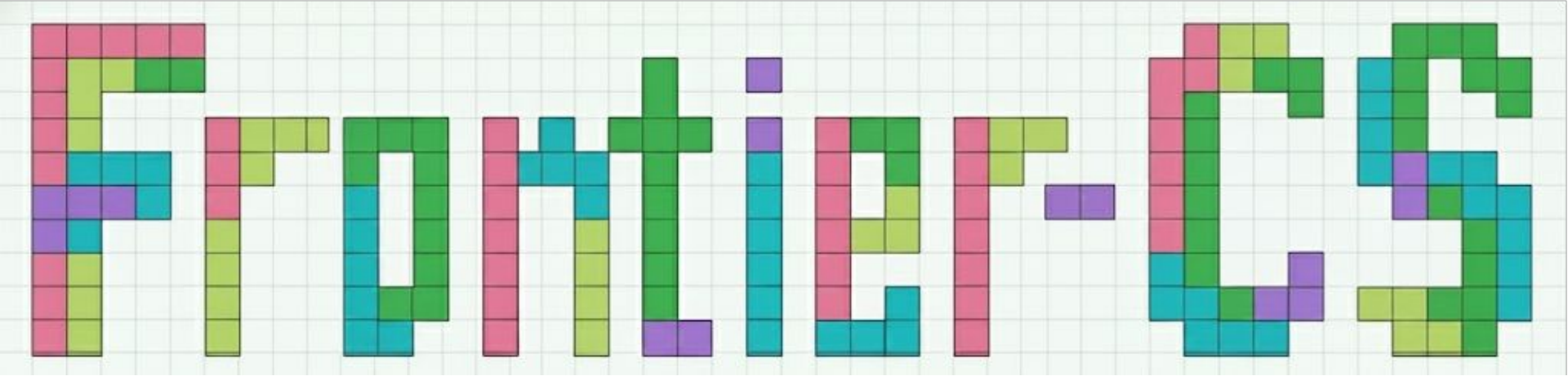


Open-ended Code Generation Challenges for Evolving Intelligence

Qiuyang Mang *on behalf of Frontier-CS Team*

{SLICE, SKY} @ UC Berkeley

SWE-bench Verified ≈ **80%**

AIME ≈ **100%**

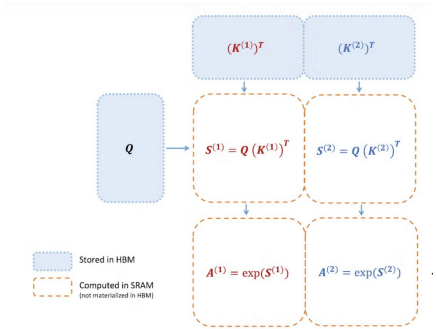
Humanity Last Exam ≈ **45%**

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

What's next?

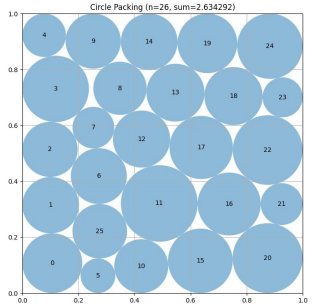
Open-ended Problems

Beyond Passing Exams: Exploring New Frontiers



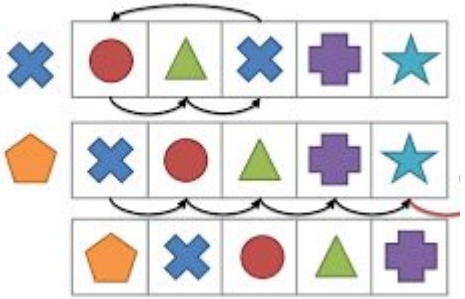
LLM as a performance optimizer

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?

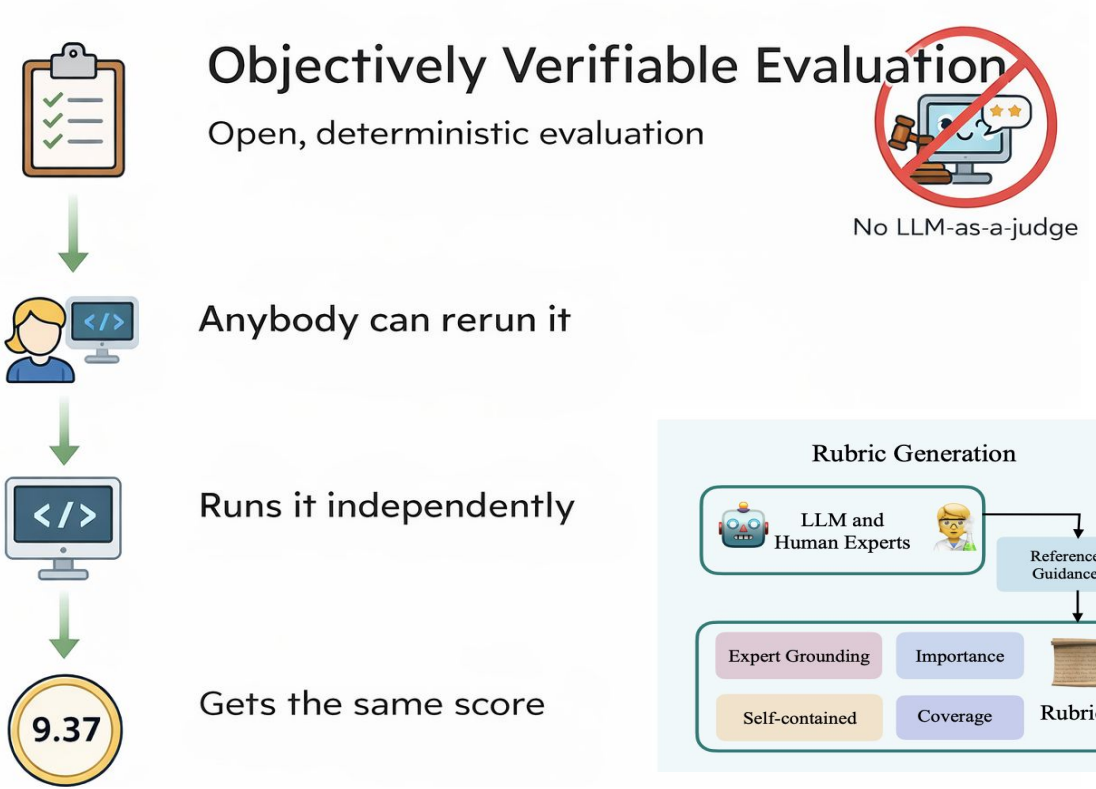


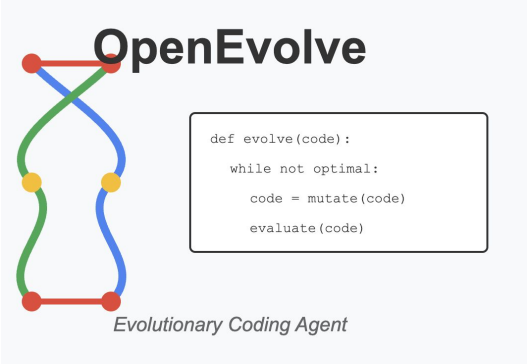
LLM as a system researcher

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

No optimal solutions, only better frontiers.

We Don’t Know the **Best**, but We Know How **Good**



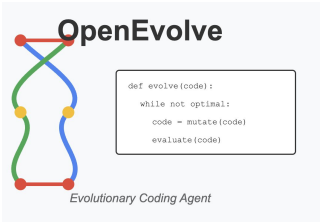


⇒ *Continuous scores guide agentic evolution and support post-training*

LLMs for Open-Ended Problems: Where We Are Limited

KernelBench: ~200 GPU programming problems (single-domain)

ADRS: ~10 system research problems

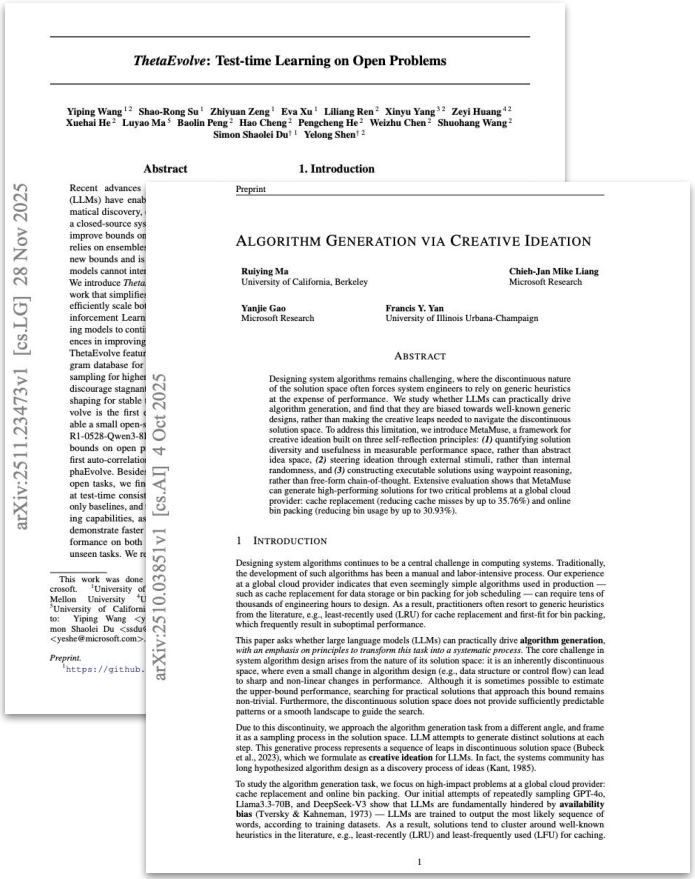


OpenEvolve: 14 example problems

ALE-bench: 40 Atcoder Heuristic Contest problems (single-source)

We need a **large-scale benchmark** for open-ended problems, spanning multiple domains, sources, and problem types.

LLMs for Open-Ended Problems: Where We Are Limited



ThetaEvolve: Test-time Learning on Open Problems

Yiping Wang [1 2] Shao-Rong Su [1] Zhiyuan Zeng [1] Eva Xu [1] Liliang Ren [2] Xinyu Yang [3 2] Zeyi Huang [4 2] Xuehai He [2] Luyao Ma [5] Baolin Peng [2] Hao Cheng [2] Pengcheng He [2] Weizhu Chen [2] Shuohang Wang [2] Simon Shaolei Du [† 1] Yelong Shen [† 2]

Abstract

1. Introduction

arXiv:2511.23473v1 [cs.LG] 28 Nov 2025

arXiv:2510.03851v1 [cs.AI] 4 Oct 2025

Preprint

ALGORITHM GENERATION VIA CREATIVE IDEATION

Ruiying Ma
University of California, Berkeley

Chieh-Jan Mike Liang
Microsoft Research

Yanjie Gao
Microsoft Research

Francis Y. Yan
University of Illinois Urbana-Champaign

ABSTRACT

Designing system algorithms remains challenging, where the discontinuous nature of the solution space often forces system engineers to rely on generic heuristics at the expense of performance. We study whether LLMs can practically drive algorithm generation, and find that they are biased towards well-known generic designs, rather than making the creative leaps needed to navigate the discontinuous solution space. To address this limitation, we introduce MetaMuse, a framework for creative ideation built on three self-reflection principles: *(1)* quantifying solution diversity and usefulness in measurable performance space, rather than abstract idea space, *(2)* steering ideation through external stimuli, rather than internal randomness, and *(3)* constructing executable solutions using waypoint reasoning, rather than free-form chain-of-thought. Extensive evaluation shows that MetaMuse can generate high-performing solutions for two critical problems at a global cloud provider: cache replacement (reducing cache misses by up to 35.76%) and online bin packing (reducing bin usage by up to 30.93%).

1 INTRODUCTION

Designing system algorithms continues to be a central challenge in computing systems. Traditionally, the development of such algorithms has been a manual and labor-intensive process. Our experience at a global cloud provider indicates that even seemingly simple algorithms used in production — such as cache replacement for data storage or bin packing for job scheduling — can require tens of thousands of engineering hours to design. As a result, practitioners often resort to generic heuristics from the literature, e.g., least-recently used (LRU) for cache replacement and first-fit for bin packing, which frequently result in suboptimal performance.

This paper asks whether large language models (LLMs) can practically drive **algorithm generation**, *with an emphasis on principles to transform this task into a systematic process*. The core challenge in system algorithm design arises from the nature of its solution space: it is an inherently discontinuous space, where even a small change in algorithm design (e.g., data structure or control flow) can lead to sharp and non-linear changes in performance. Although it is sometimes possible to estimate the upper-bound performance, searching for practical solutions that approach this bound remains non-trivial. Furthermore, the discontinuous solution space does not provide sufficiently predictable patterns or a smooth landscape to guide the search.

Due to this discontinuity, we approach the algorithm generation task from a different angle, and frame it as a sampling process in the solution space. LLM attempts to generate distinct solutions at each step. This generative process represents a sequence of leaps in discontinuous solution space (Bubeck et al., 2023), which we formulate as **creative ideation** for LLMs. In fact, the systems community has long hypothesized algorithm design as a discovery process of ideas (Kant, 1985).

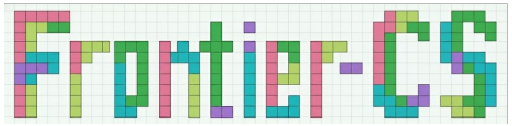
To study the algorithm generation task, we focus on high-impact problems at a global cloud provider: cache replacement and online bin packing. Our initial attempts of repeatedly sampling GPT-4o, Llama3.3-70B, and DeepSeek-V3 show that LLMs are fundamentally hindered by **availability bias** (Tversky & Kahneman, 1973) — LLMs are trained to output the most likely sequence of words, according to training datasets. As a result, solutions tend to cluster around well-known heuristics in the literature, e.g., least-recently (LRU) and least-frequently used (LFU) for caching.

1

ThetaEvolve: Evaluated on only **5** tasks from OpenEvolve Examples: CirclePacking-T / FirstAutoCorrIneq /SecondAutoCorrIneq / ThirdAutoCorrIneq /HadamardMatrix

MetaMuse: Evaluated on only **2** tasks by their authors: Bin Packing / Cache Replacement

Introducing Frontier-CS



= **200+** open-ended problems adapted from competitive programming and real CS research

- **Algorithmic Track**: 10+ experts converse exam-style problems from multiple sources into an open-ended style by *changing objectives*, *adding constraints* . . .

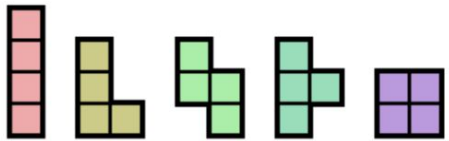
- **Research Track**: CS PhDs extract the core algorithmic problems from their research interests, including OS, HPC, AI, DB, PL, . . .

An **Open-Ended**, **Verifiable** Example with **Continuous** Score

FrontierCS

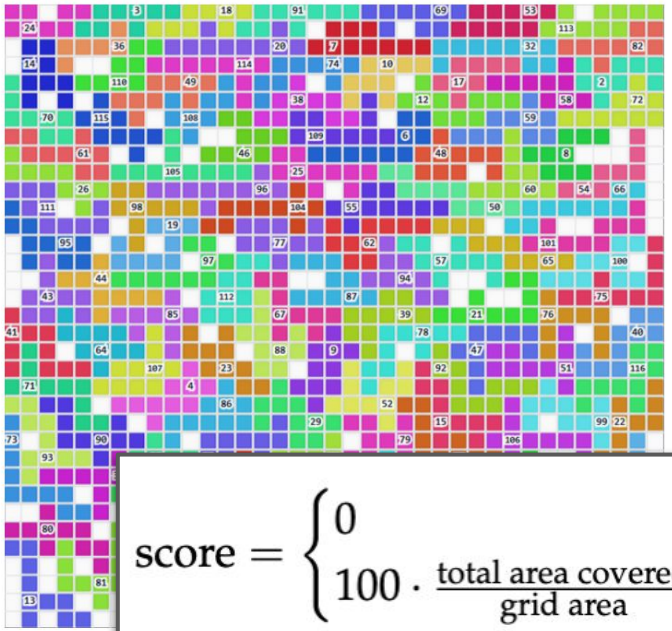
open-ended

verifiable

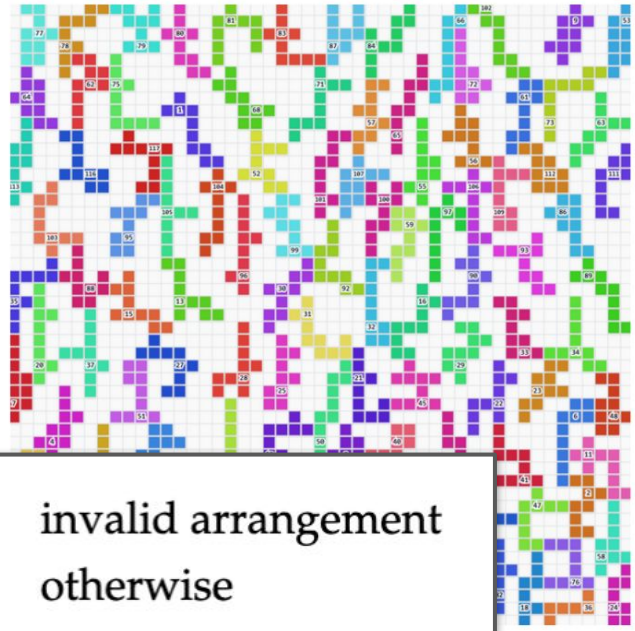


Example: Polyomino Packing
Pack all polyominoes as tightly as possible into the grid.

Human Expert 87%



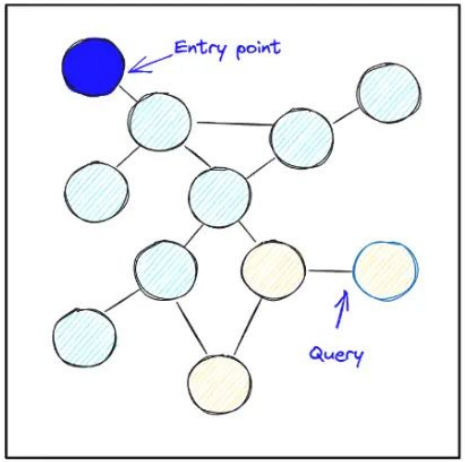
GPT-5 Thinking 47%



$$\text{score} = \begin{cases} 0 & \text{invalid arrangement} \\ 100 \cdot \dfrac{\text{total area covered}}{\text{grid area}} & \text{otherwise} \end{cases}$$

Research Problem Example: VectorDB Design

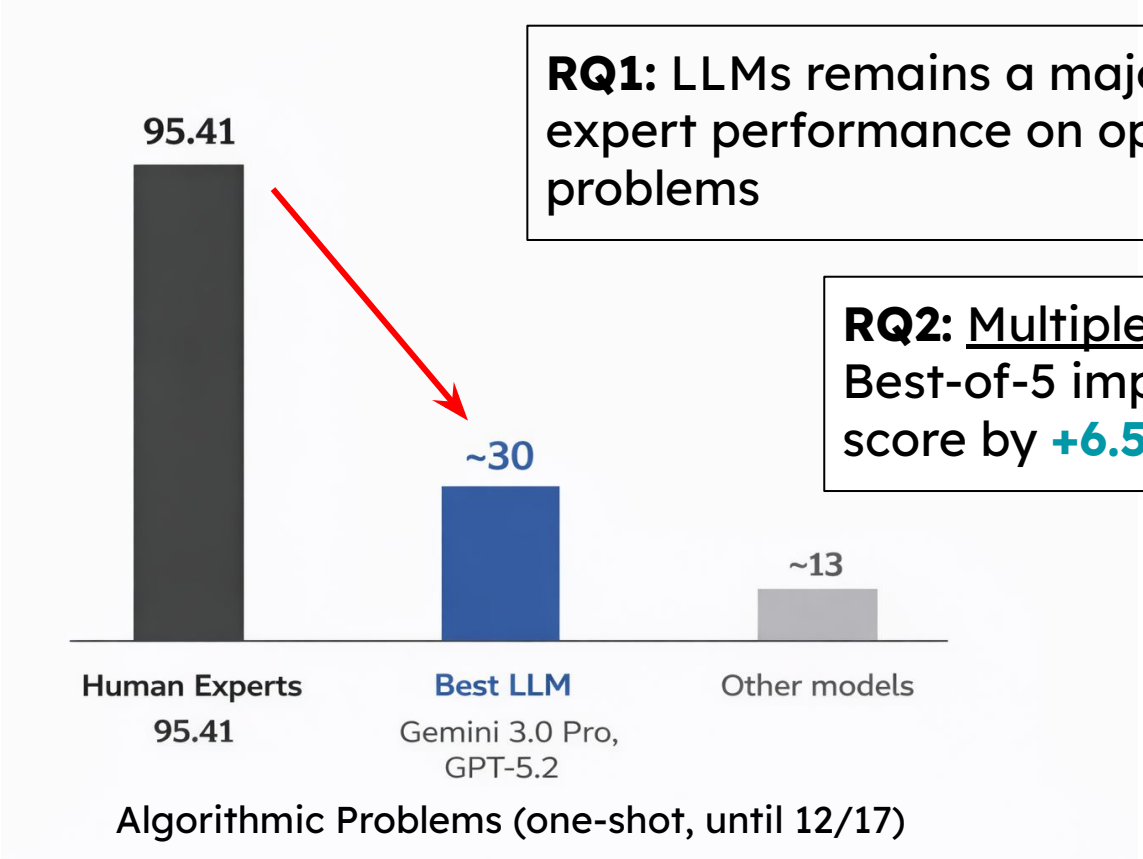
- Given: Millions of vectors and a set of query vectors.
- Task: For each query, return the nearest stored vector as fast as possible, while being correct at least 80% of the time.
- Scoring: Recall@1 ≥ 80% is required; passing solutions receive a 0–100 score normalized by average query latency relative to a human SOTA.



Research Questions on Frontier-CS

- RQ1: As LLMs saturate code exams, how do they perform on open-ended CS problems?
- RQ2: Does test-time scaling via multiple attempts still hold for open-ended CS problems?
- RQ3: Does longer chain-of-thought test-time scaling still work?
- RQ4: What failure modes do LLMs exhibit on open-ended CS problems?

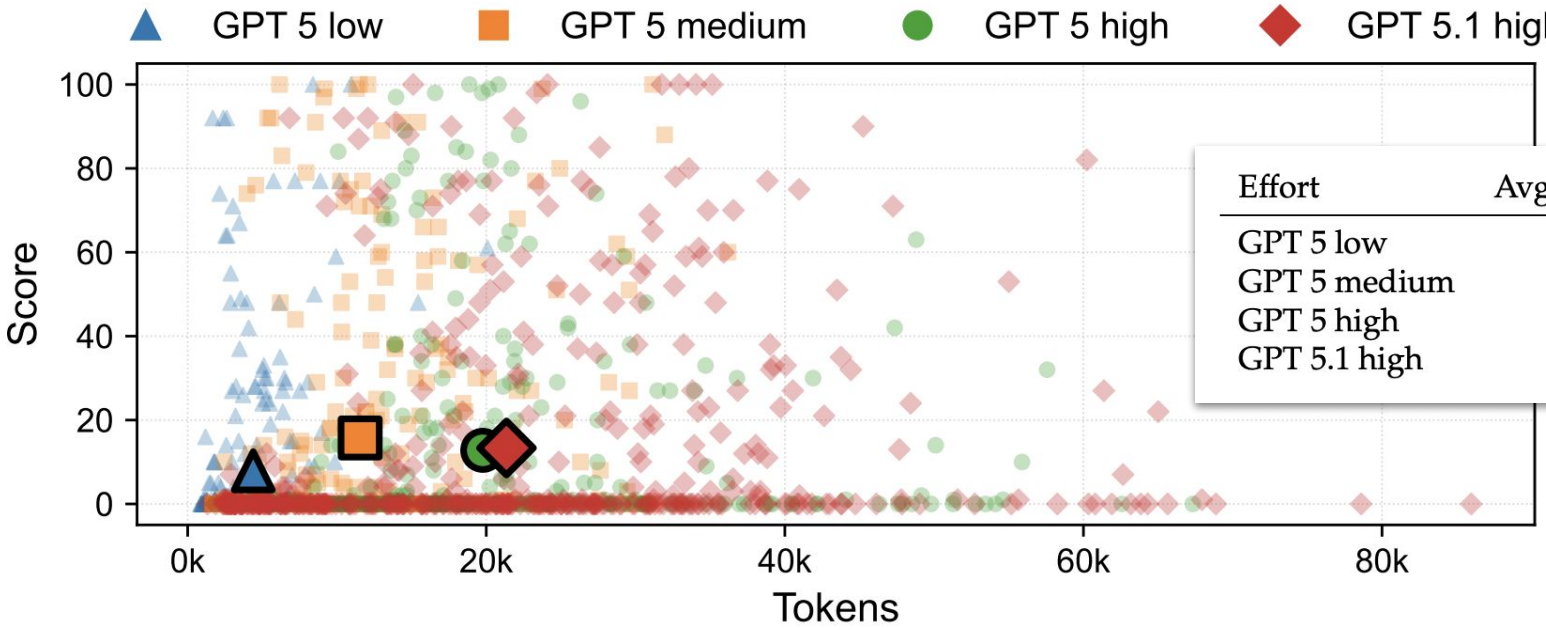
Open-ended Problems Remain Challenging for Frontier LLMs



RQ1: LLMs remains a major gap to human expert performance on open-ended problems

RQ2: Multiple attempts matter
Best-of-5 improves average score by **+6.5 – +22.7**

Improving Reasoning Effort Does Not Yield Further Gains



Effort	Avg tokens	Avg score
GPT 5 low	4,389	7.903
GPT 5 medium	11,554	**15.336**
GPT 5 high	19,763	12.626
GPT 5.1 high	**20,402**	12.508

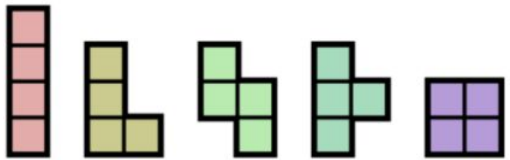
RQ3: Longer chain-of-thought not always help open-ended problem solving.

Misleading Micro-Optimization Trap

- LLMs often fail to identify which choices are algorithmically meaningful, becoming trapped in micro optimizations.
- **GPT-5 solution:** starts with a suboptimal core idea, augmented with several low-level memory and efficiency optimizations ⇒ ~40 points

Human solution & prompt fix:

Please use a 2D array to maintain the rectangle state, and convert to the required format only at the end ⇒ ~80 points



Example: Polyomino Packing
Pack all polyominoes as tightly as possible into the grid.

Harbor Integration

Agentic Benchmarks

Impact and Future Directions of Frontier-CS

Impact:

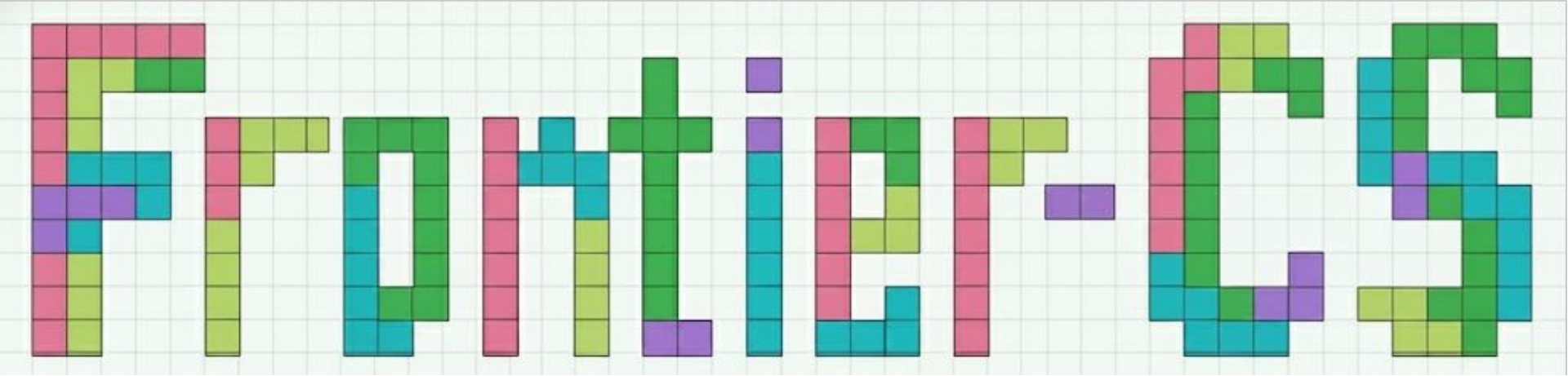
- Provides a large, diverse collection of open-ended problems
- Enables reproducible, fine-grained evaluation via deterministic, verifiable and continuous scoring
- The project has attracted early industry interest (e.g., xAI, ByteDance), and welcomes contributions of problems or compute resources.

What's next?

- **Evolving:** From domain-specific tricks to regularized, general evolving frameworks.
- **Synthesis:** Can LLMs synthesize new open-ended problems?
- **Post-training:** Can models learn new reasoning abilities from open-ended problems with continuous reward signals?

GitHub

Twitter/X



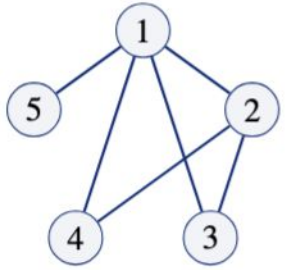
Thanks!

Website: frontier-cs.org

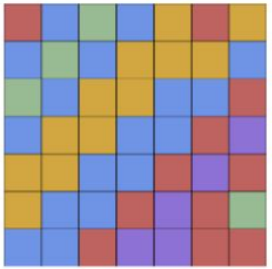
GitHub: https://github.com/FrontierCS/Frontier-CS

World Map (IOI 2025 ⇒ Frontier-CS)

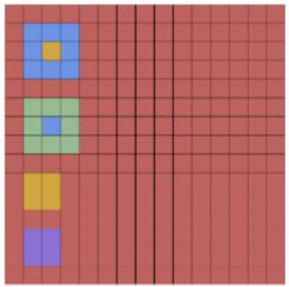
- Given several countries and which pairs must be adjacent.
- Place country labels on a square grid.
- Adjacent cells imply country adjacency.
- All required adjacencies must appear.
- Goal: use the **smallest** possible grid.



(a) Adjacency graph E.

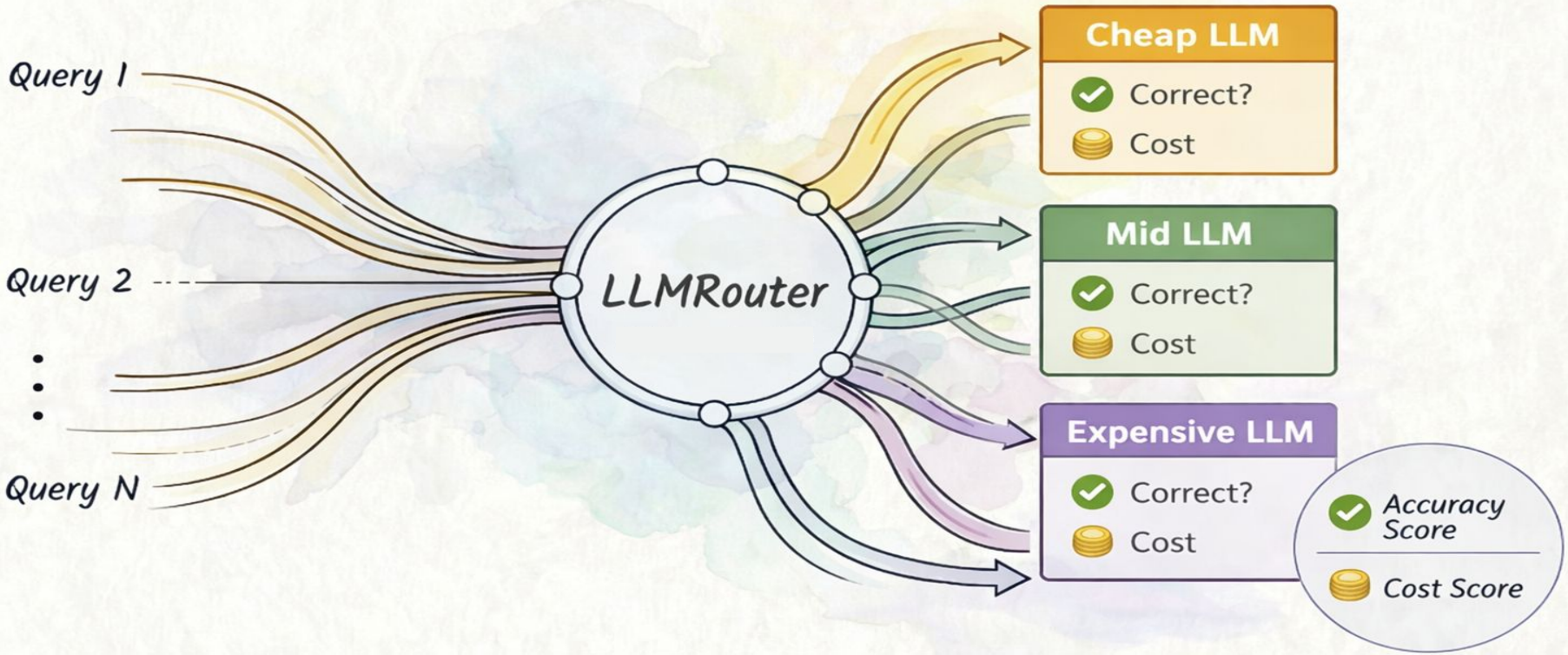


(b) Human expert ($K = 7$).



(c) GPT 5 ($K = 15$).

Research Problem: LLM Router

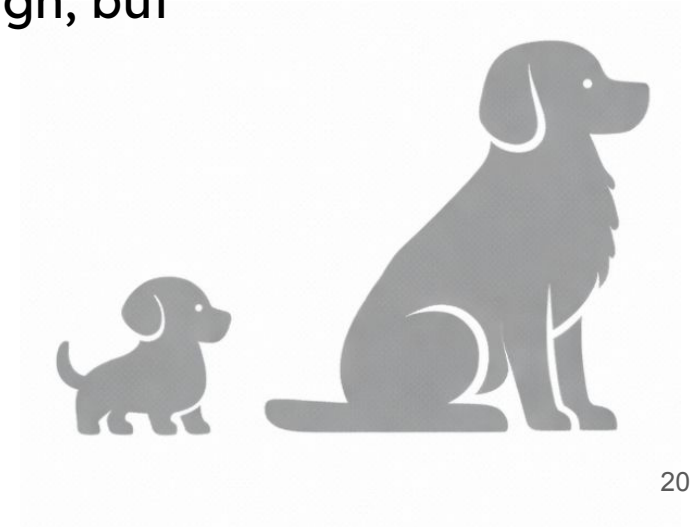


Learn routing policy from a reference dataset

LLMs for Open-Ended Problems: Where We Are Limited

- Toy-scale benchmarks cannot distinguish general algorithmic discovery ability from task overfitting.

- Toy-scale data can support framework design, but cannot train new reasoning abilities.



FrontierCS: Evolving Challenges for Evolving Intelligence

Contributors (*equal contribution)

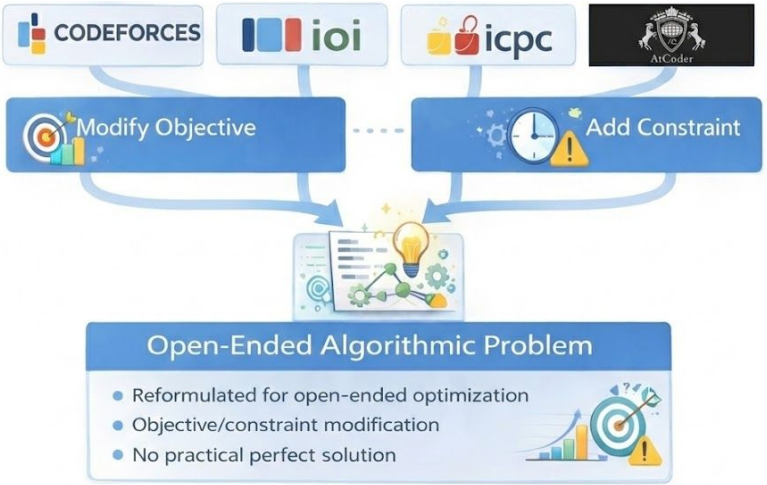
Qiuyang Mang[1,*], Wenhao Chai[2,*], Zhifei Li[1,*], Huanzhi Mao[1,*], Shang Zhou[3,*], Alexander Du[1,4,*], Hanchen Li[1,*], Shu Liu[1,*], Edwin Chen[5], Yichuan Wang[1], Xieting Chu[6], Zerui Cheng[2], Yuan Xu[4], Tian Xia[1], Zirui Wang[1], Tianneng Shi[1], Jianzhu Yao[2], Yilong Zhao[1], Qizheng Zhang[7], Charlie Ruan[1], Zeyu Shen[2], Kaiyuan Liu[8], Runyuan He[1], Dong Xing[4], Zerui Li[4], Zirong Zeng[1], Yige Jiang[9], Lufeng Cheng[10], Ziyi Zhao[11], Youran Sun[1], Wesley Zheng[1], Meiyuwang Zhang[5], Ruyi Ji[12], Xuechang Tu[6], Zihan Zheng[13], Zexing Chen[3], Kangyang Zhou[14], Zhaozi Wang[13], Jingbang Chen[5]

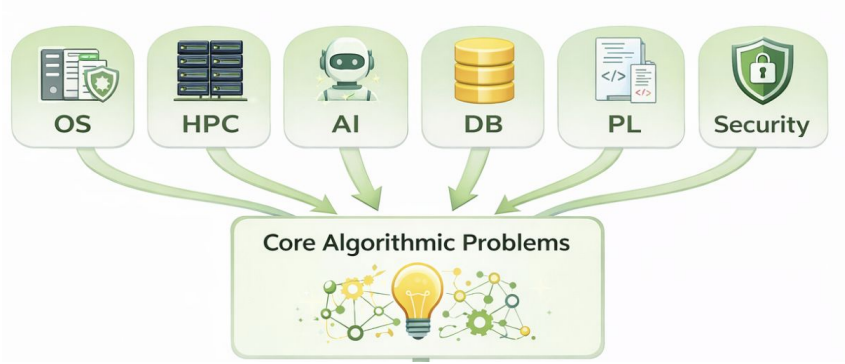
Advisors (*equal advising)

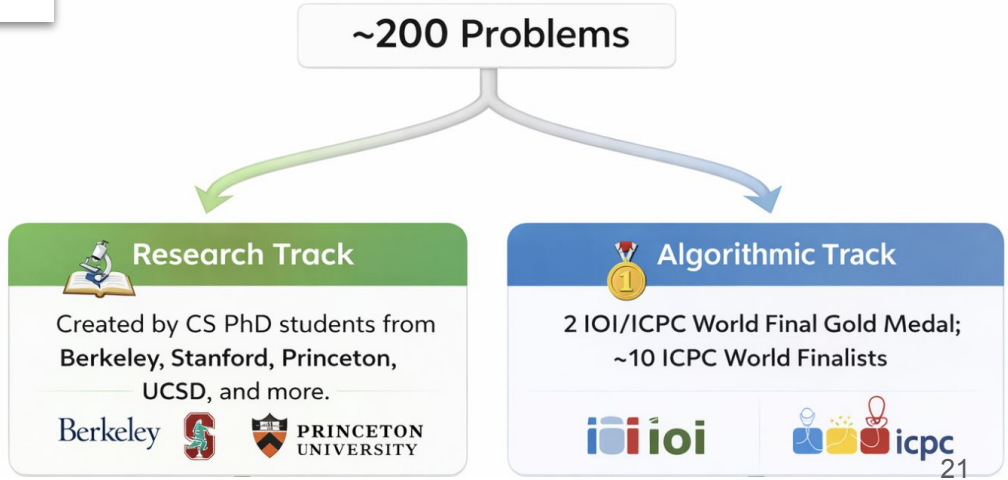
Aleksandra Korolova[2], Peter Henderson[2], Pramod Viswanath[2], Vijay Ganesh[6], Saining Xie[13], Zhuang Liu[2], Dawn Song[1], Sewon Min[1], Ion Stoica[1], Joseph E. Gonzalez[1,*], Jingbo Shang[3,*], Alvin Cheung[1,*]

Affiliations

[1]UC Berkeley [2]Princeton University [3]UCSD [4]X-camp Academy [5]Independent [6]Georgia Tech [7]Stanford University [8]University of Washington [9]Nanyang Technological University [10]University of Toronto [11]UIUC [12]University of Michigan [13]New York University [14]MIT







Frontier-CS

Evolving Challenges for Evolving Intelligence

Qiuyang Mang
Advisor: Alvin Cheung
on behalf of Frontier-CS Team

Website GitHub

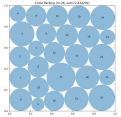
Background: LLMs Near Ceiling on exams

SWE-bench Verified ≈ **80%**
AIME ≈ **100%**
Humanity Last Exam ≈ **45%**

What's Next for Computer Science Tasks?

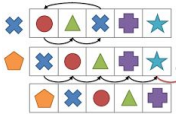
LLM as a performance optimizer

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 best solutions, only better frontiers.

Objectively Verifiable Evaluation
Open, deterministic evaluation

Anybody can rerun it

Runs it independently

9.37 Gets the same score

Unsolved
Open-ended
Verifiable
Diverse
Continuous Score

⇒ *Continuous scores guide agentic evolution and support post-training*

Where We Are Limited

Existing research on open-ended CS problems often evaluates LLMs on a few **toy tasks**.

KernelBench: ~200 GPU programming problems (single-domain)

ADRS: ~10 system research problems

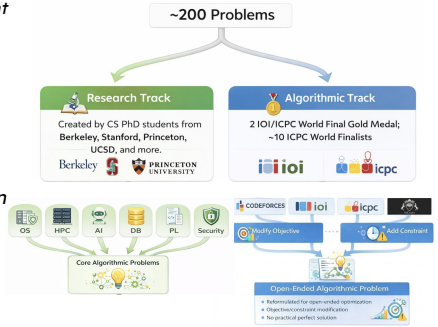
Existing open-ended CS benchmarks are often **domain-specific** or **small-scale**.

OpenEvolve: 14 example problems

ALE-bench: 40 Atcoder Heuristic Contest problems (single-source)

We need a **large-scale benchmark** for open-ended problems, spanning multiple domains, sources, and problem types.

Frontier-CS: a large, diverse datasets for open-ended CS problems



Research Track problems are constructed by distilling core algorithmic challenges from real CS research problems and asking LLMs to implement practical solutions.

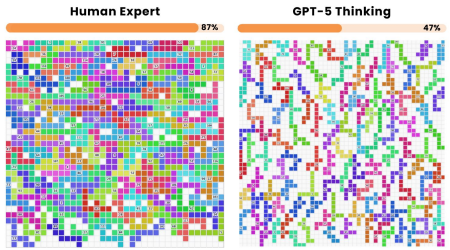
Algorithmic Track problems are created by transforming existing exam-style tasks into open-ended settings through modified objectives and added constraints, or by collecting classical NP problems and conjectures.

Frontier-CS Example Problems

FrontierCS

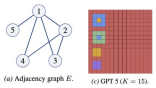
unsolved
open-ended
verifiable
diverse

Example: Polyomino Packing
Pack all polyominoes as tightly as possible into the grid.



World Map (IOI 2025 ⇒ Frontier-CS)

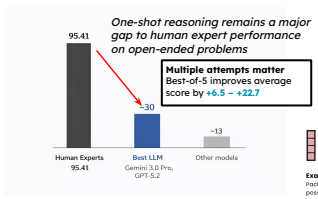
Given several countries and which pairs must be adjacent.
Place country labels on a square grid.
Adjacent cells imply country adjacency.
All required adjacencies must appear.
Goal: use the **smallest** possible grid.



Research Problems:
Cloud Scheduling,
VectorDB Design,
POC generation,
Grammar Fuzzing,
RL Algorithms,
GPU Kernel Design,
and more

Key Findings

Open-ended Problems Remain Challenging for LLMs



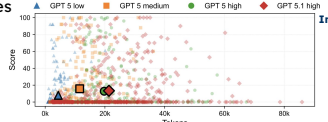
Misleading Micro-Optimization Trap

LLMs often fail to identify which choices are algorithmically meaningful, becoming trapped in micro optimizations.

GPT-5 solution: Maintains the search state as a list of polyomino placements, augmented with several low-level memory and efficiency optimizations ⇒ -40 score

Human solution & prompt fix:
Please use a 2D array to maintain the rectangle state, and convert to the required format only at the end
⇒ -80 score

Improving Reasoning Effort Does Not Yield Further Gains



Effort	Avg tokens	Avg score
GPT 5 low	4,389	7.9
GPT 5 medium	11,554	15.336
GPT 5 high	19,763	12.626
GPT 5.1 high	**20,402**	12.508