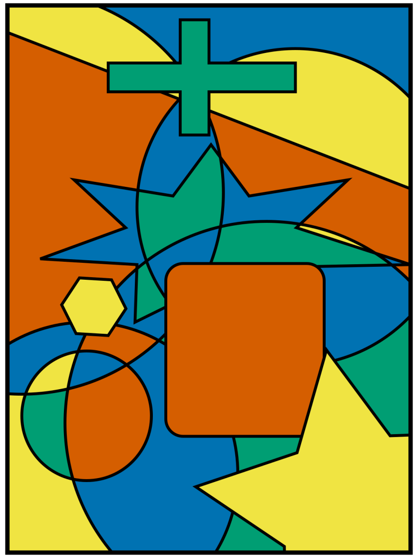
Theorem Proving via Machine Learning

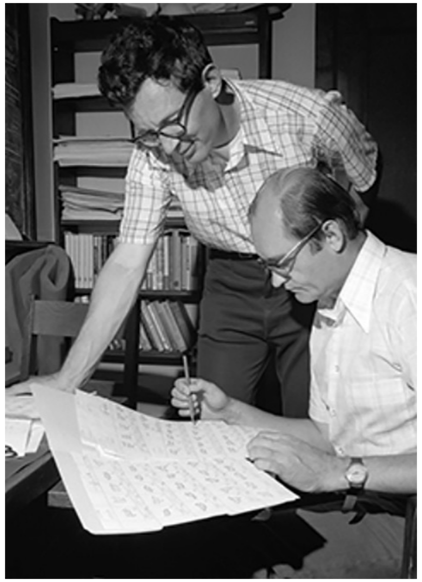
Kaiyu Yang

Postdoc @ Computing + Mathematical Sciences

Caltech

Computer-Aided Proofs in Mathematics



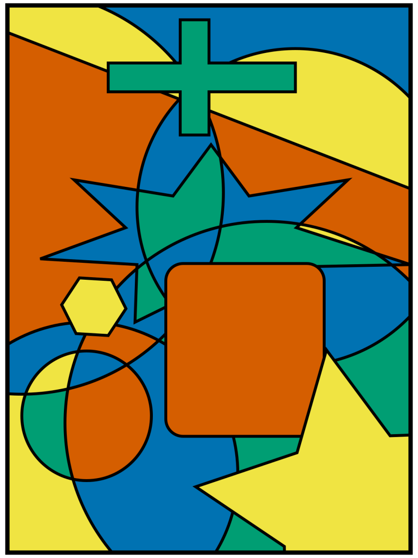


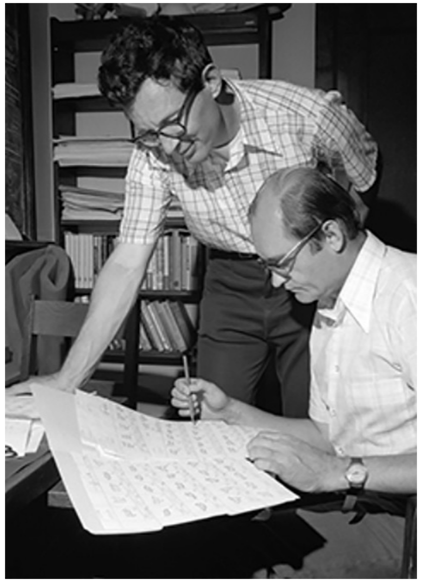
Four Color Theorem

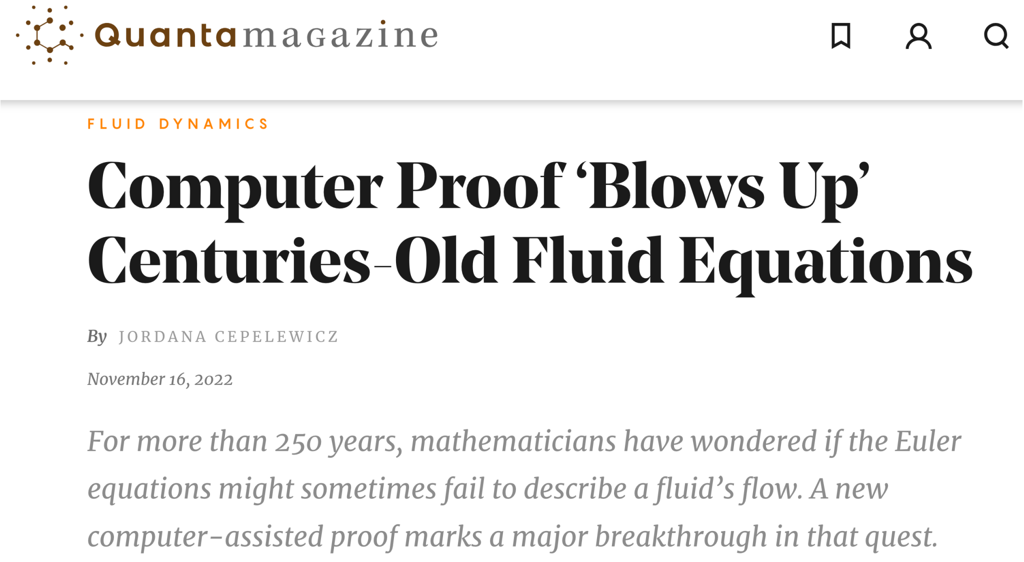
Use computers to check 1000+ configurations

[Appel and Haken, "Every Planar Map Is Four Colorable", 1976]

Computer-Aided Proofs in Mathematics







Quantamagazine

FLUID DYNAMICS

Computer Proof 'Blows Up' Centuries-Old Fluid Equations

By JORDANA CEPELEWICZ

November 16, 2022

For more than 250 years, mathematicians have wondered if the Euler equations might sometimes fail to describe a fluid's flow. A new computer-assisted proof marks a major breakthrough in that quest.

Four Color Theorem

Computers check 1000+ configurations

[Appel and Haken, "Every Planar Map Is Four Colorable", 1976]

Blowup of the Euler Equations

Computers calculate bounds of integrals

[Chen and Thomas, "Stable Nearly Self-similar Blowup Of The 2D Boussinesq And 3D Euler Equations With Smooth Data", 2022]

Automated Theorem Proving

$$1 + 2 + \cdots + n = \frac{(n+1)n}{2}$$

- Generate the proof fully automatically

Automated Theorem Proving

$$1 + 2 + \cdots + n = \frac{(n+1)n}{2}$$

$$\neg q \vee p \vee \neg r$$

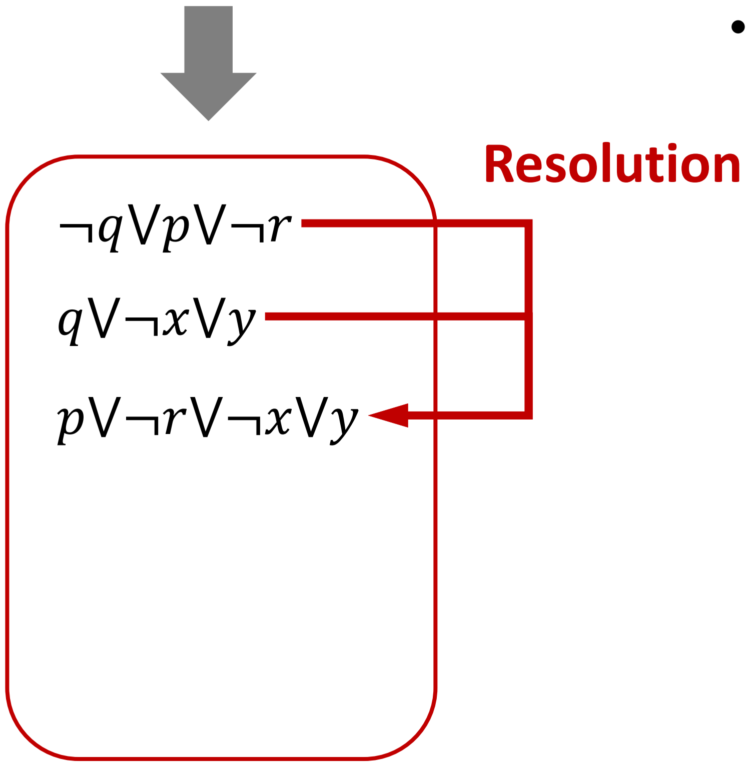
$$q \vee \neg x \vee y$$

Conjunctive normal form (CNF)

- Generate the proof fully automatically
- Low-level: First-order logic, CNFs, and resolution

Automated Theorem Proving

$$1 + 2 + \cdots + n = \frac{(n+1)n}{2}$$

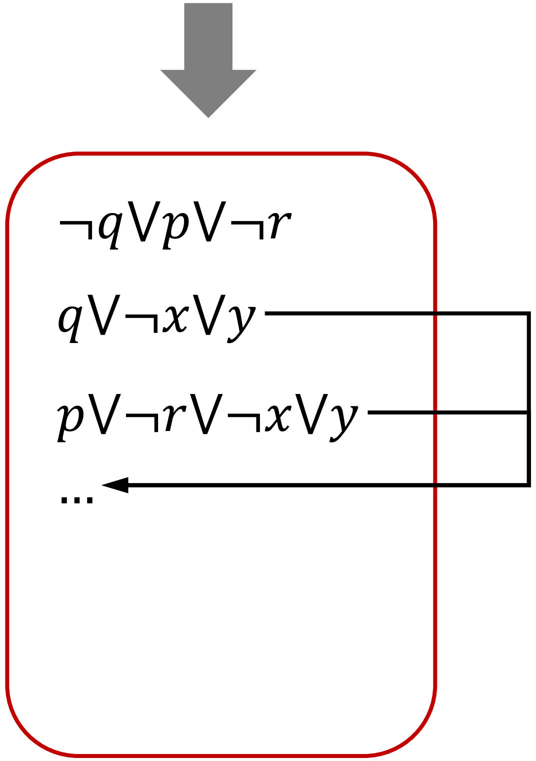


Conjunctive normal form (CNF)

- Generate the proof fully automatically
- Low-level: First-order logic, CNFs, and resolution

Automated Theorem Proving

$$1 + 2 + \cdots + n = \frac{(n+1)n}{2}$$

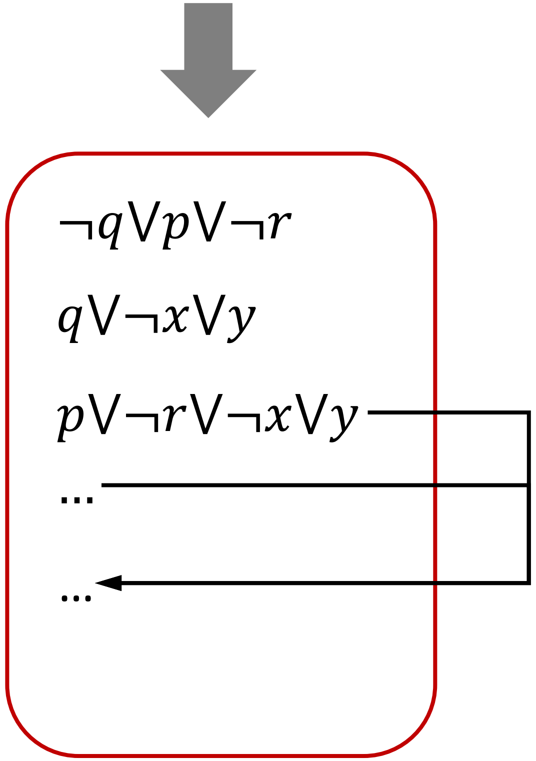


Conjunctive normal form (CNF)

- Generate the proof fully automatically
- Low-level: First-order logic, CNFs, and resolution

Automated Theorem Proving

$$1 + 2 + \cdots + n = \frac{(n+1)n}{2}$$

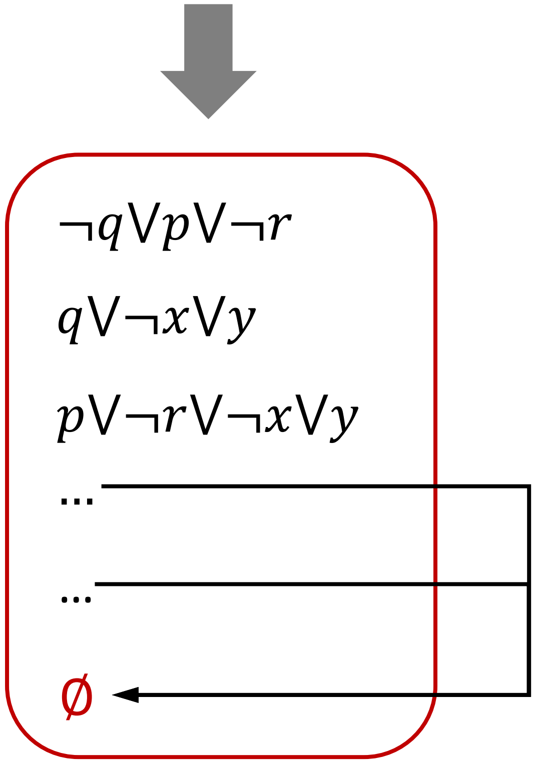


Conjunctive normal form (CNF)

- Generate the proof fully automatically
- Low-level: First-order logic, CNFs, and resolution

Automated Theorem Proving

$$1 + 2 + \cdots + n = \frac{(n+1)n}{2}$$



Conjunctive normal form (CNF)

- Generate the proof fully automatically
- Low-level: First-order logic, CNFs, and resolution

Automated Theorem Proving

$$1 + 2 + \cdots + n = \frac{(n+1)n}{2}$$

$\neg q \lor p \lor \neg r$

$q \lor \neg x \lor y$

$p \lor \neg r \lor \neg x \lor y$

…

…

$\emptyset$

Conjunctive normal form (CNF)

- Generate the proof fully automatically
- Low-level: First-order logic, CNFs, and resolution
- Main challenge: Large search space

[Haken, "The Intractability of Resolution", Theoretical Computer Science, 1985]

- Heuristics for pruning the search space

[Kovács and Voronkov, CAV 2013]
[Schulz et al. CADE 2019].
[Korovin, IJCAR 2008]
[Urban et al. TABLEAUX 2011]
[Loos et al. LPAR-21]
[Kaliszyk et al. NeurIPS 2018]

- Successful examples: Robbins Conjecture

[McCune, "Solution of the Robbins Problem", 1997]

- Intractable for most theorems

Automated Theorem Proving

$$1 + 2 + \cdots + n = \frac{(n+1)n}{2}$$

$\neg q \vee p \vee \neg r$

$q \vee \neg x \vee y$

$p \vee \neg r \vee \neg x \vee y$

…

…

$\emptyset$

Conjunctive normal form (CNF)

- Generate the proof fully automatically
- Low-level: First-order logic, CNFs, and resolution
- Main challenge: Large search space

[Haken, “The Intractability of Resolution”, Theoretical Computer Science, 1985]

- Heuristics for pruning the search space

[Kovács and Voronkov, CAV 2013] [Urban et al. TABLEAUX 2011]
[Schulz et al. CADE 2019]. [Loos et al. LPAR-21]
[Korovin, IJCAR 2008] [Kaliszyk et al. NeurIPS 2018]

- Successful examples: Robbins Conjecture

[McCune, “Solution of the Robbins Problem”, 1997]

- Intractable for most theorems in math
- **Lack high-level intuitions of mathematicians**

Theorem Proving in Proof Assistants

Proof assistant

Theorem Proving in Proof Assistants

Human

```
theorem gcd_self (n : nat) : gcd n n = n :=
```

Proof assistant

Theorem Proving in Proof Assistants

Human

```
theorem gcd_self (n : nat) : gcd n n = n :=
```

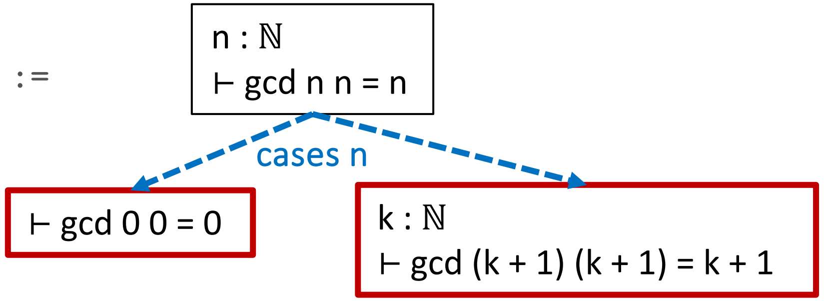
$n : \mathbb{N}$
$\vdash \text{gcd } n\ n = n$

Proof assistant

Theorem Proving in Proof Assistants

Human

```
theorem gcd_self (n : nat) : gcd n n = n :=
begin
  cases n,
```

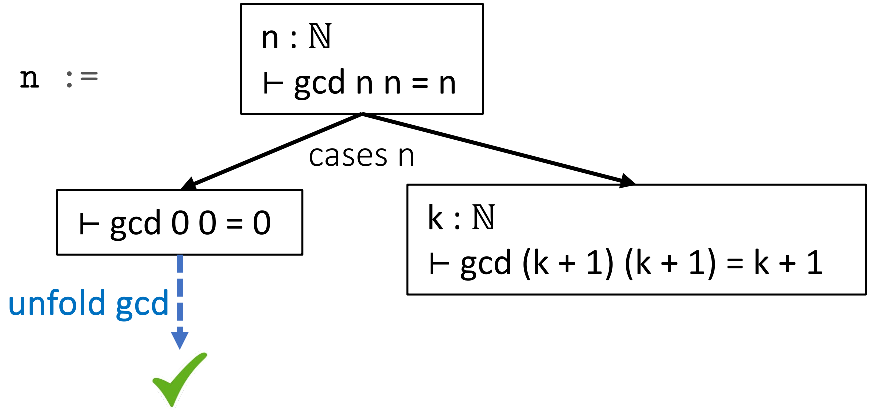


Proof assistant

Theorem Proving in Proof Assistants

Human

```
theorem gcd_self (n : nat) : gcd n n = n :=
begin
  cases n,
  { unfold gcd },
```

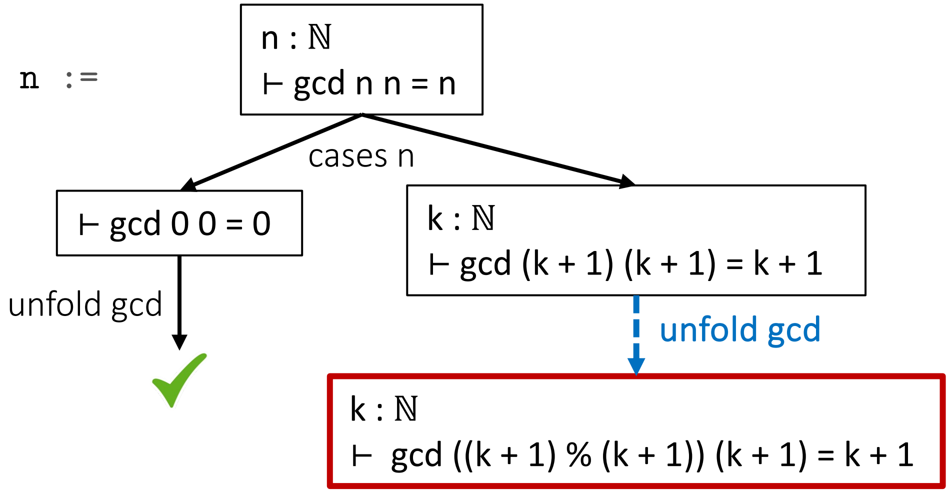


Proof assistant

Theorem Proving in Proof Assistants

Human

```
theorem gcd_self (n : nat) : gcd n n = n :=
begin
  cases n,
  { unfold gcd },
  unfold gcd,
```

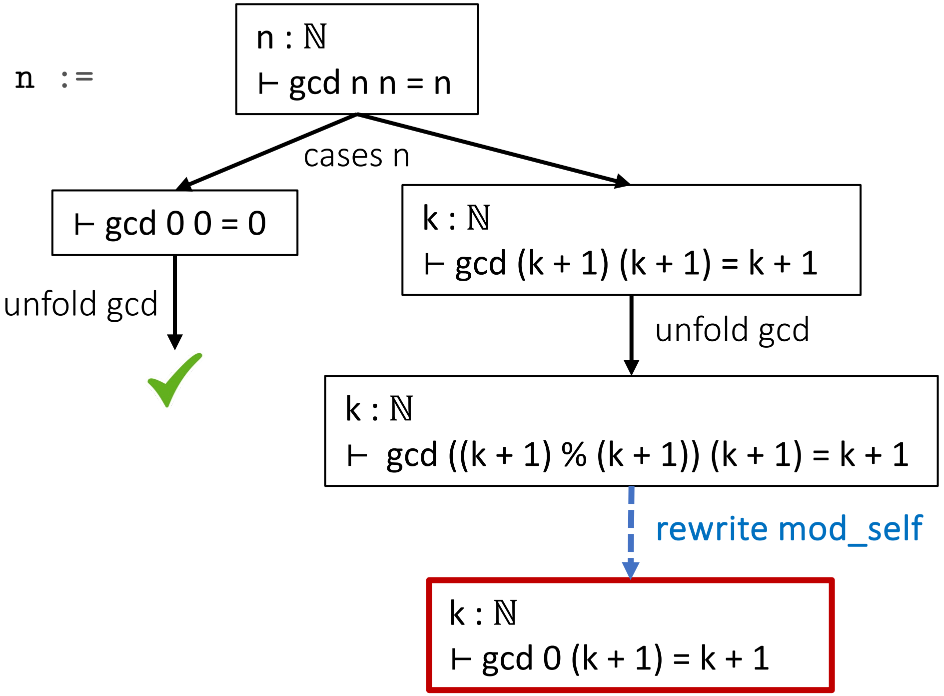


Proof assistant

Theorem Proving in Proof Assistants

Human

```
theorem gcd_self (n : nat) : gcd n n = n :=
begin
  cases n,
  { unfold gcd },
  unfold gcd,
  rewrite mod_self,
```

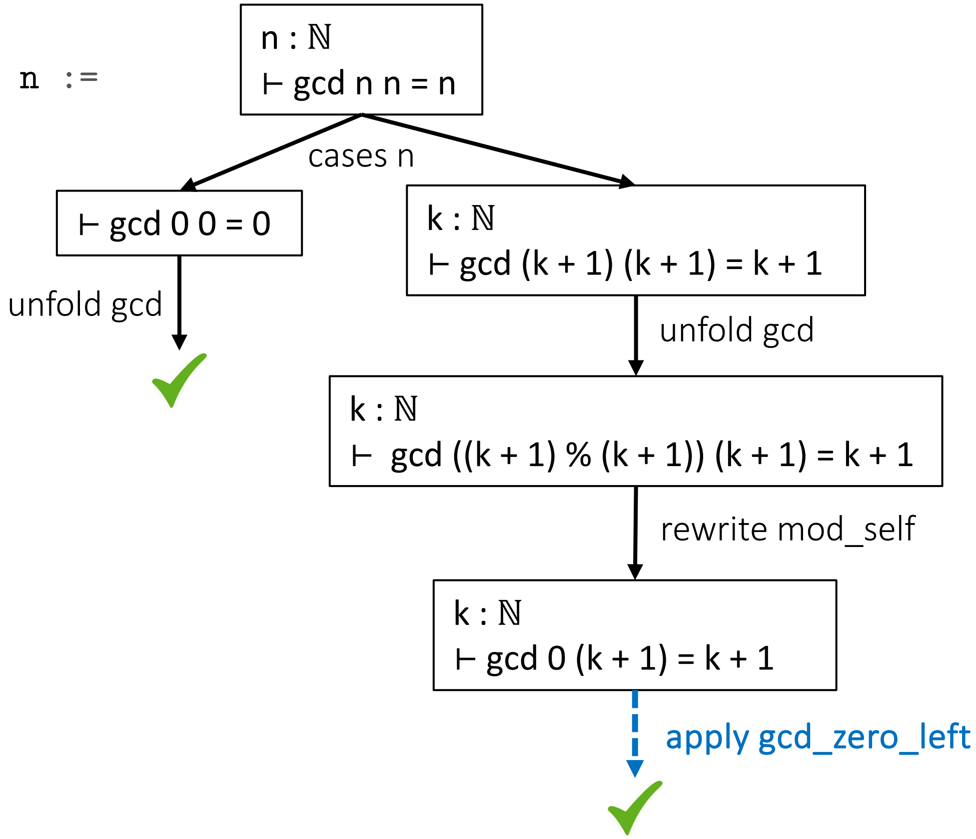


Proof assistant

Theorem Proving in Proof Assistants

Human

```
theorem gcd_self (n : nat) : gcd n n = n :=
begin
  cases n,
  { unfold gcd },
  unfold gcd,
  rewrite mod_self,
  apply gcd_zero_left
end
```



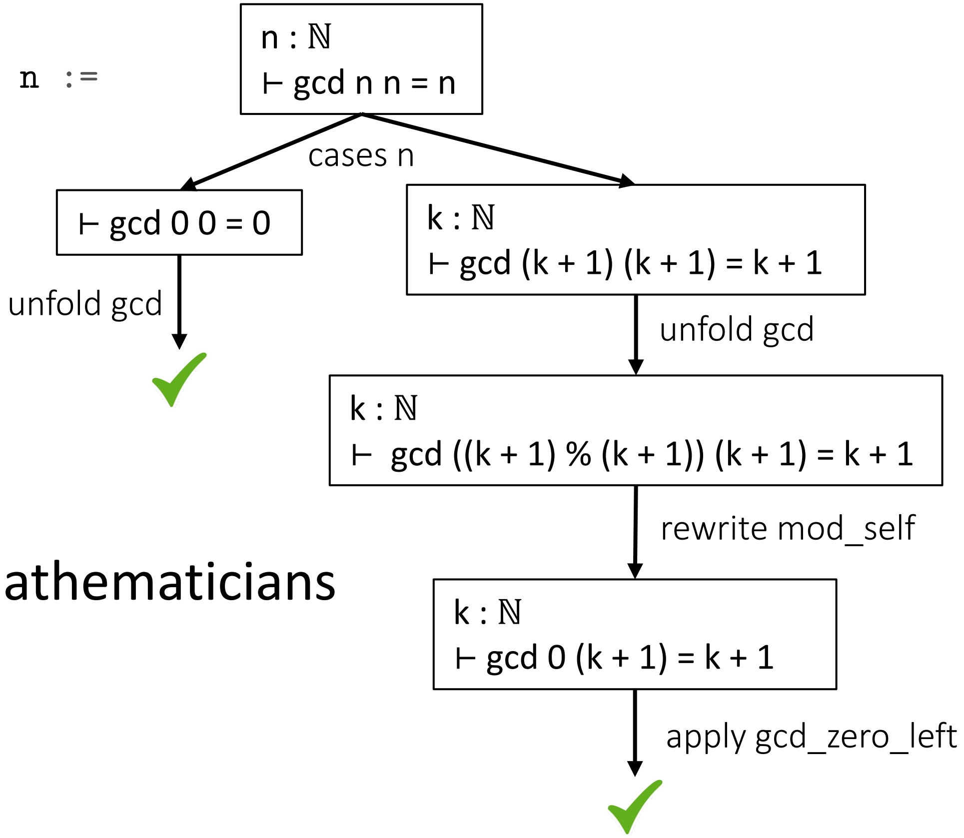
Proof assistant

Theorem Proving in Proof Assistants

Human

```
theorem gcd_self (n : nat) : gcd n n = n :=
begin
  cases n,
  { unfold gcd },
  unfold gcd,
  rewrite mod_self,
  apply gcd_zero_left
end
```

✓ High-level guidance from mathematicians

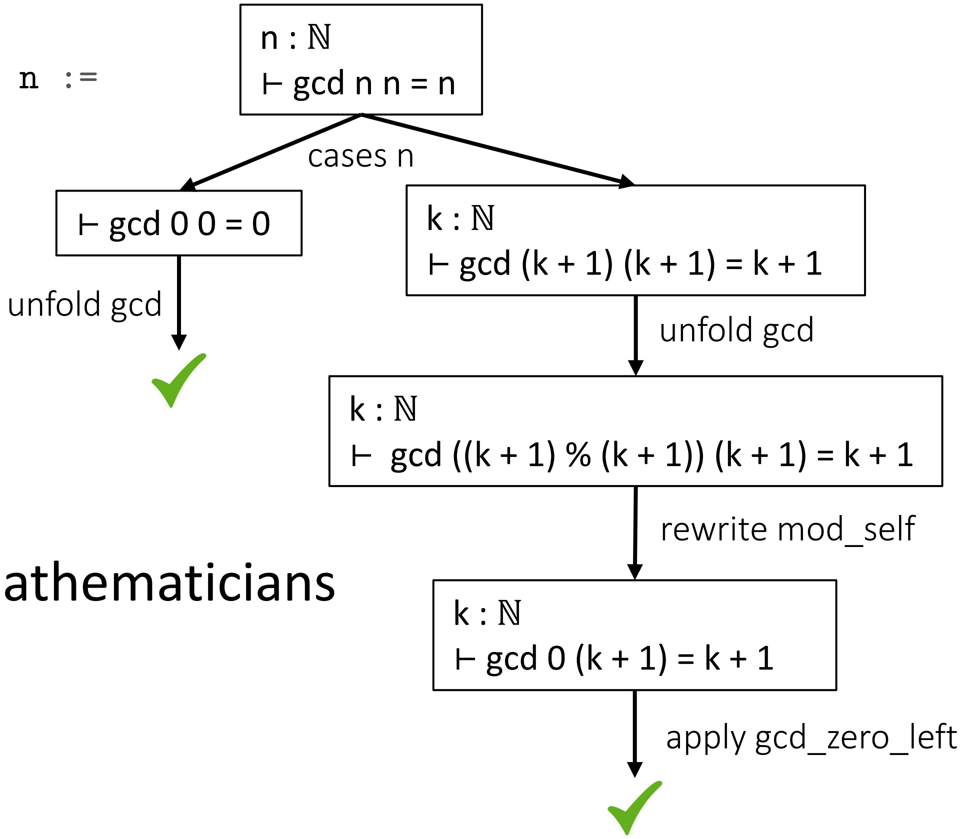


Proof assistant

Theorem Proving in Proof Assistants

Human

```
theorem gcd_self (n : nat) : gcd n n = n :=
begin
  cases n,
  { unfold gcd },
  unfold gcd,
  rewrite mod_self,
  apply gcd_zero_left
end
```



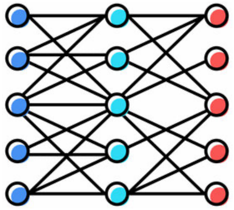
Proof assistant

✓ High-level guidance from mathematicians

✗ Labor-intensive

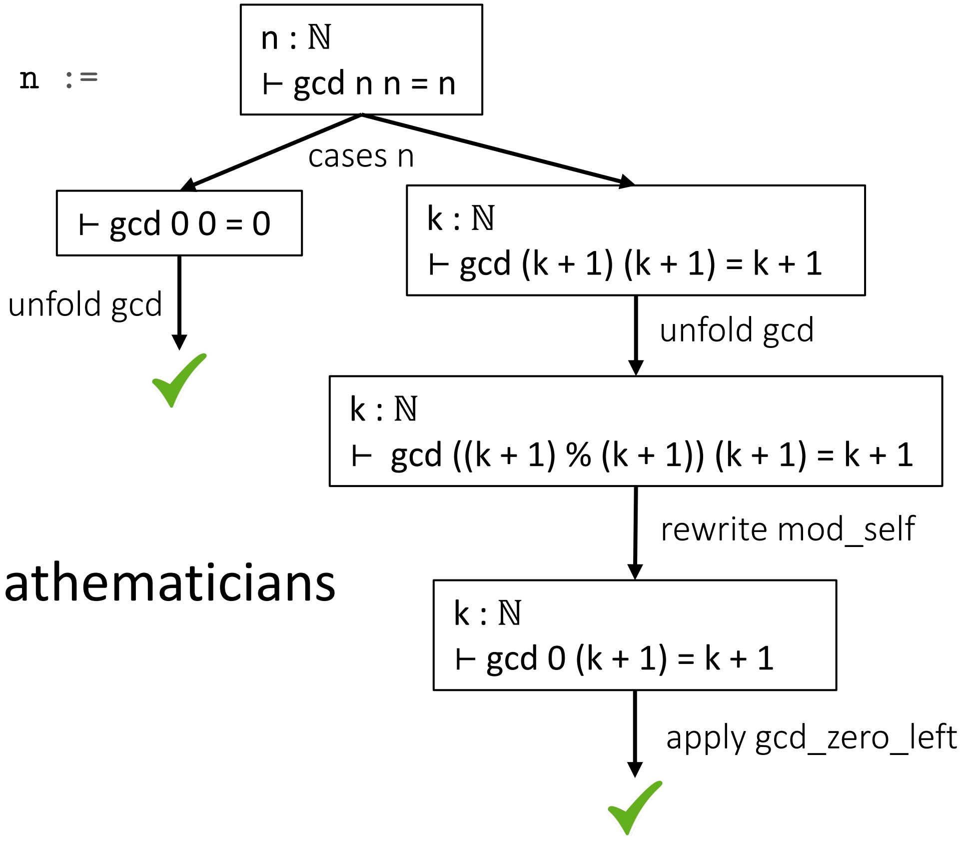
Theorem Proving in Proof Assistants

Human



Machine learning

```
theorem gcd_self (n : nat) : gcd n n = n :=
begin
  cases n,
  { unfold gcd },
  unfold gcd,
  rewrite mod_self,
  apply gcd_zero_left
end
```



Proof assistant

✓ High-level guidance from mathematicians

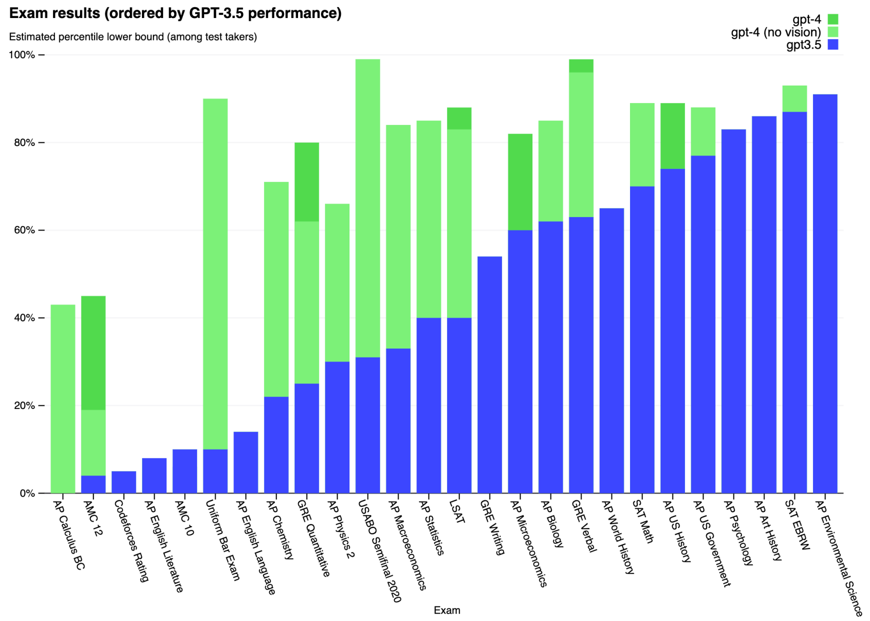
✗ ~~Labor-intensive~~

Lean to interact with proof assistants

Large Language Models (LLMs)

Large Language Models (LLMs) for Math

- GPT-4: 89th percentile in SAT Math among human test takers



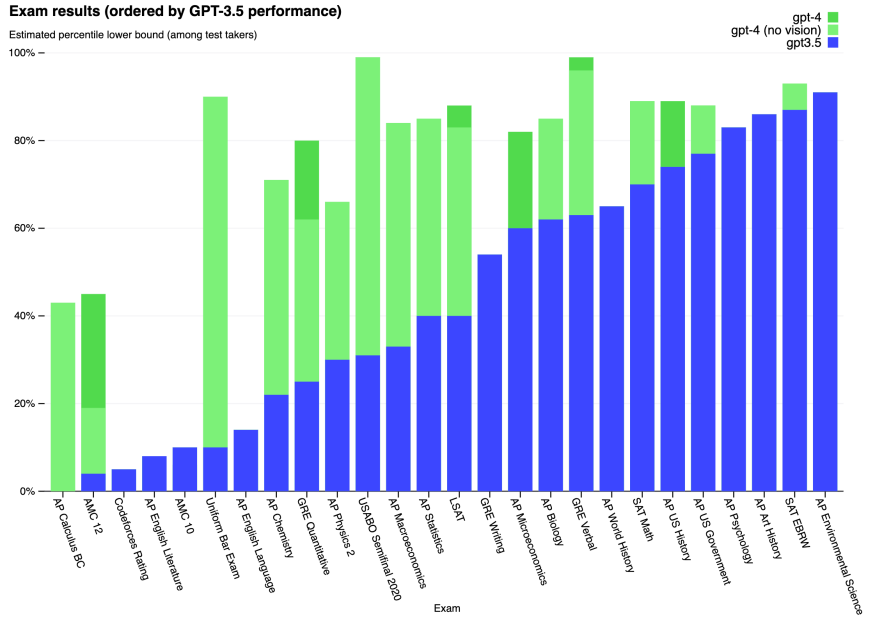
[OpenAI, "GPT-4 Technical Report", 2023]

Large Language Models (LLMs) for Math

- GPT-4: 89th percentile in SAT Math among human test takers
- Minerva: Google's LLM specialized in math

Question: For every $a, b, b \neq a$ prove that

$$\frac{a^2 + b^2}{2} > \left(\frac{a+b}{2}\right)^2.$$

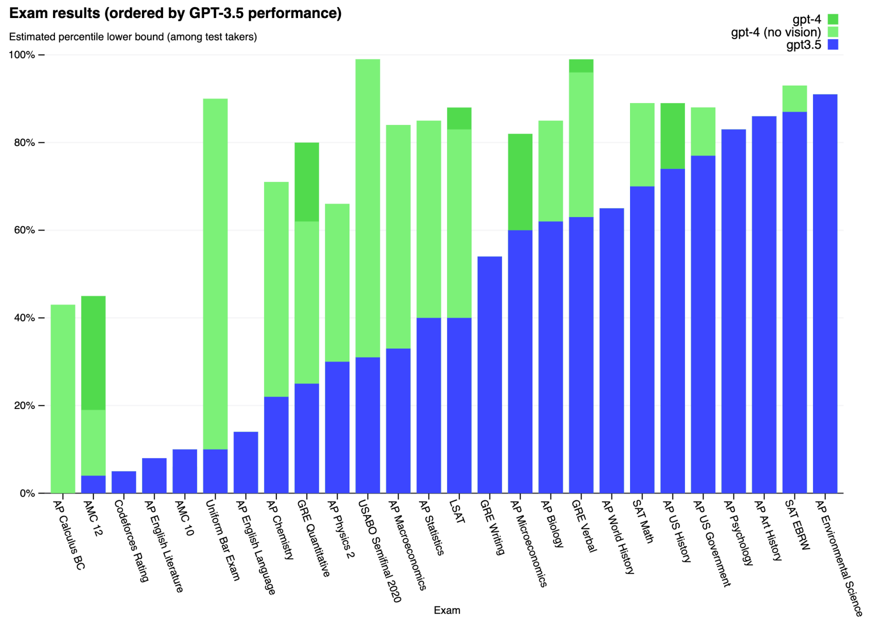


[OpenAI, "GPT-4 Technical Report", 2023]

[Lewkowycz et al., "Solving Quantitative Reasoning Problems with Language Models", 2022]

Large Language Models (LLMs) for Math

- GPT-4: 89th percentile in SAT Math among human test takers
- Minerva: Google's LLM specialized in math



[OpenAI, "GPT-4 Technical Report", 2023]

Question: For every $a, b, b \neq a$ prove that

$$\frac{a^2 + b^2}{2} > \left(\frac{a+b}{2}\right)^2.$$

Model output:

$$\frac{a^2+b^2}{2} > \left(\frac{a+b}{2}\right)^2$$

$$\iff \frac{a^2+b^2}{2} > \frac{a^2+b^2+2ab}{4}$$

$$\iff a^2 + b^2 > \frac{a^2+b^2+2ab}{2}$$

$$\iff 2a^2 + 2b^2 > a^2 + b^2 + 2ab$$

$$\iff a^2 + b^2 > 2ab$$

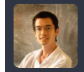
$$\iff a^2 + b^2 - 2ab > 0$$

$$\iff (a-b)^2 > 0$$

which is true, because the square of a real number is positive.

[Lewkowycz et al., "Solving Quantitative Reasoning Problems with Language Models", 2022]

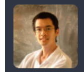
Large Language Models (LLMs) for Math



Terence Tao
@tao@mathstodon.xyz

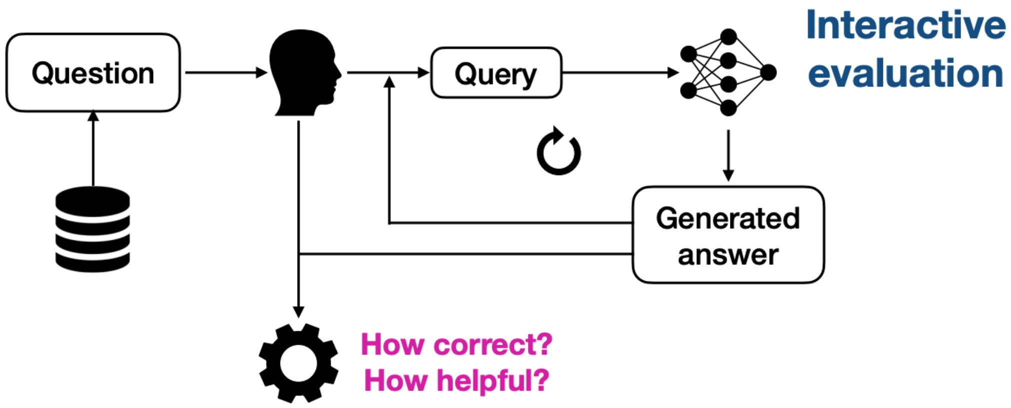
As an experiment, I recently tried consulting #GPT4 on a question I found on #MathOverflow prior to obtaining a solution. The question is at mathoverflow.net/questions/449... and my conversation with GPT-4 is at chat.openai.com/share/53aab67e... . Based on past experience, I knew to not try to ask the #AI to answer the question directly (as this would almost surely lead to nonsense), but instead to have it play the role of a collaborator and offer strategy suggestions. It did end up suggesting eight approaches, one of which (generating functions) being the one that was ultimately successful. In this particular case, I would probably have tried a generating function approach eventually, and had no further need of GPT-4 once I started doing so (relying instead on a lengthy MAPLE worksheet, and some good old-fashioned hand calculations at the blackboard and with pen and paper), but it was slightly helpful nevertheless (I had initially thought of pursuing the asymptotic analysis approach instead to gain intuition, but this turned out to be unnecessary). I also asked an auxiliary question in which GPT-4 pointed out the relevance of Dyck paths (and some related structures), which led to one of my other comments on the OP's question. I decided to share my experience in case it encourages others to perform similar experiments.

Large Language Models (LLMs) for Math



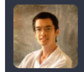
Terence Tao
@tao@mathstodon.xyz

As an experiment, I recently tried consulting #GPT4 on a question I found on #MathOverflow prior to obtaining a solution. The question is at mathoverflow.net/questions/449... and my conversation with GPT-4 is at chat.openai.com/share/53aab67e... . Based on past experience, I knew to not try to ask the #AI to answer the question directly (as this would almost surely lead to nonsense), but instead to have it play the role of a collaborator and offer strategy suggestions. It did end up suggesting eight approaches, one of which (generating functions) being the one that was ultimately successful. In this particular case, I would probably have tried a generating function approach eventually, and had no further need of GPT-4 once I started doing so (relying instead on a lengthy MAPLE worksheet, and some good old-fashioned hand calculations at the blackboard and with pen and paper), but it was slightly helpful nevertheless (I had initially thought of pursuing the asymptotic analysis approach instead to gain intuition, but this turned out to be unnecessary). I also asked an auxiliary question in which GPT-4 pointed out the relevance of Dyck paths (and some related structures), which led to one of my other comments on the OP's question. I decided to share my experience in case it encourages others to perform similar experiments.



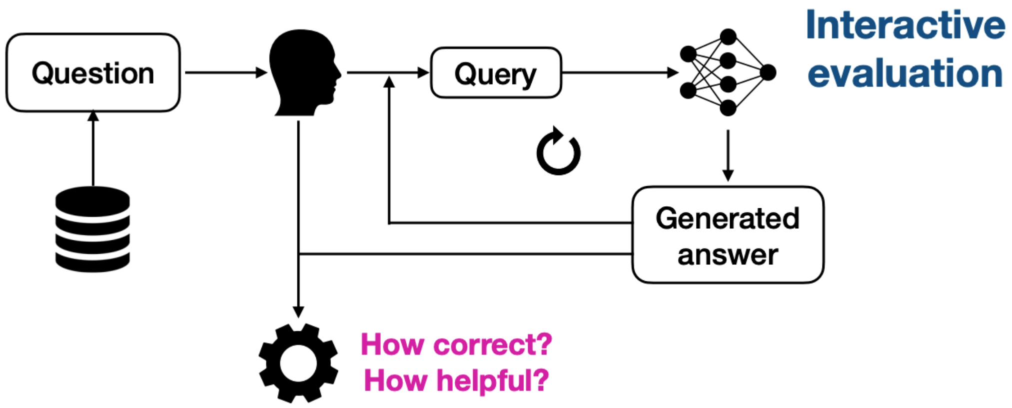
[Collins et al., "Evaluating Language Models for Mathematics through Interactions", 2023]

Large Language Models (LLMs) for Math



Terence Tao
@tao@mathstodon.xyz

As an experiment, I recently tried consulting #GPT4 on a question I found on #MathOverflow prior to obtaining a solution. The question is at mathoverflow.net/questions/449... and my conversation with GPT-4 is at chat.openai.com/share/53aab67e... . Based on past experience, I knew to not try to ask the #AI to answer the question directly (as this would almost surely lead to nonsense), but instead to have it play the role of a collaborator and offer strategy suggestions. It did end up suggesting eight approaches, one of which (generating functions) being the one that was ultimately successful. In this particular case, I would probably have tried a generating function approach eventually, and had no further need of GPT-4 once I started doing so (relying instead on a lengthy MAPLE worksheet, and some good old-fashioned hand calculations at the blackboard and with pen and paper), but it was slightly helpful nevertheless (I had initially thought of pursuing the asymptotic analysis approach instead to gain intuition, but this turned out to be unnecessary). I also asked an auxiliary question in which GPT-4 pointed out the relevance of Dyck paths (and some related structures), which led to one of my other comments on the OP's question. I decided to share my experience in case it encourages others to perform similar experiments.



LLMs can be useful for theorem proving in Lean

[Collins et al., "Evaluating Language Models for Mathematics through Interactions", 2023]

What are LLMs?

$$y = f(x;\ \theta)$$

- Map the **input string x** to the **output string y**
- y is generated word by word

What are LLMs?

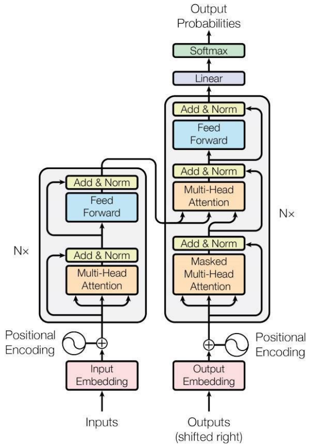
$$y = f(x;\ \theta)$$

- Map the input string x to the output string y
- y is generated word by word
- **The mapping is parameterized by $\boldsymbol{\theta} \in \mathbb{R}^{\boldsymbol{n}}$ with $\boldsymbol{n} \gg \mathbf{1}$**

What are LLMs?

$$y = f(x;\ \theta)$$

- Map the input string x to the output string y
- y is generated word by word
- The mapping is parameterized by $\theta \in \mathbb{R}^n$ with $n \gg 1$
- State-of-the-art LLMs are mostly based on a class of mappings called **Transformer**



[Vaswani et al., 2017]

Training LLMs

Initialize θ randomly

$$y = f(x;\ \theta_0)$$

Training LLMs

How are you?

$$y = f(x;\ \theta_0)$$

D@saf;o;k#fd?H …

Initialize θ randomly

Training LLMs

[Radford et al., "Improving Language Understanding By Generative Pre-training", 2018]

How are you?

$y = f(x;\ \theta_0)$

D@saf;o;k#fd?H …

Pretrain on Internet texts

$y = f(x;\ \theta_1)$

Training LLMs

How are you?

$y = f(x;\ \theta_0)$

D@saf;o;k#fd?H …

Pretrain on Internet texts

How are you?

$y = f(x;\ \theta_1)$

I am a senior web developer with extensive experience in building high quality sites. I want to work with you for a long time. Please send me a message so that we can discuss more. Best regards.

Training LLMs

[Ouyang et al., "Training Language Models To Follow Instructions With Human Feedback", 2022]

How are you?

$y = f(x;\ \theta_0)$

D@saf;o;k#fd?H …

Pretrain on Internet texts

How are you?

$y = f(x;\ \theta_1)$

I am a senior web developer with extensive experience in building high quality sites. I want to work with you for a long time. Please send me a message so that we can discuss more. Best regards.

Align to humans (optional)

$y = f(x; \theta_2)$

Training LLMs

How are you?

$y = f(x;\ \theta_0)$

D@saf;o;k#fd?H …

Pretrain on Internet texts

How are you?

$y = f(x;\ \theta_1)$

I am a senior web developer with extensive experience in building high quality sites. I want to work with you for a long time. Please send me a message so that we can discuss more. Best regards.

Align to humans (optional)

How are you?

$y = f(x; \theta_2)$

Hello! I don't have feelings, but I'm here and ready to assist you with any information or help you may need. How can I assist you today?

Training LLMs

How are you?

$y = f(x;\ \theta_0)$

D@saf;o;k#fd?H …

Pretrain on Internet texts

How are you?

$y = f(x;\ \theta_1)$

I am a senior web developer with extensive experience in building high quality sites. I want to work with you for a long time. Please send me a message so that we can discuss more. Best regards.

Align to humans (optional)

How are you?

$y = f(x; \theta_2)$

Hello! I don't have feelings, but I'm here and ready to assist you with any information or help you may need. How can I assist you today?

GPT-4

Claude

Google Bard

Using LLMs: Prompting vs. Finetuning

How are you?

$y = f(x;\ \theta_0)$

D@saf;o;k#fd?H …

Pretrain on Internet texts

How are you?

$y = f(x;\ \theta_1)$

I am a senior web developer with extensive experience in building high quality sites. I want to work with you for a long time. Please send me a message so that we can discuss more. Best regards.

Align to humans (optional)

$y = f(x; \theta_2)$

Using LLMs: Prompting vs. Finetuning

How are you?

$y = f(x;\ \theta_0)$

D@saf;o;k#fd?H …

Pretrain on Internet texts

How are you?

$y = f(x;\ \theta_1)$

I am a senior web developer with extensive experience in building high quality sites. I want to work with you for a long time. Please send me a message so that we can discuss more. Best regards.

Align to humans (optional)

Please classify the sentiment in this product review by replying either "positive" or "negative": ``` This tent was missing its stakes, tarp, and fly cover. I had to cover it in leaves.```

$y = f(x; \theta_2)$

negative

Using LLMs: Prompting vs. Finetuning

How are you?

$y = f(x;\ \theta_0)$

D@saf;o;k#fd?H …

Pretrain on Internet texts

How are you?

$y = f(x;\ \theta_1)$

I am a senior web developer with extensive experience in building high quality sites. I want to work with you for a long time. Please send me a message so that we can discuss more. Best regards.

Align to humans (optional)

How are you?

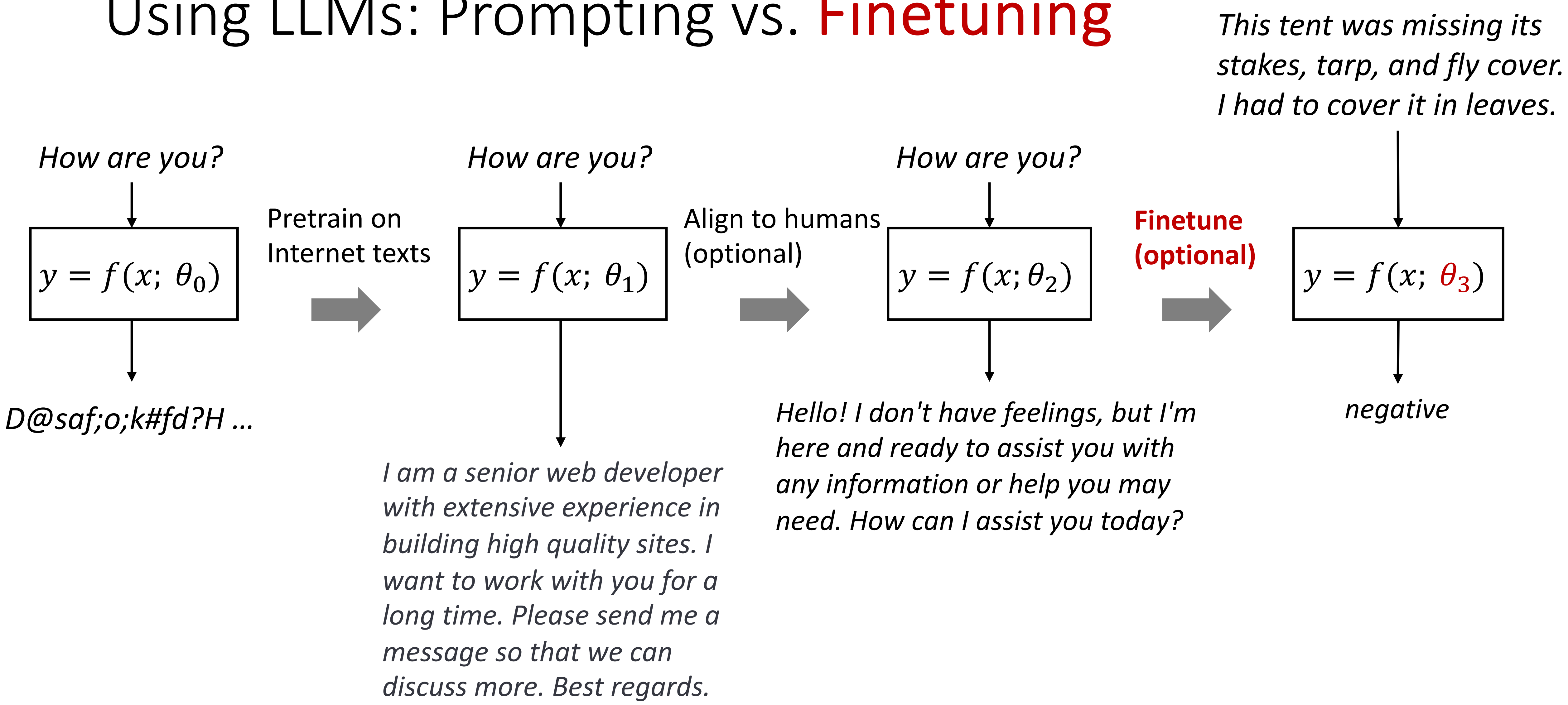
$y = f(x; \theta_2)$

Hello! I don't have feelings, but I'm here and ready to assist you with any information or help you may need. How can I assist you today?

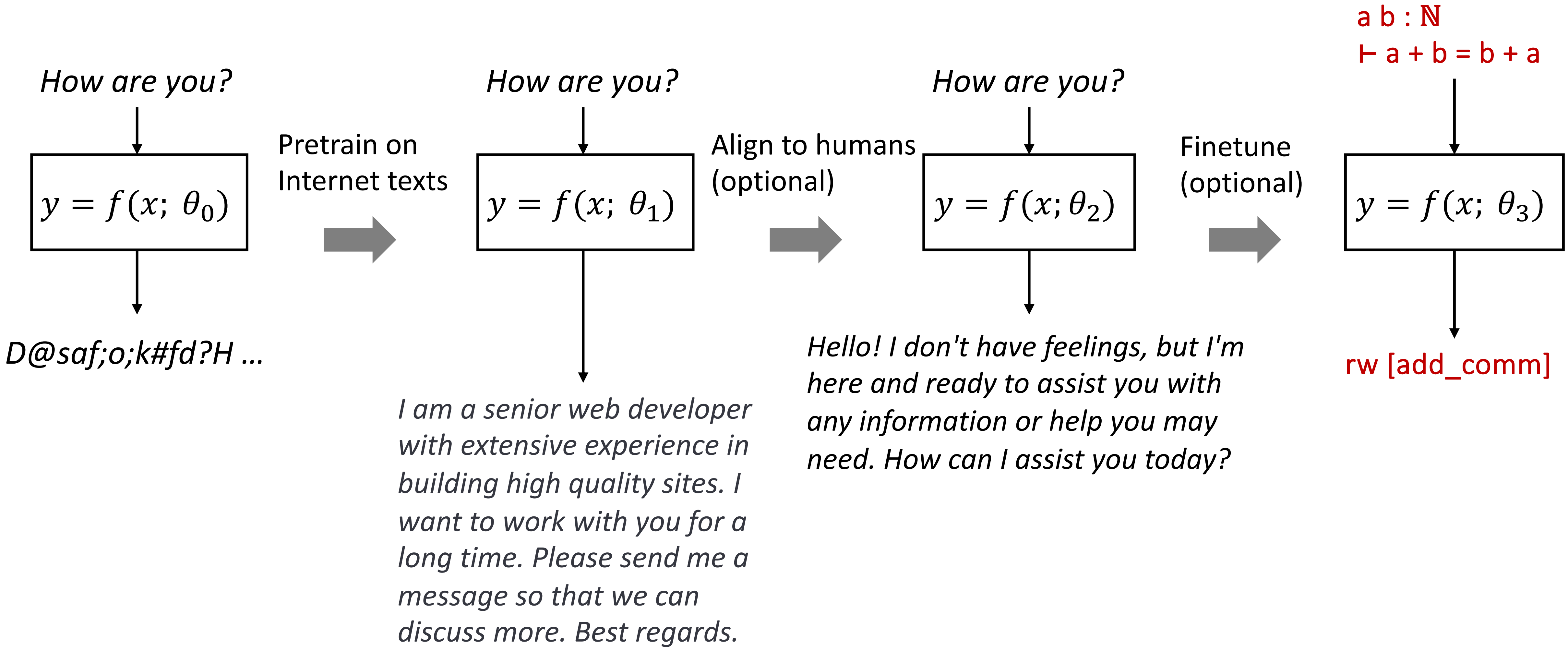
Finetune (optional)

$y = f(x;\ \theta_3)$

Using LLMs: Prompting vs. Finetuning



Using LLMs: Prompting vs. Finetuning

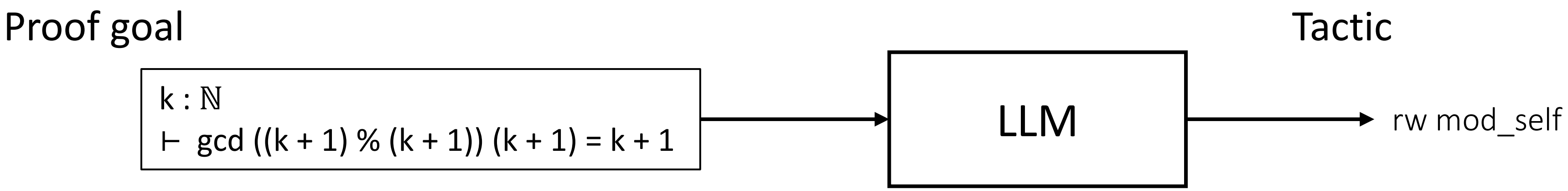


LLMs for Theorem Proving

Using LLMs to Generate Tactics

[Polu and Sutskever, "Generative Language Modeling for Automated Theorem Proving", 2020]

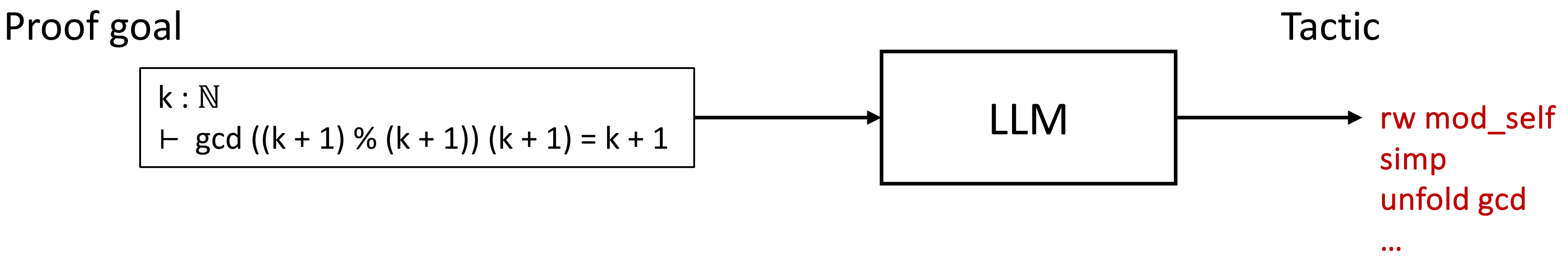
- Training
  1. Pretrain on generic texts from the Internet
  2. Optional: Pretrain on domain-specific texts, e.g., MathOverflow and GitHub
  3. Finetune on (goal, tactic) pairs from formal math libraries, e.g., AFP or mathlib



Using LLMs to Generate Tactics

[Polu and Sutskever, "Generative Language Modeling for Automated Theorem Proving", 2020]

- Training
  1. Pretrain on generic texts from the Internet
  2. Optional: Pretrain on domain-specific texts, e.g., MathOverflow and GitHub
  3. Finetune on (goal, tactic) pairs from formal math libraries, e.g., AFP or mathlib
- Testing
  - Sample multiple tactic suggestions at each step and search for proofs
  - Evaluate on % of theorems proved under a fixed compute budget



[Han et al., "Proof Artifact Co-training for Theorem Proving with Language Models", 2022]

Proof Artifact Co-training

- LLMs are data-hungry, but human-written proofs are limited (~100K proofs in mathlib)
- 9 auxiliary tasks
  - **Next lemma prediction**: Proof goal -> the next lemma to be applied
  - **Type prediction**: Partial proof term -> its type
  - **Theorem naming**: theorem statement -> its name
  - …

Proof Artifact Co-training

- LLMs are data-hungry, but human-written proofs are limited (~100K proofs in mathlib)
- 9 auxiliary tasks
  - **Next lemma prediction**: Proof goal -> the next lemma to be applied
  - **Type prediction**: Partial proof term -> its type
  - **Theorem naming**: theorem statement -> its name
  - …

Model	Tokens elapsed	`mix1`	`mix2`	`tactic`	Pass-rate
Baselines					
`refl`					1.1%
`tidy-bfs`					9.9%
`WebMath > tactic`	1B			1.02	32.2%
Co-training (PACT)					
`WebMath > mix1 + tactic`	18B	*0.08*		0.94	40.0%
`WebMath > mix2 + tactic`	75B		*0.09*	0.93	46.0%
`WebMath > mix1 + mix2 + tactic`	71B	*0.09*	*0.09*	**0.91**	**48.4%**

- **Key insight: Training on tactic generation + auxiliary tasks is better than tactic generation alone**

[Zheng et al., "MiniF2F: A Cross-System Benchmark for Formal Olympiad-Level Mathematics", 2022]

MiniF2F Benchmark

- Math olympiads problems from AMC, AIME, IMO, etc.
- 488 theorems (many w/o proof) for evaluation; no training

MiniF2F Benchmark

- Math olympiads problems from AMC, AIME, IMO, etc.
- 488 theorems (many w/o proof) for evaluation; no training
- Open problems:
  - How to formalize problems asking for numerical answers?
  - How to deal with geometry?

Informal

Solve for a: $\sqrt{4+\sqrt{16+16a}}+\sqrt{1+\sqrt{1+a}}=6$. Show that it is 8.

Lean

```
theorem mathd_algebra_17
  (a : ℝ)
  (h₀ : real.sqrt (4 + real.sqrt (16 + 16 * a)) + real.sqrt (1 + real.sqrt (1 + a))
  a = 8 :=
begin
  sorry
end
```

[Polu et al., "Formal Mathematics Statement Curriculum Learning", 2023]

OpenAI

Solving (some) formal math olympiad problems

Expert Iteration

- Specialized domains without sufficient existing proofs for training, e.g., MiniF2F
- LLMs perform badly on out-of-domain data

[Polu et al., "Formal Mathematics Statement Curriculum Learning", 2023]

OpenAI M

Expert Iteration

Solving (some) formal math olympiad problems

- Specialized domains without sufficient existing proofs for training, e.g., MiniF2F
- LLMs perform badly on out-of-domain data
- **Solution: Iteratively improve the prover on the new domain**
  1. Train the prover
  2. Use the prover to find new proofs
  3. Add new proofs to the training data and go back to step 1

[Polu et al., "Formal Mathematics Statement Curriculum Learning", 2023]

OpenAI

Solving (some) formal math olympiad problems

Expert Iteration

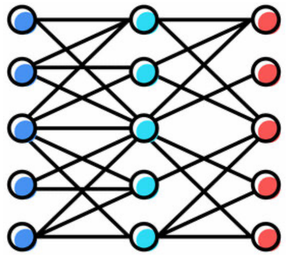
- Specialized domains without sufficient existing proofs for training, e.g., MiniF2F
- LLMs perform badly on out-of-domain data
- **Solution: Iteratively improve the prover on the new domain**
    1. Train the prover
    2. Use the prover to find new proofs
    3. Add new proofs to the training data and go back to step 1

Model	d	e	*pass@1*	*pass@8*
mathlib-valid				
PACT	512	16	48.4%	
θ_0^*	512	16	48.5%	57.6%
θ_0	512	8	46.7%	57.5%
θ_1	512	8	**56.3%**	**66.3%**

Model	d	e	*pass@1*	*pass@8*
miniF2F-valid				
miniF2F	128	16	23.9%	29.3%
θ_0^*	128	16	27.6%	31.8%
θ_0	512	8	28.4%	33.6%
θ_1	512	8	**28.5%**	**35.5%**

Pipeline for Learning-based Theorem Proving

- Learning-based provers are complicated



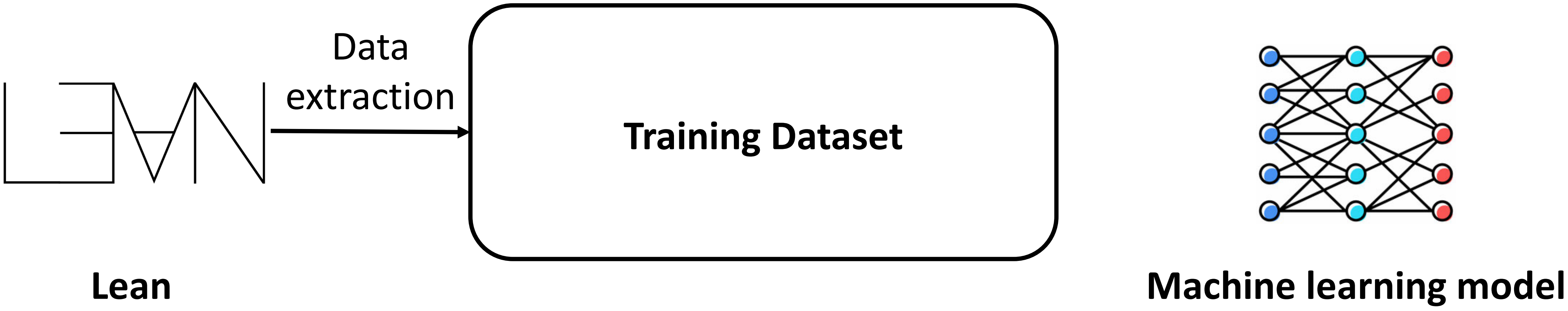
Lean

Machine learning model

[Yang et al., "LeanDojo: Theorem Proving with Retrieval-Augmented Language Models", 2023]

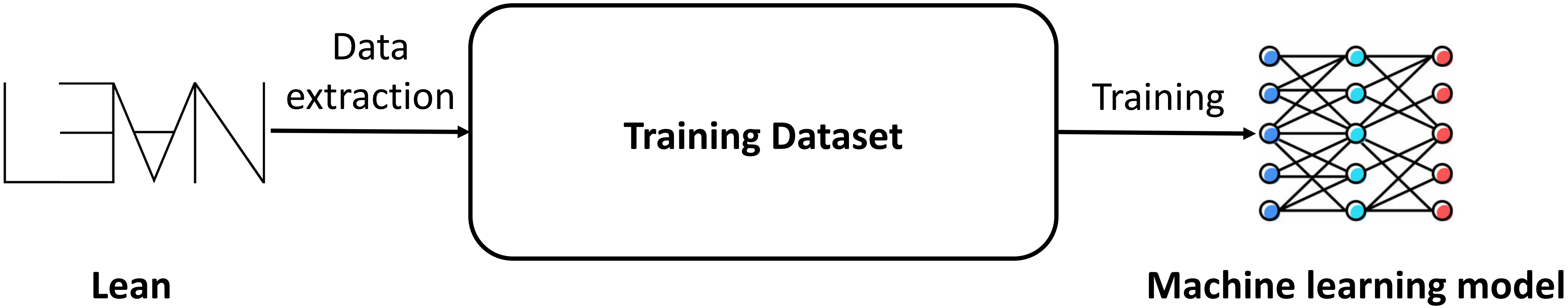
Pipeline for Learning-based Theorem Proving

- Learning-based provers are complicated



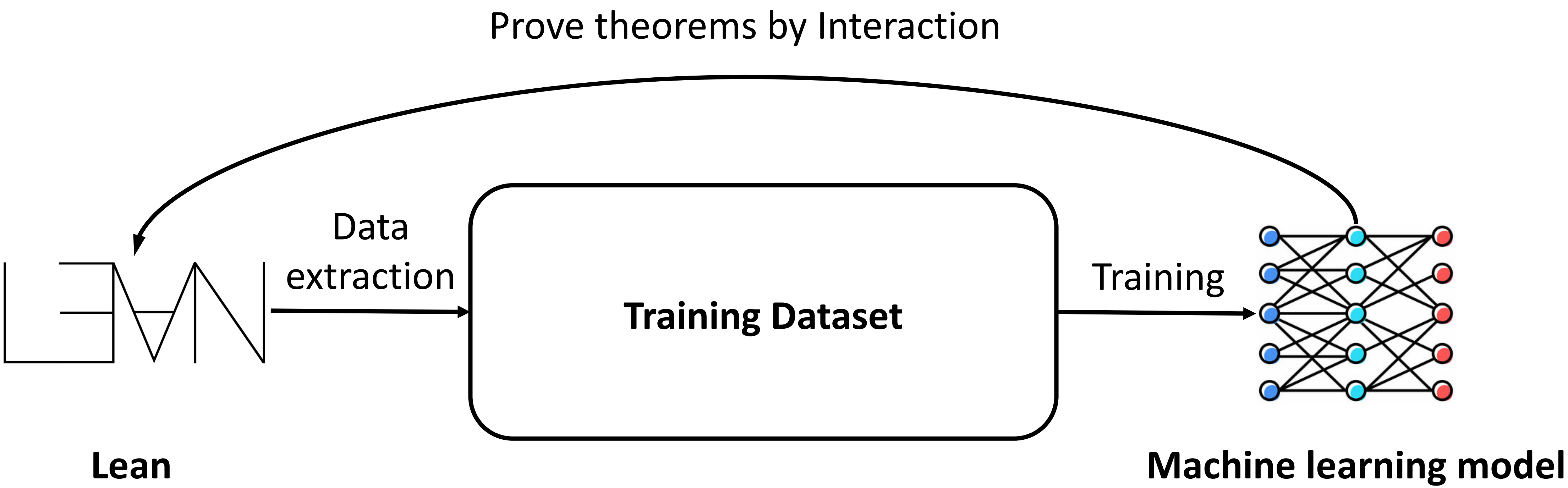
Pipeline for Learning-based Theorem Proving

- Learning-based provers are complicated



Pipeline for Learning-based Theorem Proving

- Learning-based provers are complicated



[Yang et al., "LeanDojo: Theorem Proving with Retrieval-Augmented Language Models", 2023]

Breaking the Barriers in LLMs for Theorem Proving

Jiang et al., LISA, 2021
Jiang et al., Thor, 2022
First et al., Baldur, 2023
Polu and Sutskever, GPT-f, 2020
Han et al., PACT, 2022
Polu et al., 2023
Lample et al., HTPS 2022
Wang et al., DT-Solver, 2023

[Yang et al., "LeanDojo: Theorem Proving with Retrieval-Augmented Language Models", 2023]

Breaking the Barriers in LLMs for Theorem Proving

	Dataset available
Jiang et al., LISA, 2021	✓
Jiang et al., Thor, 2022	✓
First et al., Baldur, 2023	✗
Polu and Sutskever, GPT-f, 2020	✗
Han et al., PACT, 2022	✗
Polu et al., 2023	✗
Lample et al., HTPS 2022	✗
Wang et al., DT-Solver, 2023	✓

Breaking the Barriers in LLMs for Theorem Proving

	Dataset available	Model available	Code available
Jiang et al., LISA, 2021	✓	✗	✗
Jiang et al., Thor, 2022	✓	✗	✗
First et al., Baldur, 2023	✓✗	✗	✗
Polu and Sutskever, GPT-f, 2020	✗	✗	✗
Han et al., PACT, 2022	✓✗	✗	✗
Polu et al., 2023	✓✗	✗	✗
Lample et al., HTPS 2022	✓✗	✗	✗
Wang et al., DT-Solver, 2023	✓	✗	✗

[Yang et al., "LeanDojo: Theorem Proving with Retrieval-Augmented Language Models", 2023]

Breaking the Barriers in LLMs for Theorem Proving

	Dataset available	Model available	Code available	Interaction tool available
Jiang et al., LISA, 2021	✓	✗	✗	✓
Jiang et al., Thor, 2022	✓	✗	✗	✓
First et al., Baldur, 2023	✓✗	✗	✗	✓
Polu and Sutskever, GPT-f, 2020	✗	✗	✗	✗
Han et al., PACT, 2022	✓✗	✗	✗	✓
Polu et al., 2023	✓✗	✗	✗	✓
Lample et al., HTPS 2022	✓✗	✗	✗	✗
Wang et al., DT-Solver, 2023	✓	✗	✗	✗

Breaking the Barriers in LLMs for Theorem Proving

	Dataset available	Model available	Code available	Interaction tool available	Model size (# params)	Compute (hours)
Jiang et al., LISA, 2021	✓	✗	✗	✓	163M	-
Jiang et al., Thor, 2022	✓	✗	✗	✓	700M	1K on TPU
First et al., Baldur, 2023	✓/✗	✗	✗	✓	62,000M	-
Polu and Sutskever, GPT-f, 2020	✗	✗	✗	✗	774M	40K on GPU
Han et al., PACT, 2022	✓/✗	✗	✗	✓	837M	1.5K on GPU
Polu et al., 2023	✓/✗	✗	✗	✓	774M	48K on GPU
Lample et al., HTPS 2022	✓/✗	✗	✗	✗	600M	34K on GPU
Wang et al., DT-Solver, 2023	✓	✗	✗	✗	774M	1K on GPU

[Yang et al., "LeanDojo: Theorem Proving with Retrieval-Augmented Language Models", 2023]

Breaking the Barriers in LLMs for Theorem Proving

	Dataset available	Model available	Code available	Interaction tool available	Model size (# params)	Compute (hours)
Jiang et al., LISA, 2021	✓	✗	✗	✓	163M	-
Jiang et al., Thor, 2022	✓	✗	✗	✓	700M	1K on TPU
First et al., Baldur, 2023	✓✗	✗	✗	✓	62,000M	-
Polu and Sutskever, GPT-f, 2020	✗	✗	✗	✗	774M	40K on GPU
Han et al., PACT, 2022	✓✗	✗	✗	✓	837M	1.5K on GPU
Polu et al., 2023	✓✗	✗	✗	✓	774M	48K on GPU
Lample et al., HTPS 2022	✓✗	✗	✗	✗	600M	34K on GPU
Wang et al., DT-Solver, 2023	✓	✗	✗	✗	774M	1K on GPU
LeanDojo (ours)	✓	✓	✓	✓	**517M**	**120 on GPU**

[Yang et al., "LeanDojo: Theorem Proving with Retrieval-Augmented Language Models", 2023]

Breaking the Barriers in LLMs for Theorem Proving

	Dataset available	Model available	Code available	Interaction tool available	Model size (# params)	Compute (hours)
Jiang et al., LISA, 2021	✓	✗	✗	✓	163M	-
Jiang et al., Thor, 2022	✓	✗	✗	✓	700M	1K on TPU
First et al., Baldur, 2023	×	✗	✗	✓	62,000M	-
Polu and Sutskever, GPT-f, 2020	✗	✗	✗	✗	774M	40K on GPU
Han et al., PACT, 2022	×	✗	✗	✓	837M	1.5K on GPU
Polu et al., 2023	×	✗	✗	✓	774M	48K on GPU
Lample et al., HTPS 2022	×	✗	✗	✗	600M	34K on GPU
Wang et al., DT-Solver, 2023	✓	✗	✗	✗	774M	1K on GPU
LeanDojo (ours)	✓	✓	✓	✓	**517M**	**120 on GPU**

Give researchers access to state-of-the-art LLM-based provers with modest computational costs

[Yang et al., "LeanDojo: Theorem Proving with Retrieval-Augmented Language Models", 2023]

Retrieval-Augmented Prover

- Existing methods only see the current proof state, w/o knowledge of premises

All ***accessible premises*** in the math library

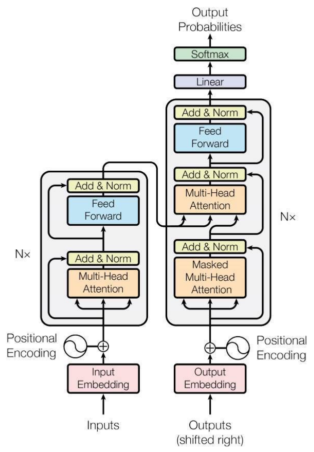
```
theorem mod_self (n : nat) : n % n = 0
theorem gcd_zero_left (x : nat) : gcd 0 x = x

   ⋮              33K on average            ⋮

def gcd : nat → nat → nat                 …
```

State

$k : \mathbb{N}$
$\vdash \text{gcd}\ ((k + 1)\ \%\ (k + 1))\ (k + 1) = k + 1$



[Vaswani et al., NeurIPS 2017]

→ rewrite mod_self

Tactic

Retrieval-Augmented Prover

- Existing methods only see the current proof state, w/o knowledge of premises
- Given a state, we retrieve premises from the set of **all accessible premises**

State

$k : \mathbb{N}$
$\vdash$ gcd ((k + 1) % (k + 1)) (k + 1) = k + 1

All ***accessible premises***
in the math library

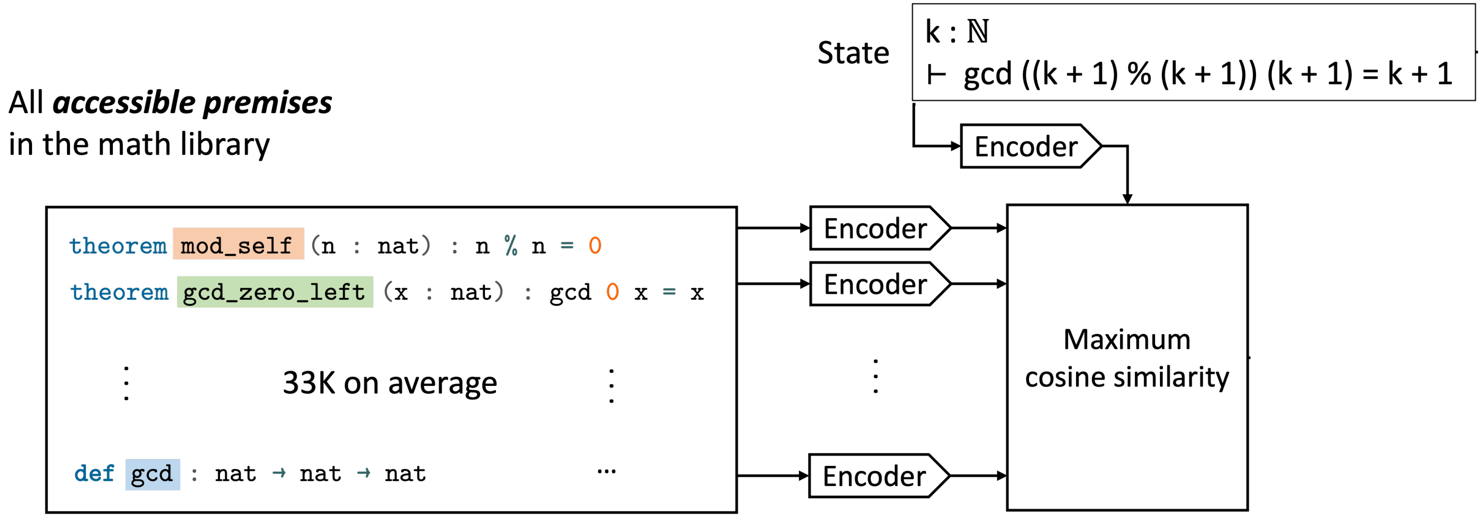
```
theorem mod_self (n : nat) : n % n = 0
theorem gcd_zero_left (x : nat) : gcd 0 x = x

  ⋮            33K on average            ⋮

def gcd : nat → nat → nat               ...
```

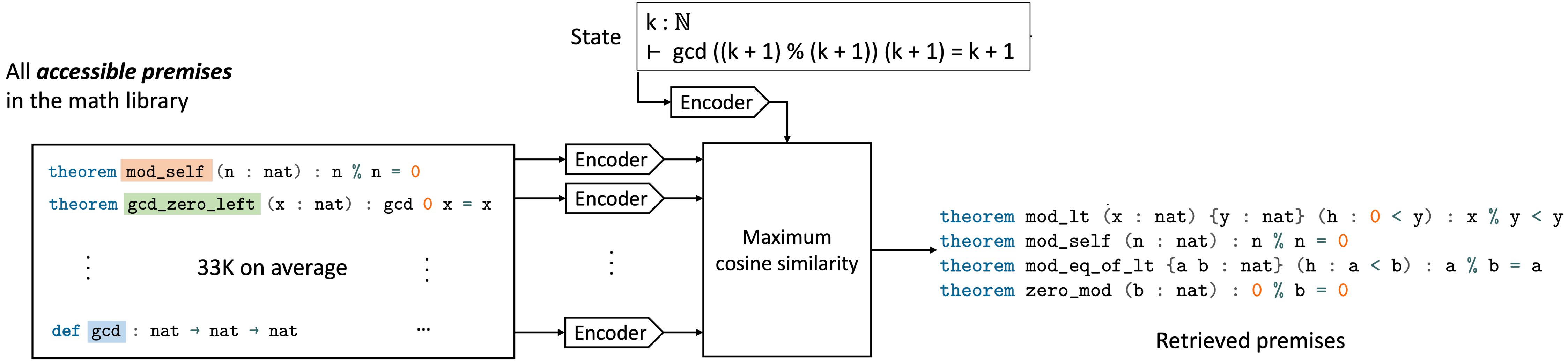
Retrieval-Augmented Prover

- Existing methods only see the current proof state, w/o knowledge of premises
- Given a state, we retrieve premises from the set of **all accessible premises**



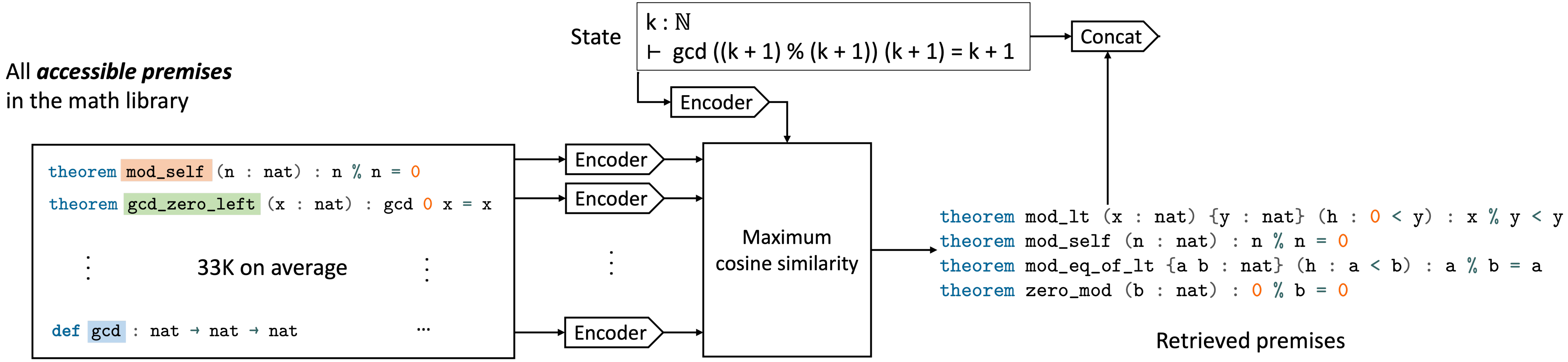
Retrieval-Augmented Prover

- Existing methods only see the current proof state, w/o knowledge of premises
- Given a state, we retrieve premises from the set of **all accessible premises**



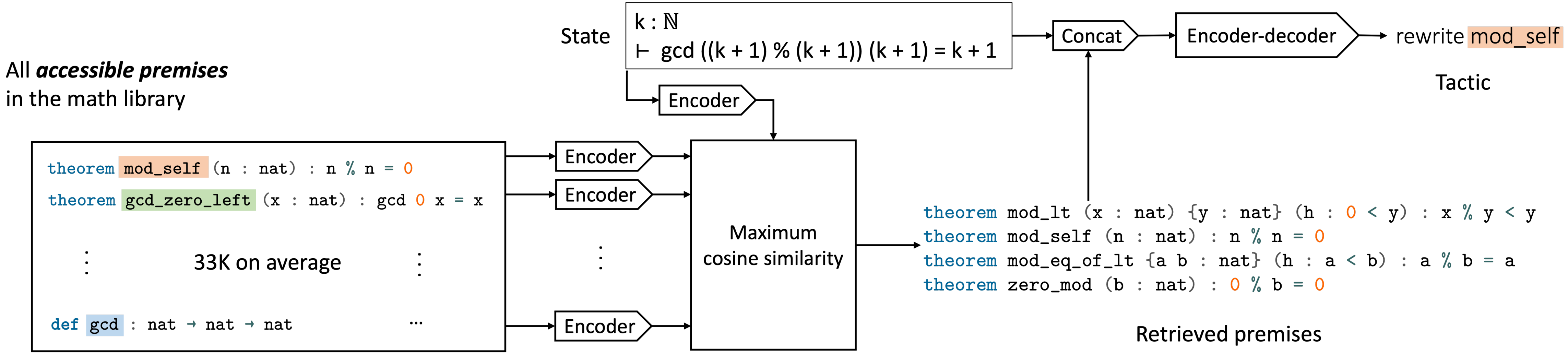
Retrieval-Augmented Prover

- Existing methods only see the current proof state, w/o knowledge of premises
- Given a state, we retrieve premises from the set of **all accessible premises**
- Retrieved premises are concatenated with the state and used for tactic generation

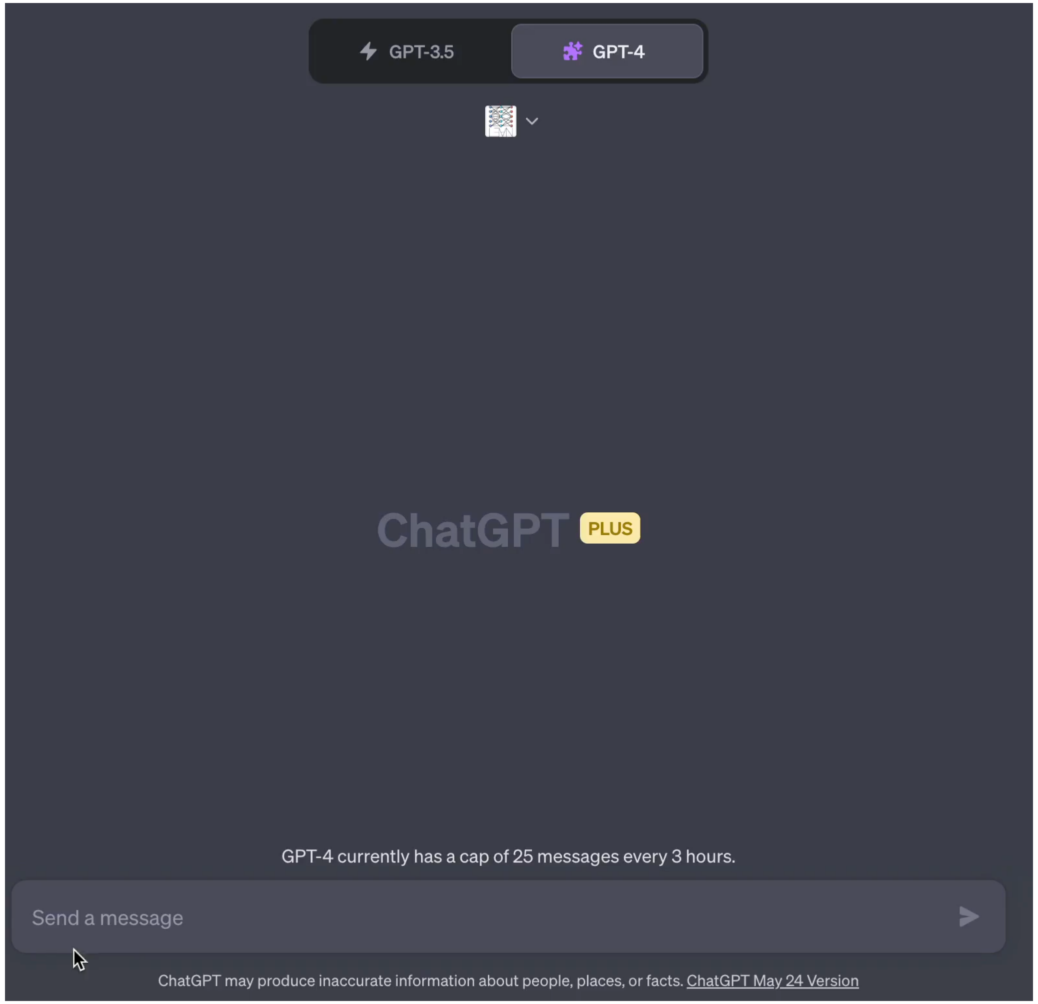


Retrieval-Augmented Prover

- Existing methods only see the current proof state, w/o knowledge of premises
- Given a state, we retrieve premises from the set of **all accessible premises**
- Retrieved premises are concatenated with the state and used for tactic generation



ChatGPT for Theorem Proving in Lean



[Wu et al., "Autoformalization with Large Language Models", 2022]

Autoformalization

- LLMs translate informal math into formal math
- Very useful task, but less well-defined. Hard to evaluate

Case Study 2 Question:
Natural Language version: "*When all the girls at Madeline's school line up in rows of eight, there are seven left over. If instead they line up in rows of four, how many are left over? The final answer is 3.*" Translate the natural language version to an Isabelle version:

PaLM output:

```
theorem
   fixes n::nat
   assumes "n mod 8 = 7"
   shows "n mod 4 = 3"
```

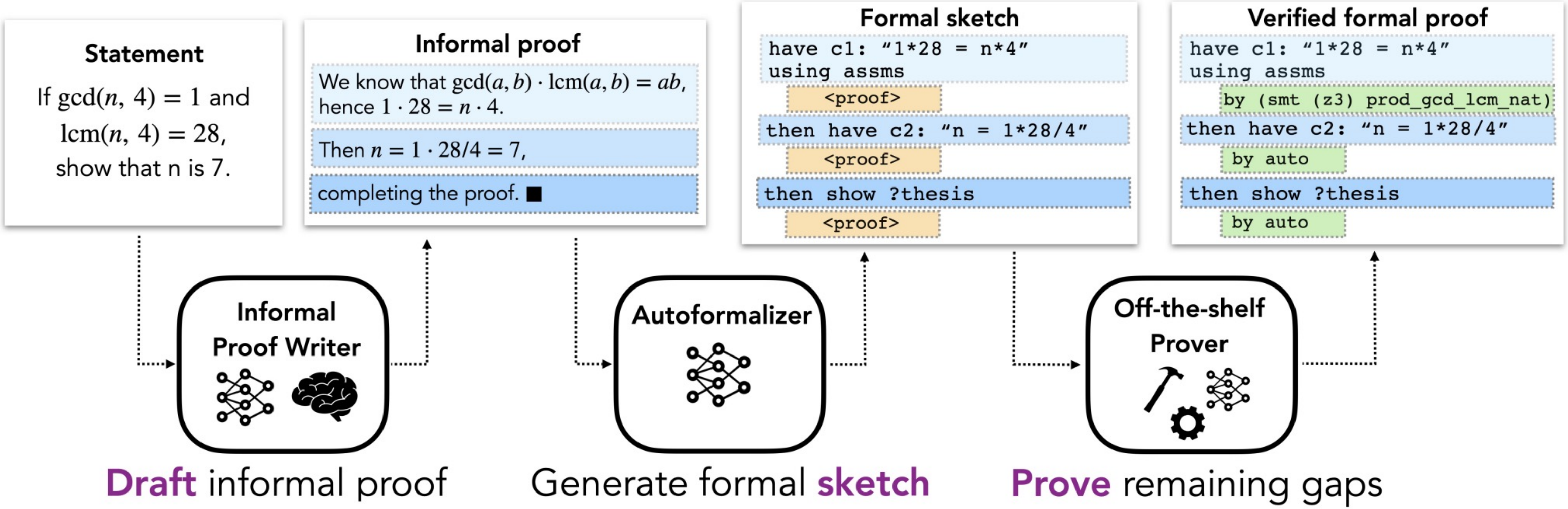
Case Study 3 Question:
Natural language version: "*Let f be a linear function for which $f(6) - f(2) = 12$. What is $f(12) - f(2)$? The final answer is 30.*" Translate the natural language version to an Isabelle version:

Codex output:

```
theorem
   fixes f :: "real \<Rightarrow> real"
   assumes "linear f"
           "f 6 - f 2 = 12"
   shows "f 12 - f 2 = 30"
```

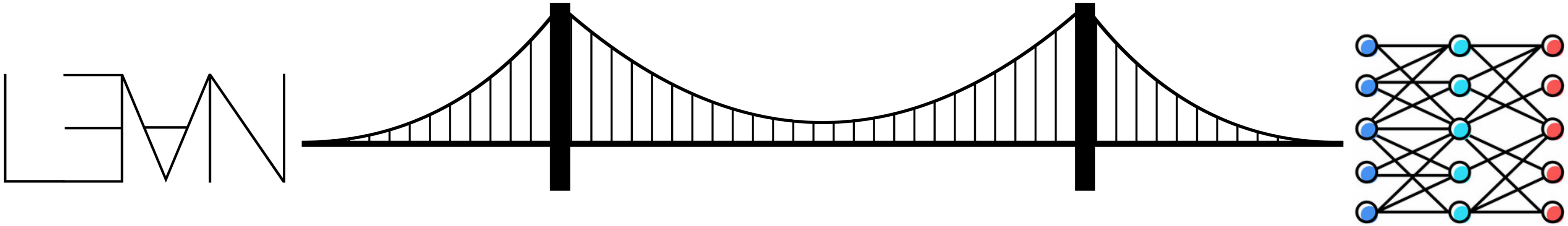
[Jiang et al., "Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs", 2023]

Guiding Formal Provers with Informal Proofs



LLM-based Proof Automation Tools for Lean

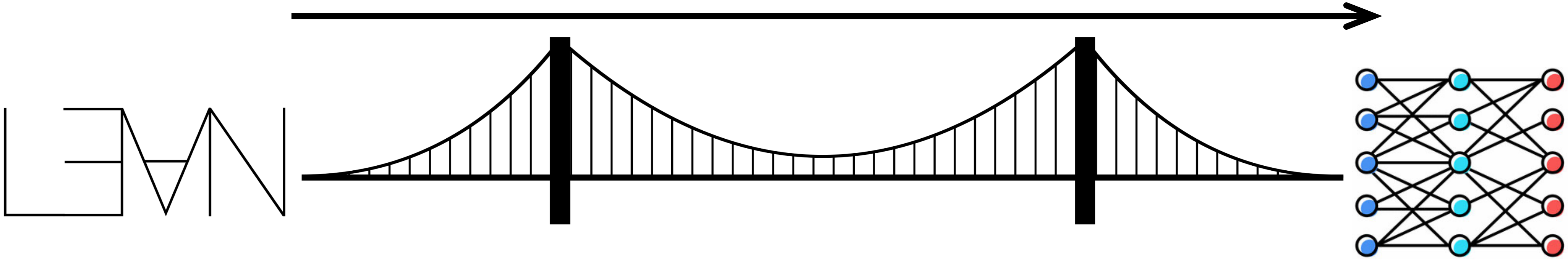
Bridging Machine Learning and Theorem Proving



Lean **Machine learning model**

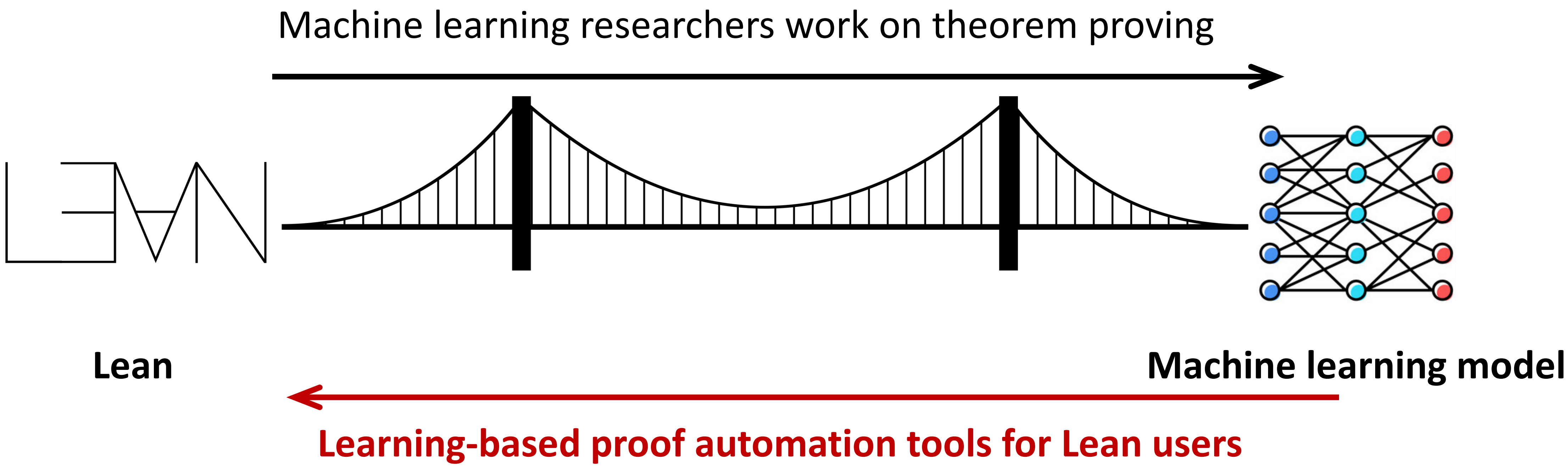
Bridging Machine Learning and Theorem Proving

Machine learning researchers work on theorem proving



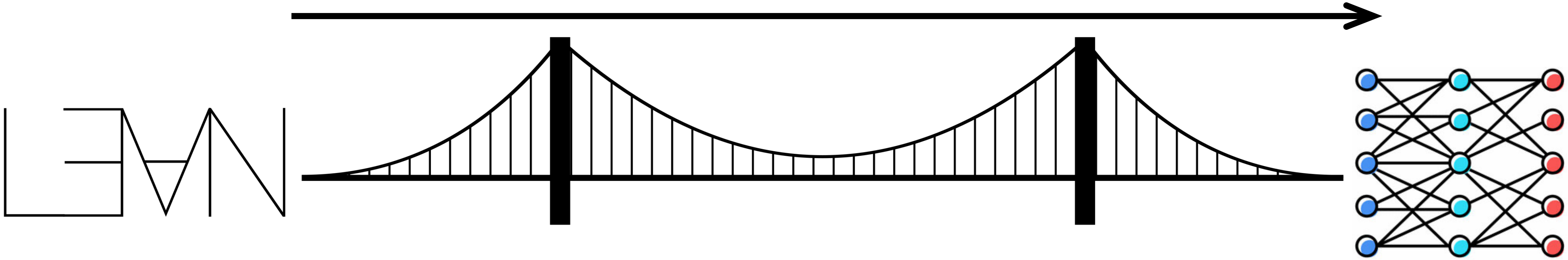
Lean **Machine learning model**

Bridging Machine Learning and Theorem Proving



Bridging Machine Learning and Theorem Proving

Machine learning researchers work on theorem proving



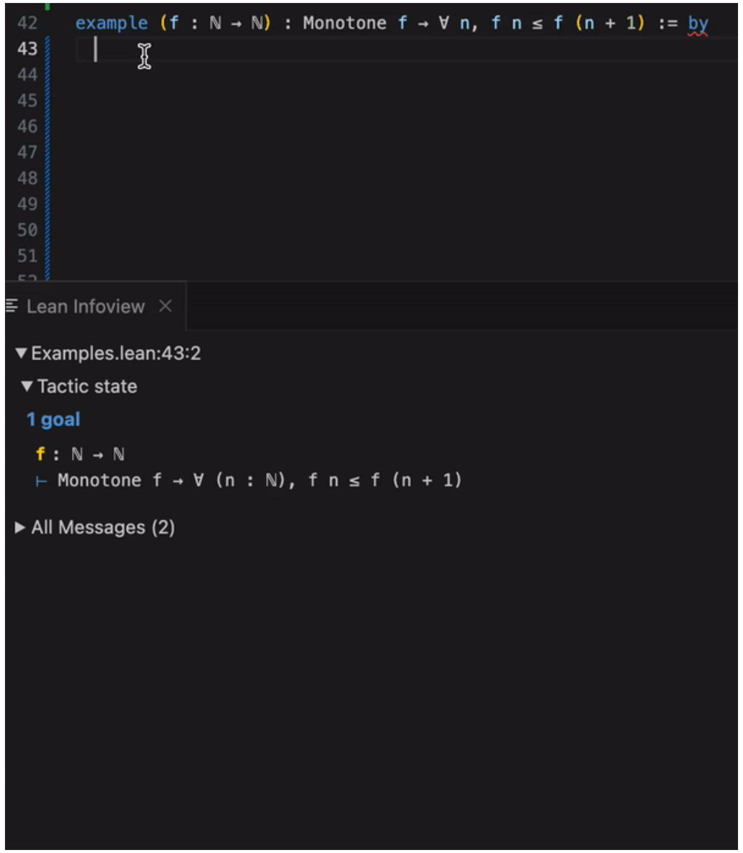
Lean

Machine learning model

Learning-based proof automation tools for Lean users

- Run on CPUs reasonably fast
- Integrated into VSCode
- Care about a specific domain, not aggregated performance on mathlib

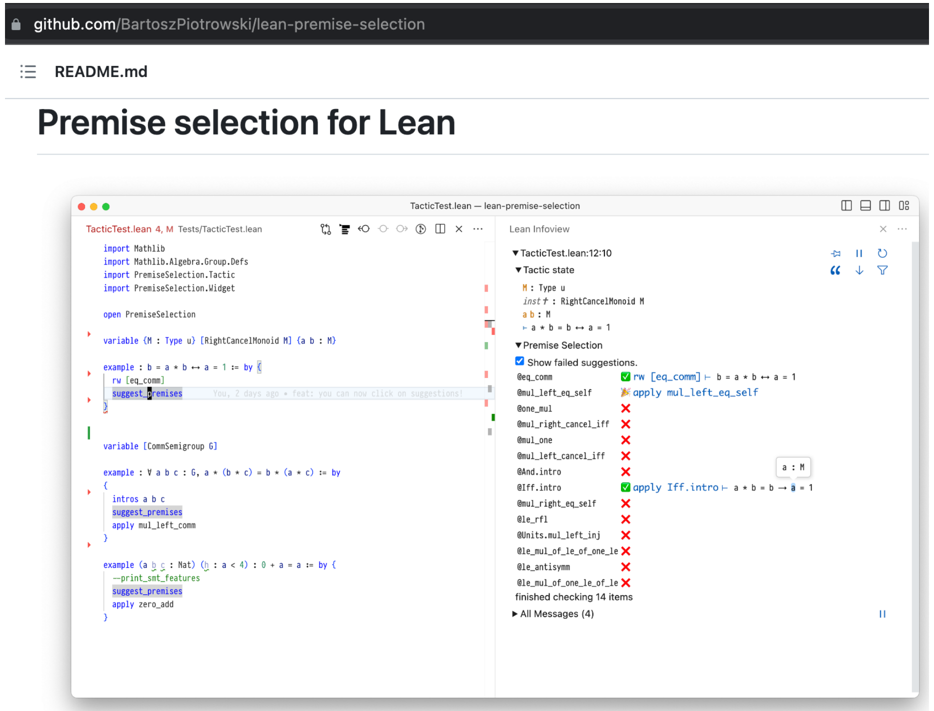
Tools for Tactic Suggestion



[Welleck and Saha, “llmstep: LLM proofstep suggestions in Lean”]
https://github.com/wellecks/llmstep

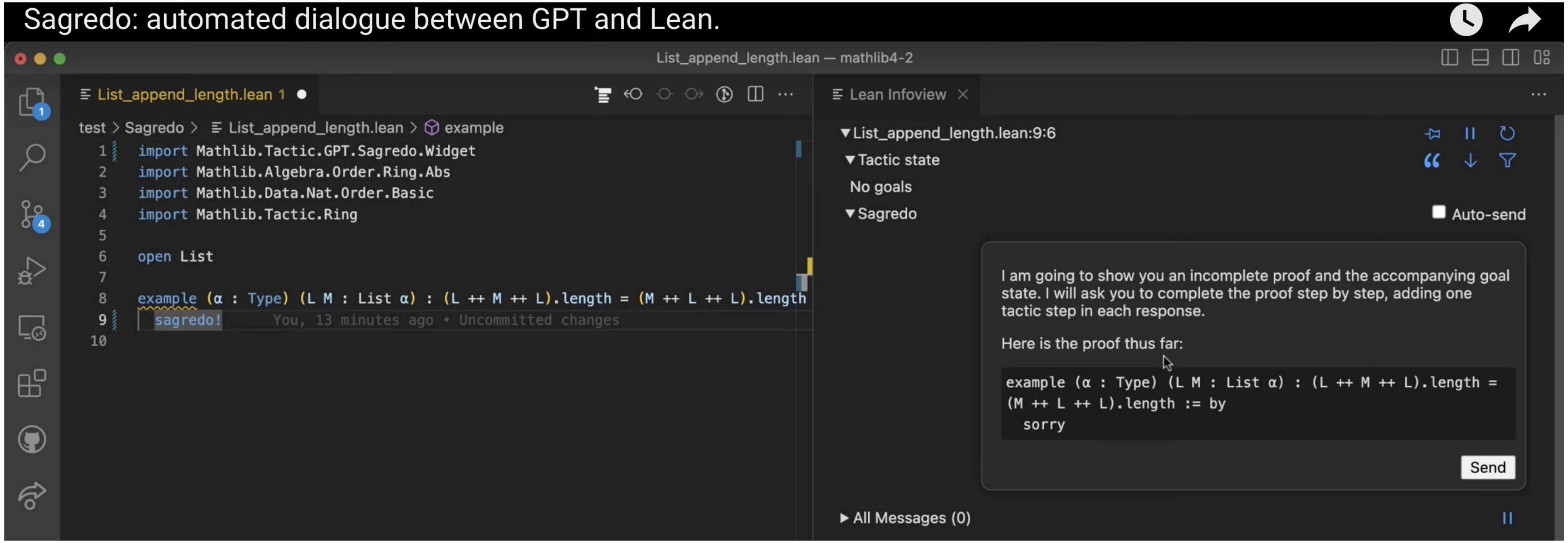
Tools for Premise Selection

- Built-in tactics such as library_search, apply?, exact?



[Piotrowski et al. "Machine-Learned Premise Selection for Lean"]
https://github.com/BartoszPiotrowski/lean-premise-selection

Tools for Interfacing with GPT-4



[Morrison et al., "Sagredo: automated dialogue between GPT and Lean"]
https://www.youtube.com/watch?v=CEwRMT0GpKo

Thank You