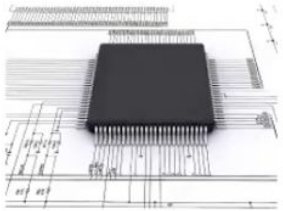
AI-Assisted Theorem Proving in Lean

Evaluating three benchmarks

Theorem Proving and Automated Reasoning

Formal mathematics

Software verification



Hardware verification

Cyber-physical systems

ATPs (Automated Theorem Provers)

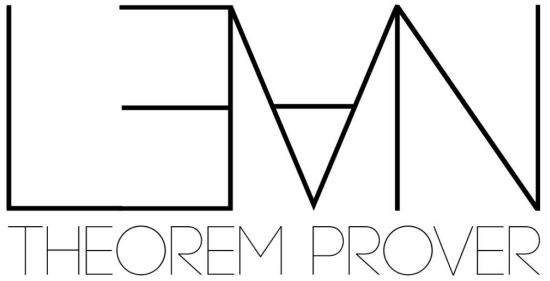
Characteristics:

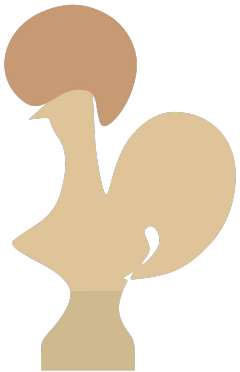
- **First Order Logic**
- **SMT solvers, model checkers**
- **Low human intervention**
- **Limited expressiveness**

ITPs (Interactive Theorem Provers)

Characteristics:

- **Higher Order Logic**
- **Dependent Type Theory**
- **High human intervention**
- **Expressive logic**
- **Human-readable**
- **Machine-checkable**
- **Proof automation is critical for wider adoption**





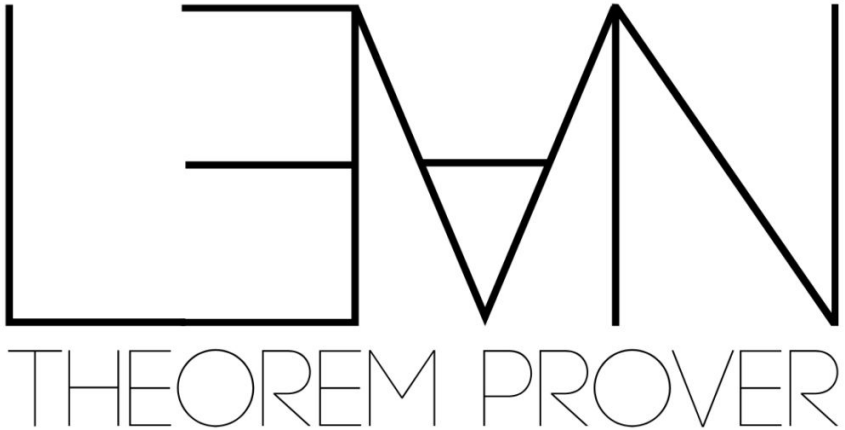
Why Lean?

Lean's foundation:

• Dependent type theory, CIC, proof irrelevance.

Dual role:

• Functional programming + interactive theorem proving.



Ecosystem:

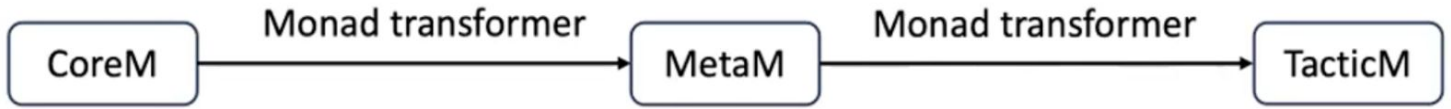
• Mathlib: community-driven library for pure mathematics.

• Active research and development in metaprogramming and automation.

Lean's verification paradigm:

State Monads of Lean

- Environments, local contexts, and goals implemented within Lean as a hierarchy of state monads



Environment
- Definitions, lemmas, etc.

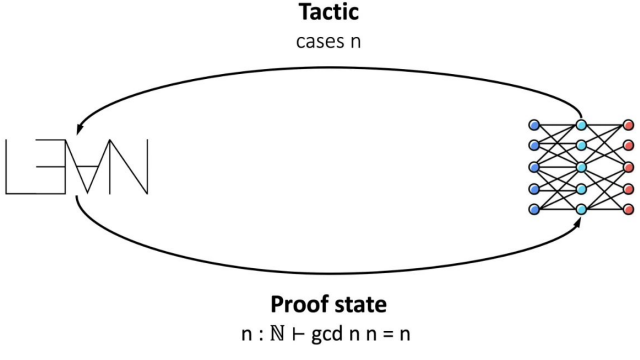
Local context
- Local hypotheses & goals

Current goals

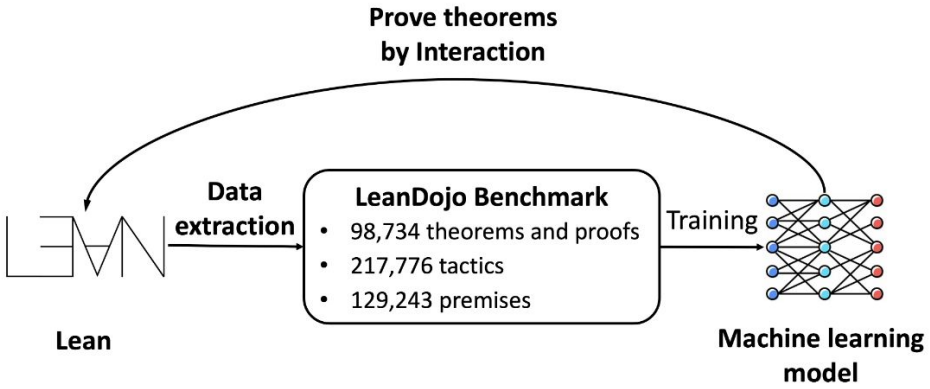
AI-Assisted Theorem Proving in Lean

Objectives:

- **Leveraging Machine Learning and LLMs to automate the process of ITPs**
- **Suggest tactics**
- **Generate proofs**
- **Guide search**
- **Formalize informal proofs**
- **Acts as a copilot to the human user**

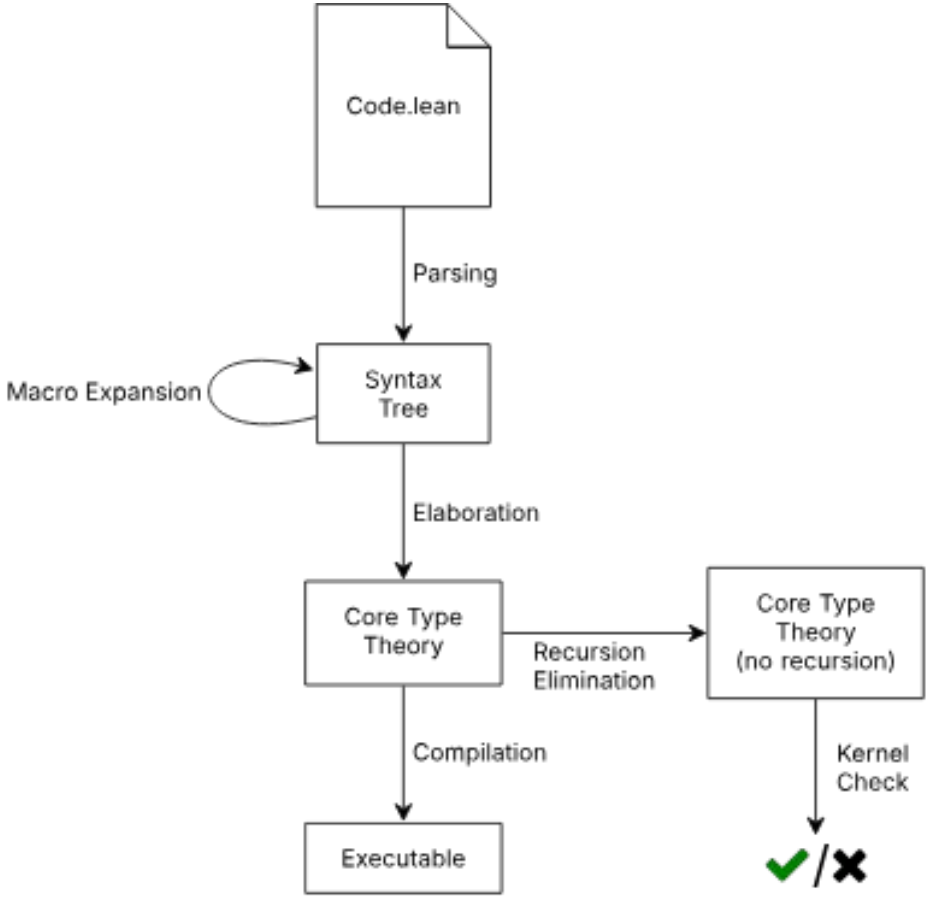


LeanDojo Overview

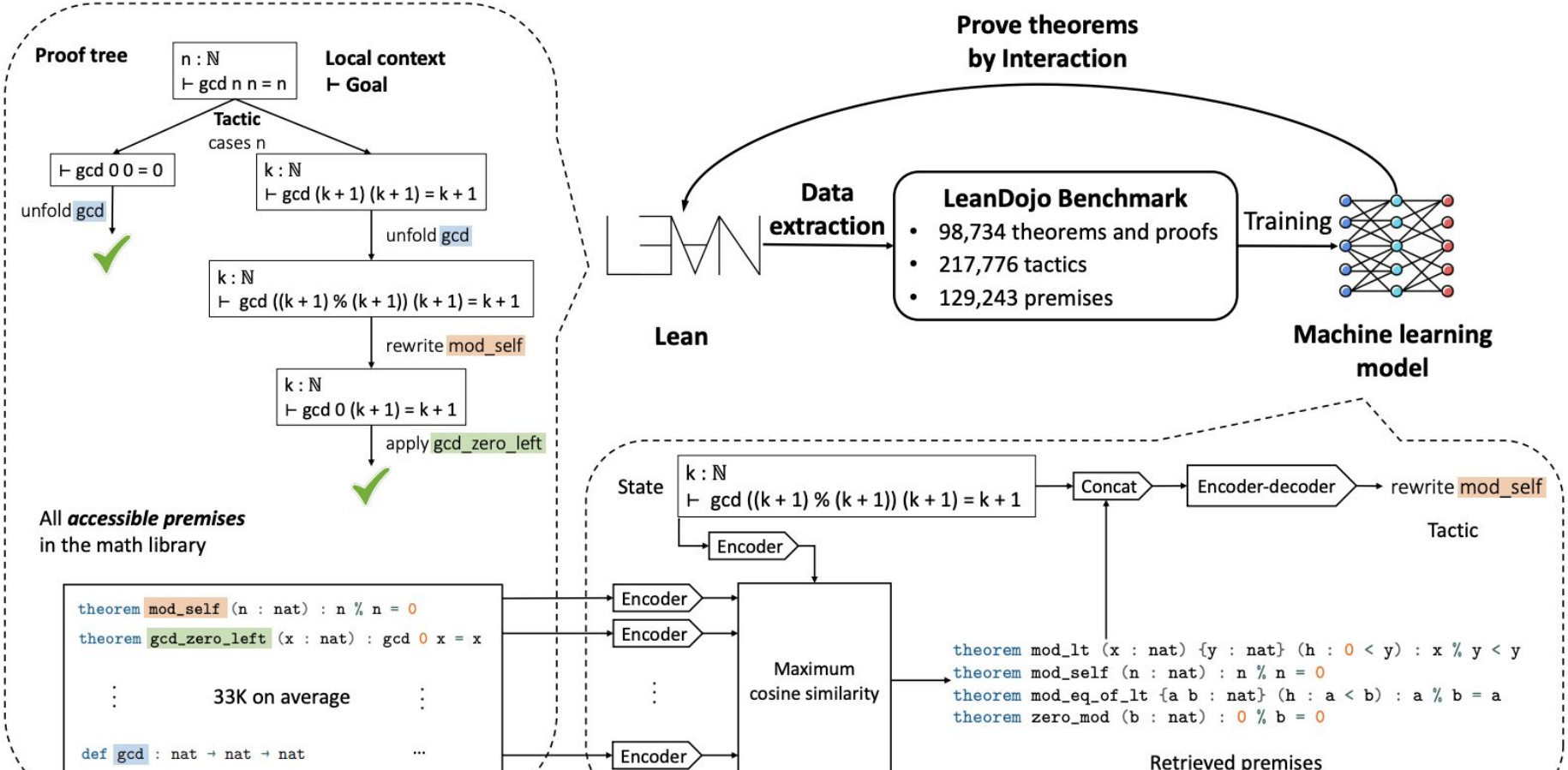


Lean 4 benchmark: 122,517 theorems, 259,580 tactics, 167,779 premises

Gym Concept	LeanDojo Analog
Environment	The Lean proof state and proof engine
State	Current proof goal, hypotheses, context
Action	Tactic (e.g. `intro`, `rw`, `apply`, etc.)
Reward	Success/failure signals, proof progress
Done	Whether the theorem has been fully proved



Model Training and Theorem Proving: ReProver



Results of ReProver

Performance increase:

- Random split: 47.6% success rate => 51.2% success rate
- Novel premises split: 23.2% success rate => 26.3% success rate

Reliability increase:

- 21.1% error rate => 1.4% error rate

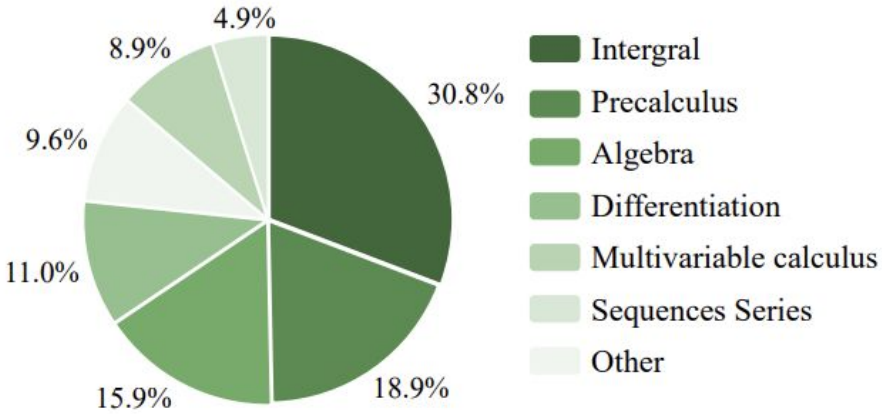
Community contribution:

- discovered Lean proofs for 65 theorems

FormalMath Overview

5,560 formally verified problems from:

high school olympiad question and undergraduate level math theorems



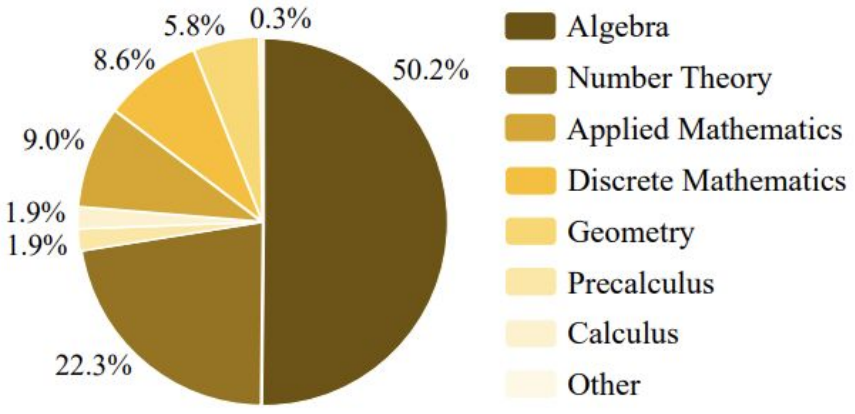
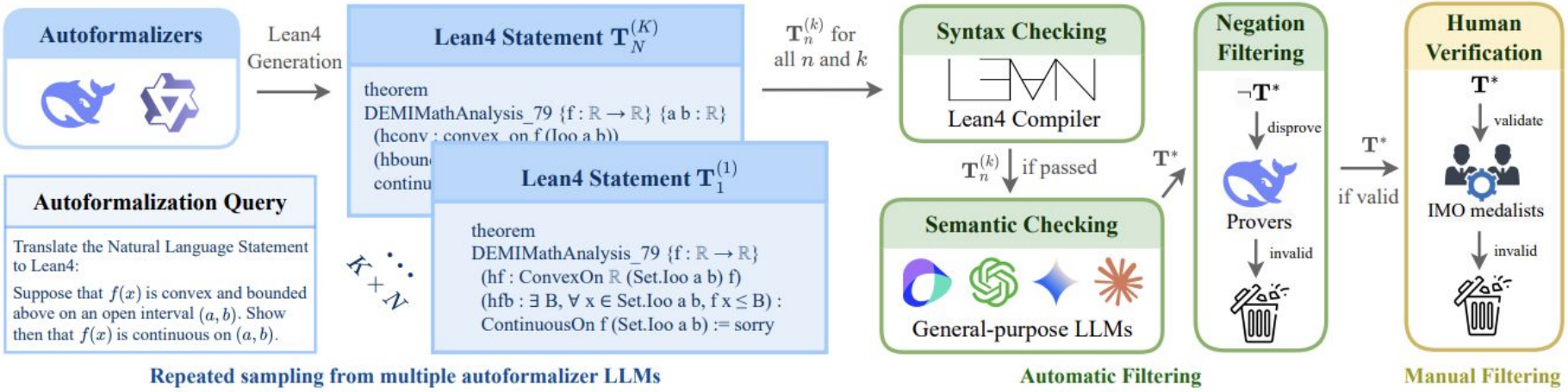
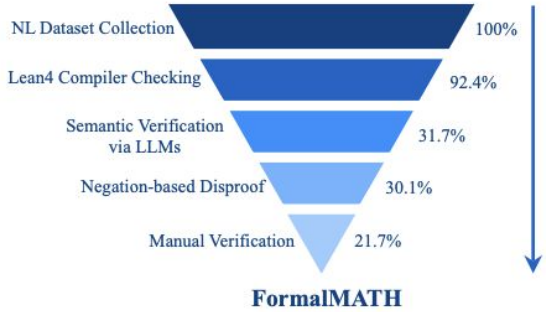


Figure 3: The distribution of mathematical domains in the full set of FormalMATH.

Human-in-the-loop pipeline





(b) Data preservation rate

21.7% ÷ 30.1% = 72.09%

72.09% of statements before manual verification are retained

Evaluation process

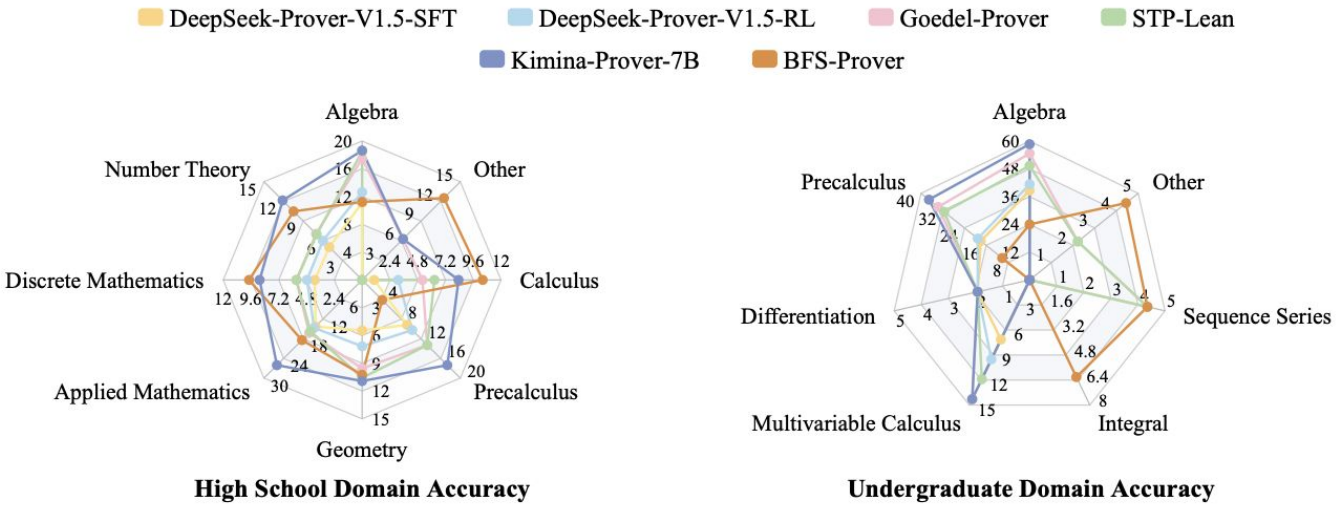
Best-First Tree Search (BFS)

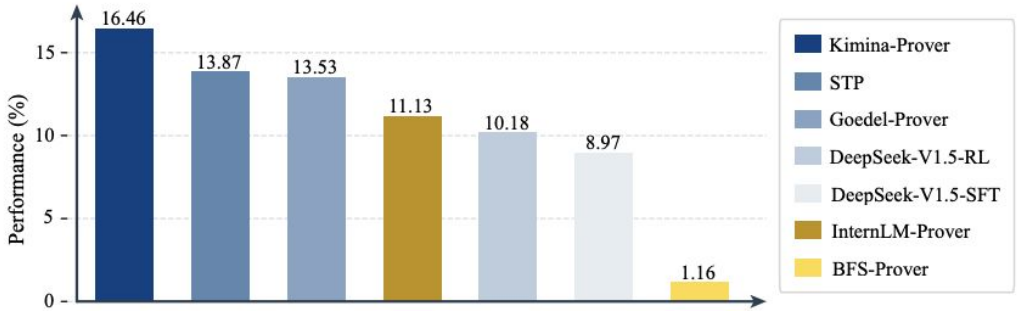
Method	Sampling budget	Pass@K (%)
Best-First Tree Search Methods		
BFS(DeepSeek-Prover-V1.5-RL) [XRS+24]	$1 \times 32 \times 100$	4.91
	$4 \times 32 \times 100$	10.29
	$8 \times 32 \times 100$	12.16
	$16 \times 32 \times 100$	14.96
	$32 \times 32 \times 100$	17.41
BFS(InternLM-V2.5) [WHZ+24]	$1 \times 32 \times 100$	7.87
	$4 \times 32 \times 100$	15.79
	$8 \times 32 \times 100$	20.02
	$16 \times 32 \times 100$	22.74
	$32 \times 32 \times 100$	25.65
BFS(BFS-Prover) [XXY+25]	$1 \times 32 \times 100$	27.10
	$4 \times 32 \times 100$	34.04
	$8 \times 32 \times 100$	37.56
	$16 \times 32 \times 100$	41.75
	$32 \times 32 \times 100$	45.88

Single-Pass Generation (SPG)

Single-Pass Generation Methods		
Kimina-Prover-7B [WUL+25]	32	48.94
STP [DM25]	32	48.59
	128	50.35
	512	51.45
	1024	52.03
	2048	52.60
	3200	53.17
DeepSeek-Prover-V1.5-SFT [XRS+24]	32	40.40
	128	42.11
	512	44.17
	1024	45.08
	2048	46.12
	3200	46.82
DeepSeek-Prover-V1.5-RL [XRS+24]	32	47.98
	128	48.75
	512	49.27
	1024	49.68
	2048	50.08
	3200	50.35
Goedel-Prover [LTL+25]	32	46.70
	128	48.02
	512	48.68
	1024	49.04
	2048	49.20
	3200	49.41
Ensemble of All SPG Methods	4×3200	54.11

Evaluation results of LLM-based theorem provers





(a) Performance of current provers on FormalMATH

AlphaProof's Formal Conjectures

Terence Tao
@tao@mathstodon.xyz

Google #DeepMind has launched an open repository of formalized mathematics conjectures github.com/google-deepmind/for... . For instance, the four Landau problems in analytic number theory are formalized at github.com/google-deepmind/for... . They are soliciting further contributions to the database (at present it only contains a tiny fraction of the conjectures in the literature).

It is of course far easier to state an open problem than it is to prove it, but having some sort of standardized formulation of such problems is important first step if one is to hope to use automated tools to help make progress on these problems. If one naively asks an AI tool to formally solve an informally specified problem, it is far more likely that it could succeed on a technicality - for instance, by establishing a formal statement which contains an unintended edge case that can be handled trivially (e.g., if some key parameter is intended to be non-zero, but the formalization permits the parameter to vanish) - than it is to solve the problem as intended.

google-deepmind/**formal-conjectures**

A collection of formalized statements of conjectures in Lean.

6 Contributors · 23 Issues · 8 Stars · 1 Fork

Key features of formal conjectures repository

- Category attribute

```
@[category research open]
theorem foo : Transcendental ℚ (rexp 1 + π) := by
  sorry

@[category research solved]
theorem bar : FermatLastTheorem := by
  sorry
```

- AMS attribute

```
@[AMS 11] -- `11` means "Number Theory"
theorem flt : FermatLastTheorem := by
  sorry
```

Problem Category Statistics

Count	Category
289	Research (open)
208	Research (solved)
5	Graduate
31	Undergraduate
8	High School
57	API
44	Tests

- answer() elaborator

```
@[category research open]
theorem HadwigerNelsonProblem :
    UnitDistancePlaneGraph.chromaticNumber = answer(sorry) := by
  sorry
```

Three Benchmarks summarized

- **LeanDojo**: framework that extracts training data from Lean's mathematical libraries and provides a gym-like interface for RL AI agents to interact with

- **FormalMath**: Benchmark for measuring formal mathematical reasoning abilities of LLM-Prover by evaluating proof generation

- **Formal Conjectures**: Repository of mathematical conjectures at different levels, formalized in Lean, used to evaluate generalization and progress of AI theorem provers