Google DeepMind

AlphaProof

When RL meets Formal Maths

Thomas HUBERT, March 2025

Google DeepMind

AlphaProof

当 RL 遇见形式数学

托马斯·HUBERT，2025 年 3 月

Google DeepMind


Mathematics
AlphaProof
数学
AlphaProof

We have used maths to both describe, predict and shape the natural world

我们用数学来描述、预测和塑造自然世界

3

Mathematics, a root node to intelligence?

Reasoning & Planning

Generalisation & Abstraction

Knowledge & Creativity

Open ended & Unbounded complexity

Even requires an eye for beauty...

数学，智能的根节点?

推理与规划

泛化与抽象

知识与创造力

开放式与 Unbounded complexity

甚至需要一双发现美的眼睛 ...

A Brief History of Formalisation in Mathematics

Proofs & Axiomatisation

Importance of proofs discovered by the ancient Greeks

Since then, what constitutes a mathematical proof has evolved.

数学形式化的简史

证明与公理化

古希腊人发现的证明的重要性

从那时起，构成数学证明的内容已经演变。

A Brief History of Formalisation in Mathematics

Proofs & Axiomatisation

Importance of proofs discovered by the ancient Greeks

Since then, what constitutes a mathematical proof has evolved.

From words to symbols

A square and ten times its root are equal to thirty-nine.

$x^2 + 10x = 39$

Take the coefficient of the root, divide it into two parts, and multiply one of them by itself.
Add this to thirty-nine
Take the square root of sixty-four which is eight
Subtract from this one of the halves of the coefficient of the root.
The result is the value of the root which is three.

数学形式化的简史

Proofs & Axiomatisation

古希腊人发现的证明的重要性

从那时起，构成数学证明的内容已经演变。

从文字到符号

一个平方数和它的十倍根相等於三十九。

$x^2 + 10x = 39$

取根的系数，分成两部分，将其中一部分自乘。加上三十九。取六十四的平方根，它是八。从这个数中减去根系数的一半。结果是根的值，它是三。

A Brief History of Formalisation in Mathematics

Proofs & Axiomatisation

Importance of proofs discovered by the ancient Greeks

Since then, what constitutes a mathematical proof has evolved.

From words to symbols

A square and ten times its root are equal to thirty-nine.

$x^2 + 10x = 39$

Take the coefficient of the root, divide it into two parts, and multiply one of them by itself.
Add this to thirty-nine
Take the square root of sixty-four which is eight
Subtract from this one of the halves of the coefficient of the root.
The result is the value of the root which is three.

$-\frac{b}{2} \pm \sqrt{\frac{b}{2}^2 - c}$

数学形式化的简史

Proofs & Axiomatisation

古希腊人发现的证明的重要性

从那时起，构成数学证明的内容已经演变。

从文字到符号

一个平方数和它的十倍根相等於三十九。

$x^2 + 10x = 39$

取根的系数，分成两部分，将其中一部分自乘。加上三十九。取六十四的平方根，它是八。从这个数中减去根系数的一半。结果是根的值，它是三。

$-\frac{b}{2} \pm \sqrt{\frac{b}{2}^2 - c}$

A Brief History of Formalisation in Mathematics

1. Rigor and clarity
2. Efficiency and Communication
3. Abstraction and Generalisation
4. Unification
5. Created new fields

数学形式化的简史

1. 严谨性与清晰性
2. 效率与交流
3. 抽象与泛化
4. 统一性
5. 创造了新的领域

Computer Formalisation of Mathematics

Mathematics in code

```
/-- Euclid's theorem on the **infinitude of primes**.
Here given in the form: for every `n`, there exists a prime number `p ≥
n`. -/
theorem exists_infinite_primes (n : ℕ) : ∃ p, n ≤ p ∧ Prime p :=
  let p := minFac (n ! + 1)
  have f1 : n ! + 1 ≠ 1 := ne_of_gt <| succ_lt_succ <| factorial_pos _
  have pp : Prime p := minFac_prime f1
```

▼ Expected type

```
n : ℕ
p : ℕ := (n ! + 1).minFac
f1 : n ! + 1 ≠ 1
pp : Prime p
⊢ ∃ p, n ≤ p ∧ Prime p
```


数学的计算机形式化

代码中的数学


```
/-- Euclid's theorem on the **infinitude of primes**.
Here given in the form: for every `n`, there exists a prime number `p ≥
n`. -/
theorem exists_infinite_primes (n : ℕ) : ∃ p, n ≤ p ∧ Prime p :=
  let p := minFac (n ! + 1)
  have f1 : n ! + 1 ≠ 1 := ne_of_gt <| succ_lt_succ <| factorial_pos _
  have pp : Prime p := minFac_prime f1
```

▼ Expected type

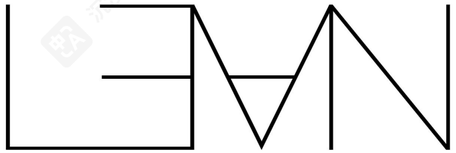
```
n : ℕ
p : ℕ := (n ! + 1).minFac
f1 : n ! + 1 ≠ 1
pp : Prime p
⊢ ∃ p, n ≤ p ∧ Prime p
```

Lean

Lean is a programming language, theorem prover, and interactive proof assistant.

It has a vibrant open source community of mathematicians.

Lean has been used to formalize fields medal mathematics.
A huge quest the community is on it to build the math library of the future.



Programming Language and Theorem Prover

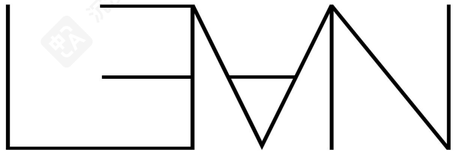
Lean

Lean 是一种编程语言、定理证明器和交互式证明助手。

它拥有一个 充满活力的开源社区的数学家。

Lean 已被用于形式化 菲尔兹奖数学。

社区正在努力构建未来的 数学库。



Programming Language and Theorem Prover

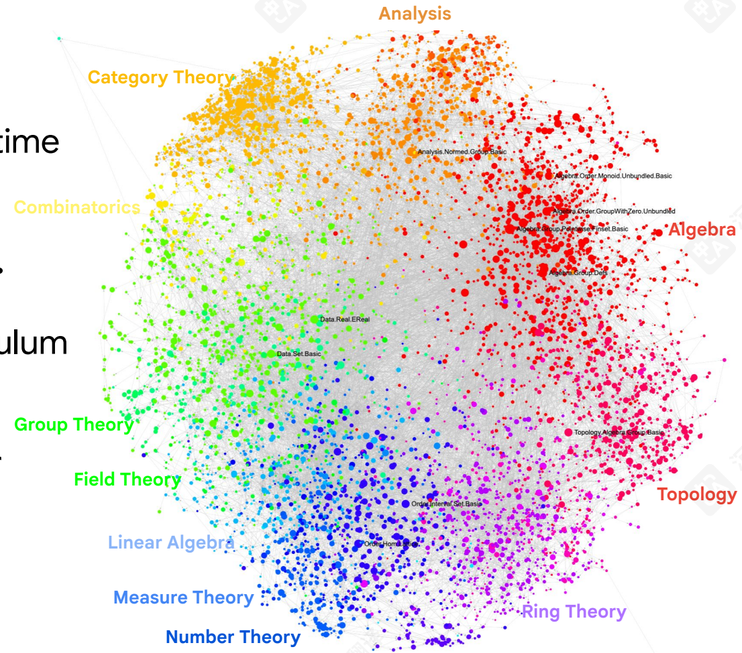
Mathlib

Built **open source** on own free time of mathematicians!

It aims to be **General** & **Unified.**

Covers ~ 80% undergrad curriculum

Most of the library is above undergraduate but has irregular coverage with big holes.



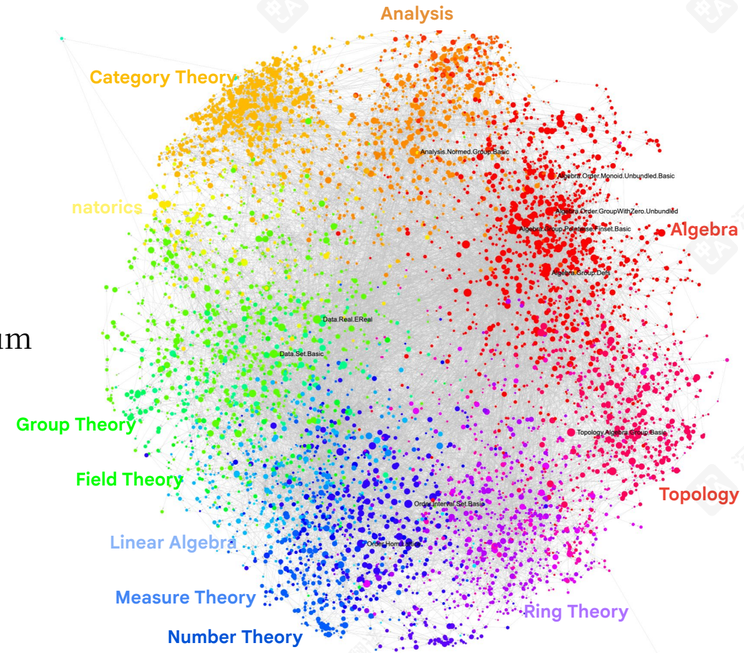
Mathlib

在数学家的业余时间里构建的 **开源** 项目！Co
mbi

其目标旨在 **通用** & **统一**。

Covers ~ 80% undergrad curriculum

该库大部分内容超出本科水平，但覆盖范围不规律，存在较大空白。



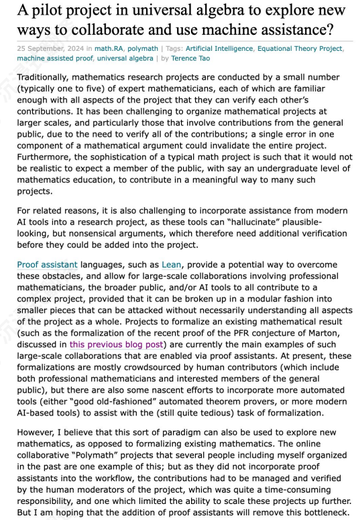
11

Computer Formalisation unlocks enormous synergies

Perfect Verification + Software Stack

Instant Wins
- Correctness concerns disappear
- Giant Proofs can be trusted
- Proof checking can be delegated entirely to the computer

Game Changer
- Transforms mathematics into a video game 🎉!
- For Education
- All the tools of Software Engineering for Maths
- Unlocks Massive Collaboration



A pilot project in universal algebra to explore new ways to collaborate and use machine assistance?

25 September, 2024 in math.RA, polymath | Tags: Artificial Intelligence, Equational Theory Project, machine assisted proof, universal algebra | by Terence Tao

Traditionally, mathematics research projects are conducted by a small number (typically one to five) of expert mathematicians, each of which are familiar enough with all aspects of the project that they can verify each other's contributions. It has been challenging to organize mathematical projects at larger scales, and particularly those that involve contributions from the general public, due to the need to verify all of the contributions; a single error in one component of a mathematical argument could invalidate the entire project. Furthermore, the sophistication of a typical math project is such that it would not be realistic to expect a member of the public, with say an undergraduate level of mathematics education, to contribute in a meaningful way to many such projects.

For related reasons, it is also challenging to incorporate assistance from modern AI tools into a research project, as these tools can "hallucinate" plausible-looking, but nonsensical arguments, which therefore need additional verification before they could be added into the project.

Proof assistant languages, such as Lean, provide a potential way to overcome these obstacles, and allow for large-scale collaborations involving professional mathematicians, the broader public, and/or AI tools to all contribute to a complex project, provided that it can be broken up in a modular fashion into smaller pieces that can be attacked without necessarily understanding all aspects of the project as a whole. Projects to formalize an existing mathematical result (such as the formalization of the recent proof of the PFR conjecture of Marton, discussed in this previous blog post) are currently the main examples of such large-scale collaborations that are enabled via proof assistants. At present, these formalizations are mostly crowdsourced by human contributors (which include both professional mathematicians and interested members of the general public), but there are also some nascent efforts to incorporate more automated tools (either "good old-fashioned" automated theorem provers, or more modern AI-based tools) to assist with the (still quite tedious) task of formalization.

However, I believe that this sort of paradigm can also be used to explore new mathematics, as opposed to formalizing existing mathematics. The online collaborative "Polymath" projects that several people including myself organized in the past are one example of this; but as they did not incorporate proof assistants into the workflow, the contributions had to be managed and verified by the human moderators of the project, which was quite a time-consuming responsibility, and one which limited the ability to scale these projects up further. But I am hoping that the addition of proof assistants will remove this bottleneck.

计算机形式化解锁巨大的协同效应

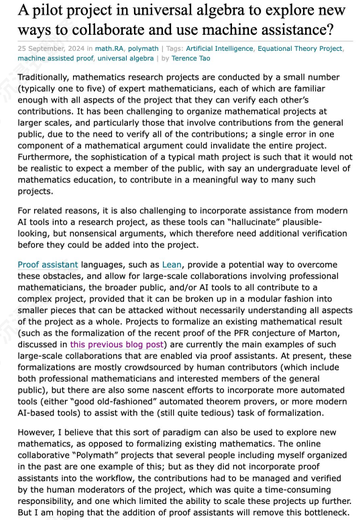
完美验证 + 软件栈

即时胜利 - 正确性

ss 问题消失 - 巨大的证明可以被信任 - 证明检查可以完全委托给计算机

改变游戏规则 - 转换 m

数学变成一个视频游戏 ！ - 用于教育 - 数学所有软件工程的工具 - 解锁大规模协作



A pilot project in universal algebra to explore new ways to collaborate and use machine assistance?

25 September, 2024 in math.RA, polymath | Tags: Artificial Intelligence, Equational Theory Project, machine assisted proof, universal algebra | by Terence Tao

Traditionally, mathematics research projects are conducted by a small number (typically one to five) of expert mathematicians, each of which are familiar enough with all aspects of the project that they can verify each other's contributions. It has been challenging to organize mathematical projects at larger scales, and particularly those that involve contributions from the general public, due to the need to verify all of the contributions; a single error in one component of a mathematical argument could invalidate the entire project. Furthermore, the sophistication of a typical math project is such that it would not be realistic to expect a member of the public, with say an undergraduate level of mathematics education, to contribute in a meaningful way to many such projects.

For related reasons, it is also challenging to incorporate assistance from modern AI tools into a research project, as these tools can "hallucinate" plausible-looking, but nonsensical arguments, which therefore need additional verification before they could be added into the project.

Proof assistant languages, such as Lean, provide a potential way to overcome these obstacles, and allow for large-scale collaborations involving professional mathematicians, the broader public, and/or AI tools to all contribute to a complex project, provided that it can be broken up in a modular fashion into smaller pieces that can be attacked without necessarily understanding all aspects of the project as a whole. Projects to formalize an existing mathematical result (such as the formalization of the recent proof of the PFR conjecture of Marton, discussed in this previous blog post) are currently the main examples of such large-scale collaborations that are enabled via proof assistants. At present, these formalizations are mostly crowdsourced by human contributors (which include both professional mathematicians and interested members of the general public), but there are also some nascent efforts to incorporate more automated tools (either "good old-fashioned" automated theorem provers, or more modern AI-based tools) to assist with the (still quite tedious) task of formalization.

However, I believe that this sort of paradigm can also be used to explore new mathematics, as opposed to formalizing existing mathematics. The online collaborative "Polymath" projects that several people including myself organized in the past are one example of this; but as they did not incorporate proof assistants into the workflow, the contributions had to be managed and verified by the human moderators of the project, which was quite a time-consuming responsibility, and one which limited the ability to scale these projects up further. But I am hoping that the addition of proof assistants will remove this bottleneck.

Challenges for Computer Formalisation

Only 0.1%-1% of mathematicians have adopted Lean.

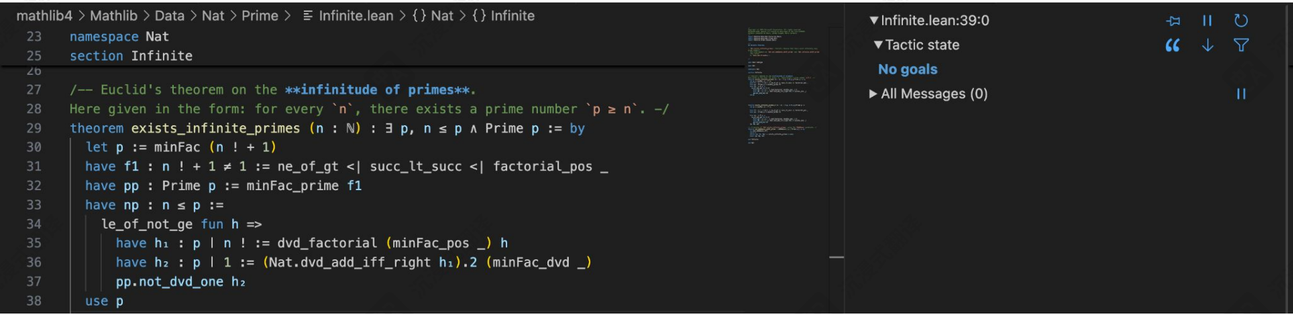
Not yet a good tool for mainstream research
- Steep Learning Curve
- Significant Time Investment
- Tooling and Library maturity (though rapidly growing)
- Perceived lack of necessity for research.

计算机形式化面临的挑战

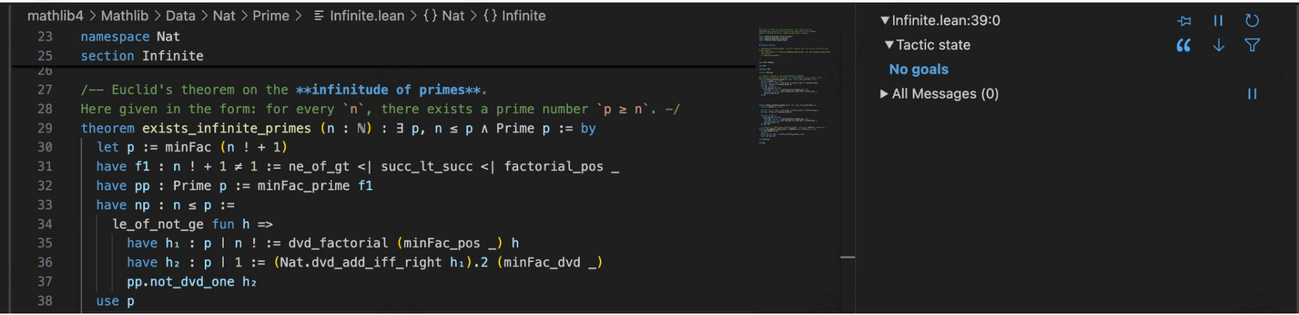
只有 0.1%-1% 的数学家采用了 Lean。

尚未成为主流研究的好工具
- 学习曲线陡峭 - 需要大量时间投入 - 工具和库的成熟度（尽管正在快速成长） - 研究中感知不到必要性。

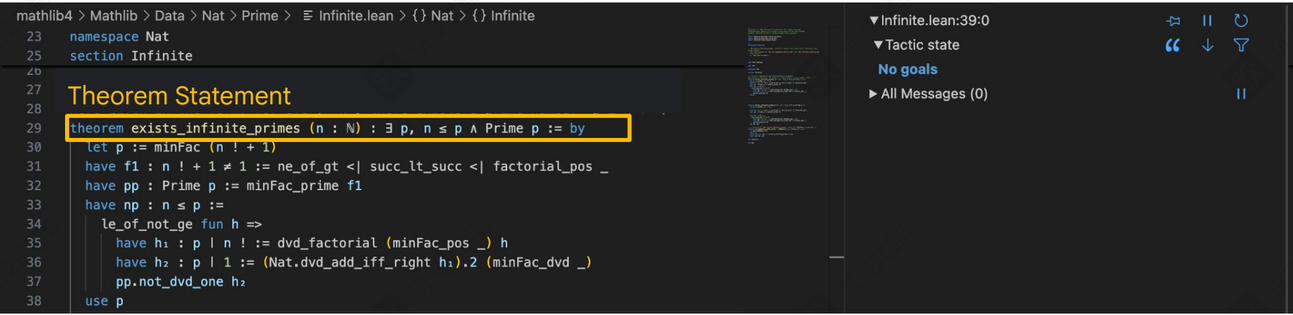
Demo - Infinitude of Primes



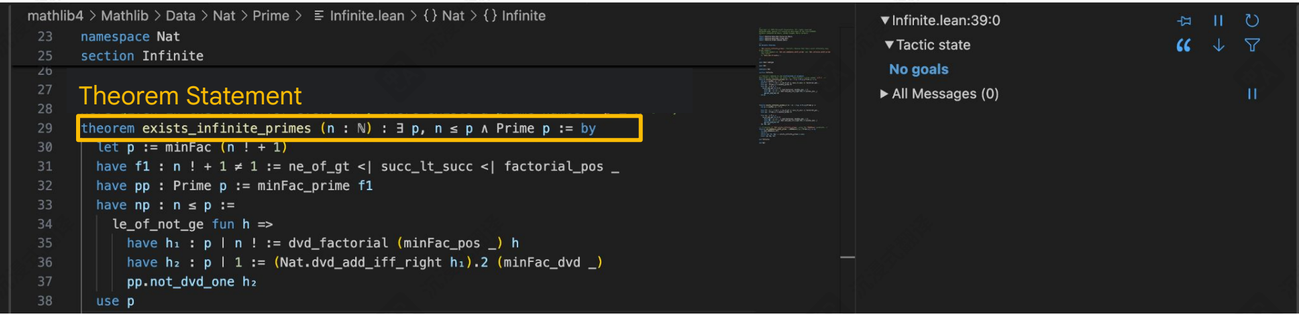
Demo - 无限素数



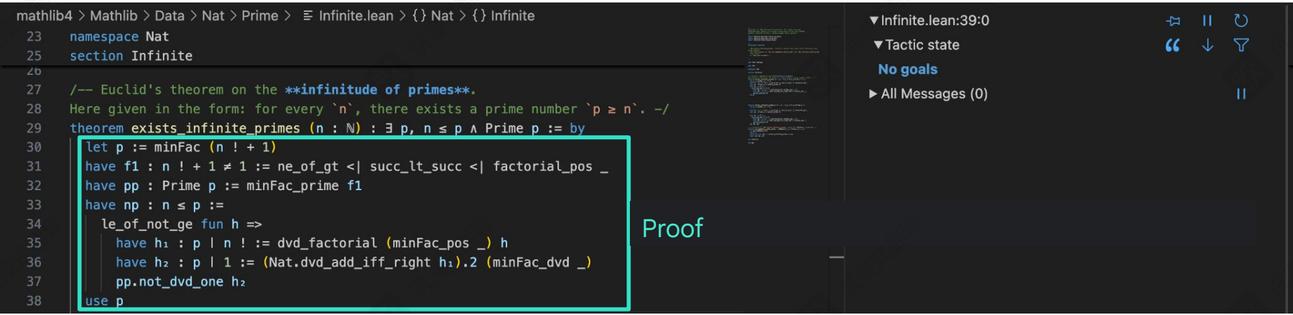
Demo - Infinitude of Primes



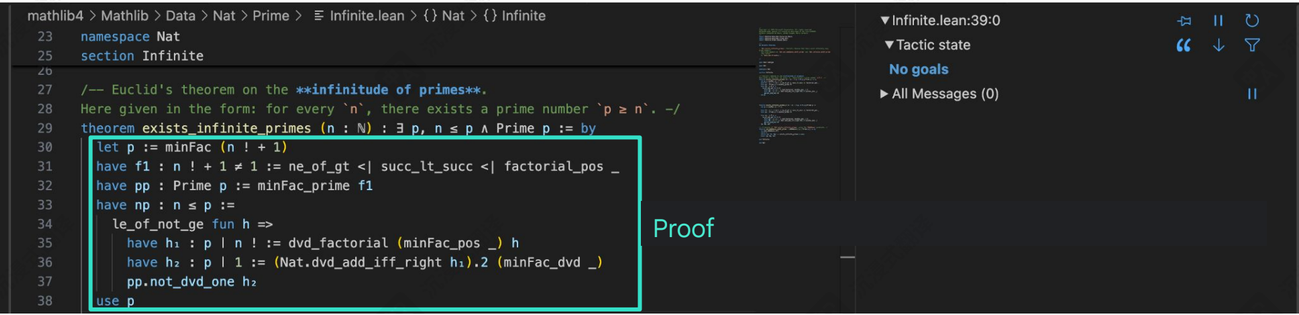
Demo - 无限素数



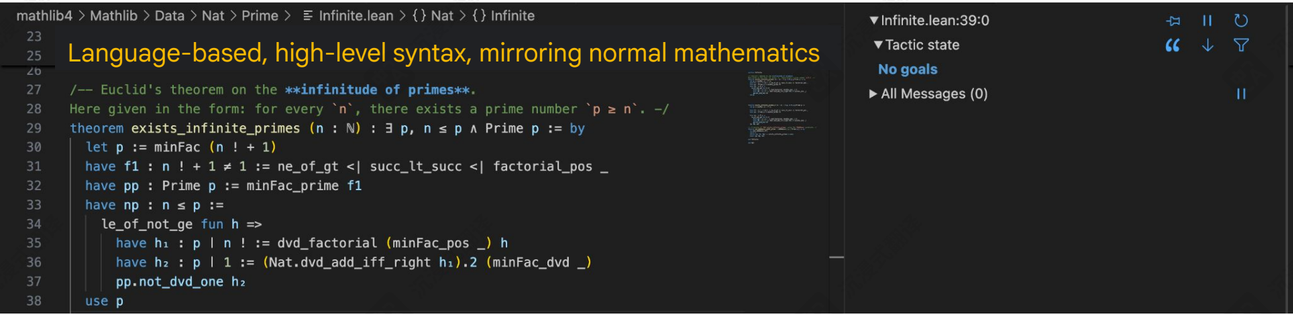
Demo - Infinitude of Primes



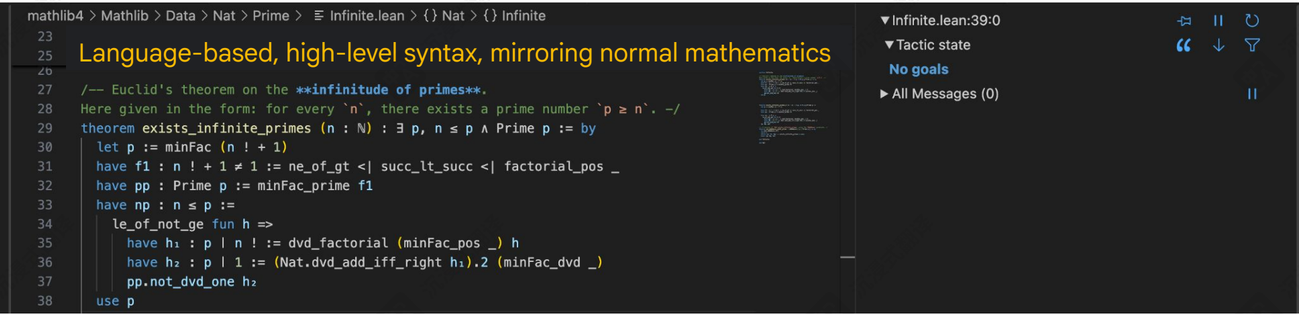
Demo - 无限素数



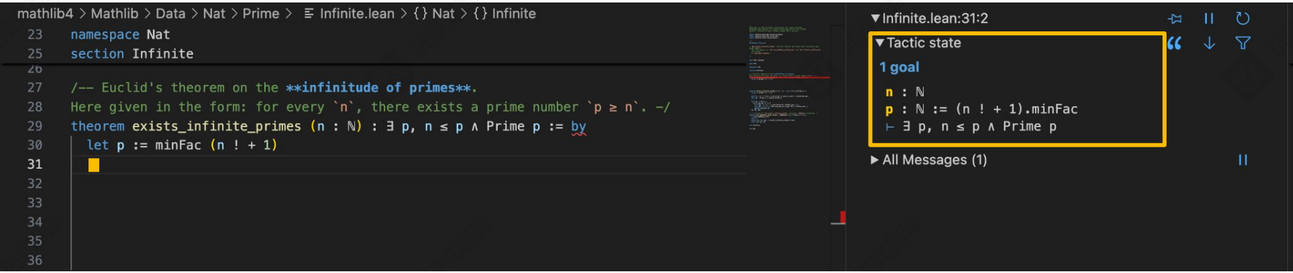
Demo - Infinitude of Primes



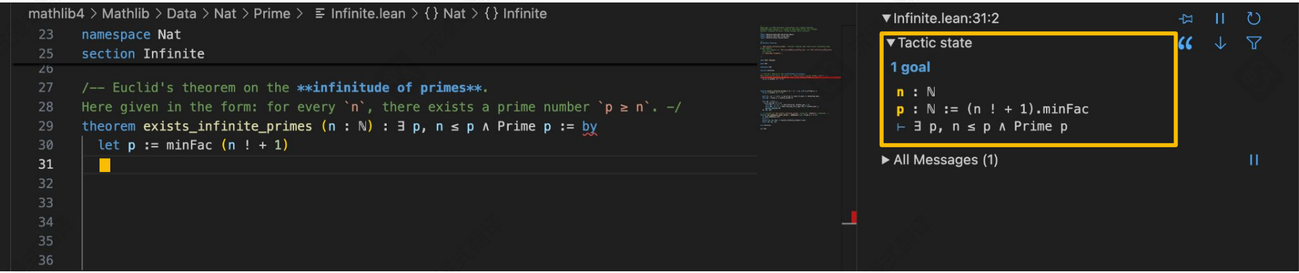
Demo - 无限素数



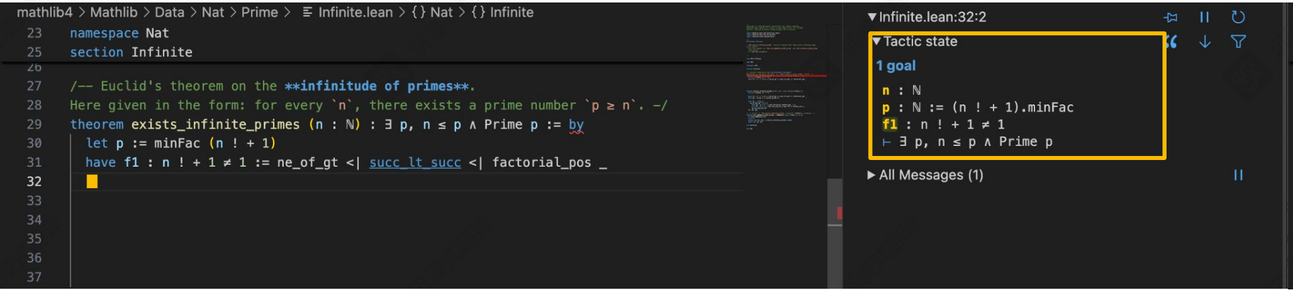
Demo - Infinitude of Primes



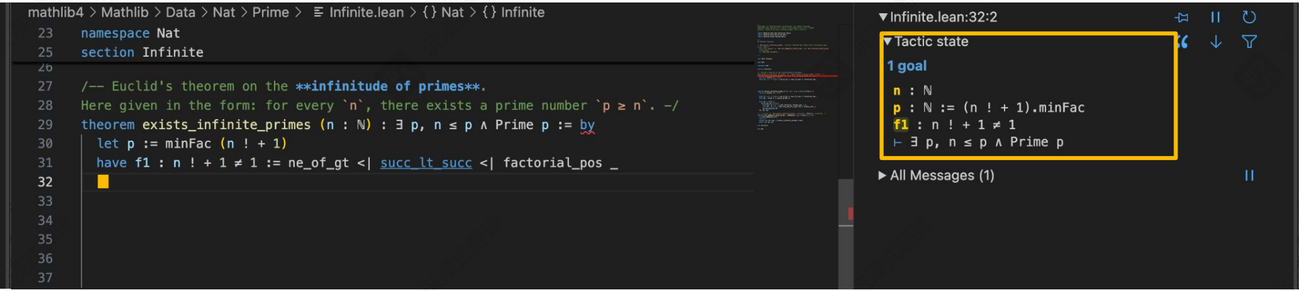
Demo - 无限素数



Demo - Infinitude of Primes

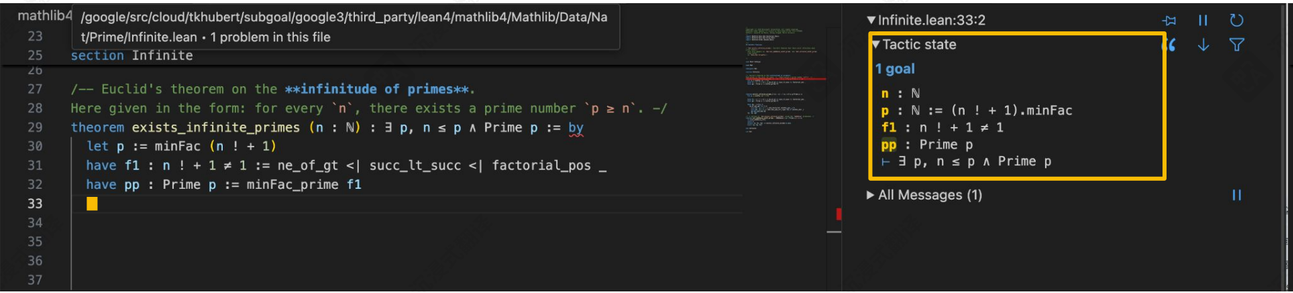


Demo - 无限素数

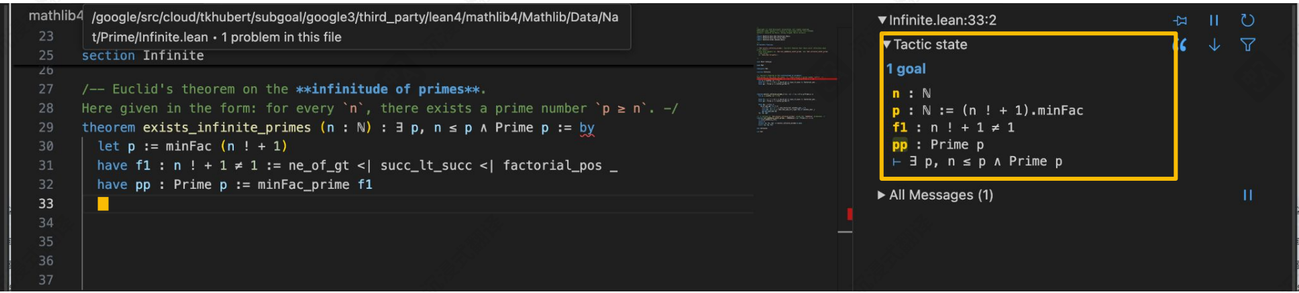


19

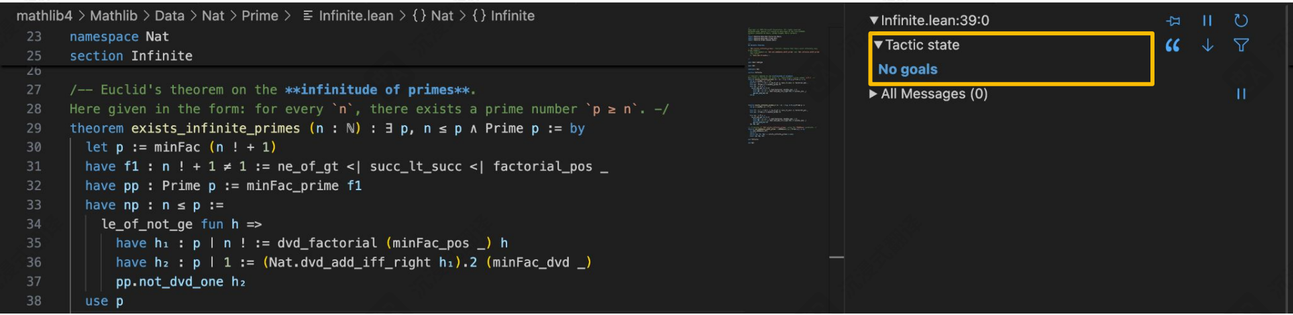
Demo - Infinitude of Primes



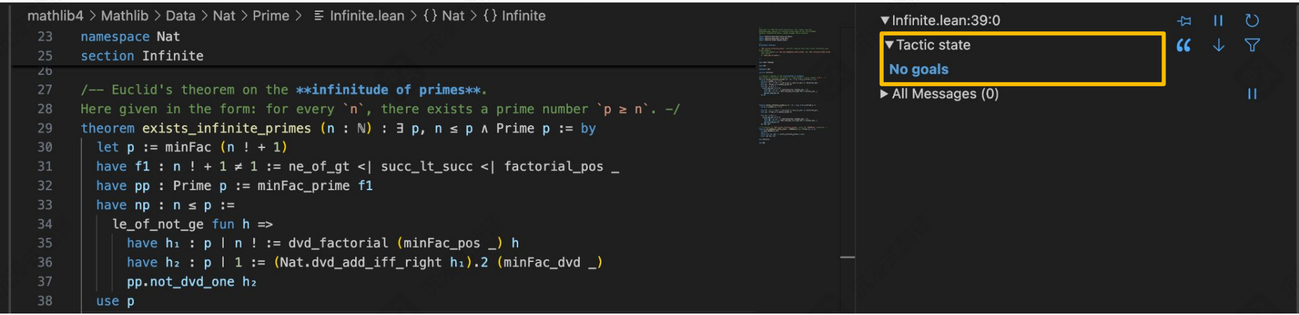
Demo - 无限素数



Demo - Infinitude of Primes



Demo - 无限素数



Computer Formalisation

Huge enthusiasm from a vibrant community

Building the library of the future

Unlocks enormous synergies

Still has some way to go

Belief this will inevitably play an integral role in the future of Mathematics

计算机形式化

来自充满活力的社区的热情高涨

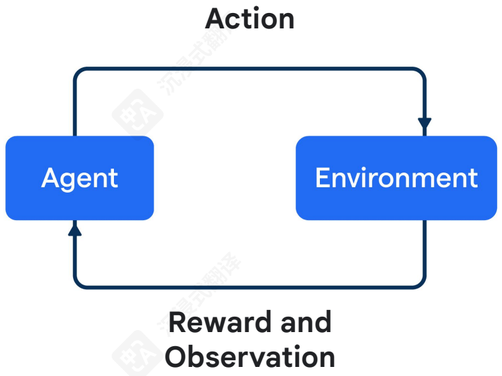
构建未来的图书馆

释放巨大的协同效应

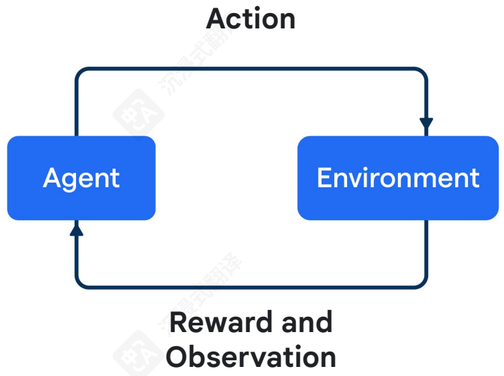
仍需努力

相信这不可避免地将在未来数学中发挥重要作用

Reinforcement Learning



强化学习



What's RL

RL is trial and error learning.

An **agent** learns by **interacting** with an **environment** to maximize **reward**.

RL 是什么

RL 是试错学习。

一个 **智能体**通过**与一个**环境**互动**来最大化**奖励**。

SuperScale RL: A Proven Recipe to Superintelligence

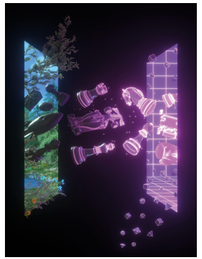
Atari DQN

AlphaGo

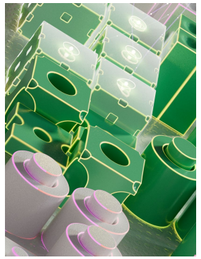

AlphaGoZero

AlphaZero

AlphaStar


MuZero


Stratego


AlphaDev

AlphaTensor


Fusion

SuperScale RL：通往超级智能的成熟方案

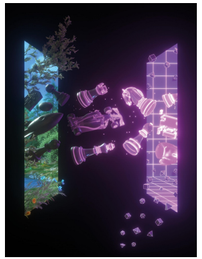
Atari DQN

AlphaGo


AlphaGoZero

AlphaZero

AlphaStar


MuZero


Stratego


AlphaDev

AlphaTensor

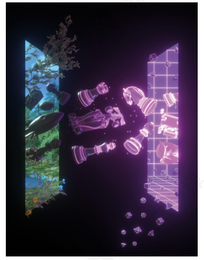

Fusion

The Zero Series

AlphaGo

AlphaGoZero

AlphaZero



MuZero

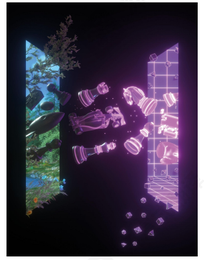
AlphaTensor

The Zero Series

AlphaGo

AlphaGoZero

AlphaZero



MuZero

AlphaTensor

The Zero Philosophy

If an agent can learn to master an environment **tabula rasa**, then you demonstrably have a system that is **discovering and learning new knowledge by itself**.

It also means that this algorithm is **general** and should apply to other domains.

This is particularly important to **advance science**.

零哲学

如果一个智能体能够学会掌控一个环境 **白板**，那么你显然就拥有一个能够 **自我发现和学习新知识**的系统

这也意味着这个算法是 **通用的**，并且应该适用于其他领域。

这对 **推进科学**尤为重要。

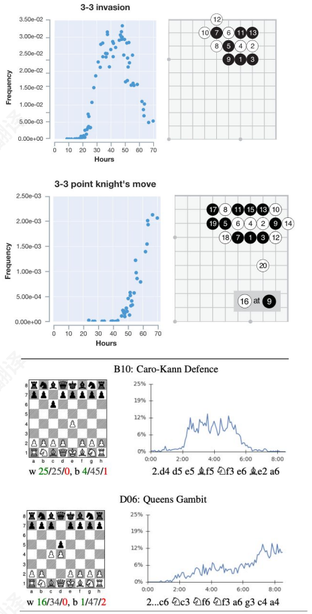
27

AlphaGoZero & AlphaZero

Learned simply by playing games against itself, starting from completely random play.

Rediscovered human patterns, accumulating thousands of years of human knowledge in a matter of days.

Ultimately discarded some in preference of **new discoveries** that humans don't even know about.

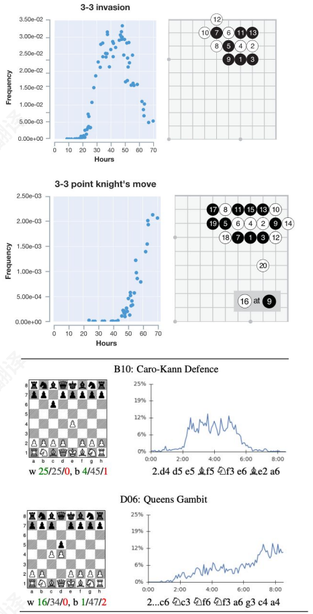


AlphaGoZero & AlphaZero

通过与自己玩游戏来学习，从一开始就是完全随机的玩法。

重新发现人类模式，在几天之内积累了数千年的人类知识。

最终放弃了一些，而选择了**新的发现**人类甚至不知道的。



28

AlphaTensor

Not only board games!

AlphaTensor **discovered novel**, efficient, and provably correct **algorithms** for fundamental tasks such as **matrix multiplication**.

Bonus: Can tackle huge action spaces $O(10^{30})$

Size (n, m, p)	Best method known	Best rank known	AlphaTensor rank Modular	AlphaTensor rank Standard
(2, 2, 2)	(Strassen, 1969)[2]	7	7	7
(3, 3, 3)	(Laderman, 1976)[15]	23	23	23
(4, 4, 4)	(Strassen, 1969)[2] (2, 2, 2) ⊗ (2, 2, 2)	49	47	49
(5, 5, 5)	(3, 5, 5) + (2, 5, 5)	98	96	98
(2, 2, 3)	(2, 2, 2) + (2, 2, 1)	11	11	11
(2, 2, 4)	(2, 2, 2) + (2, 2, 2)	14	14	14
(2, 2, 5)	(2, 2, 2) + (2, 2, 3)	18	18	18
(2, 3, 3)	(Hopcroft and Kerr, 1971)[16]	15	15	15
(2, 3, 4)	(Hopcroft and Kerr, 1971)[16]	20	20	20
(2, 3, 5)	(Hopcroft and Kerr, 1971)[16]	25	25	25
(2, 4, 4)	(Hopcroft and Kerr, 1971)[16]	26	26	26
(2, 4, 5)	(Hopcroft and Kerr, 1971)[16]	33	33	33
(2, 5, 5)	(Hopcroft and Kerr, 1971)[16]	40	40	40
(3, 3, 4)	(Smirnov, 2013)[18]	29	29	29
(3, 3, 5)	(Smirnov, 2013)[18]	36	36	36
(3, 4, 4)	(Smirnov, 2013)[18]	38	38	38
(3, 4, 5)	(Smirnov, 2013)[18]	48	47	47
(3, 5, 5)	(Sedoglavic and Smirnov, 2021)[19]	58	58	58
(4, 4, 5)	(4, 4, 2) + (4, 4, 3)	64	63	63
(4, 5, 5)	(2, 5, 5) ⊗ (2, 1, 1)	80	76	76

AlphaTensor

不仅限于棋盘游戏！

AlphaTensor **发现了新颖**，高效，且可证明正确**算法**用于基本任务，例如**矩阵乘法**。

额外： 能够处理巨大的动作空间 $O(10_{30})$

Size (n, m, p)	Best method known	Best rank known	AlphaTensor rank Modular	AlphaTensor rank Standard
(2, 2, 2)	(Strassen, 1969)[2]	7	7	7
(3, 3, 3)	(Laderman, 1976)[15]	23	23	23
(4, 4, 4)	(Strassen, 1969)[2] (2, 2, 2) ⊗ (2, 2, 2)	49	47	49
(5, 5, 5)	(3, 5, 5) + (2, 5, 5)	98	96	98
(2, 2, 3)	(2, 2, 2) + (2, 2, 1)	11	11	11
(2, 2, 4)	(2, 2, 2) + (2, 2, 2)	14	14	14
(2, 2, 5)	(2, 2, 2) + (2, 2, 3)	18	18	18
(2, 3, 3)	(Hopcroft and Kerr, 1971)[16]	15	15	15
(2, 3, 4)	(Hopcroft and Kerr, 1971)[16]	20	20	20
(2, 3, 5)	(Hopcroft and Kerr, 1971)[16]	25	25	25
(2, 4, 4)	(Hopcroft and Kerr, 1971)[16]	26	26	26
(2, 4, 5)	(Hopcroft and Kerr, 1971)[16]	33	33	33
(2, 5, 5)	(Hopcroft and Kerr, 1971)[16]	40	40	40
(3, 3, 4)	(Smirnov, 2013)[18]	29	29	29
(3, 3, 5)	(Smirnov, 2013)[18]	36	36	36
(3, 4, 4)	(Smirnov, 2013)[18]	38	38	38
(3, 4, 5)	(Smirnov, 2013)[18]	48	47	47
(3, 5, 5)	(Sedoglavic and Smirnov, 2021)[19]	58	58	58
(4, 4, 5)	(4, 4, 2) + (4, 4, 3)	64	63	63
(4, 5, 5)	(2, 5, 5) ⊗ (2, 1, 1)	80	76	76

29

What made those systems Superhuman?

Scaled up trial and error

Grounded feedback signal

Search

Curriculum

What made those systems Superhuman?

Scaled up trial and error

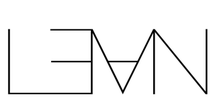
Grounded feedback signal

Search

Curriculum

What made those systems Superhuman?

✅ **Scaled up trial and error**



Grounded feedback signal

Search

Curriculum

What made those systems Superhuman?

✅ **Scaled up trial and error**

Grounded feedback sign al

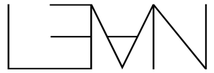
Search

Curriculum

31

What made those systems Superhuman?

✅ **Scaled up trial and error**

✅ **Grounded feedback signal** 

Search

Curriculum

What made those systems Superhuman?

✅ Scaled up trial and error

✅ Grounded feedback signal

Search

Curriculum

32

What made those systems Superhuman?

✅ **Scaled up trial and error**

✅ **Grounded feedback signal**

✅ Search

Curriculum

What made those systems Superhuman?

✅ **Scaled up trial and error**

✅ **Grounded feedback signal**

✅ Search

Curriculum

33

What made those systems Superhuman?

✅ **Scaled up trial and error**

✅ **Grounded feedback signal**

✅ Search

⁉️ Curriculum

What made those systems Superhuman?

✅ **Scaled up trial and error**

✅ **Grounded feedback signal**

✅ Search

⁉️ Curriculum

34

AlphaZero for Mathematics

When RL meets Formal Maths

AlphaZero is the agent.

+

Lean is the environment.

AlphaZero for Mathematics

当强化学习遇到形式数学

AlphaZero is the agent.

+

Lean is the environment.

35

AlphaProof: Master Plan

1. Lean gives us a way to **scale up trial and error** with
   a. An **environment** to explore mathematics completely in silico
   b. A **perfect feedback** signal for proving

2. We can therefore **reach superhuman intelligence and discover new truths**, just like in Go/chess/Tensor decomposition… provided:
   - We can generate high enough quality + quantity problems
   - RL works

AlphaProof: Master Plan

1. Lean 给我们提供了一种 **扩展试错** 的方法，
   a. 一个 **环境**，可以在纯计算中完全探索数学 b. 一个 **完美反馈** 信号，用于证明，

2. 因此我们可以 **达到超人类智能并发现新的真理**，就像在 Go/ 棋类 / 张量分解中一样 …… 只要满足：
   - 我们可以生成足够高质量 + 数量的问题，
   - 强化学习有效，

AlphaProof: Master Plan

1. Lean gives us a way to **scale up trial and error** with
   a. An **environment** to explore mathematics completely in silico
   b. A **perfect feedback** signal for proving

2. We can therefore **reach superhuman intelligence and discover new truths**, just like in Go/chess/Tensor decomposition… provided:
   ⁉️ We can generate high enough quality + quantity problems
   ✅ RL works

AlphaProof: Master Plan

1. Lean 给我们提供了一种 **扩展试错** 的方法，
   a. 一个 **环境**，可以在纯计算中完全探索数学 b. 一个 **完美反馈** 信号，用于证明，

2. 因此我们可以 **达到超人类智能并发现新的真理**，就像在 Go/ 棋类 / 张量分解中一样 …… 只要满足：
   ⁉️ 我们可以生成足够高质量 + 数量的问题，
   ✅ 强化学习有效，

Where do the problems come from?

Humans define the problems

- We want to help human mathematics
- There are already a lot of problems!

问题从何而来?

人类定义了问题

- 我们想要帮助人类数学 -

已经有很多

问题!

Where do the problems come from?

Humans define the problems

- We want to help human mathematics
- There are already a lot of problems!

AlphaProof defines the problems

- Around human mathematics
  - Auto-formalisation
  - Test-time RL
- Augment human mathematics

问题从何而来?

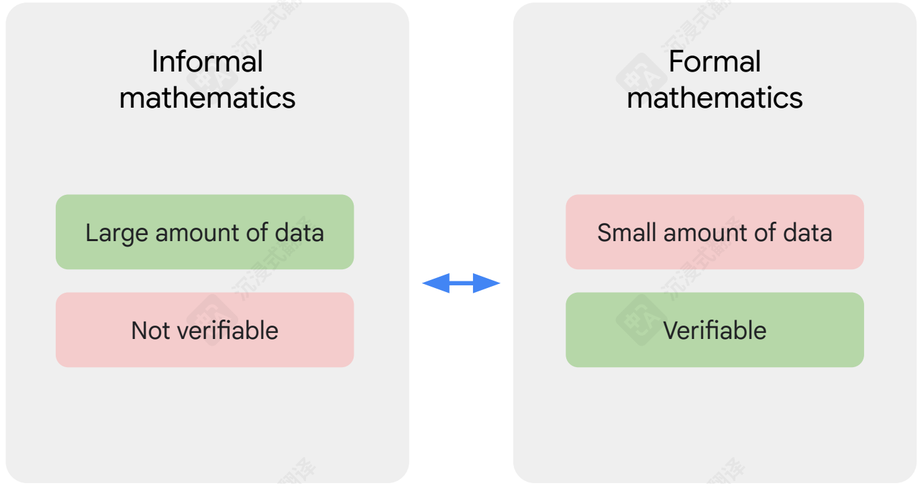
人类定义问题

- 我们想帮助人类数学 - 已经有很多问题了!

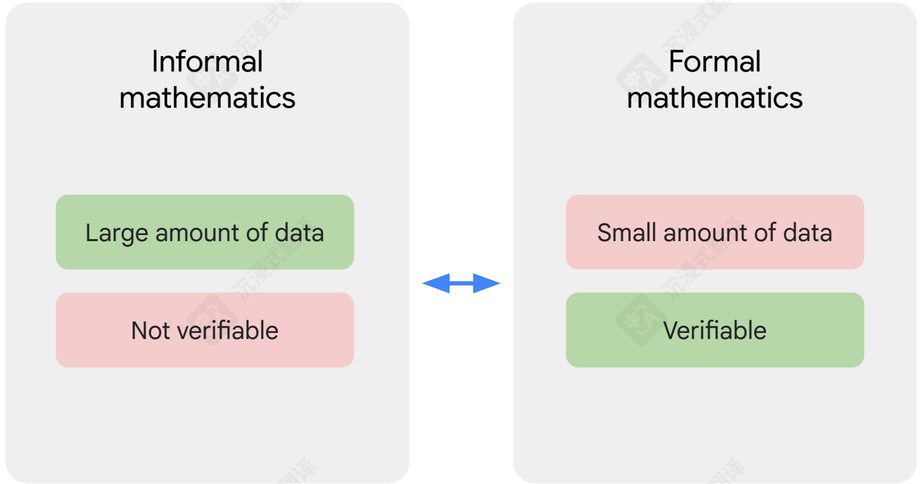
AlphaProof 定义问题

- 围绕人类数学 - 自动形式化
  - 测试时强化学习
- 增强人类数学

Let's take a step back...

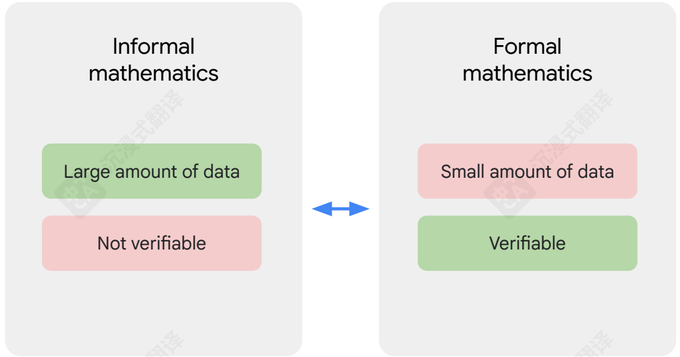


让我们退一步 ...



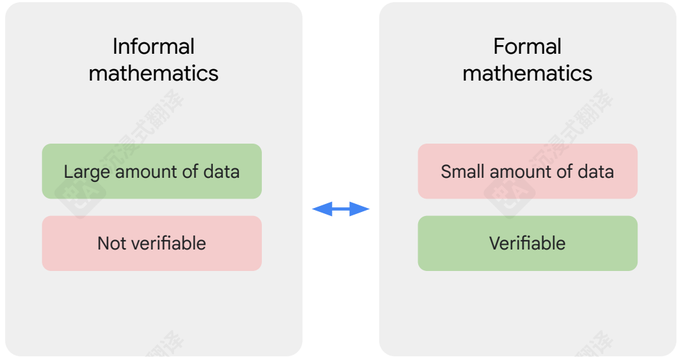
AlphaProof: Foundational Bet

Perfect verification will in the long run be the most important property for mathematics.



AlphaProof: 基础赌注

完美的验证从长远来看将是对数学最重要的属性。



RL meets Formal Math

- We have a **proven recipe** to build agent that can discover knowledge by themselves and reach superhuman intelligence in certain domains
- Key ingredients are **scaled up trial and error** and **grounded feedback**
- **Formal Maths** provides us with those key ingredients

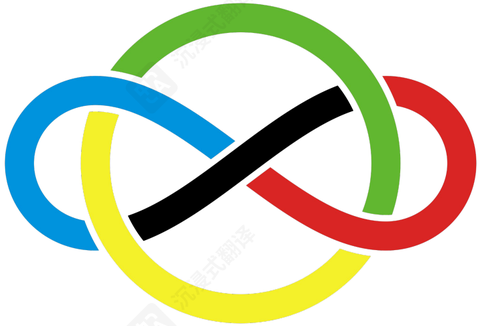
RL meets Formal Math

- 我们有一个**经过验证的方案**来构建能够自我发现知识的智能体，并在特定领域达到超人类智能

- 关键成分是 **扩大规模的试错**和**基于事实的反馈**
- **形式数学**为我们提供了这些关键成分

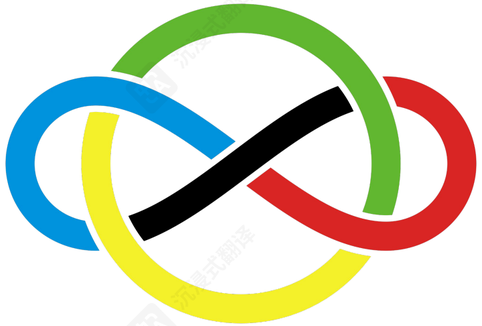
42

IMO 2024



International Mathematics Olympiad

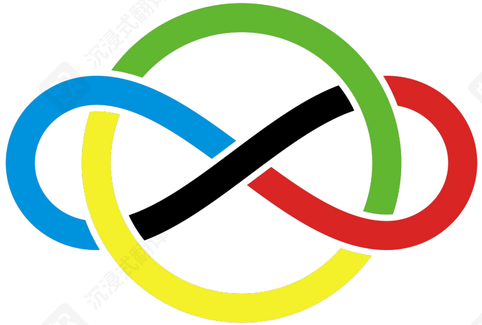
IMO 2024



国际数学奥林匹克

What's the IMO

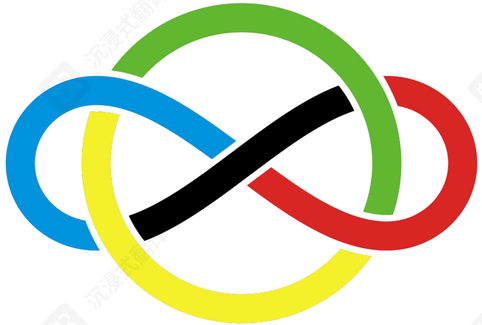
The world-championship level event in mathematics that brings together the best young minds from around the world.



International Mathematics Olympiad

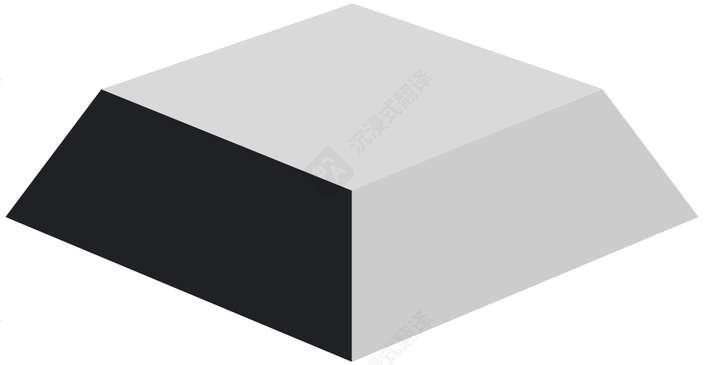
IMO 是什么

一项数学世界冠军级别的事件，汇聚了来自世界各地的顶尖年轻才俊。



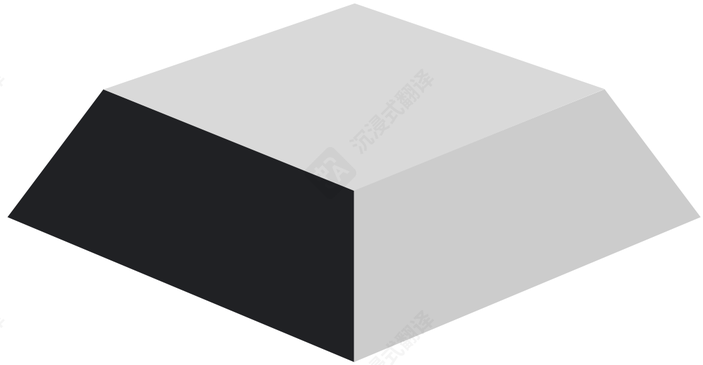
国际数学奥林匹克

USA team selection



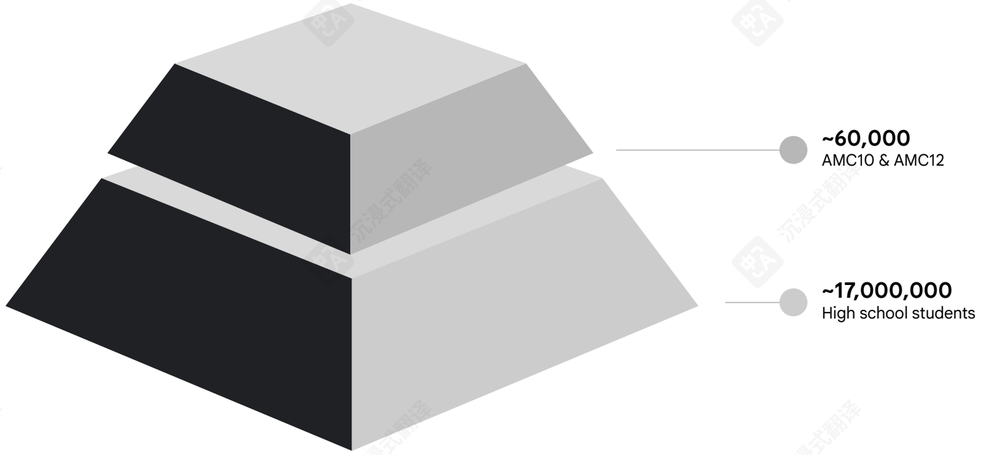
~17,000,000
High school students

美国队选拔

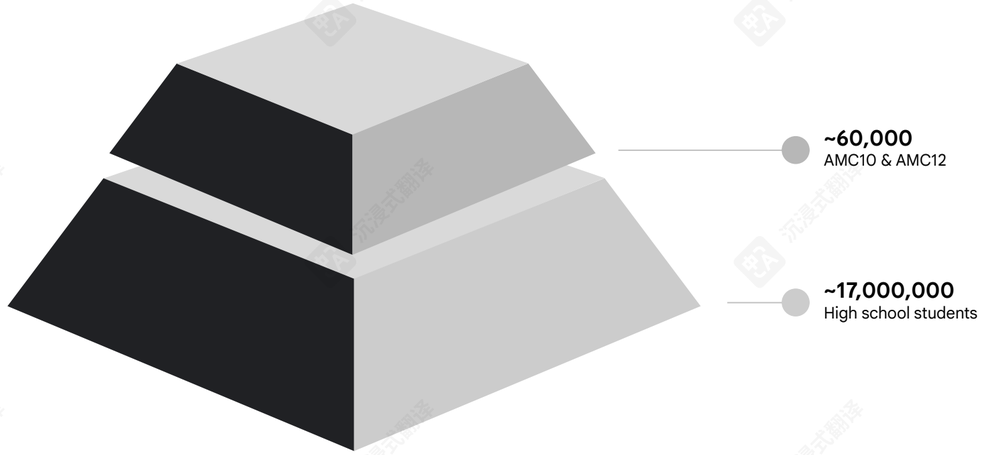


~17,000,000
High school students

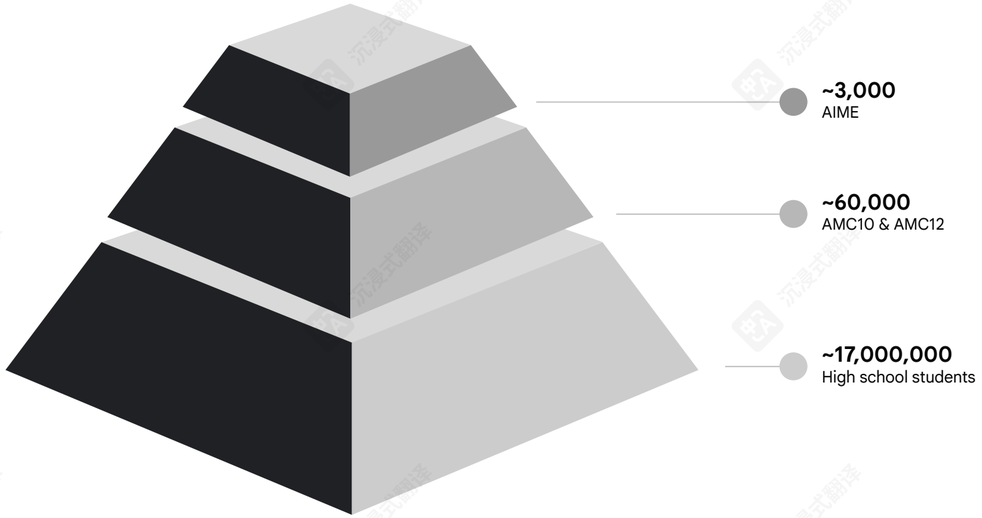
USA team selection



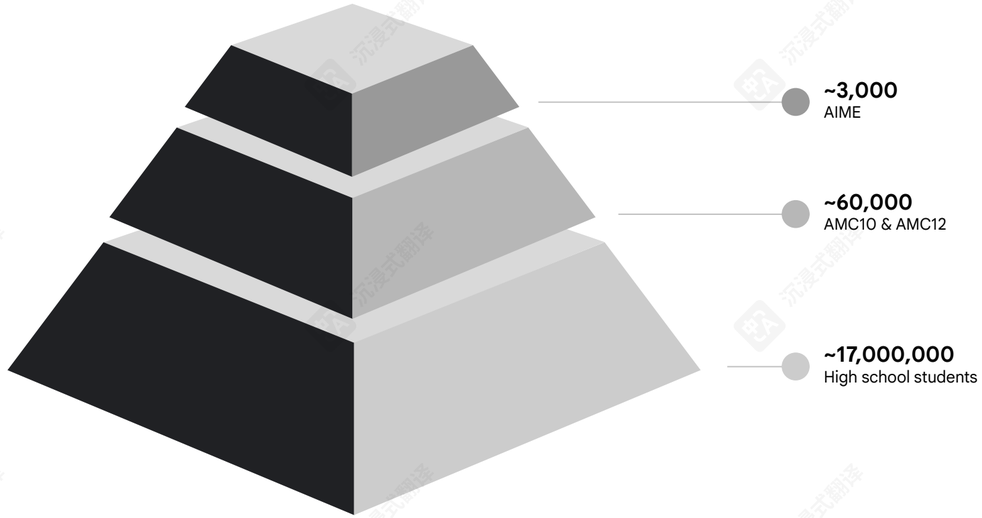
USA team selection



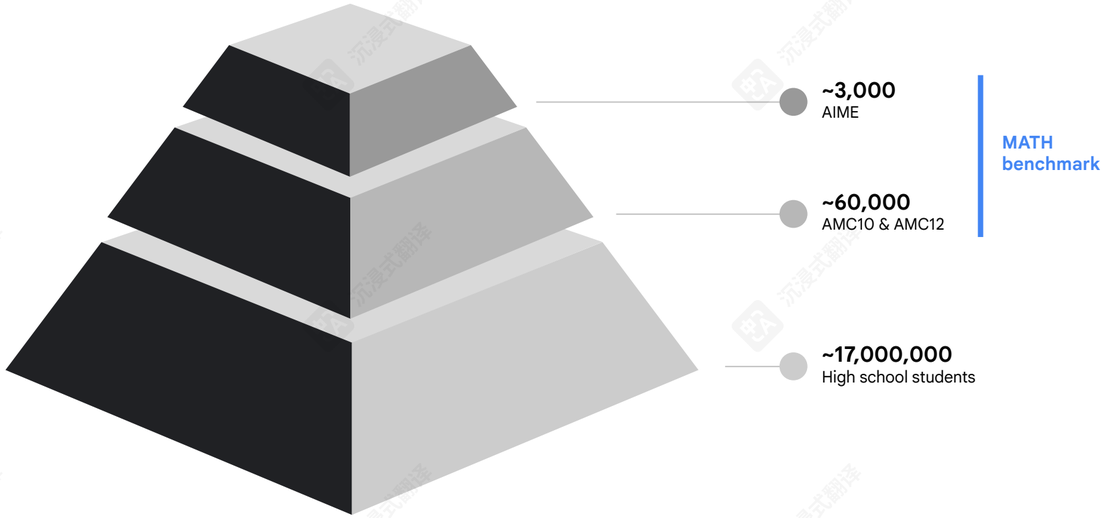
USA team selection



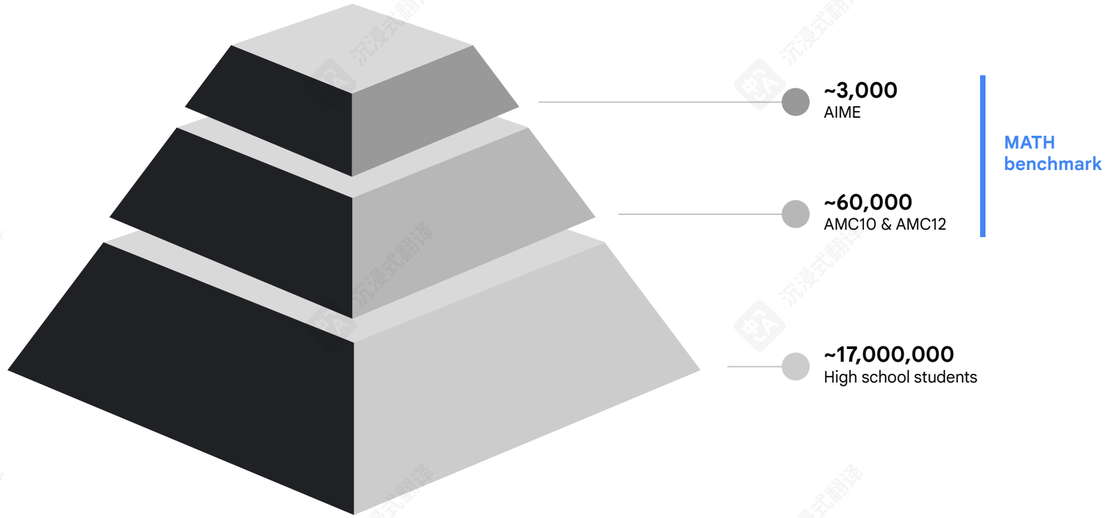
USA team selection



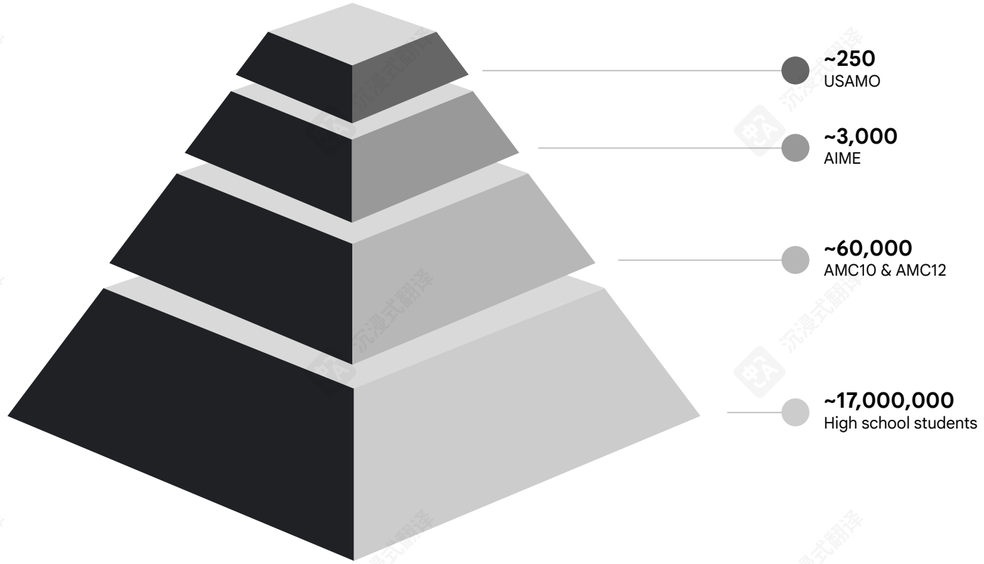
USA team selection



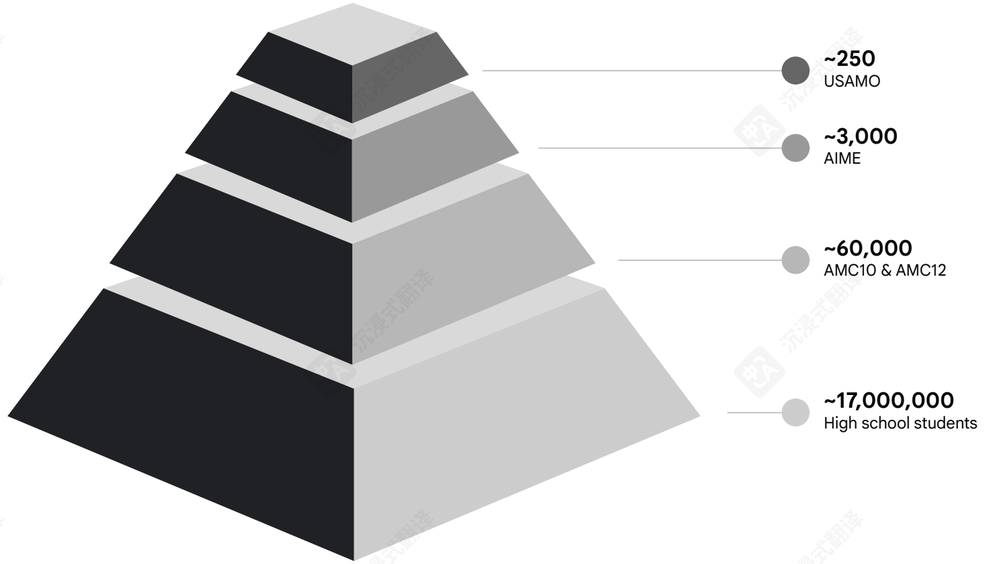
USA team selection

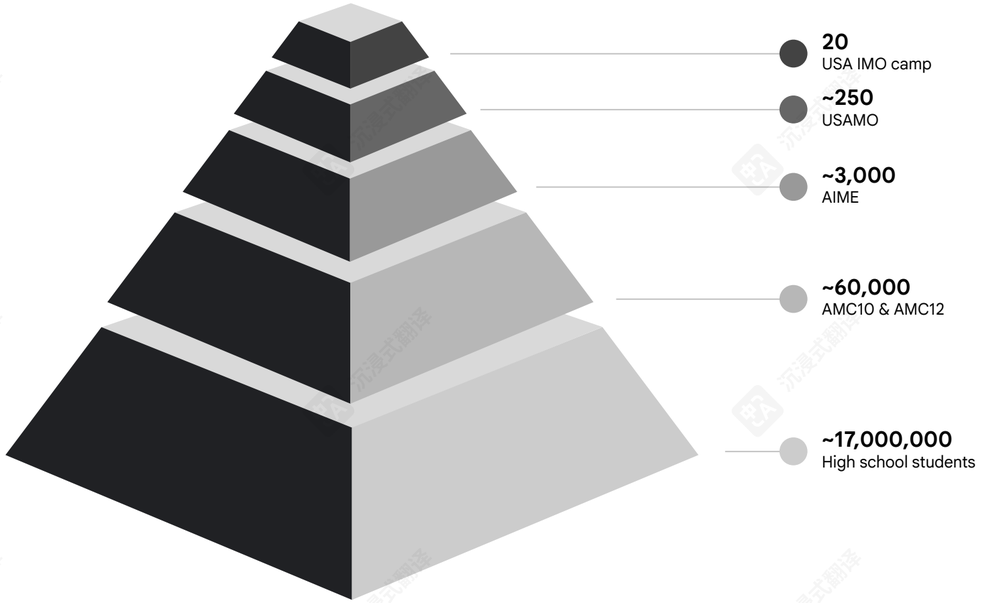


USA team selection

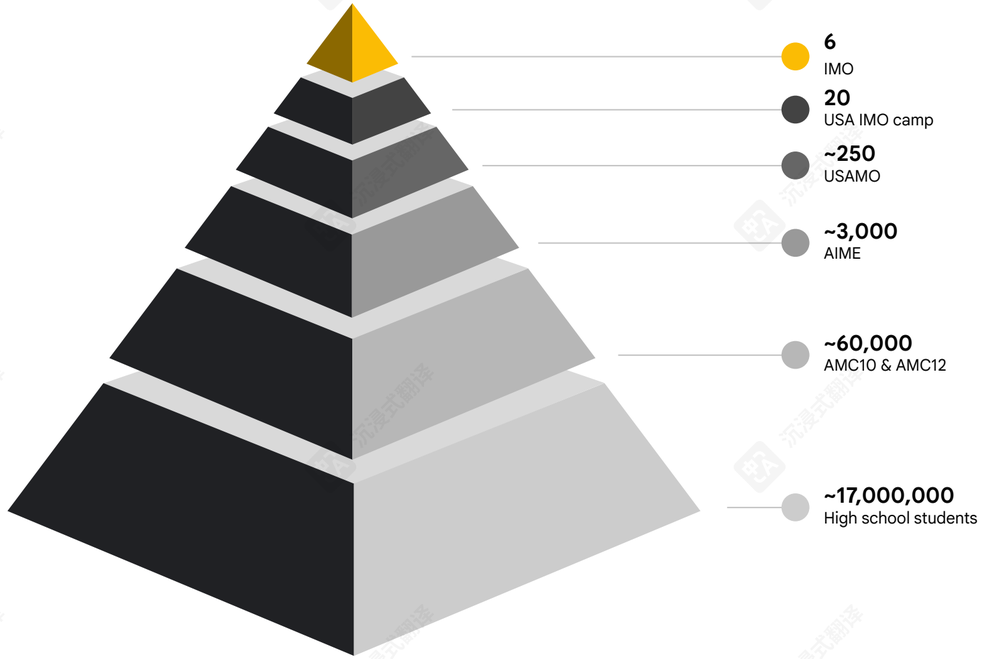


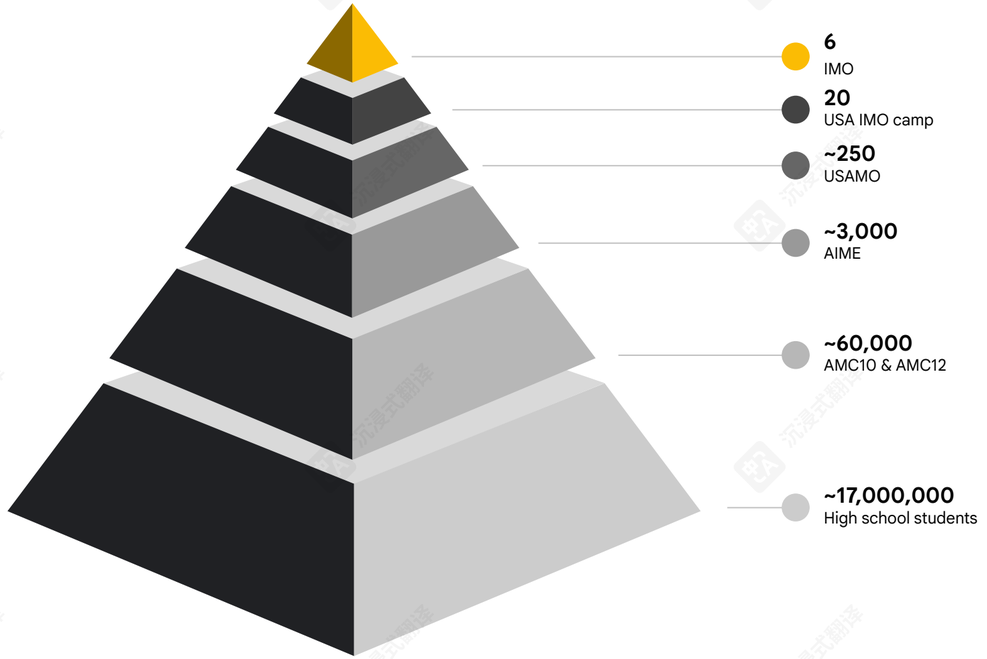
USA team selection





20
USA IMO camp
~250
USAMO
~3,000
AIME
~60,000
AMC10 & AMC12
~17,000,000
High school students


6
IMO
20
USA IMO camp
~250
USAMO
~3,000
AIME
~60,000
AMC10 & AMC12
~17,000,000
High school students


6
IMO
20
USA IMO camp
~250
USAMO
~3,000
AIME
~60,000
AMC10 & AMC12
~17,000,000
High school students

The IMO Problems

These are hard problems!

- (Easy, Medium, Hard)
- Test of reasoning, not knowledge
- Take hours even for specialists
- Not leaked

Half of the participants, solve 2 or less problems.

Po-Shen Loh • 3rd+
Carnegie Mellon Math Professor • IMO Foundation VP for A...
1mo •

+ Follow ···

I tried this year's problems while at the International #Math Olympiad myself. Took me hours.

Context about IMO: problems are specifically selected to be non-standard. For the previous 10 years, I served as the national coach for USA (https://lnkd.in/ggXvNQHY). During IMO itself, national coaches meet to pick the problems to appear on the exam. One of most important tasks of that group is: avoid problems similar to problems that appeared anywhere before. National coaches would dig up an old obscure math contest with a similar problem, and show it to the group, after which the proposed problem would be struck down.

The IMO Problems

这些是很难的问题！

- (简单，中等，困难) - 测试推理能力，而非知识 - 即使对专家来说也要花费数小时 - 不会泄露

一半的参与者，只能解决 2 个或更少的问题。

Po-Shen Loh • 3rd+
Carnegie Mellon Math Professor • IMO Foundation VP for A...
1mo •

+ Follow ···

I tried this year's problems while at the International #Math Olympiad myself. Took me hours.

Context about IMO: problems are specifically selected to be non-standard. For the previous 10 years, I served as the national coach for USA (https://lnkd.in/ggXvNQHY). During IMO itself, national coaches meet to pick the problems to appear on the exam. One of most important tasks of that group is: avoid problems similar to problems that appeared anywhere before. National coaches would dig up an old obscure math contest with a similar problem, and show it to the group, after which the proposed problem would be struck down.

IMO2024

65TH INTERNATIONAL
MATHEMATICAL OLYMPIAD

16-17TH JULY 2024

IMO2024

第 65 届国际数学奥林匹克

2024 年 7 月 16-17 日

Our IMO Participation - Apollo program

Can we reach the moon?

Can our system solve the 2024 IMO problems at all, given

- the compute available to us.
- and enough time.

我们的 IMO 参与 - 阿波罗计划

我们能到达月球吗?

我们的系统能解决 2024 年的 IMO 问题吗
在所有情况下，考虑到

- 我们可用的计算能力。 - 以及足够的时间。

January 2024

We are 7 months away, we need to make a decision about Geometry:

- Mathlib is very sparse in 2D euclidean geometry
- Should we fill it ourselves?

2024 年 1 月

我们还有 7 个月的时间，我们需要就几何学做出决定：- Mathlib 在二维欧几里得几何方面非常稀疏 - 我们应该自己填充吗？

January 2024

We are 7 months away, we need to make a decision about Geometry:
- Mathlib is very sparse in 2D euclidean geometry
- Should we fill it ourselves?

Miraculously, another group within Google DeepMind has been working on IMO Geometry problem and the system is already very strong!

We decide to join forces for July:
- **AlphaGeometry** will tackle geometry
- **AlphaProof** will tackle algebra, number theory, combinatorics

2024 年 1 月

我们还有 7 个月的时间，我们需要就几何学做出决定：- Mathlib 在二维欧几里得几何方面非常稀疏 - 我们是否应该自己填补它?

奇迹般地，另一个团队在 Google DeepMind 内部一直在研究 IMO 几何问题，系统已经非常强大！

我们决定在 7 月份联手：- **AlphaGeometry** 将负责几何学 - **AlphaProof** 将负责代数、数论、组合学

March 2024

We have been training AlphaProof on proving theorems but not all questions are of the form *"Prove that"*. Some require an answer to be determined, e.g. *“Find all X such that …”*

We debate what we should do:
Easy mode: Correct answer is given by an oracle; AI proves it is correct
Hard mode: AI determines the answer and proves it is correct

2024 年 3 月

我们一直在训练 AlphaProof 来证明定理，但并非所有问题都以“证明”的形式出现。有些问题需要确定答案，例如“找出所有满足 … 的 X”

我们讨论我们应该做什么：**简单模式：** 神秘的答案由预言者给出；AI 证明它是正确的 **困难模式：** AI 确定答案并证明它是正确的

March 2024

We have been training AlphaProof on proving theorems but not all questions are of the form *"Prove that"*. Some require an answer to be determined, e.g. *“Find all X such that …”*

We debate what we should do:
Easy mode: Correct answer is given by an oracle; AI proves it is correct
Hard mode: AI determines the answer and proves it is correct

AlphaProof will operate in **hard mode**:
- Generates candidate answers using Gemini
- Attempts to prove/disprove all candidates

2024 年 3 月

我们一直在训练 AlphaProof 来证明定理，但并非所有问题都以“证明”的形式出现。有些问题需要确定答案，例如“找出所有满足 … 的 X”

我们讨论我们应该做什么：**简单模式：** 神秘的预言者给出正确答案；AI 证明它是正确的 **困难模式：** AI 确定答案并证明它是正确的

AlphaProof 将以 **困难模式** 运行：
- 使用 Gemini 生成候选答案 - 尝试证明 / 反驳所有候选答案

Formalised Problems in Input

We have a final debate around if the system should receive as input
- The natural language description of the problem or
- A manually formalised description of the problem

输入中的形式化问题

我们有一个问题的自然语言描述或是否应该接收输入
形式化描述
- 手动形式化的问题描述

Formalised Problems in Input

We have a final debate around if the system should receive as input
- The natural language description of the problem or
- A manually formalised description of the problem

After concertation with our judges, we decided to give the **formalised problems as input** to the system.
- The ability we cared about was mathematical reasoning and problem solving.
- We asked our lean experts to manually formalise the problems for the system.

输入中的形式化问题

我们有一个问题的自然语言描述，或者
手动形式化的问题描述
- 一个手动形式化的问题描述

在与我们的评委协商后，我们决定将**形式化的输入问题**提供给系统。

- 我们关心的能力是数学推理和问题
解决。 - 我们请我们的精益专家手动为系统形式化问题。

Our Protocol

After officially receiving the problems at 1PM,

1. Lean experts manually formalize problems
2. Generate O(100) answer candidates with Gemini
3. Filter the easily disprovable ones
4. Run test-time RL

我们的协议

在下午 1 点正式收到问题后，

1. 人类专家手动形式化问题
2. 使用 Gemini 生成 O(100) 个答案候选
3. 过滤掉容易证伪的
4. 运行测试时的 RL

16th July, Day 1, Tuesday

1PM

P1*: Algebra
P2*: Number Theory
P3: Combinatorics

* require an answer to be determined

Problem 1. Determine all real numbers α such that, for every positive integer n, the integer

$$\lfloor \alpha \rfloor + \lfloor 2\alpha \rfloor + \cdots + \lfloor n\alpha \rfloor$$

is a multiple of n. (Note that $\lfloor z \rfloor$ denotes the greatest integer less than or equal to z. For example, $\lfloor -\pi \rfloor = -4$ and $\lfloor 2 \rfloor = \lfloor 2.9 \rfloor = 2$.)

Problem 2. Determine all pairs (a, b) of positive integers for which there exist positive integers g and N such that

$$\gcd(a^n + b,\, b^n + a) = g$$

holds for all integers $n \geqslant N$. (Note that $\gcd(x, y)$ denotes the greatest common divisor of integers x and y.)

Problem 3. Let $a_1, a_2, a_3, \ldots$ be an infinite sequence of positive integers, and let N be a positive integer. Suppose that, for each $n > N$, a_n is equal to the number of times a_{n-1} appears in the list $a_1, a_2, \ldots, a_{n-1}$.

Prove that at least one of the sequences $a_1, a_3, a_5, \ldots$ and $a_2, a_4, a_6, \ldots$ is eventually periodic.

(An infinite sequence $b_1, b_2, b_3, \ldots$ is *eventually periodic* if there exist positive integers p and M such that $b_{m+p} = b_m$ for all $m \geqslant M$.)

16th July, Day 1, Tuesday

1PM

P1*: Algebra
P2*: Number Theory
P3: Combinatorics

* 需要确定答案

Problem 1. Determine all real numbers α such that, for every positive integer n, the integer

$$\lfloor \alpha \rfloor + \lfloor 2\alpha \rfloor + \cdots + \lfloor n\alpha \rfloor$$

is a multiple of n. (Note that $\lfloor z \rfloor$ denotes the greatest integer less than or equal to z. For example, $\lfloor -\pi \rfloor = -4$ and $\lfloor 2 \rfloor = \lfloor 2.9 \rfloor = 2$.)

Problem 2. Determine all pairs (a, b) of positive integers for which there exist positive integers g and N such that

$$\gcd(a^n + b,\, b^n + a) = g$$

holds for all integers $n \geqslant N$. (Note that $\gcd(x, y)$ denotes the greatest common divisor of integers x and y.)

Problem 3. Let $a_1, a_2, a_3, \ldots$ be an infinite sequence of positive integers, and let N be a positive integer. Suppose that, for each $n > N$, a_n is equal to the number of times a_{n-1} appears in the list $a_1, a_2, \ldots, a_{n-1}$.

Prove that at least one of the sequences $a_1, a_3, a_5, \ldots$ and $a_2, a_4, a_6, \ldots$ is eventually periodic.

(An infinite sequence $b_1, b_2, b_3, \ldots$ is *eventually periodic* if there exist positive integers p and M such that $b_{m+p} = b_m$ for all $m \geqslant M$.)

16th July, Day 1, Tuesday

1PM

P1*: Algebra ⟶ Disproves 99% of guesses
P2*: Number Theory ⟶ Disproves 98% of guesses
P3: Combinatorics

* require an answer to be determined

16th July, Day 1, Tuesday

1PM

P1*: Algebra ⟶ Disproves 99% of guesses
P2*: Number Theory ⟶ Disproves 98% of guesses
P3: Combinatorics

* 需要确定答案

17th July, Day 2, Wednesday

1PM

P4: Geometry
P5*: Combinatorics
P6*: Algebra

* require an answer to be determined

Problem 4. Let ABC be a triangle with $AB < AC < BC$. Let the incentre and incircle of triangle ABC be I and ω, respectively. Let X be the point on line BC different from C such that the line through X parallel to AC is tangent to ω. Similarly, let Y be the point on line BC different from B such that the line through Y parallel to AB is tangent to ω. Let AI intersect the circumcircle of triangle ABC again at $P \neq A$. Let K and L be the midpoints of AC and AB, respectively.

Prove that $\angle KIL + \angle YPX = 180°$.

Problem 5. Turbo the snail plays a game on a board with 2024 rows and 2023 columns. There are hidden monsters in 2022 of the cells. Initially, Turbo does not know where any of the monsters are, but he knows that there is exactly one monster in each row except the first row and the last row, and that each column contains at most one monster.

Turbo makes a series of attempts to go from the first row to the last row. On each attempt, he chooses to start on any cell in the first row, then repeatedly moves to an adjacent cell sharing a common side. (He is allowed to return to a previously visited cell.) If he reaches a cell with a monster, his attempt ends and he is transported back to the first row to start a new attempt. The monsters do not move, and Turbo remembers whether or not each cell he has visited contains a monster. If he reaches any cell in the last row, his attempt ends and the game is over.

Determine the minimum value of n for which Turbo has a strategy that guarantees reaching the last row on the n^{th} attempt or earlier, regardless of the locations of the monsters.

Problem 6. Let $\mathbb{Q}$ be the set of rational numbers. A function $f\colon \mathbb{Q} \to \mathbb{Q}$ is called *aquaesulian* if the following property holds: for every $x, y \in \mathbb{Q}$,

$$f(x + f(y)) = f(x) + y \qquad \text{or} \qquad f(f(x) + y) = x + f(y).$$

Show that there exists an integer c such that for any aquaesulian function f there are at most c different rational numbers of the form $f(r) + f(-r)$ for some rational number r, and find the smallest possible value of c.

17th July, Day 2, Wednesday

1PM

P4: Geometry
P5*: Combinatorics
P6*: Algebra

* 需要确定答案

Problem 4. Let ABC be a triangle with $AB < AC < BC$. Let the incentre and incircle of triangle ABC be I and ω, respectively. Let X be the point on line BC different from C such that the line through X parallel to AC is tangent to ω. Similarly, let Y be the point on line BC different from B such that the line through Y parallel to AB is tangent to ω. Let AI intersect the circumcircle of triangle ABC again at $P \neq A$. Let K and L be the midpoints of AC and AB, respectively.

Prove that $\angle KIL + \angle YPX = 180°$.

Problem 5. Turbo the snail plays a game on a board with 2024 rows and 2023 columns. There are hidden monsters in 2022 of the cells. Initially, Turbo does not know where any of the monsters are, but he knows that there is exactly one monster in each row except the first row and the last row, and that each column contains at most one monster.

Turbo makes a series of attempts to go from the first row to the last row. On each attempt, he chooses to start on any cell in the first row, then repeatedly moves to an adjacent cell sharing a common side. (He is allowed to return to a previously visited cell.) If he reaches a cell with a monster, his attempt ends and he is transported back to the first row to start a new attempt. The monsters do not move, and Turbo remembers whether or not each cell he has visited contains a monster. If he reaches any cell in the last row, his attempt ends and the game is over.

Determine the minimum value of n for which Turbo has a strategy that guarantees reaching the last row on the n^{th} attempt or earlier, regardless of the locations of the monsters.

Problem 6. Let $\mathbb{Q}$ be the set of rational numbers. A function $f\colon \mathbb{Q} \to \mathbb{Q}$ is called *aquaesulian* if the following property holds: for every $x, y \in \mathbb{Q}$,

$$f(x + f(y)) = f(x) + y \qquad \text{or} \qquad f(f(x) + y) = x + f(y).$$

Show that there exists an integer c such that for any aquaesulian function f there are at most c different rational numbers of the form $f(r) + f(-r)$ for some rational number r, and find the smallest possible value of c.

17th July, Day 2, Wednesday

1PM

P4: Geometry → Solved in secs by AlphaGeometry
P5*: Combinatorics → Failed to formalise on the day
P6*: Algebra → Disproves 7% of guesses

* require an answer to be determined

17th July, Day 2, Wednesday

1PM

P4: Geometry → Solved in secs by AlphaGeometry
P5*: Combinatorics → Failed to formalise on the day
P6*: Algebra → Disproves 7% of guesses

* 需要确定答案

18th July, Day 3, Thursday

We had a way to track progress and we observe:

- P1: **AlphaProof** solved half of the problem **3/7 points**
- P2: **AlphaProof** arguably solved the whole problem **6/7 points**
- P3: **AlphaProof** makes no progress

Day1: **9/21** points

18th July, Day 3, Thursday

我们有一种方法来跟踪进度，我们观察到：

- P1: **AlphaProof** 解决了问题的一半 **3/7 分** - P2: **AlphaProof** 按理说解决了整个问题 **6/7 分** - P3: **AlphaProof** 没有任何进展

Day1: **9/21** 分

18th July, Day 3, Thursday

We had a way to track progress and we observe:

- P1: **AlphaProof** solved half of the problem **3/7 points**
- P2: **AlphaProof** arguably solved the whole problem **6/7 points**
- P3: **AlphaProof** makes no progress

- P4: **AlphaGeometry** solved in seconds **7/7 points**
- P5: **AlphaProof** makes no progress
- P6: **AlphaProof** proves a case **2/7 points**

Day1: **9/21** points, Day2: **9/21** points, Total **18/42** points

18th July, Day 3, Thursday

我们有一种方法来跟踪进度，我们观察到：

- P1: **AlphaProof** 解决了半个问题 **3/7 分** P2: **AlphaProof** 按理
- 说解决了整个问题 **6/7 分** P3: **AlphaProof** 没有进展
-

- P4: **AlphaGeometry** 几秒钟内解决 **7/7 分** P5:
- **AlphaProof** 没有进展 P6: **AlphaProof** 证明了一个案例
- **2/7 分**

Day1: **9/21** 分，Day2: **9/21** 分，总计 **18/42** 分

19th July, Day 4, Friday

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p6.parts.ii.only_1588200855951509015 that this email is too short to contain.

19th July, Day 4, Friday

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p6.parts.ii.only_1588200855951509015 that this email is too short to contain.

19th July, Day 4, Friday

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p6.parts.ii.only_1588200855951509015 that this email is too short to contain.
Jk, here it is:

```
theorem imo_2024_p6.parts.ii.only_1588200855951509015 (IsAquaesulian : (Q → Q) → Prop) (IsAquaesulian_def : ∀ f, IsAquaesulian f ↔ ∀ x y, f (x + f
y) = f x + y ∨ f (f x + y) = x + f y) : IsLeast {(c : ℤ) | ∀ f, IsAquaesulian f → {(f r + f (-r)) | (r : Q)}.Finite ∧ {(f r + f (-r)) | (r :
Q)}.ncard ≤ c} 2 := by
  exists@?_
  · use λu b=>if j:u 0=0then by_contra λc=>?_ else ?_
    · suffices:({J|∃k,u k+u (-k)= J}) ⊆{0}
      · simp_all[this.antisymm]
      rintro - (a, rfl)
      contrapose! c
      simp_all
      suffices:{U|∃examples6, (u) ‹Q› +u ( -‹_›)= U} ⊆{0,(u (a : Rat)+ (u<|@@↑(( (-a )))) } ..
      · use ( Set.toFinite ( _) ).subset ↑@@this , (Set.ncard_le_ncard$ (((this )) ) ).trans (Set.ncard_pair$ Ne.symm (↑ ( (c)) ) ).le
      rintro-(hz, rfl)
      induction b @hz a
      · have:=b (-a)$ hz+u a
        have:=b hz hz
        simp_all[add_comm]
        have:=b (-hz) (hz+u ↑(hz))
        simp_all[ add_assoc, C]
        induction this
        · simp_all
          have:=b hz (hz+(u a+u (-a)))
          have:=b (hz+(u a+u (-a)))$ hz+(u a+u (-a))
          use .inr$ by_contra$ by hint
        have:=b hz$ hz+(u hz+u (-hz))
        cases b (hz+(u hz+u (-hz)))$ hz+(u hz+u (-hz))with|_=>hint
      have:=b (-hz) (u hz+a)
      have:=b$ -a
      specialize this (u hz+a)
      simp_all[ ←add_assoc]
      have:=b 0
      have:=b
      specialize b a a
      simp_all[add_comm]
      have:=(this<| -a) (↑a + (((u a))): (↑_ :((( _) ) ) )) ..
      simp_all[add_assoc]
```

19th July, Day 4, Friday

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p6.parts.ii.only_1588200855951509015 that this email is too short to contain.
Jk, here it is:

```
theorem imo_2024_p6.parts.ii.only_1588200855951509015 (IsAquaesulian : (Q → Q) → Prop) (IsAquaesulian_def : ∀ f, IsAquaesulian f ↔ ∀ x y, f (x + f
y) = f x + y ∨ f (f x + y) = x + f y) : IsLeast {(c : ℤ) | ∀ f, IsAquaesulian f → {(f r + f (-r)) | (r : Q)}.Finite ∧ {(f r + f (-r)) | (r :
Q)}.ncard ≤ c} 2 := by
  exists@?_
  · use λu b=>if j:u 0=0then by_contra λc=>?_ else ?_
    · suffices:({J|∃k,u k+u (-k)= J}) ⊆{0}
      · simp_all[this.antisymm]
      rintro - (a, rfl)
      contrapose! c
      simp_all
      suffices:{U|∃examples6, (u) ‹Q› +u ( -‹_›)= U} ⊆{0,(u (a : Rat)+ (u<|@@↑(( (-a )))) } ..
      · use ( Set.toFinite ( _) ).subset ↑@@this , (Set.ncard_le_ncard$ (((this )) ) ).trans (Set.ncard_pair$ Ne.symm (↑ ( (c)) ) ).le
      rintro-(hz, rfl)
      induction b @hz a
      · have:=b (-a)$ hz+u a
        have:=b hz hz
        simp_all[add_comm]
        have:=b (-hz) (hz+u ↑(hz))
        simp_all[ add_assoc, C]
        induction this
        · simp_all
          have:=b hz (hz+(u a+u (-a)))
          have:=b (hz+(u a+u (-a)))$ hz+(u a+u (-a))
          use .inr$ by_contra$ by hint
        have:=b hz$ hz+(u hz+u (-hz))
        cases b (hz+(u hz+u (-hz)))$ hz+(u hz+u (-hz))with|_=>hint
      have:=b (-hz) (u hz+a)
      have:=b$ -a
      specialize this (u hz+a)
      simp_all[ ←add_assoc]
      have:=b 0
      have:=b
      specialize b a a
      simp_all[add_comm]
      have:=(this<| -a) (↑a + (((u a))): (↑_ :((( _) ) ) )) ..
      simp_all[add_assoc]
```

19th July, Day 4, Friday

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p6.parts.ii.only_1588200855951509015 that this email is too short to contain.

Thomas Hubert <tkhubert@google.com>
to Demis, Sergey

19th July, Day 4, Friday

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p6.parts.ii.only_1588200855951509015 that this email is too short to contain.

Thomas Hubert <tkhubert@google.com>
to Demis, Sergey

19th July, Day 4, Friday

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p6.parts.ii.only_1588200855951509015 that this email is too short to contain.
Jk, here it is:

```
theorem imo_2024_p6.parts.ii.only_1588200855951509015 (IsAquaesulian : (ℚ → ℚ) → Prop) (IsAquaesulian_def : ∀ f, IsAquaesulian f ↔ ∀ x y, f (x + f y) = f x + y ∨ f (f x + y) = x + f y) : IsLeast {(c : ℤ) | ∀ f, IsAquaesulian f → {(f r + f (-r)) | (r : ℚ)}.Finite ∧ {(f r + f (-r)) | (r : ℚ)}.ncard ≤ c} 2 := by
```

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p2_7949354807137552108 that this email is too short to contain.
Jk, here it is:

```
theorem imo_2024_p2_7949354807137552108 : {(a, b) | 0 < a ∧ 0 < b ∧ ∃ g N, 0 < g ∧ 0 < N ∧ ∀ n ≥ N, Nat.gcd (a ^ n + b) (b ^ n + a) = g} = {(a, b)
| Nat.gcd a b = 1 ∧ a = b} := by
  use subset antisymm (λt S=>S.2.2.rec λw o=>? ) ?   ..
```

19th July, Day 4, Friday

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p6.parts.ii.only_1588200855951509015 that this email is too short to contain.
Jk, here it is:

```
theorem imo_2024_p6.parts.ii.only_1588200855951509015 (IsAquaesulian : (ℚ → ℚ) → Prop) (IsAquaesulian_def : ∀ f, IsAquaesulian f ↔ ∀ x y, f (x + f y) = f x + y ∨ f (f x + y) = x + f y) : IsLeast {(c : ℤ) | ∀ f, IsAquaesulian f → {(f r + f (-r)) | (r : ℚ)}.Finite ∧ {(f r + f (-r)) | (r : ℚ)}.ncard ≤ c} 2 := by
```

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p2_7949354807137552108 that this email is too short to contain.
Jk, here it is:

```
theorem imo_2024_p2_7949354807137552108 : {(a, b) | 0 < a ∧ 0 < b ∧ ∃ g N, 0 < g ∧ 0 < N ∧ ∀ n ≥ N, Nat.gcd (a ^ n + b) (b ^ n + a) = g} = {(a, b)
| Nat.gcd a b = 1 ∧ a = b} := by
  use subset antisymm (λt S=>S.2.2.rec λw o=>? ) ?   ..
```

19th July, Day 4, Friday

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p6.parts.ii.only_1588200855951509015 that this email is too short to contain.
Jk, here it is:

```
theorem imo_2024_p6.parts.ii.only_1588200855951509015 (IsAquaesulian : (ℚ → ℚ) → Prop) (IsAquaesulian_def : ∀ f, IsAquaesulian f ↔ ∀ x y, f (x + f
y) = f x + y ∨ f (f x + y) = x + f y) : IsLeast {(c : ℤ) | ∀ f, IsAquaesulian f → {(f r + f (-r)) | (r : ℚ)}.Finite ∧ {(f r + f (-r)) | (r :
ℚ)}.ncard ≤ c} 2 := by
```

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p2_7949354807137552108 that this email is too short to contain.
Jk, here it is:

```
theorem imo_2024_p2_7949354807137552108 : {(a, b) | 0 < a ∧ 0 < b ∧ ∃ g N, 0 < g ∧ 0 < N ∧ ∀ n ≥ N, Nat.gcd (a ^ n + b) (b ^ n + a) = g} = {(a, b)
| Nat.gcd a b = 1 ∧ a = b} := by
  use subset antisymm (λt S=>S.2.2.rec λw o=>? ) ?   ..
```

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p1_189795200876102478 5 that this email is too short to contain.
Jk, here it is:

```
theorem imo_2024_p1_1897952008761024785 : {(α : ℝ) | ∀ (n : ℕ), 0 < n → (n : ℤ) | (∑ i in Finset.Icc 1 n, ⌊i * α⌋)} = {2 * k | k ∈ Set.range
(Int.cast : ℤ → ℝ)} := by
```

19th July, Day 4, Friday

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p6.parts.ii.only_1588200855951509015 that this email is too short to contain.
Jk, here it is:

```
theorem imo_2024_p6.parts.ii.only_1588200855951509015 (IsAquaesulian : (ℚ → ℚ) → Prop) (IsAquaesulian_def : ∀ f, IsAquaesulian f ↔ ∀ x y, f (x + f
y) = f x + y ∨ f (f x + y) = x + f y) : IsLeast {(c : ℤ) | ∀ f, IsAquaesulian f → {(f r + f (-r)) | (r : ℚ)}.Finite ∧ {(f r + f (-r)) | (r :
ℚ)}.ncard ≤ c} 2 := by
```

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p2_7949354807137552108 that this email is too short to contain.
Jk, here it is:

```
theorem imo_2024_p2_7949354807137552108 : {(a, b) | 0 < a ∧ 0 < b ∧ ∃ g N, 0 < g ∧ 0 < N ∧ ∀ n ≥ N, Nat.gcd (a ^ n + b) (b ^ n + a) = g} = {(a, b)
| Nat.gcd a b = 1 ∧ a = b} := by
  use subset antisymm (λt S=>S.2.2.rec λw o=>? ) ?   ..
```

mehtarishi+alphaproof-agent@google.com

I have discovered a truly marvelous proof of imo_2024_p1_189795200876102478 5 that this email is too short to contain.
Jk, here it is:

```
theorem imo_2024_p1_1897952008761024785 : {(α : ℝ) | ∀ (n : ℕ), 0 < n → (n : ℤ) | (∑ i in Finset.Icc 1 n, ⌊i * α⌋)} = {2 * k | k ∈ Set.range
(Int.cast : ℤ → ℝ)} := by
```


Final Results

P1, P2, P6 fully solved by AlphaProof
P4 fully solved by AlphaGeometry

We keep running over the weekend in the hope of getting one point on P3. The agent made some progress but not enough for a partial point.

最终结果

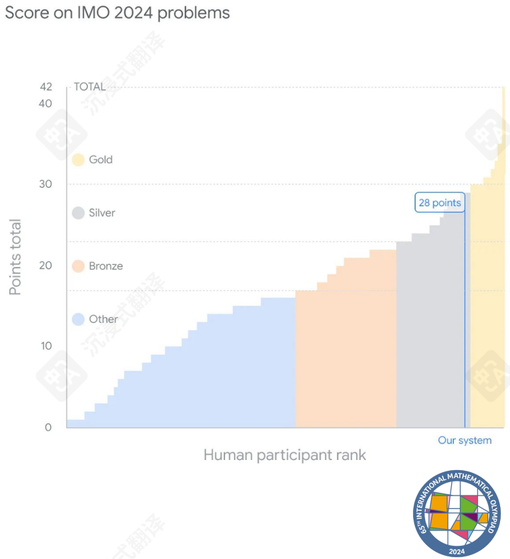
P1、P2、P6 由 AlphaProof 完全解决，P4 由 AlphaGeometry 完全解决

我们周末继续努力，希望能获得 P3 的一分。代理取得了一些进展，但不足以获得部分分数。

Final Results

P1, P2, P6 fully solved by AlphaProof
P4 fully solved by AlphaGeometry

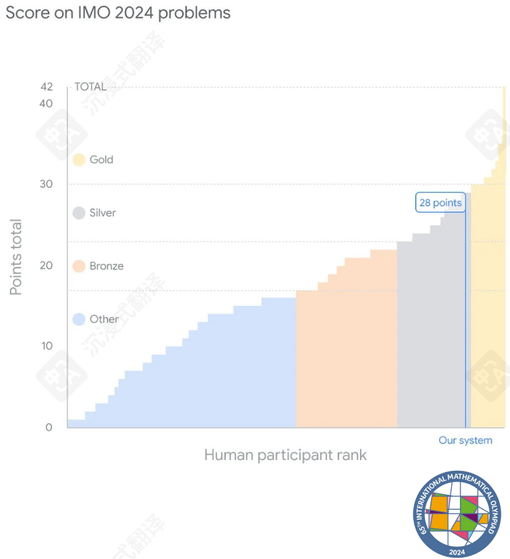
We reached the score of a **Silver** medallist and missed the **Gold** threshold by one point (with more time and more compute!).



最终结果

P1、P2、P6 由 AlphaProof 完全解决，P4 由 AlphaGeometry 完全解决

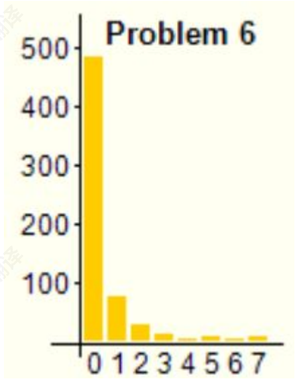
我们达到了**银牌**选手的分数，但错过了**金牌**标准，仅差一分（如果有更多时间和更多计算能力！）。



P6

P6 was arguably one of the hardest problems in the last 10 years at the IMO. Only 5 / 609 solved the problem fully.

***"I spent a couple of hours on this problem and did not solve it**. [...] I find the fact that the program can come up with a somewhat complicated construction like this very impressive"*
Prof Sir Timothy Gowers, Fields medalist and IMO gold medalist

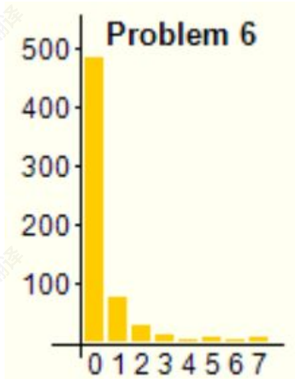


P6

P6 无疑是过去十年 IMO 中最难的问题之一。只有 5 / 609 人完全解决了这个问题。

" 我花了几小时在这个问题上，但没有解决它。[...] 我发现程序能想出像这样稍微复杂一点的构造，这非常令人印象深刻 "

戈登·卡拉汉奖得主、 *IMO* 金牌得主



Challenges

Gaps in Mathlib:

- Geometry, even P1

Combinatorics problems were difficult:

- P5 was extremely hard to formalise
- Did not make progress on P3 & P5.

Used many orders of magnitude more compute than human contestants:

- Successfully landed on the moon
- And will want to be more efficient for the next trip!

挑战

Mathlib 中的差距：-
几何学，甚至 P1

组合问题很困难：- P5 非常难以形式化 - 在
P3 & P5 上没有进展。

使用了比人类参赛者高几个数量级的计算量：

- 成功登上了月球 - 并且希望在下次旅行中更加高效！

AlphaProof

Methods

AlphaZero is the agent.

+

Lean is the environment.

AlphaProof

方法

AlphaZero is the agent.

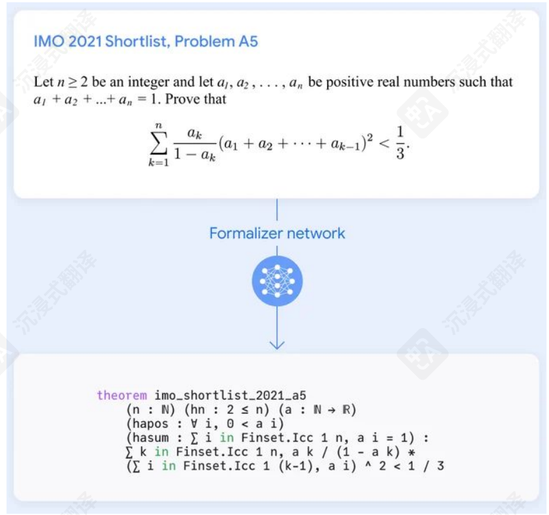
+

Lean is the environment.

77

Formaliser Model

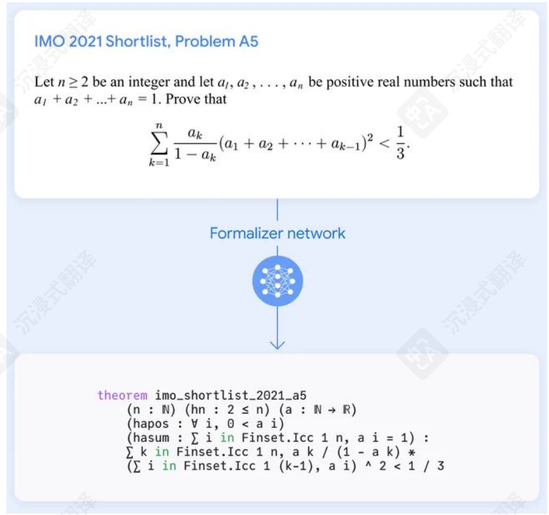
Input: a problem/theorem described in natural language



Output: a Lean formalisation

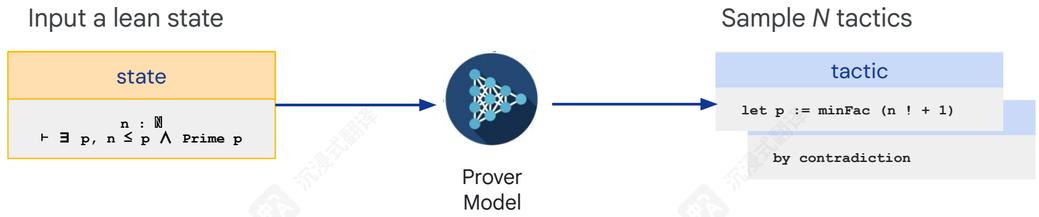
Formaliser Model

输入： 用自然语言描述的问题 / 定理

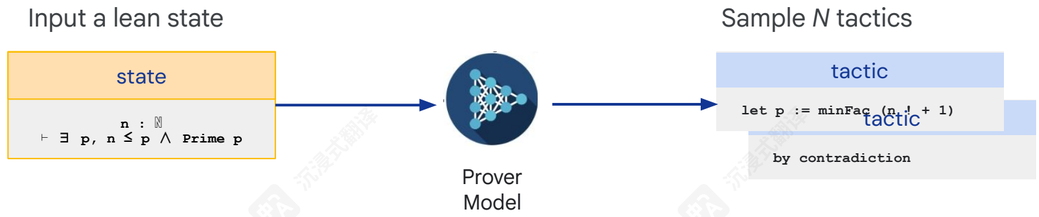


输出： Lean 形式化

Prover Model



证明模型

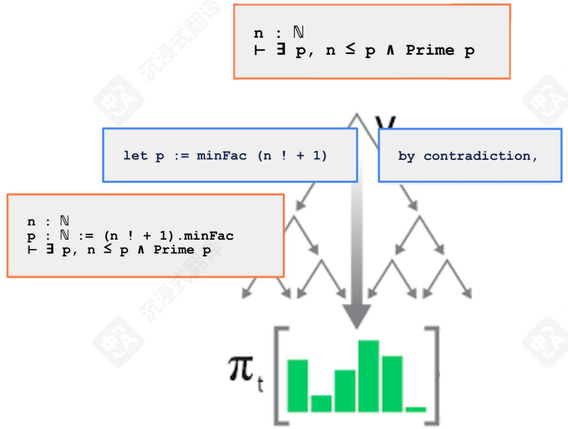


Prover Model + AlphaZero Search

Search over actions = Lean tactic

Compute new Lean state after every action / tactic application

Exploit high prior and high value paths
Explore low visited paths

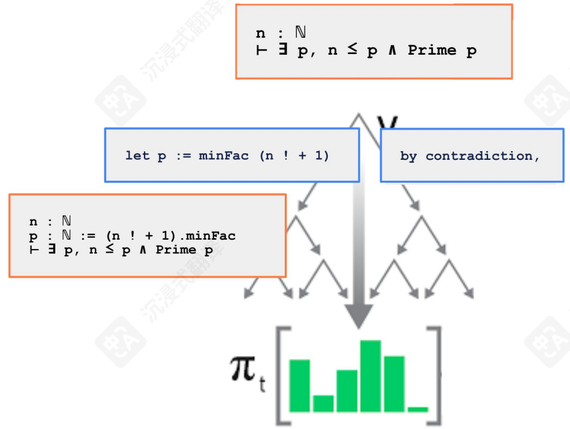


证明者模型 + AlphaZero 搜索

在 动作 = Lean 策略

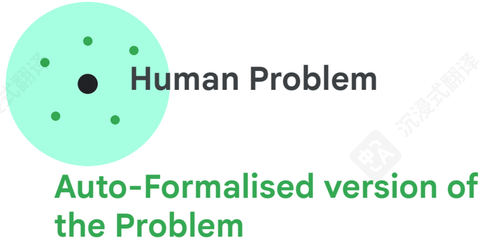
计算新的 Lean 状态在每次动作 / 策略应用后

利用高先验和高价值路径 探索低访问路径



Step 1: Auto formalisation

1: Train a formalisation model and auto-formalise human created problems

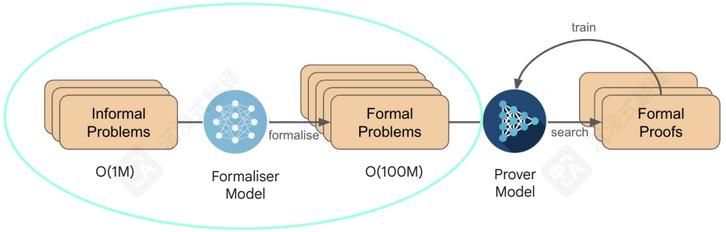


步骤 1：自动格式化

1: 训练一个格式化模型并自动格式化人类创建的问题

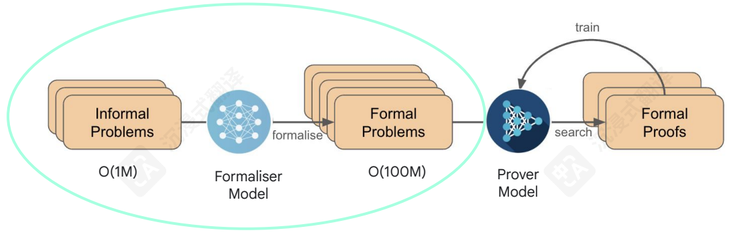
Step 1: Auto formalisation

1: Train a statement formalisation model and auto-formalise human created problems



步骤 1： 自动格式化

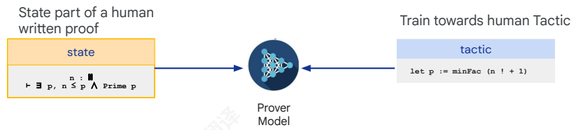
1: 训练一个陈述格式化模型并自动格式化人类创建的问题



Step 2: Build on top of Mathlib

2: Train the prover model supervised on Mathlib

- 100k definitions
- 200k theorems
- 300k lines of proofs

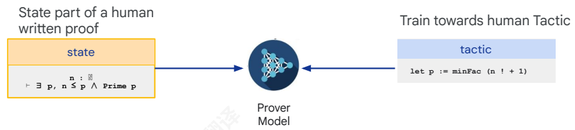


Learn a good prior of actions to take.

Step 2: 基于 Mathlib 进行构建

2: 在 Mathlib 上进行监督训练证明者模型

- 100k 个定义 - 200k 个定理 - 300k 行证明

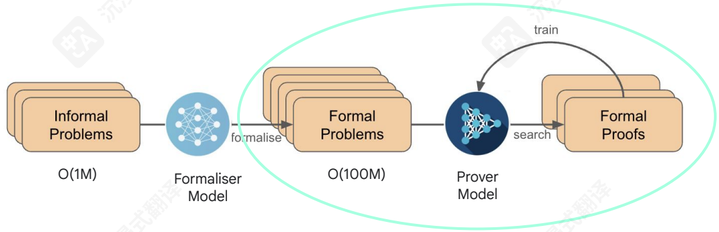


学习采取行动的良好先验。

Step 3: AlphaZero Reinforcement Learning

3: Train the prover model by RL

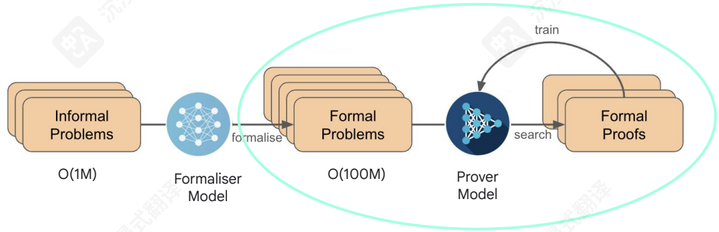
For each formal problem
- Generate experience of (dis)proving by searching over Lean steps.
- Use Lean to verify proofs
- Reinforce the prover network with each success



Step 3: AlphaZero 强化学习

3: 通过强化学习训练证明者模型

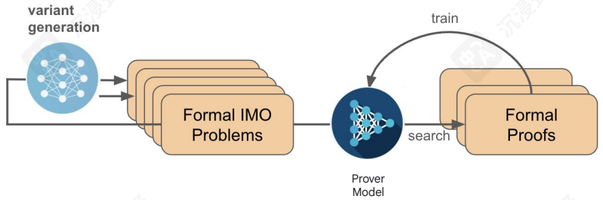
对于每个形式化问题
- 通过搜索 Lean 步骤生成 (反) 证明经验。 - 使用 Lean 验证证明 - 每次成功时强化证明网络



Final Step: Test-Time RL

4: Train the prover model on specific problems by RL

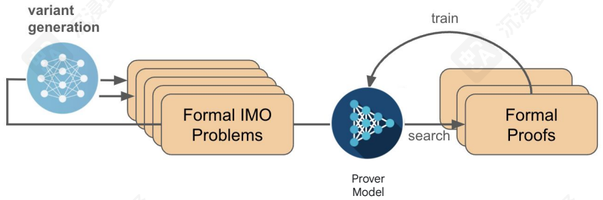
For each problem:
- Generate variants of the problem
- Run RL exactly as previous step



Final Step: 测试时强化学习

4: 通过强化学习训练证明者模型在特定问题上

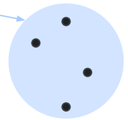
对于每个问题：
- 生成问题的变体
- 按照上一步完全运行强化学习



Final Step: Test-Time RL

Very hard problem

Bubble of problems the agent can solve easily

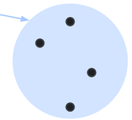


Generalist Checkpoint

Final Step: 测试时强化学习

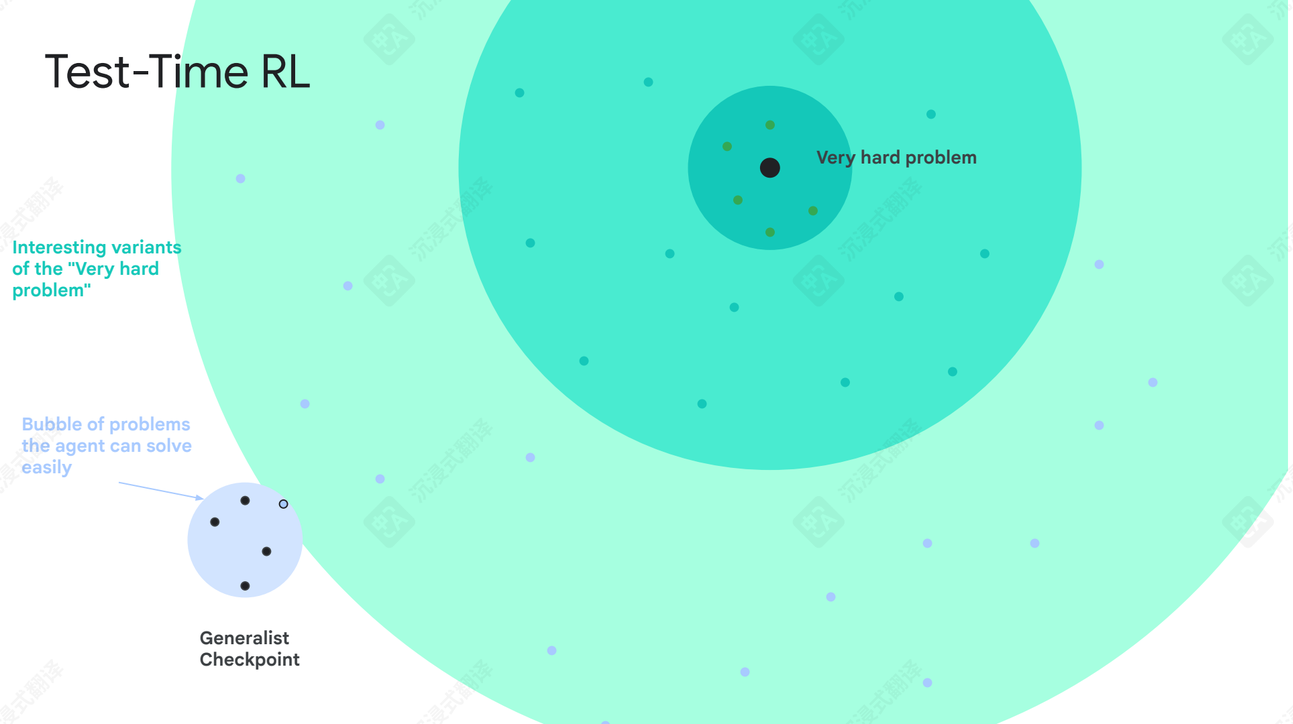
Very hard problem

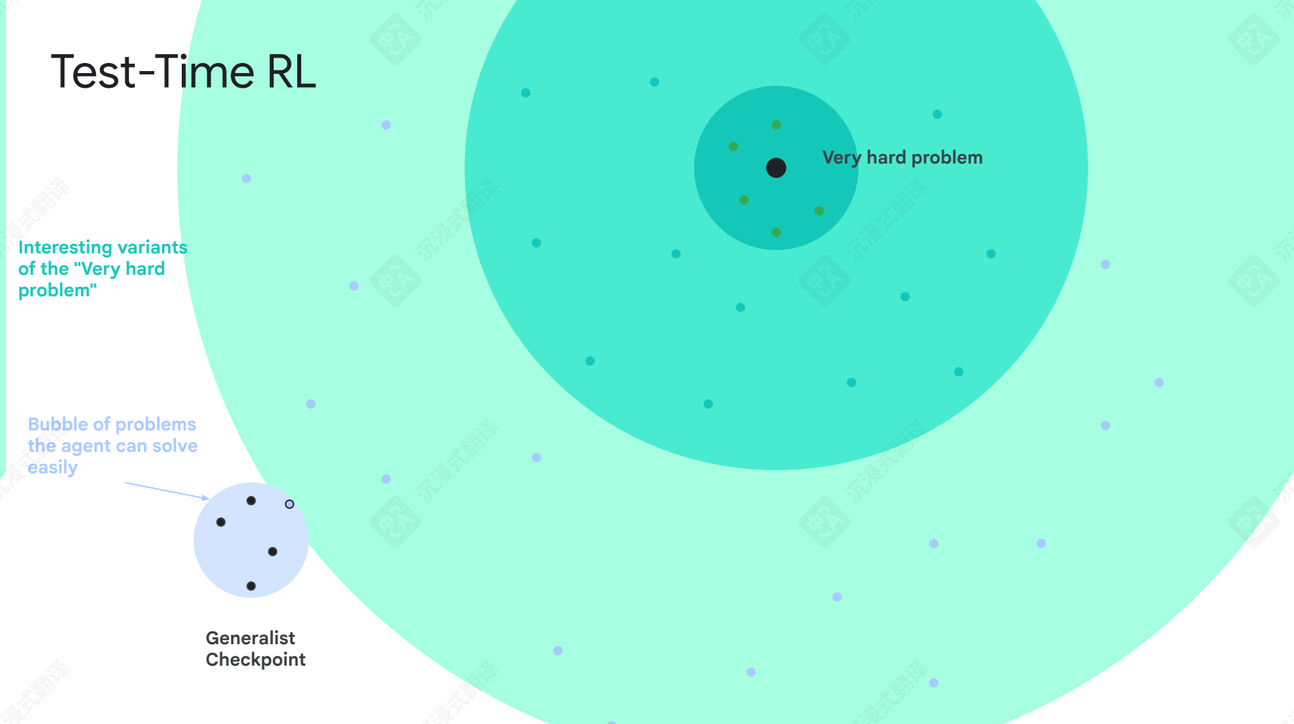
Bubble of problems the agent can solve easily

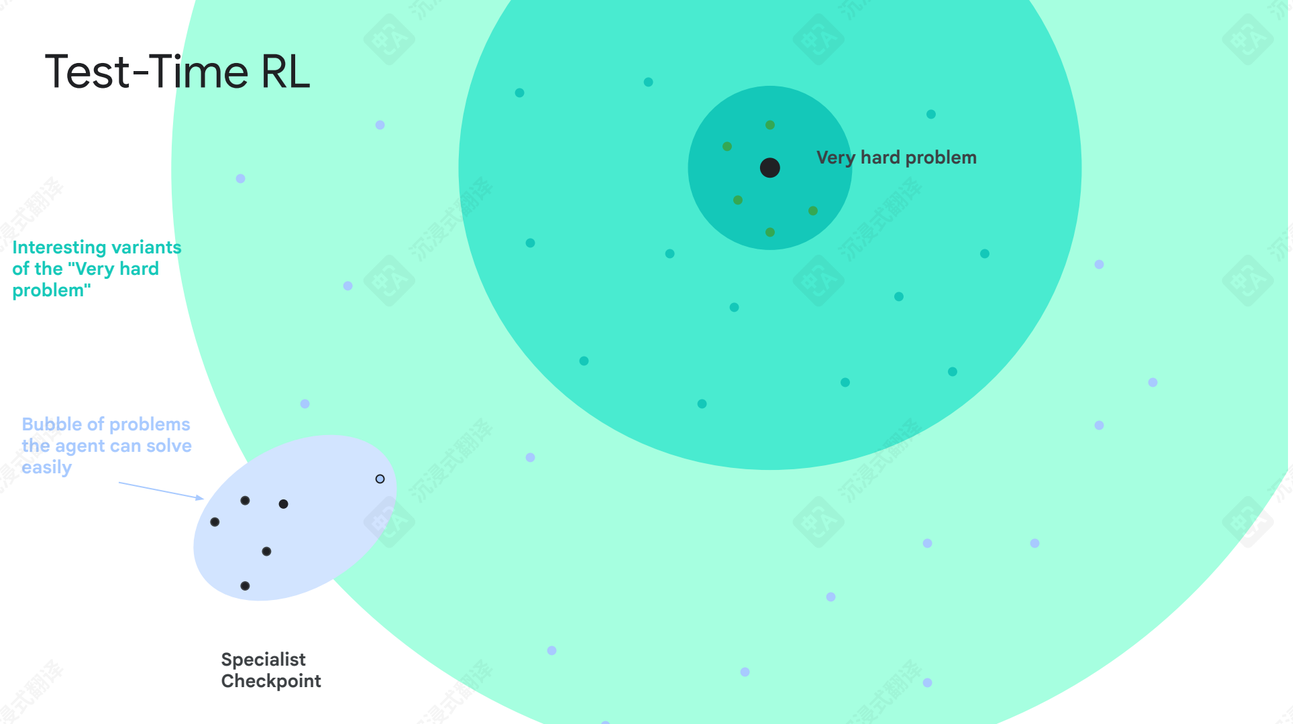


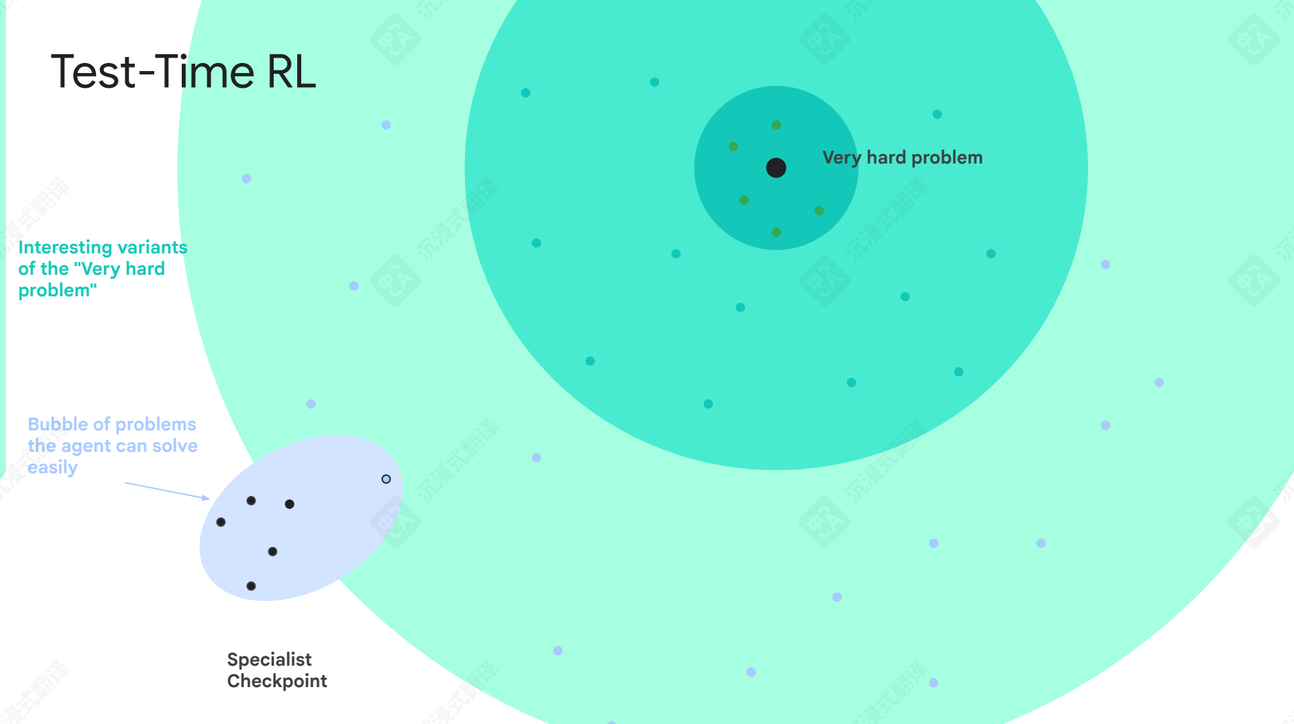
通用检查点

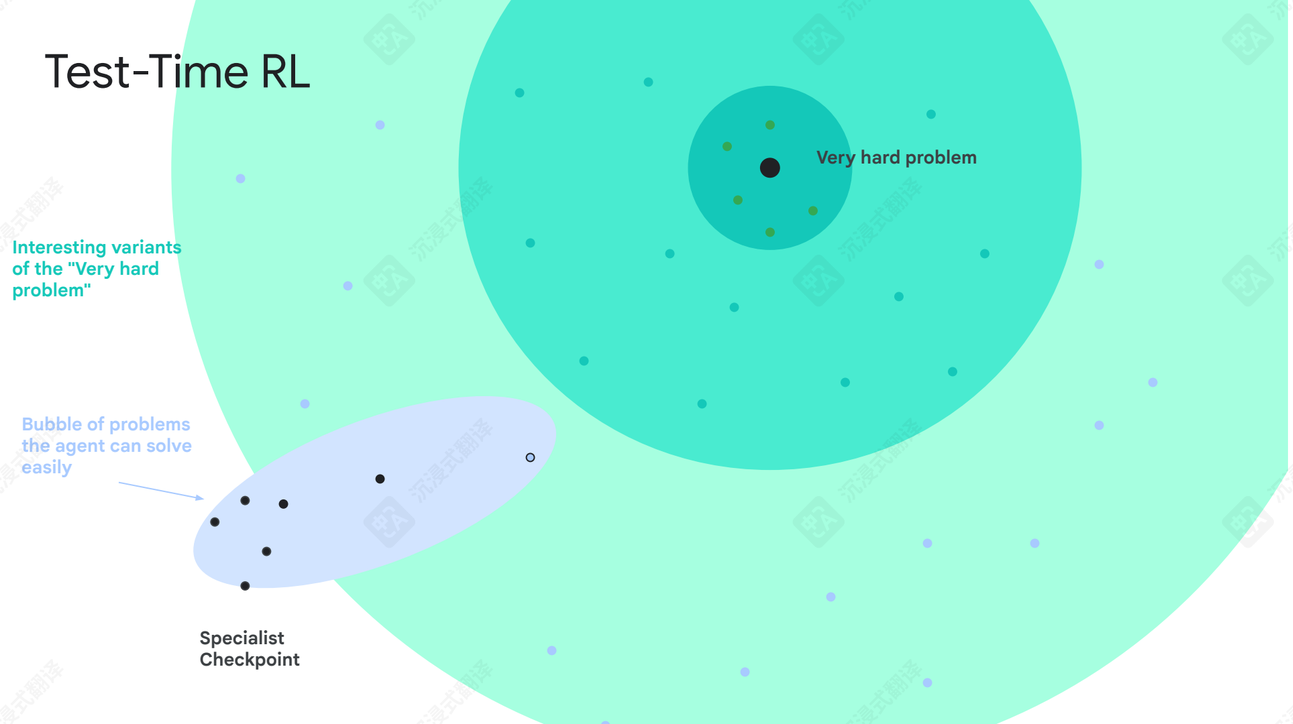
86

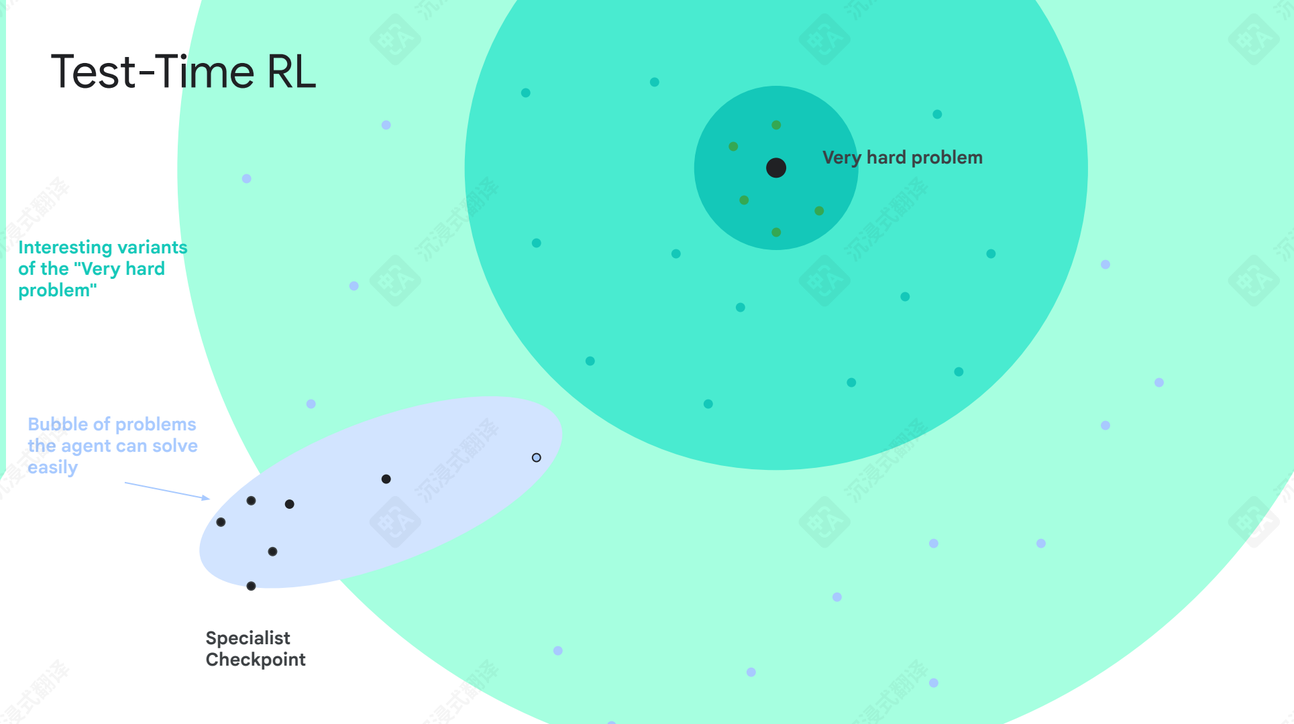

Test-Time RL
Very hard problem
Interesting variants of the "Very hard problem"
Bubble of problems the agent can solve easily
Generalist Checkpoint

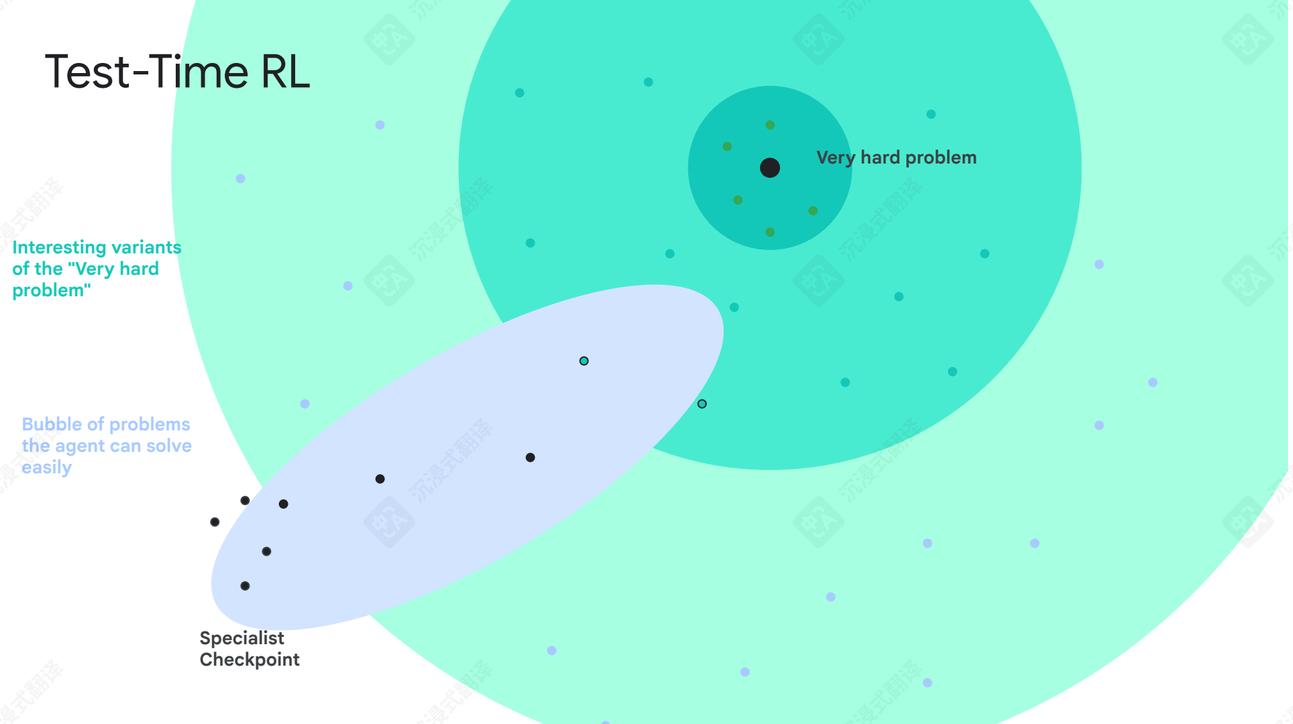

Test-Time RL
Very hard problem
Interesting variants of the "Very hard problem"
Bubble of problems the agent can solve easily
Generalist Checkpoint

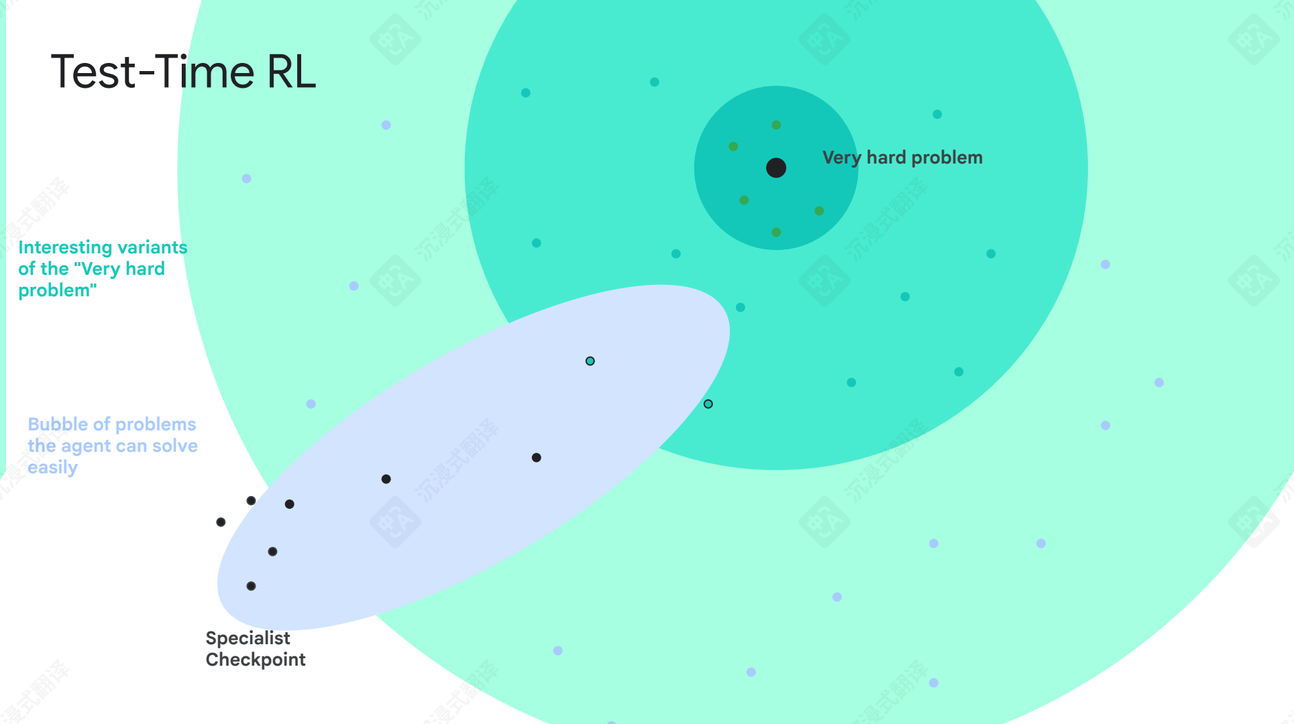

Test-Time RL
Very hard problem
Interesting variants of the "Very hard problem"
Bubble of problems the agent can solve easily
Specialist Checkpoint

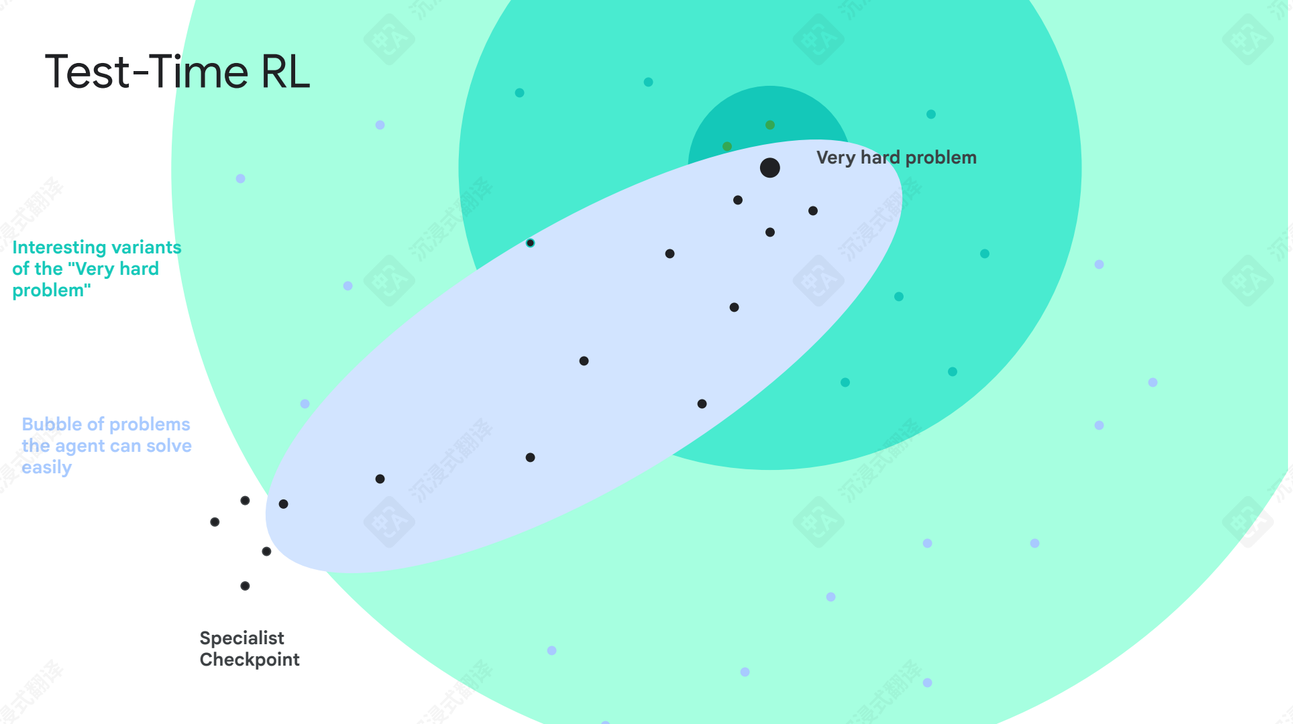

Test-Time RL
Very hard problem
Interesting variants of the "Very hard problem"
Bubble of problems the agent can solve easily
Specialist Checkpoint

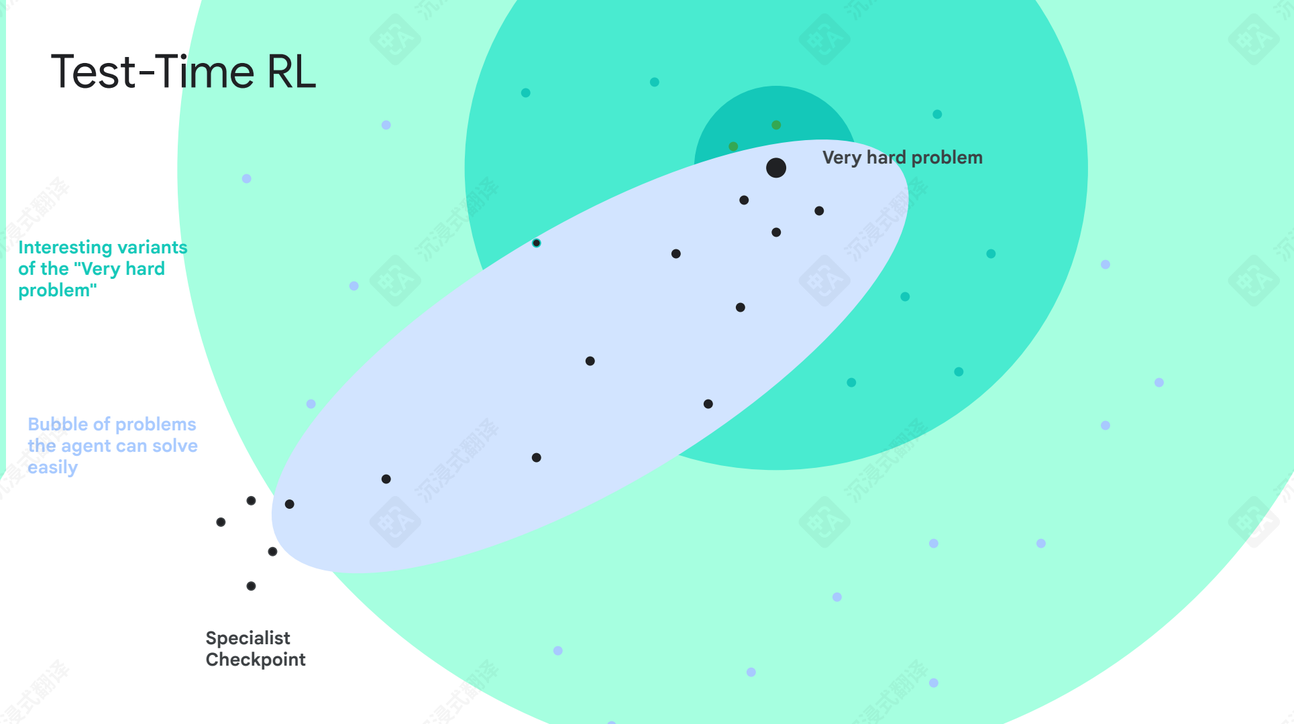

Test-Time RL
Very hard problem
Interesting variants of the "Very hard problem"
Bubble of problems the agent can solve easily
Specialist Checkpoint


Test-Time RL
Very hard problem
Interesting variants of the "Very hard problem"
Bubble of problems the agent can solve easily
Specialist Checkpoint


Test-Time RL
Very hard problem
Interesting variants of the "Very hard problem"
Bubble of problems the agent can solve easily
Specialist Checkpoint


Test-Time RL
Very hard problem
Interesting variants of the "Very hard problem"
Bubble of problems the agent can solve easily
Specialist Checkpoint


Test-Time RL
Very hard problem
Interesting variants of the "Very hard problem"
Bubble of problems the agent can solve easily
Specialist Checkpoint


Test-Time RL
Very hard problem
Interesting variants of the "Very hard problem"
Bubble of problems the agent can solve easily
Specialist Checkpoint

Challenges for AlphaProof

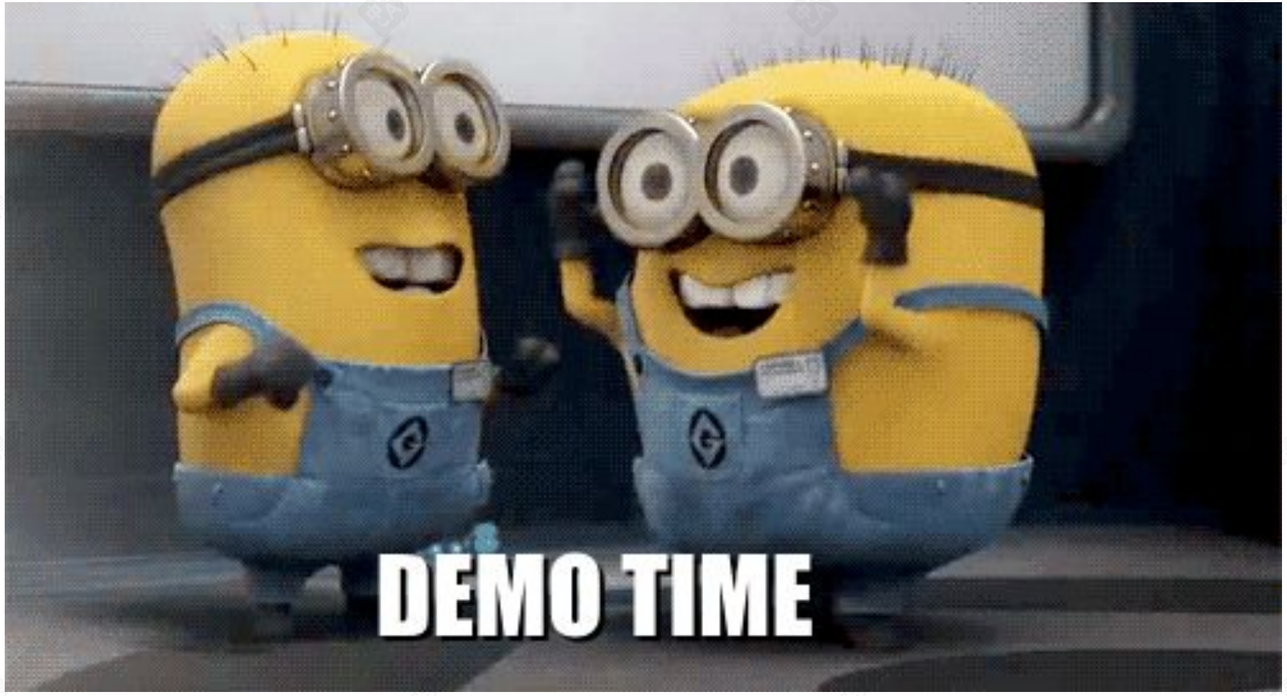
Inherited challenges from Formal Mathematics
- Most of human data in natural language
- Cannot easily learn and work on areas not supported by Mathlib.

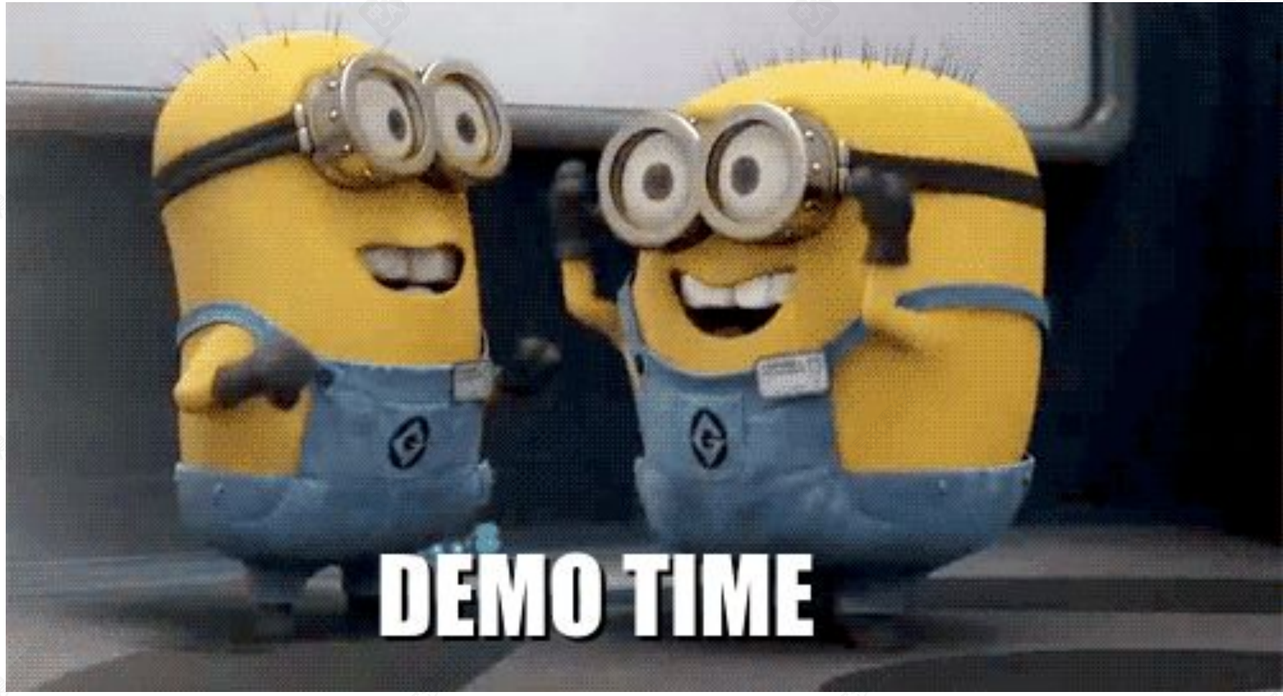
Generally,
- Creative building of new objects and theories, interestingness and beauty in mathematics

AlphaProof 的挑战

从形式数学继承的挑战
- 大部分人类自然语言数据 - 无法轻松学习和处理 Mathlib 不支持领域的知识。

通常来说，- 创造性地构建新对象和理论，数学中的趣味性和美感


DEMO TIME


DEMO TIME

Warm up: back to the infinitude of primes

Warm up: 回归素数的无限性

AIME 2020 ii p10

Problem

Find the sum of all positive integers n such that when $1^3+2^3+3^3+\cdots+n^3$ is divided by $n+5$, the remainder is 17.

Solution 1

The formula for the sum of cubes, also known as Nicomachus's Theorem, is as follows:

$$1^3+2^3+3^3+\cdots+k^3=(1+2+3+\cdots+k)^2=\left(\frac{k(k+1)}{2}\right)^2$$

for any positive integer k.

So let's apply this to this problem.

Let $m=n+5$. Then we have

$$\begin{aligned}1^3+2^3+3^3+\cdots+(m-5)^3&\equiv 17 \mod m\\ \left(\frac{(m-5)(m-4)}{2}\right)^2&\equiv 17 \mod m\\ \left(\frac{m(m-9)+20}{2}\right)^2&\equiv 17 \mod m\\ (m(m-9)+20)^2&\equiv 4\cdot 17 \mod m\\ (20)^2&\equiv 68 \mod m\\ 332&\equiv 0 \mod m\end{aligned}$$

So, $m\in\{83,166,332\}$. Testing the cases, only 332 fails. This leaves $78+161=\boxed{239}$.

AIME 2020 ii p10

Problem

Find the sum of all positive integers n such that when $1^3+2^3+3^3+\cdots+n^3$ is divided by $n+5$, the remainder is 17.

Solution 1

The formula for the sum of cubes, also known as Nicomachus's Theorem, is as follows:

$$1^3+2^3+3^3+\cdots+k^3=(1+2+3+\cdots+k)^2=\left(\frac{k(k+1)}{2}\right)^2$$

for any positive integer k.

So let's apply this to this problem.

Let $m=n+5$. Then we have

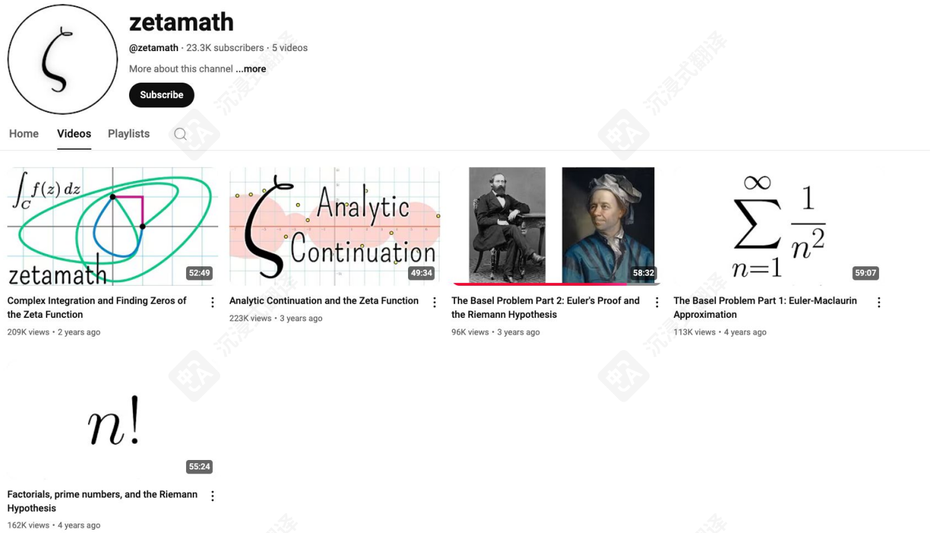
$$\begin{aligned}1^3+2^3+3^3+\cdots+(m-5)^3&\equiv 17 \mod m\\ \left(\frac{(m-5)(m-4)}{2}\right)^2&\equiv 17 \mod m\\ \left(\frac{m(m-9)+20}{2}\right)^2&\equiv 17 \mod m\\ (m(m-9)+20)^2&\equiv 4\cdot 17 \mod m\\ (20)^2&\equiv 68 \mod m\\ 332&\equiv 0 \mod m\end{aligned}$$

So, $m\in\{83,166,332\}$. Testing the cases, only 332 fails. This leaves $78+161=\boxed{239}$.

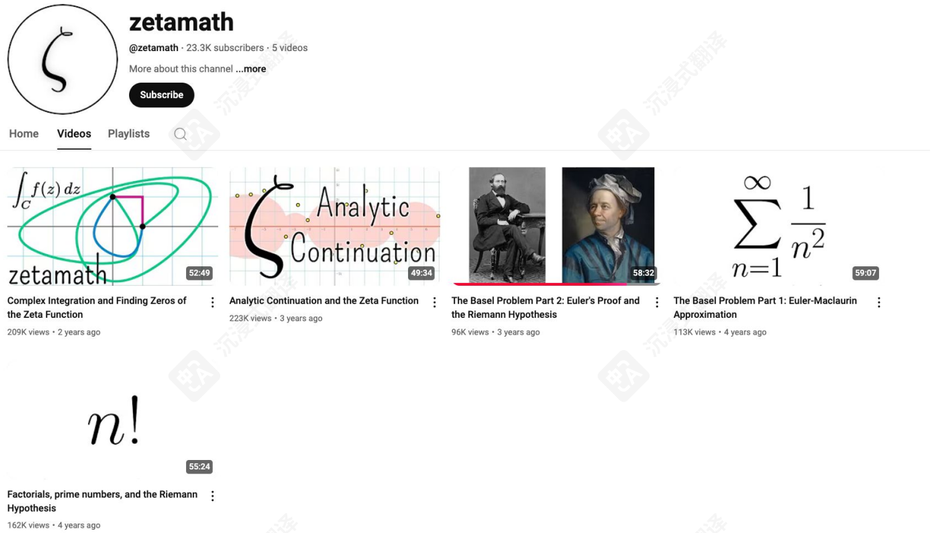
Warm up: back to the infinitude of primes

Warm up: 回归素数的无限性

Demo: Link between the Riemann Zeta and the primes



Demo：黎曼ζ函数与质数之间的联系



Link between the Riemann Zeta

$$\sum_{n=1}^{\infty} \frac{1}{n} = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \frac{1}{11} + \frac{1}{12} + \frac{1}{13} + \frac{1}{14} + \frac{1}{15} + \cdots$$

黎曼ζ函数之间的联系

$$\sum_{n=1}^{\infty} \frac{1}{n} = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \frac{1}{11} + \frac{1}{12} + \frac{1}{13} + \frac{1}{14} + \frac{1}{15} + \cdots$$

$$\sum_{n=1}^{\infty} \frac{1}{n} = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{2^2} + \frac{1}{5} + \frac{1}{2 \cdot 3} + \frac{1}{7} + \frac{1}{2^3} + \frac{1}{3^2} + \frac{1}{2 \cdot 5} + \frac{1}{11} + \frac{1}{2^2 \cdot 3} + \frac{1}{13} + \frac{1}{2 \cdot 7} + \frac{1}{3 \cdot 5} + \cdots$$

$$\sum_{n=1}^{\infty} \frac{1}{n} = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{2^2} + \frac{1}{5} + \frac{1}{2 \cdot 3} + \frac{1}{7} + \frac{1}{2^3} + \frac{1}{3^2} + \frac{1}{2 \cdot 5} + \frac{1}{11} + \frac{1}{2^2 \cdot 3} + \frac{1}{13} + \frac{1}{2 \cdot 7} + \frac{1}{3 \cdot 5} + \cdots$$

$$\left(1+\frac{1}{2}+\frac{1}{2^2}+\frac{1}{2^3}+\ldots\right)$$
$$\cdot\left(1+\frac{1}{3}+\frac{1}{3^2}+\frac{1}{3^3}+\ldots\right)$$
$$\cdot\left(1+\frac{1}{5}+\frac{1}{5^2}+\frac{1}{5^3}+\ldots\right)$$
$$\cdot\ \ldots$$

$$\left(1+\frac{1}{2}+\frac{1}{2^2}+\frac{1}{2^3}+\ldots\right)$$
$$\cdot\left(1+\frac{1}{3}+\frac{1}{3^2}+\frac{1}{3^3}+\ldots\right)$$
$$\cdot\left(1+\frac{1}{5}+\frac{1}{5^2}+\frac{1}{5^3}+\ldots\right)$$
$$\cdot\ \ldots$$

$$\prod_{p} \left(1 + \frac{1}{p} + \frac{1}{p^2} + \frac{1}{p^3} + \dots\right) = \sum_{n=1}^{\infty} \frac{1}{n}$$

$$\prod_{p} \left(1 + \frac{1}{p} + \frac{1}{p^2} + \frac{1}{p^3} + \dots\right) = \sum_{n=1}^{\infty} \frac{1}{n}$$

$$\prod_{p} \frac{1}{1-\frac{1}{p}} = \sum_{n=1}^{\infty} \frac{1}{n}$$

$$\prod_{p} \frac{1}{1-\frac{1}{p}} = \sum_{n=1}^{\infty} \frac{1}{n}$$

The Riemann zeta function

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} = \prod_{p} \left(1 + \frac{1}{p^s} + \frac{1}{p^{2s}} + \dots\right)$$

黎曼ζ函数

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} = \prod_{p} \left(1 + \frac{1}{p^s} + \frac{1}{p^{2s}} + \dots\right)$$

Let's prove the finite case

$$\left(1+\frac{1}{2}+\frac{1}{2^2}+\frac{1}{2^3}+\dots\right)$$
$$\cdot\left(1+\frac{1}{3}+\frac{1}{3^2}+\frac{1}{3^3}+\dots\right)$$
$$\cdot\left(1+\frac{1}{5}+\frac{1}{5^2}+\frac{1}{5^3}+\dots\right)$$

让我们证明有限情况

$$\left(1+\frac{1}{2}+\frac{1}{2^2}+\frac{1}{2^3}+\dots\right)$$
$$\cdot\left(1+\frac{1}{3}+\frac{1}{3^2}+\frac{1}{3^3}+\dots\right)$$
$$\cdot\left(1+\frac{1}{5}+\frac{1}{5^2}+\frac{1}{5^3}+\dots\right)$$

Final Link with the root of the zeta function

$$\zeta(s) = \frac{e^{a+bs}}{s-1} \prod_{\zeta(\alpha)=0} \left(1 - \frac{s}{\alpha}\right) e^{s/\alpha}$$

Final Link with the root of the zeta function

$$\zeta(s) = \frac{e^{a+bs}}{s-1} \prod_{\zeta(\alpha)=0} \left(1 - \frac{s}{\alpha}\right) e^{s/\alpha}$$

Final Link with the root of the zeta function

$$\zeta(s) = \prod_{p} \frac{1}{1 - p^{-s}}$$

$$\zeta(s) = \frac{e^{a+bs}}{s-1} \prod_{\zeta(\alpha)=0} \left(1 - \frac{s}{\alpha}\right) e^{s/\alpha}$$

Final Link with the root of the zeta function

$$\zeta(s) = \prod_{p} \frac{1}{1 - p^{-s}}$$

$$\zeta(s) = \frac{e^{a+bs}}{s-1} \prod_{\zeta(\alpha)=0} \left(1 - \frac{s}{\alpha}\right) e^{s/\alpha}$$

Final Link with the root of the zeta function

$$\sum_{p^k \leq x} \log(p) = x - \log(2\pi) - \sum_{\zeta(\alpha)=0} \frac{x^\alpha}{\alpha}$$

Final Link with the root of the zeta function

$$\sum_{p^k \leq x} \log(p) = x - \log(2\pi) - \sum_{\zeta(\alpha)=0} \frac{x^\alpha}{\alpha}$$


What’s next
接下来是什么

What's Next for AlphaProof?

Broaden to the entire Mathematical landscape

Contribute to the Frontiers of Research Math

AlphaProof as a useful tool for every thinker

AlphaProof 的下一步是什么?

扩展到整个数学领域

为研究数学的前沿做出贡献

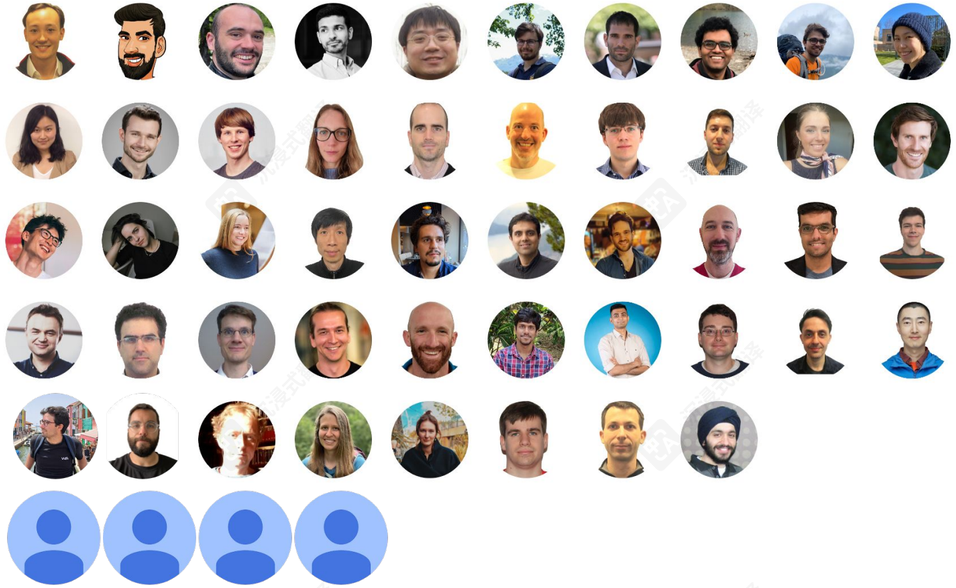
AlphaProof 作为每个思考者的有用工具

Individually, this was almost impossible

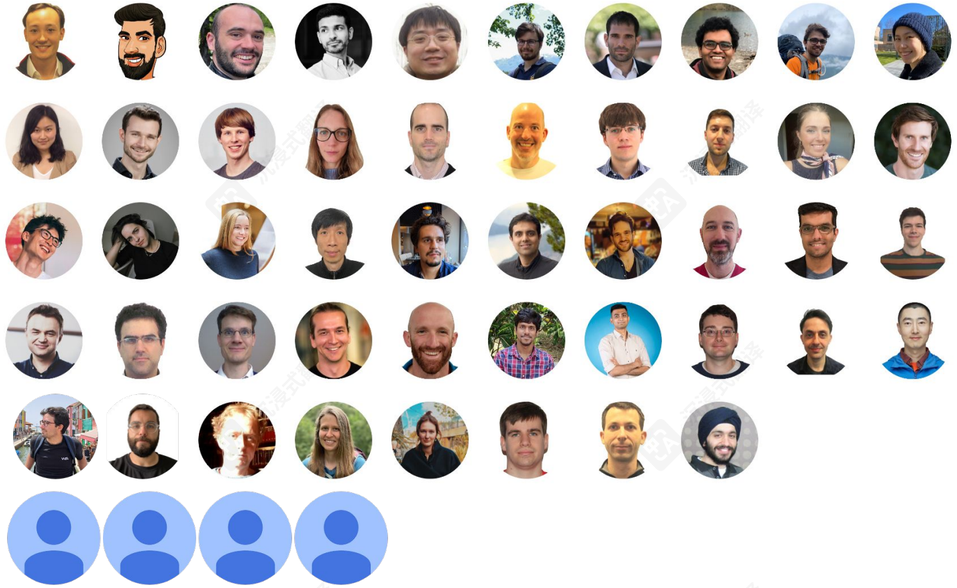
Together, it felt impossible to fail

单独来看，这几乎不可能

一起的话，感觉不可能失败



Google DeepMind



Google DeepMind

Thank you for Listening

Questions?

Google DeepMind

Thank you for Listening

问题?

Google DeepMind