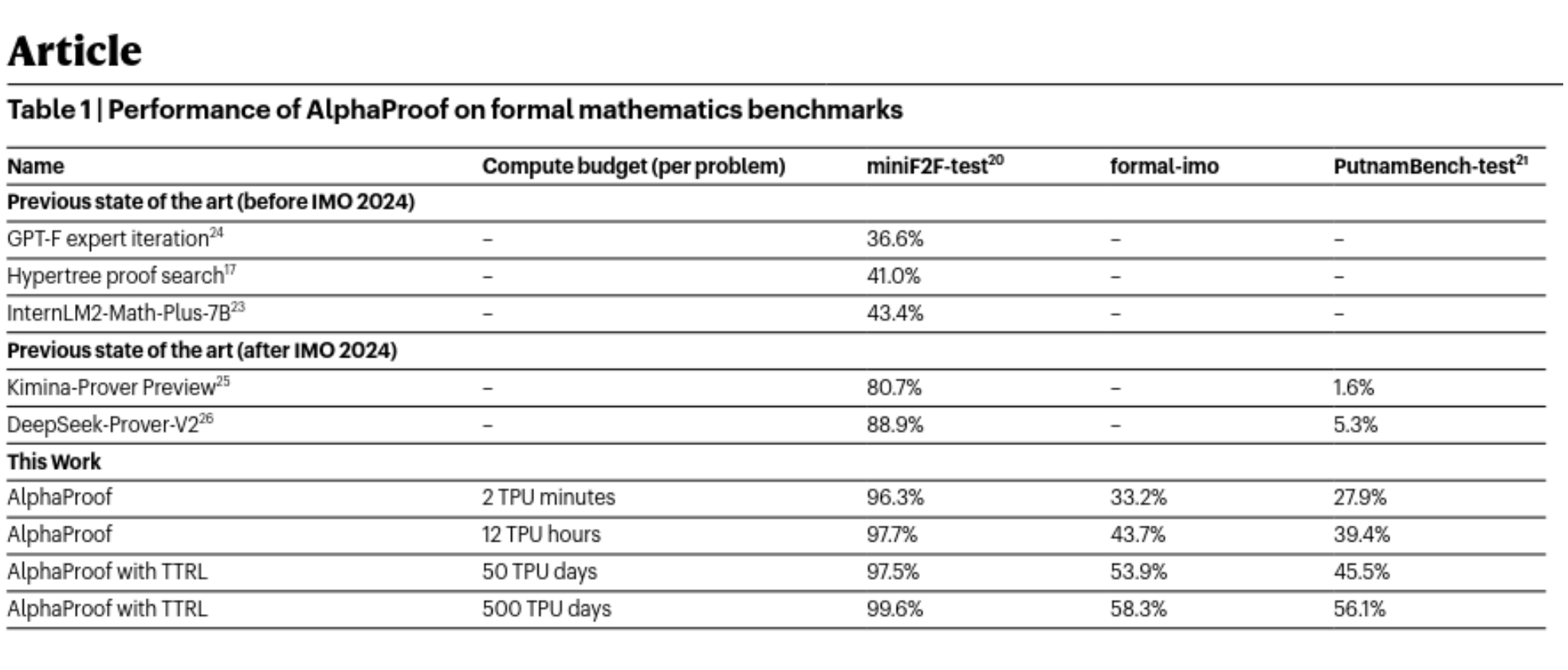

a Main RL
Train
Informal problems
Formalization system
Formal problems
Formal proofs
~1 million
~80 million
Proof network + search
AlphaZero

Test-time RL (TTRL)


a Main RL
Train
Informal problems
Formalization system
Formal problems
Formal proofs
~1 million
~80 million
Proof network + search
AlphaZero
b Test-time RL
Train
Target problem
1
Variant generator
Formal variants
~400,000
Proof network + search
Formal proofs
AlphaZero

AlphaProof: scaling

AlphaProof: results



Article

Table 1 | Performance of AlphaProof on formal mathematics benchmarks

Name	Compute budget (per problem)	miniF2F-test[20]	formal-imo	PutnamBench-test[21]
Previous state of the art (before IMO 2024)				
GPT-F expert iteration[24]	–	36.6%	–	–
Hypertree proof search[17]	–	41.0%	–	–
InternLM2-Math-Plus-7B[23]	–	43.4%	–	–
Previous state of the art (after IMO 2024)				
Kimina-Prover Preview[25]	–	80.7%	–	1.6%
DeepSeek-Prover-V2[26]	–	88.9%	–	5.3%
This Work				
AlphaProof	2 TPU minutes	96.3%	33.2%	27.9%
AlphaProof	12 TPU hours	97.7%	43.7%	39.4%
AlphaProof with TTRL	50 TPU days	97.5%	53.9%	45.5%
AlphaProof with TTRL	500 TPU days	99.6%	58.3%	56.1%

The final scorecard

	Policy	Value	Search	Data	miniF2F

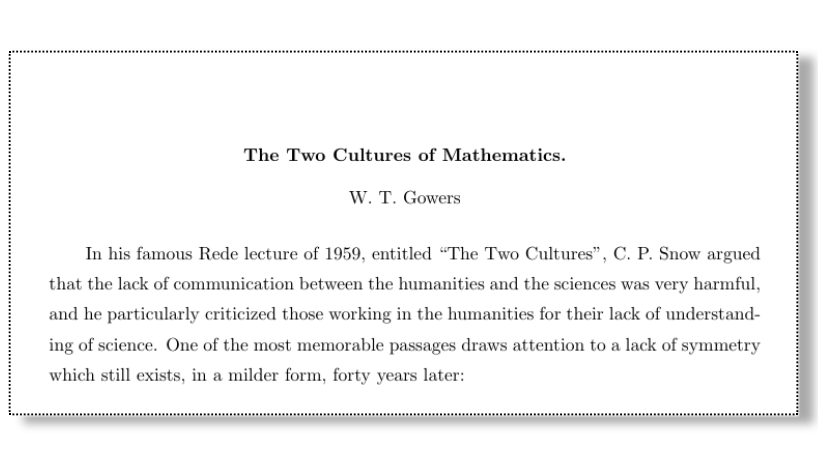
Three strategies for data

CONCLUSION

What we covered today

Beyond the IMO

The two cultures of mathematics



The Two Cultures of Mathematics.

W. T. Gowers

In his famous Rede lecture of 1959, entitled "The Two Cultures", C. P. Snow argued that the lack of communication between the humanities and the sciences was very harmful, and he particularly criticized those working in the humanities for their lack of understanding of science. One of the most memorable passages draws attention to a lack of symmetry which still exists, in a milder form, forty years later:

Erdős vs. Grothendieck

Erdős (problem solver)

Grothendieck (theory builder)

Harvard CS 285: Reinforcement Learning — April 2026

Thank you!

Gabriel Poesia
Research Fellow
Kempner Institute, Harvard University
gabriel_poesia@fas.harvard.edu
https://gpoesia.com

Questions?