The problem is the problem: Towards scalable mathematical discovery

Sean Welleck

Geometry of Machine Learning Workshop 2026

Carnegie Mellon University

The Problem Is the Problem: Towards Scalable Mathematical Discovery

Zeyu Zheng[1,*], **Shengtong Zhang**[2,*], **Jeremy Avigad**[1], **Prasad Tetali**[1], **Sean Welleck**[1]
[1]Carnegie Mellon University [2]Anysphere Co.

https://arxiv.org/abs/2608.16977

Zeyu Zheng

Shengtong Zhang

Jeremy Avigad

Prasad Tetali

Sean Welleck

If we give AI a problem, it can often help us solve it

2021: word problems

October 29, 2021 Publication

Solving math word problems

Read paper ↗ Download dataset ↗

OpenAI - October 29, 2021

If we give AI a problem, it can often help us solve it

2022: simple proofs

NaturalProver: Grounded Mathematical Proof Generation with Language Models

Let x be an even integer. Then $x + 5$ is odd.

Generated Proof

Proof by Contradiction: Aiming for a contradiction, suppose $x + 5$ is even.

Then there exists an integer k such that $x + 5 = 2k$.

This contradicts the premise that x is even.

Hence, by Proof by Contradiction, $x + 5$ is odd.

Welleck et al. - NeurIPS 2022, Figure 1

If we give AI a problem, it can often help us solve it

2026: open problems

September 8, 2026 Research Publication

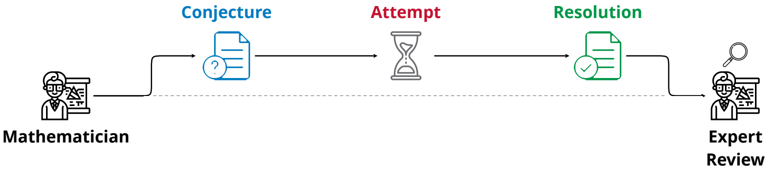
On the Navier–Stokes Millennium Prize Problem

Read the paper Link to Lean formalized proof >

OpenAI - September 8, 2026

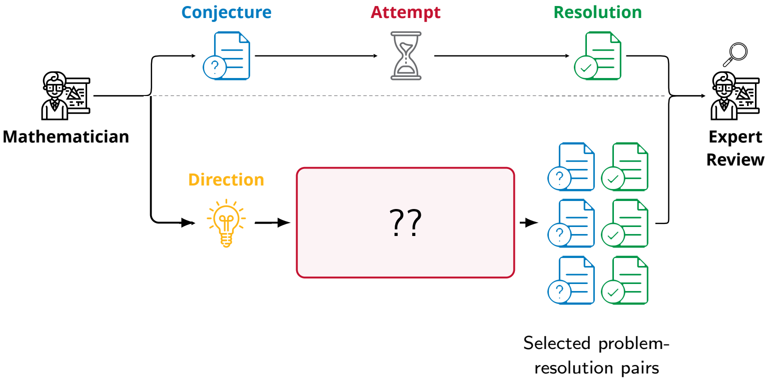
From choosing a problem to choosing a direction

Problem-level interface



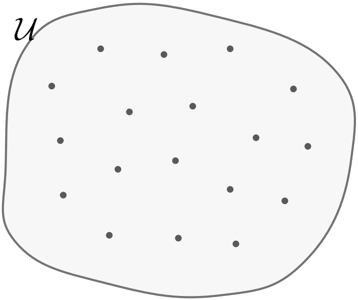
From choosing a problem to choosing a direction

Problem-level interface



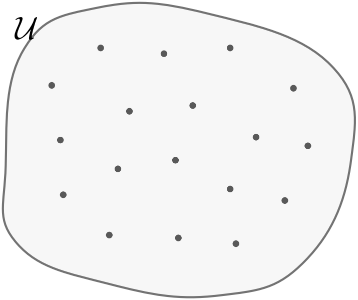
Our goal: *Direction-level* interface

Discovery as effort allocation



$\mathcal{U}$: space of all problems expressible in natural language

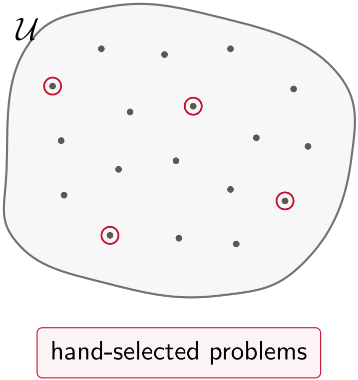
Discovery as effort allocation



$\mathcal{U}$: space of all problems expressible in natural language

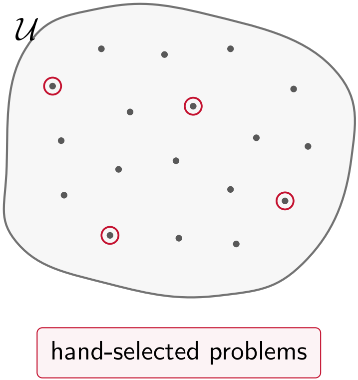
- How should we allocate human and AI effort?

Discovery as effort allocation



Option 1: have AI attempt hand-selected problems, humans review the results

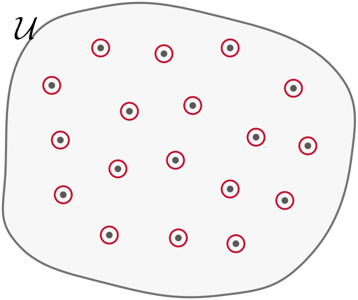
Discovery as effort allocation



Option 1: have AI attempt hand-selected problems, humans review the results

- Effort is concentrated on a small set of problems
- Problems may be too difficult for the current AI

Discovery as effort allocation

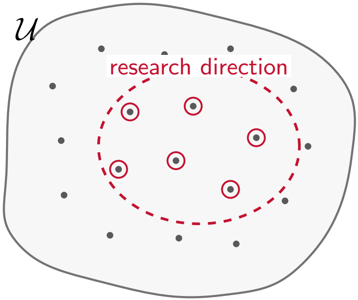


AI attempts every problem

Option 2: have AI attempt all problems, humans review the results

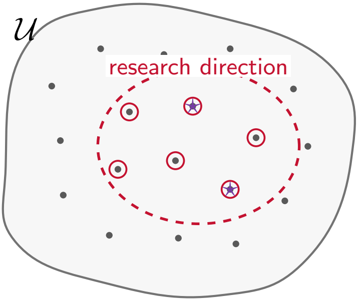
- Bottlenecks: AI effort, human effort

Discovery as effort allocation



Our goal: given a research direction, find relevant problems

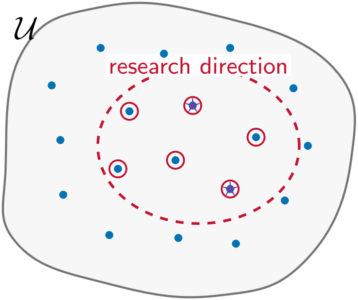
Discovery as effort allocation



Our goal: given a research direction, find relevant problems

- Humans review the most promising ones

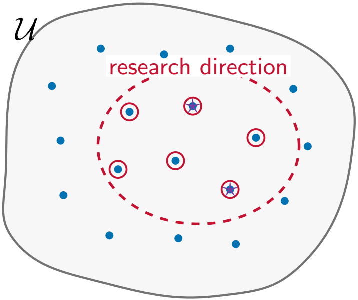
Discovery as effort allocation



Our goal: given a research direction, find relevant problems

- Humans review the most promising ones
- Which space of problems should the system consider?

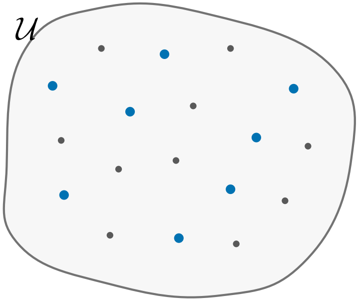
Discovery as effort allocation



Our goal: given a research direction, find relevant problems

- Humans review the most promising ones
- Which space of problems should the system consider?
- Which problems should the system recommend to humans?

What kinds of problems?

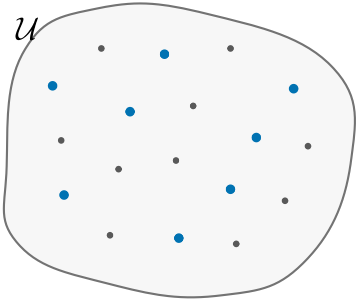


- AI-generated problems

Option 1: have AI propose problems

- Automated conjecturing

What kinds of problems?

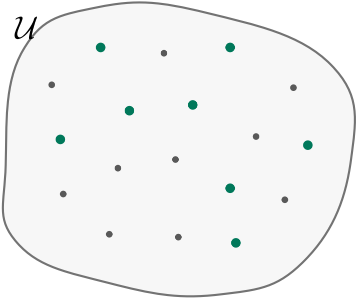


- AI-generated problems

Option 1: have AI propose problems

- Automated conjecturing
- Might not be interesting

What kinds of problems?

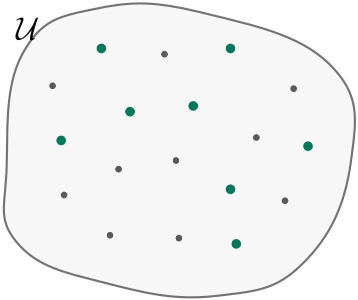


- Problems from the literature

Option 2: select problems from the literature

- Likely to be interesting

What kinds of problems?

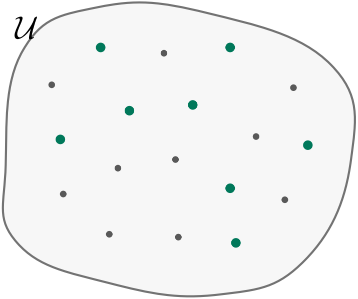


- Problems from the literature

Option 2: select problems from the literature

- Likely to be interesting
- Still a lot

What kinds of problems?



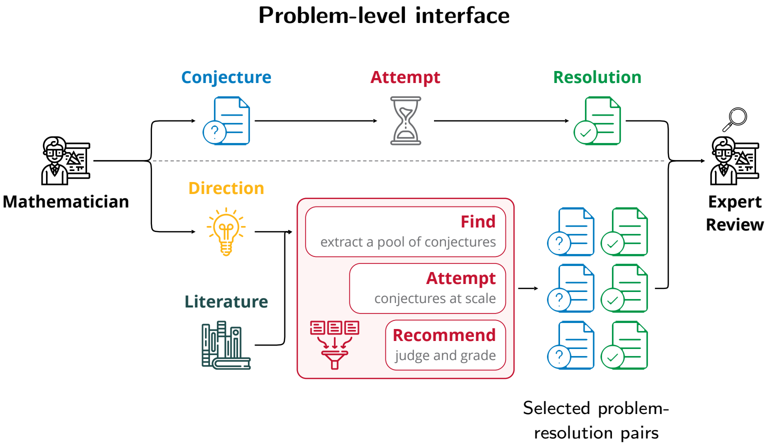
- Problems from the literature

Option 2: select problems from the literature

- Likely to be interesting
- Still a lot

Our goal: build a recommendation engine

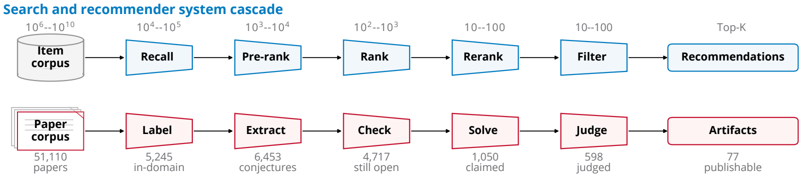
Find, Attempt, Recommend (FAR)



Find, Attempt, Recommend (FAR)

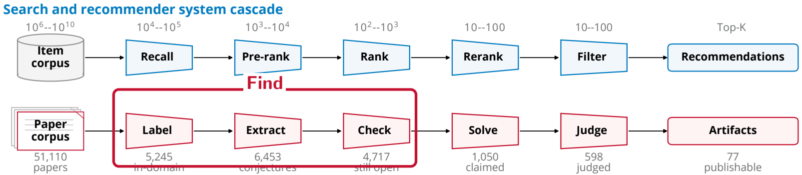
Building FAR

- An agent cascade: from cheap to expensive



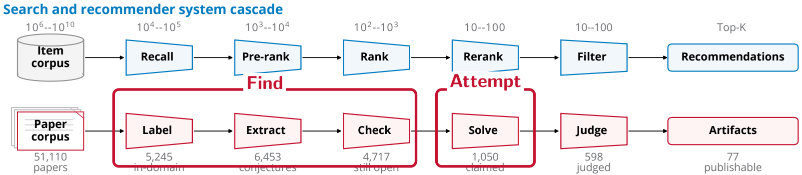
Building FAR

- An agent cascade: from cheap to expensive



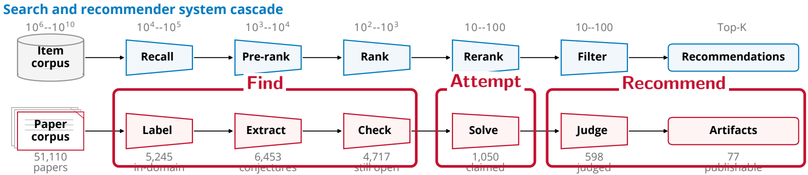
Building FAR

- An agent cascade: from cheap to expensive



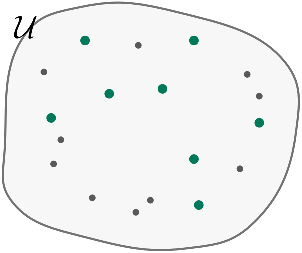
Building FAR

- An agent cascade: from cheap to expensive



Find stage

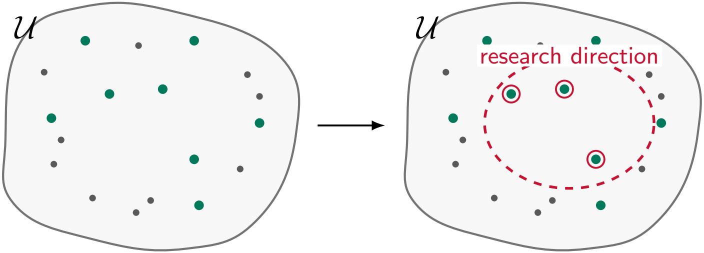
Input: a literature corpus and a research direction



- Problems from the literature

Find stage

Input: a literature corpus and a research direction



- Problems from the literature

◦ Pool $\mathcal{P}$

Output: pool of problems to potentially attempt ($\mathcal{P}$)

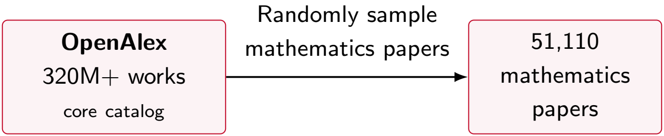
Find: our example run for combinatorics conjectures

OpenAlex

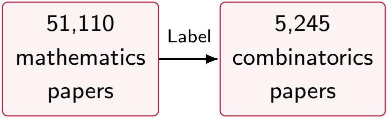
320M+ works

core catalog

Find: our example run for combinatorics conjectures

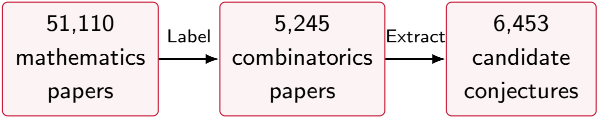


Find: our example run for combinatorics conjectures



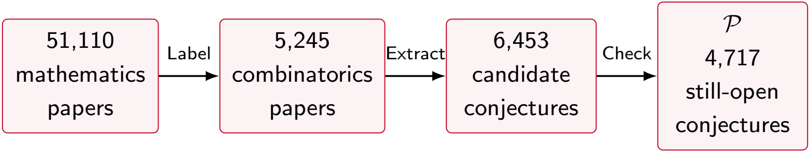
- **Label:** cheap LLM

Find: our example run for combinatorics conjectures



- **Label:** cheap LLM
- **Extract:** LLM extracts unresolved statements

Find: our example run for combinatorics conjectures

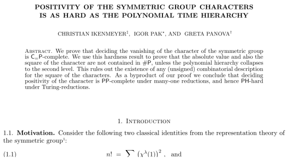


- **Label:** cheap LLM
- **Extract:** LLM extracts unresolved statements
- **Check:** agent with web search

Find: our example run for combinatorics conjectures

Source paper

Ikenmeyer, Pak, and Panova, *Positivity of the Symmetric Group Characters is as Hard as the Polynomial Time Hierarchy*


POSITIVITY OF THE SYMMETRIC GROUP CHARACTERS IS AS HARD AS THE POLYNOMIAL TIME HIERARCHY

CHRISTIAN IKENMEYER[‡], IGOR PAK[*], AND GRETA PANOVA[†]

Abstract. We prove that deciding the vanishing of the character of the symmetric group is $\mathsf{C_{=}P}$-complete. We use this hardness result to prove that the absolute value and also the square of the character are not contained in #P, unless the polynomial hierarchy collapses to the second level. This rules out the existence of any (unsigned) combinatorial description for the square of the characters. As a byproduct of our proof we conclude that deciding positivity of the character is PP-complete under many-one reductions, and hence PH-hard under Turing-reductions.

1. Introduction

1.1. **Motivation.** Consider the following two classical identities from the representation theory of the symmetric group[1]:

(1.1) $$n! = \sum (\chi^\lambda(1))^2, \quad \text{and}$$

Extracted candidate

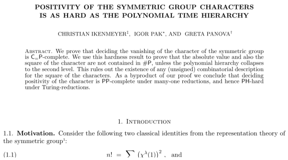
Label: Conjecture 5.2

Statement: The problem $\textsc{ComputeCharBinary}$ is GapP-complete under many-one reductions.

Find: our example run for combinatorics conjectures

Source paper

Ikenmeyer, Pak, and Panova, *Positivity of the Symmetric Group Characters is as Hard as the Polynomial Time Hierarchy*



POSITIVITY OF THE SYMMETRIC GROUP CHARACTERS IS AS HARD AS THE POLYNOMIAL TIME HIERARCHY

CHRISTIAN IKENMEYER[‡], IGOR PAK[*], AND GRETA PANOVA[†]

ABSTRACT. We prove that deciding the vanishing of the character of the symmetric group is $\mathsf{C_{=}P}$-complete. We use this hardness result to prove that the absolute value and also the square of the character are not contained in #P, unless the polynomial hierarchy collapses to the second level. This rules out the existence of any (unsigned) combinatorial description for the square of the characters. As a byproduct of our proof we conclude that deciding positivity of the character is PP-complete under many-one reductions, and hence PH-hard under Turing-reductions.

1. INTRODUCTION

1.1. **Motivation.** Consider the following two classical identities from the representation theory of the symmetric group[1]:

(1.1) $$n! = \sum (\chi^\lambda(1))^2, \text{ and}$$

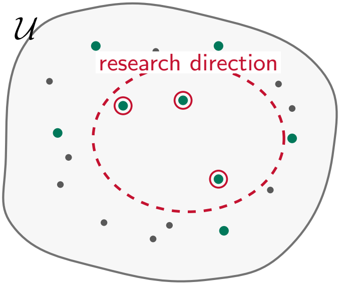
Extracted candidate

Label: Conjecture 5.2

Statement: The problem ComputeCharBinary is GapP-complete under many-one reductions.

Status: Open. The check found no credible resolution.

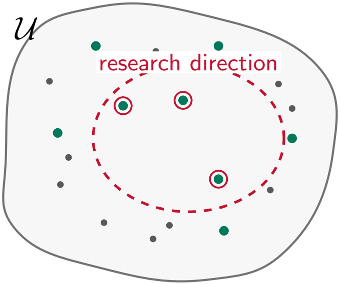
Attempt stage



◦ Pool $\mathcal{P}$

- Given a pool of problems and a budget of attempts, attempt to solve some of the problems

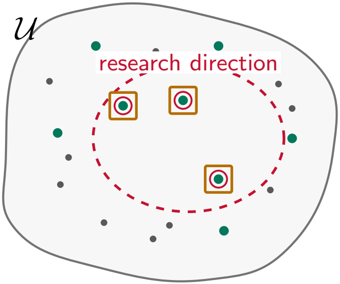
Attempt stage



○ Pool $\mathcal{P}$

- Given a pool of problems and a budget of attempts, attempt to solve some of the problems
- Similar to a bandit problem: each problem is an arm and each attempt is a pull

Attempt stage



○ Pool $\mathcal{P}$

□ Problem to attempt

- Given a pool of problems and a budget of attempts, attempt to solve some of the problems
- Similar to a bandit problem: each problem is an arm and each attempt is a pull
- Initial uninformed strategy: just try each problem once

Attempt: our example run for combinatorics conjectures

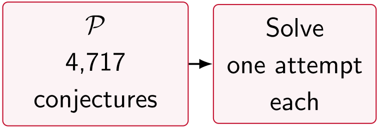
$\mathcal{P}$
4,717
conjectures

→

Solve
one attempt
each

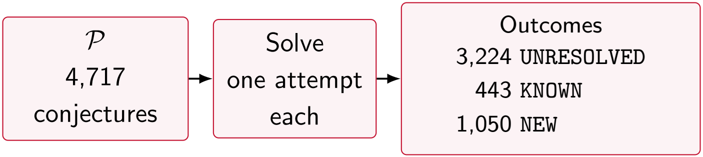
- Most powerful agent in the pipeline: GPT-5.5 with web search

Attempt: our example run for combinatorics conjectures



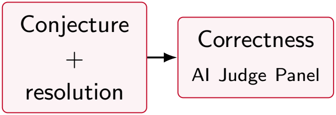
- Most powerful agent in the pipeline: GPT-5.5 with web search
    - Performs another literature review to check for overlaps

Attempt: our example run for combinatorics conjectures

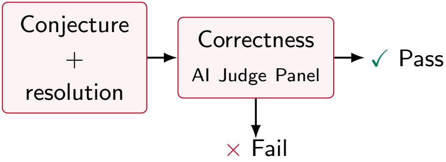


- Most powerful agent in the pipeline: GPT-5.5 with web search
  - Performs another literature review to check for overlaps

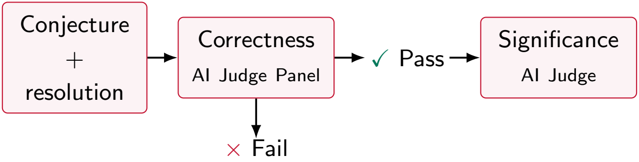
Recommend stage



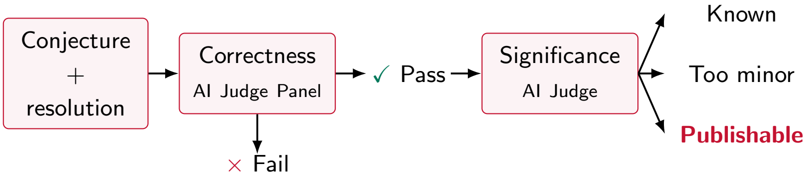
Recommend stage



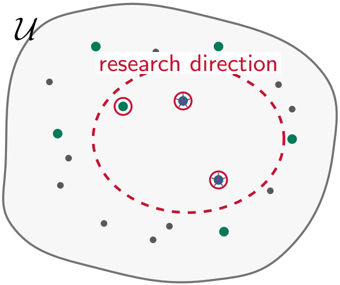
Recommend stage



Recommend stage



Recommend stage

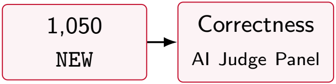


$\circ$ Pool $\mathcal{P}$

$\star$ Surfaced to the user

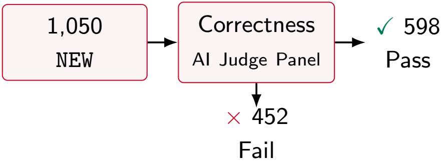
Results graded publishable are surfaced to the user

Recommend: our example run for combinatorics conjectures



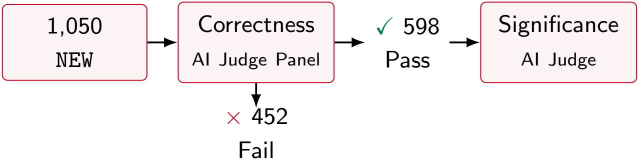
- **Correctness:** 3 independent GPT-5.5 judges (`xhigh`); all must pass

Recommend: our example run for combinatorics conjectures



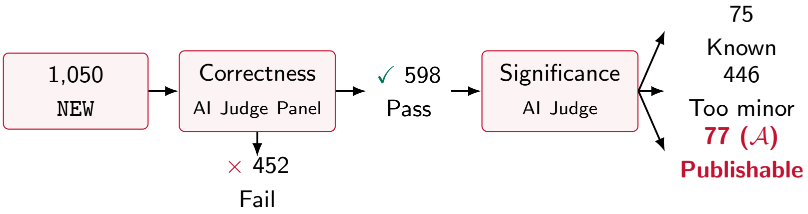
- **Correctness:** 3 independent GPT-5.5 judges (`xhigh`); all must pass

Recommend: our example run for combinatorics conjectures



- **Correctness:** 3 independent GPT-5.5 judges (`xhigh`); all must pass
- **Significance:** GPT-5.5 (`xhigh`) with web search

Recommend: our example run for combinatorics conjectures



- **Correctness:** 3 independent GPT-5.5 judges (`xhigh`); all must pass
- **Significance:** GPT-5.5 (`xhigh`) with web search

$\mathcal{A}$: conjectures with artifacts graded publishable

Studying the recommendations

- Zeyu, Shengtong, and Sylvester W. Zhang picked 15 conjecture-resolution pairs and studied the results

Studying the recommendations

- Zeyu, Shengtong, and Sylvester W. Zhang picked 15 conjecture-resolution pairs and studied the results
- Each resolution was correct

Studying the recommendations

- Zeyu, Shengtong, and Sylvester W. Zhang picked 15 conjecture-resolution pairs and studied the results
- Each resolution was correct
- Novel resolutions included conjectures of Davies-Jenssen-Perkins-Roberts, Erdős-Straus, and Ikenmeyer-Pak-Panova

C REVIEWED SOLUTIONS

This appendix collects the write-ups of the results the authors reviewed, ordered alphabetically by the mathematicians who posed the problems.

ProbXiv: view all results

ProbXiv

Search problems search

sign in

An open workspace for mathematical discovery. Check proofs, make comments, form collaborations.

Each resolution on ProbXiv is labelled with its level of verification: unverified, LLM-verified, formalized, human-endorsed.

open questions

Chen-Gendron Spin-Parity Identity for k-Differentials

posed by Dawei Chen & Quentin Gendron

“For odd k with gcd(n,k) = gcd(n+1,k) = 1, is N_k(n) ≡ ⌊ (k+1)/4 ⌋ pmod 2, where N_k(n) counts pairs 1 ≤ b_i ≤ (k-1)/2 with b_1 + b_2 ≥ (k+1)/2 and b_2 ≡ n b_1 pmod k? Conjectured by Chen and Gendron; its proof removes a…”

lean: partially-checked Lean
no ProbXiv account is credited

does the formalisation match? →

How Many Stemmata with Root Degree k?

posed by Armin Hoenen, Steffen Eger, Ralf Gehrke

“While we are not able to prove it, we think it is a very safe conjecture that R_2(m) converges to below 0.607 , as m tends toward infinity.”

model says: correct
GPT-5.5 xhigh (SMD judge 1)
no ProbXiv account is credited

say whether it holds →

RECURSIONS ASSOCIATED TO TRAPEZOID, SYMMETRIC AND ROTATION SYMMETRIC FUNCTIONS OVER GALOIS FIELDS

posed by Francis N. Castro, Robin Chapman, Luis A. Medina, L. Brehsner Sepúlveda

“Let t_k,q(n) be defined by (5.1) t_k,q(j) = q^j, for 0 ≤ j ≤ k-1 t_k,q(n) = q ∑_l=0^k-2 (q-1)^l t_k,q(n-(l+2)), for n ≥ k. Then, S_F_q(T_2,3,...,k(n)) = t_k,q(n) for all values of n ≥ k.”

model says: correct
GPT-5.5 xhigh (SMD judge 1)
no ProbXiv account is credited

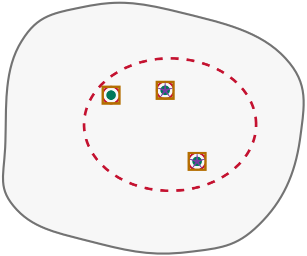
say whether it holds →

problems

https://probxiv.com

Allocating effort: scaling laws of mathematical discovery

Our run attempted every conjecture in the pool.

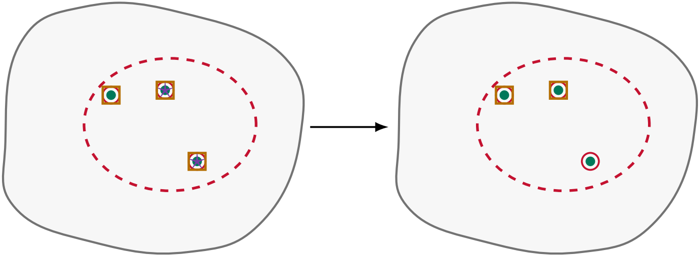


Recorded outcomes

□ Attempted problem ⋆ Graded publishable

Allocating effort: scaling laws of mathematical discovery

Our run attempted every conjecture in the pool.



Recorded outcomes

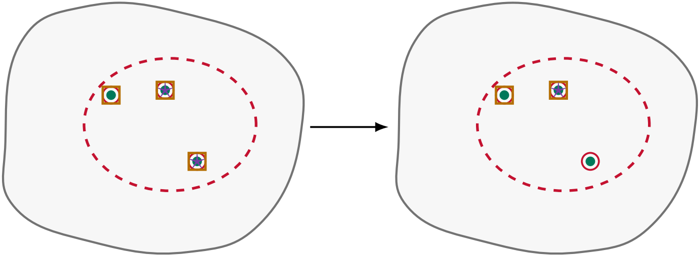
Select B attempts

□ Attempted problem ⋆ Graded publishable

- With a smaller attempt budget B, which problems should we choose?

Allocating effort: scaling laws of mathematical discovery

Our run attempted every conjecture in the pool.



Recorded outcomes

Select B attempts

□ Attempted problem ⋆ Graded publishable

- With a smaller attempt budget B, which problems should we choose?
- We consider a simple setting: 0 or 1 attempts per conjecture

Estimating the probability of a publishable result

We have the agent output a difficulty score $d(c)$ and an importance score $i(c)$ before attempting a solution.

Estimating the probability of a publishable result

We have the agent output a difficulty score $d(c)$ and an importance score $i(c)$ before attempting a solution.

Using the completed run, we fit models to predict:

1. Acceptance given difficulty $d(c)$
2. Publishability among accepted resolutions given importance $i(c)$

Estimating the probability of a publishable result

We have the agent output a difficulty score $d(c)$ and an importance score $i(c)$ before attempting a solution.

Using the completed run, we fit models to predict:

1. Acceptance given difficulty $d(c)$
2. Publishability among accepted resolutions given importance $i(c)$

Their product estimates the probability of a publishable result:

$$\hat{p}(c) = \mathrm{accept}(c) \times \mathrm{pub\text{-}given\text{-}accept}(c).$$

We hold out conjectures and use this estimate to select ones to attempt.

Discovery objectives

We can aim to maximize three different objectives:

- Expected number of publishable results

Discovery objectives

We can aim to maximize three different objectives:

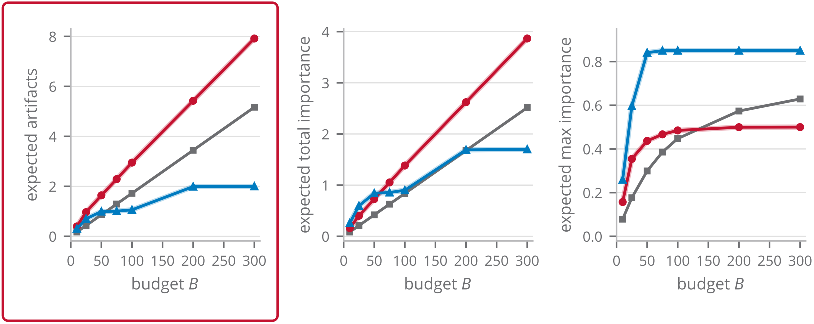
- Expected number of publishable results
- Expected total importance of publishable results

Discovery objectives

We can aim to maximize three different objectives:

- Expected number of publishable results
- Expected total importance of publishable results
- Expected importance of the single best publishable result

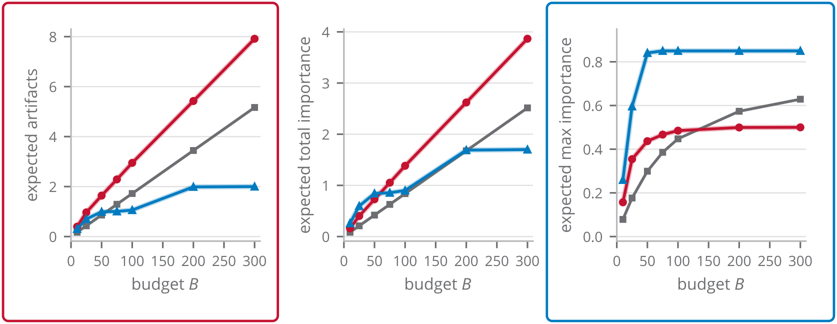
Comparing allocation strategies



Uniform random — $\hat{p}(c)$ — Top 10% by $i(c)$, then $\hat{p}(c)$

- Ranking by $\hat{p}(c)$ finds more publishable results than uniform random.

Comparing allocation strategies



Uniform random $\hat{p}(c)$ Top 10% by $i(c)$, then $\hat{p}(c)$

- The best strategy depends on the objective.

Summary

Find → Attempt → Recommend

Summary

Find → Attempt → Recommend

- **Input:** description of a direction

Summary

Find → Attempt → Recommend

- **Input:** description of a direction
- **Output:** conjecture-resolution pairs for human review

Summary

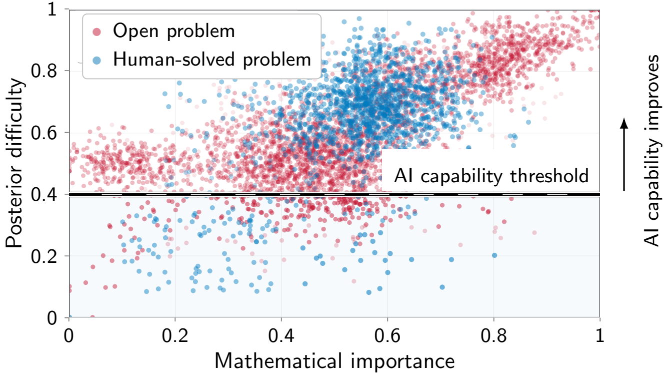
Find → Attempt → Recommend

- **Input:** description of a direction
- **Output:** conjecture-resolution pairs for human review
- To use it for your area, you need a budget of model attempts and human expertise for studying/reviewing the recommendations

Summary

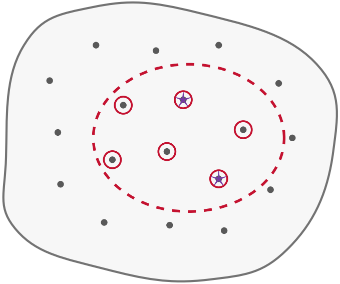
- **Input:** description of a direction
- **Output:** conjecture-resolution pairs for human review
- To use it for your area, you need a budget of model attempts and human expertise for studying/reviewing the recommendations
- Implemented as a cascade of agents; each stage can be improved

Moving forward: frontier of resolvable problems



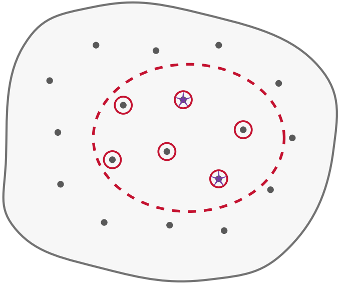
- As models improve, the frontier of resolvable problems will grow

Moving forward: which problems should AI recommend?



- Ensure that "importance" is aligned with the user's notion of importance

Moving forward: which problems should AI recommend?



- Ensure that "importance" is aligned with the user's notion of importance
- Should AI make the problems too?

ProbXiv

ProbXiv Search problems search sign in

An open workspace for mathematical discovery. Check proofs, make comments, form collaborations.

Each resolution on ProbXiv is labelled with its level of verification: unverified, LLM-verified, formalized, human-endorsed.

open questions

Chen-Gendron Spin-Parity Identity for k-Differentials

posed by Dawei Chen & Quentin Gendron

"For odd k with $\gcd(n,k) = \gcd(n+1,k) = 1$, is $N\_k(n) \equiv \lfloor (k+1)/4 \rfloor \pmod 2$, where $N\_k(n)$ counts pairs $1 \le b\_i \le (k-1)/2$ with $b\_1 + b\_2 \ge (k+1)/2$ and $b\_2 \equiv n\, b\_1 \pmod k$? Conjectured by Chen and Gendron; its proof removes a…"

lean: partially-checked Lean
no ProbXiv account is credited

does the formalisation match? →

How Many Stemmata with Root Degree k?

posed by Armin Hoenen, Steffen Eger, Ralf Gehrke

"While we are not able to prove it, we think it is a very safe conjecture that $R\_2(m)$ converges to below 0.607 , as m tends toward infinity."

model says: correct
GPT-5.5 xhigh (SMD judge 1)
no ProbXiv account is credited

say whether it holds →

RECURSIONS ASSOCIATED TO TRAPEZOID, SYMMETRIC AND ROTATION SYMMETRIC FUNCTIONS OVER GALOIS FIELDS

posed by Francis N. Castro, Robin Chapman, Luis A. Medina, L. Brehsner Sepúlveda

"Let $t\_{k,q}(n)$ be defined by (5.1) $t\_{k,q}(j) = q^j$, for $0 \le j \le k-1$ $t\_{k,q}(n) = q \sum\_{l=0}^{k-2} (q-1)^l\, t\_{k,q}(n-(l+2))$, for $n \ge k$. Then, $S\_{F\_q}(T\_{2,3,\dots,k}(n)) = t\_{k,q}(n)$ for all values of $n \ge k$."

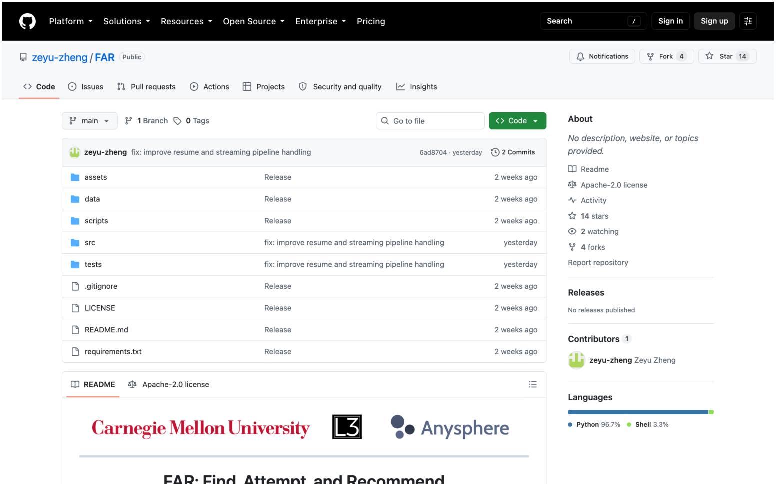
model says: correct
GPT-5.5 xhigh (SMD judge 1)
no ProbXiv account is credited

say whether it holds →

problems

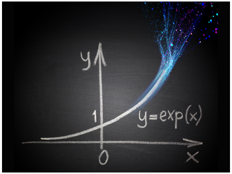
https://probxiv.com

GitHub



https://github.com/zeyu-zheng/FAR

Thank you for your support



expMath

Compute/API support from
Google, OpenAI, Anysphere

Thank you!

THE PROBLEM IS THE PROBLEM: TOWARDS SCALABLE MATHEMATICAL DISCOVERY

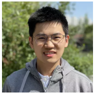
Zeyu Zheng[1]*, Shengtong Zhang[2]*, Jeremy Avigad[1], Prasad Tetali[1], Sean Welleck[1]
[1]Carnegie Mellon University [2]Anysphere Co.

`https://arxiv.org/abs/2608.16977`

`https://probxiv.com`

`https://github.com/zeyu-zheng/FAR`

Zeyu Zheng



Shengtong Zhang

Jeremy Avigad

Prasad Tetali

Sean Welleck

wellecks@cmu.edu `https://wellecks.com`

L3 Lab: `https://cmu-l3.github.io`