

# Synthesizing Open-ended Problems at Scale

---

On behalf of FrontierSmith Team

**Qiuyang Mang\***

with Runyuan He\*, Shang Zhou, Kaiyuan Liu, Hanchen Li, Huanzhi Mao, Qizheng Zhang, Zerui Li, Bo Peng, Lufeng Cheng, Tianfu Fu, Yichuan Wang, Wenhao Chai, Jingbo Shang, Alex Dimakis, Joseph E. Gonzalez, and Alvin Cheung

SWE-bench Verified  $\approx$  **80%**

AIME  $\approx$  **100%**

Humanity Last Exam  $\approx$  60%

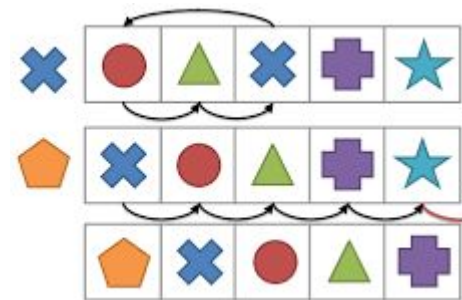
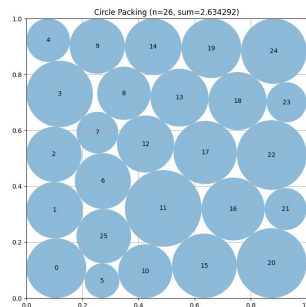
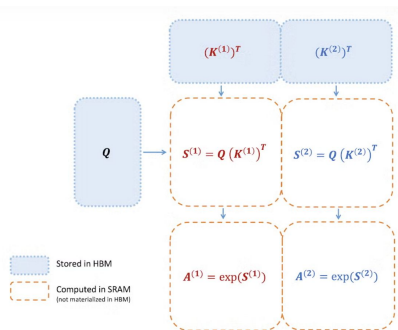


LLMs now **saturate** real exams and exam-style benchmarks.

What's next?



# Beyond Passing Exams: Open-ended Problems



LLM as a performance optimizer

*How much speedup can we achieve with a GPU kernel?*

LLM as an algorithm designer

*How many circles can we pack into a fixed region?*

LLM as a system researcher

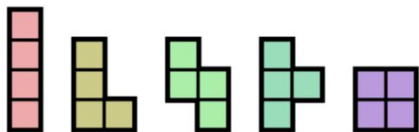
*How much cache miss cost can we reduce with a replacement policy?*

**No optimal solutions, only better frontiers.**

## FrontierCS

[ICML 26]

open-ended  
verifiable

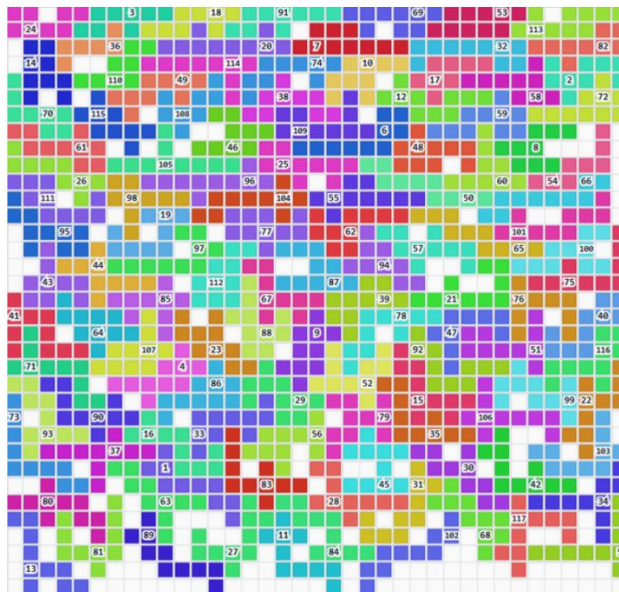


### Example: Polyomino Packing

Pack all polyominoes as tightly as possible into the grid.

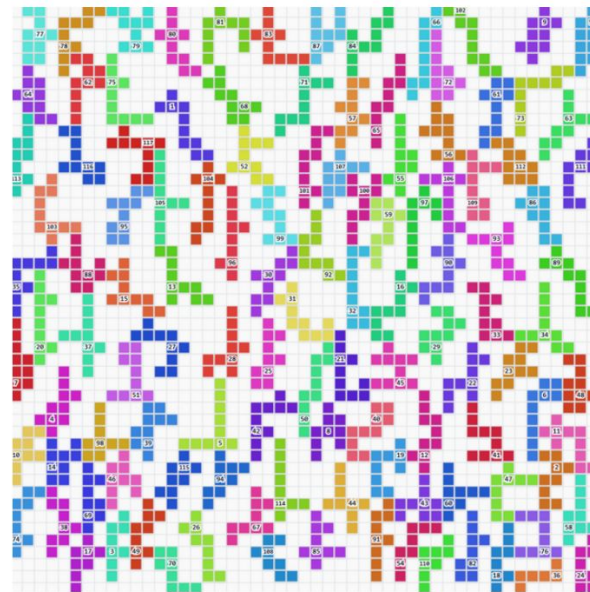
Human Expert

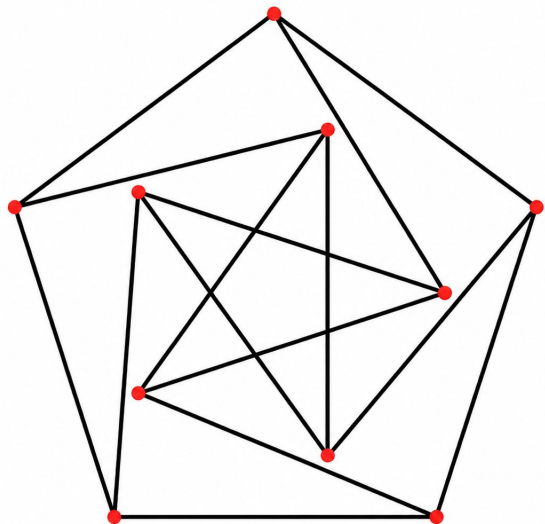
87%



GPT-5 Thinking

47%





**Erdős Planuar Unit Distance**

May 20, 2026 Research Milestone

An OpenAI model has disproved a central conjecture in discrete geometry

[Read the proof ↗](#)

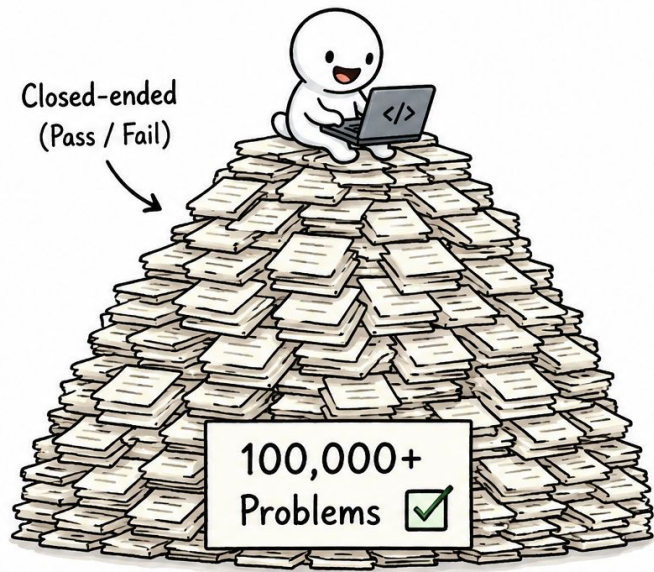
[Read the companion remarks ↗](#)

Given  $n$  points in the plane, what is the maximum number of point pairs that are exactly one unit apart?

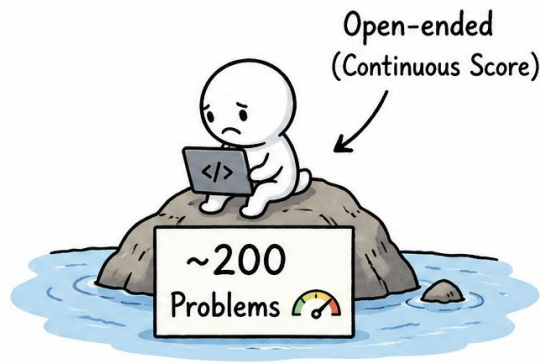
# Generating Open-ended Tasks from Closed-ended Ones

We Have Tons of Closed-Ended Code Tasks...

...But Very Few Open-Ended Ones



Thanks, Codeforces, LeetCode, etc.! 🙌



So rare... so valuable. 💎

Can LLMs generate the data needed to train themselves?

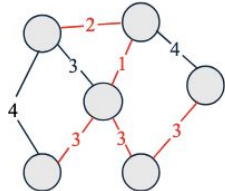
💡 Opportunity: Synthesize more high-quality open-ended code tasks!

# FrontierSmith: Synthesizing Open-ended Tasks



## Closed-ended Problem

Minimum Spanning Tree:  
Connect all nodes with  
minimum total edge  
weight.



Pass if  $Tree\ Weight = 12$

**Pass** or **Fail**

## Candidate A

Minimize total weight  
while also minimizing  
number of edges.

## Candidate B

Minimize total weight  
while satisfying the max-  
degree constraint.

## Candidate C

Minimize total weight  
while encouraging  
balanced subtree sizes.

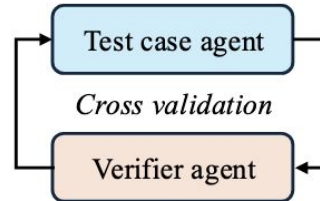
...

## Discard low- divergence candidates



Pairwise: same strategy?

Idea divergence:  $\hat{d}(c) = 0.67$   
Keep top- $N$  by divergence

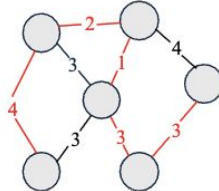


Run solutions on tests  
Score with the verifier

Keep top- $N_{final}$  by  
execution divergence

## Open-ended Problem

Connect all nodes, min  
total weight,  $s.t.$ , max  
degree  $\leq D$ .  
 $(D = 2)$



Score =  $1 - \frac{Tree\ Weight}{Total\ Weight}$

**0.67** **0.42** **0.33** **0.58**

# How to Formulate the Mutation

Task Formulation =  $(O, CI, CO)$

$O$ : Objective |  $CI$ : Input Constraints |  $CO$ : Output Constraints

## Changing Goals

$(O \rightarrow O')$

Replace exact/binary goals with optimization.

Shifts tasks from simple decision to graded performance.

### Example:

2-SAT (Decision)  $\rightarrow$  Min-True  
2-SAT (Optimization)

## Restricting Outputs

$(CO \rightarrow CO')$

Add output constraints.

Makes exact solutions infeasible at scale, favoring approximations.

### Example:

MST  $\rightarrow$  Degree-constrained  
Spanning Tree (NP-Hard)

## Generalizing Inputs

$(CI \rightarrow CI')$

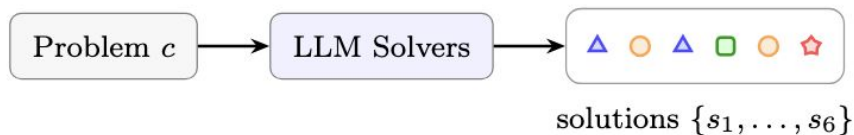
Relax constraints on input.

Transforms polynomial problems into NP-complete ones.

### Example:

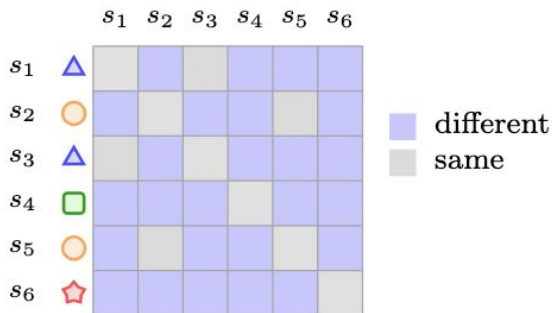
Max Independent Set: Bipartite  
(P)  $\rightarrow$  Arbitrary Graphs (NP-C)

# How Can We Ensure Mutation Quality



- △ greedy      ○ dynamic programming
- local search ☆ gradient descent

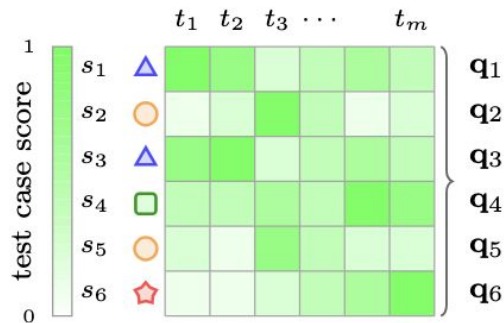
## LLM-as-a-judge-based estimation



$$\hat{d}(c) = \text{avg LLM-as-a-Judge}(s_i, s_j)$$

**LLM-based estimation.** An LLM judge labels each solution pair as same- or different-strategy.  $\hat{d}(c)$  is the fraction of pairs judged different. Blue cells: different-strategy; gray cells: same-strategy.

## Execution-based estimation



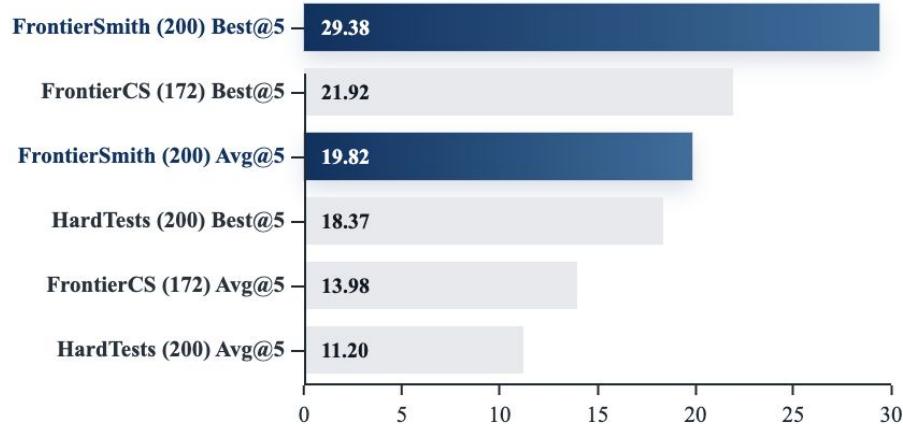
$$\hat{d}(c) = \text{avg} \frac{1}{\sqrt{m}} \|\mathbf{q}_i - \mathbf{q}_j\|_2$$

**Execution-grounded estimation.** Each solution is run on the test cases  $t_1, \dots, t_m$  and scored by the verifier, yielding a score vector  $\mathbf{q}_i$ .  $\hat{d}(c)$  is the average pairwise distance between score vectors.

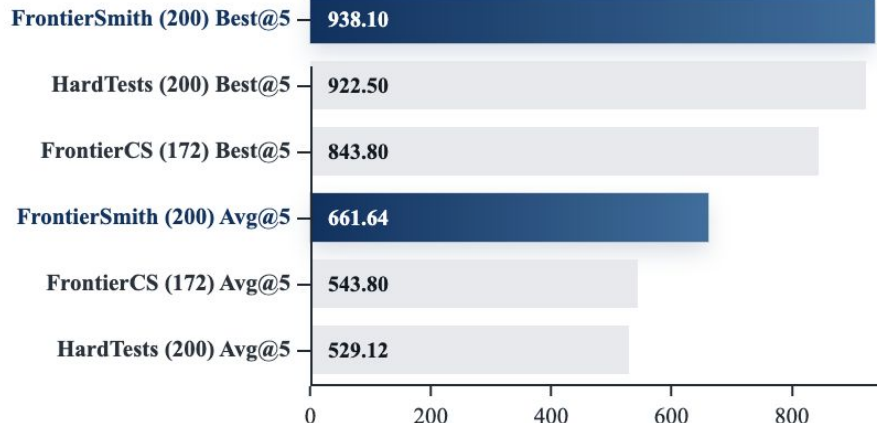
# Synthetic Data Leads Qwen-3.5-27B in Open-Ended Coding



### FrontierCS



### ALE-bench



**+12.12**

FrontierCS Avg@5 over Base

**+309.12**

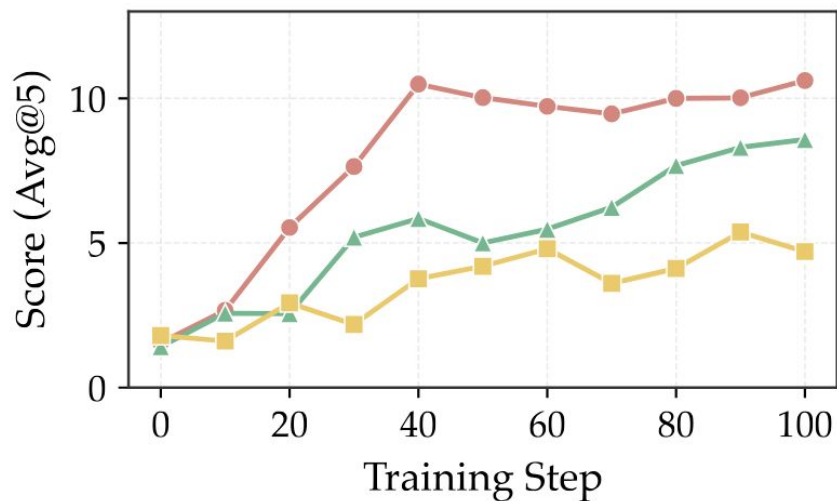
ALE-bench Avg@5 over Base

# Filtering Improves Open-Ended Training Data Quality

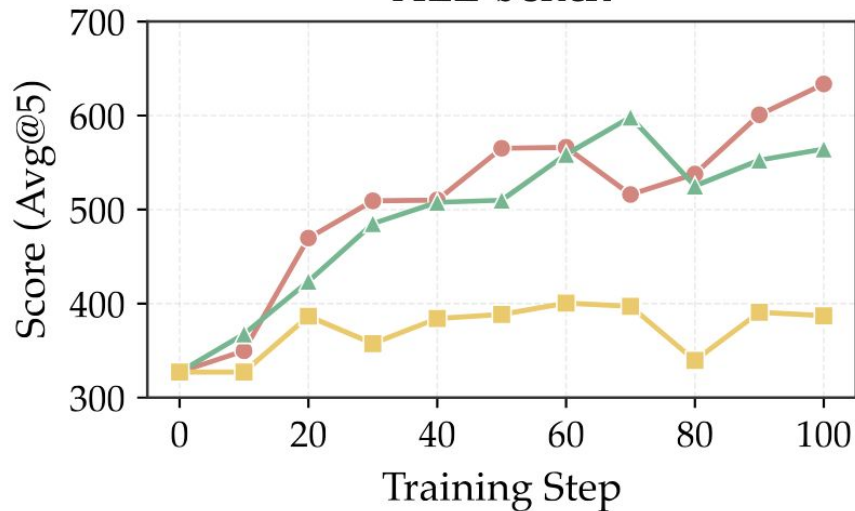


● FrontierSmith (200)    ▲ No Filter (200)    ■ HardTests (200)

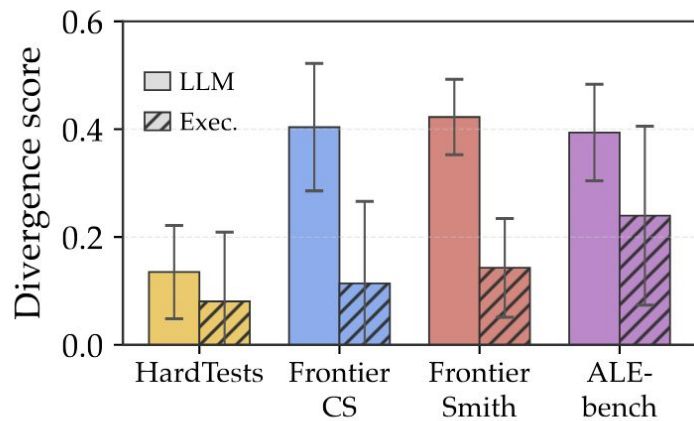
### FrontierCS



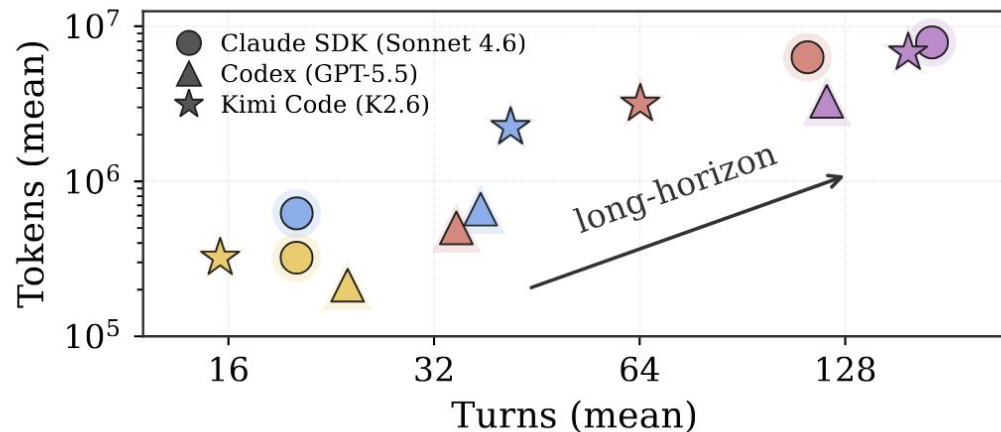
### ALE-bench



# Synthetic Tasks Match Human-curated Ones



Solution Space Diversity



Agentic Behavior

**LLMs can synthesize open-ended tasks at scale to improve their capabilities in a self-play manner**



More blog posts are now available on our website!