LiveCodeBench Pro: How Do Olympiad Medalists Judge LLMs in Competitive Programming?

Zihan Zheng [1,*,§], Zerui Cheng [2,*], Zeyu Shen [2,*], Shang Zhou [3,*], Kaiyuan Liu [4,*], Hansen He [5,*], Dongruixuan Li [6], Stanley Wei [2], Hangyi Hao [7], Jianzhu Yao [2], Peiyao Sheng [8], Zixuan Wang [2], Wenhao Chai [2,†,§], Aleksandra Korolova [2,†], Peter Henderson [2,†], Sanjeev Arora [2,†], Pramod Viswanath [2,8,†], Jingbo Shang [3,†,‡], Saining Xie [1,†,‡]

[1] New York University [2] Princeton University [3] University of California San Diego [4] University of Washington [5] Canyon Crest Academy [6] University of Waterloo [7] McGill University [8] Sentient Foundation

What Coding Tasks Can Still Challenge LLMs?

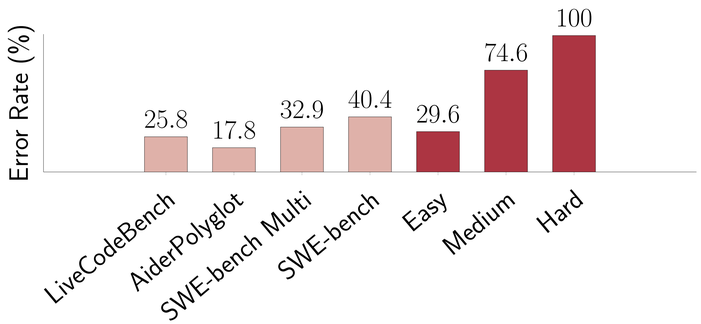

Figure 1. Gemini 2.5 Pro pass@1 error rate across coding benchmarks.

Key Findings

Our evaluation reveals significant limitations in current frontier models' competitive programming capabilities.

- LLMs perform better on **knowledge** and **logic**-heavy problems, and worse on **observation**-heavy problems or case work.
- Some LLMs make significantly more algorithm logic errors and wrong observations, and much fewer **implementation** logic errors than humans.
- Increasing the number of attempts (pass@k) significantly improves the performance of the models while **still failing** in the hard tier.
- Reasoning brings about the largest improvement in combinatorics, a large improvement in knowledge-heavy categories, and relatively low improvement in **observation**-heavy ones.

Model Performance Results

Table 1. **Pass@1 and Elo rating performance on LiveCodeBench Pro 2024Q4 split.** Each model's Elo-based *Rating* is computed from head-to-head comparisons with human participants, while the *Pct.%* column shows the model's percentile among all human contestants. *AvgTok* is the average number of tokens generated per problem and *AvgCost* is the approximate $-cost per problem. We also test `o4-mini`, although its release date was later than the benchmark curation. Additional details are in website.

Model	Hard	Medium	Easy	Rating	Pct.%	AvgTok	AvgCost
Reasoning Models							
o4-mini-high	0.0%	53.5%	83.1%	2 116	1.5%	23 819	$0.1048
Gemini 2.5 Pro	0.0%	25.4%	70.4%	1 992	2.3%	29 879	$0.2988
o3-mini	0.0%	16.9%	77.5%	1 777	4.9%	18 230	$0.0802
DeepSeek R1	0.0%	9.9%	56.3%	1 442	18.0%	16 716	$0.0366
Gemini 2.5 Flash	0.0%	12.7%	47.9%	1 334	30.3%	35 085	$0.0116
DeepSeek R1 Distill-Llama-70B	0.0%	2.8%	33.8%	999	56.0%	12 425	$0.0050
Claude 3.7 Sonnet (Max Reasoning)	0.0%	1.4%	36.6%	992	56.5%	19 075	$0.2861
Gemini 2.0 Flash Reasoning	0.0%	0.0%	29.6%	893	63.1%	11 143	$0.0390
Non-Reasoning Models							
GPT-4.1 mini	0.0%	5.6%	28.2%	1 006	55.5%	2 662	$0.0043
DeepSeek V3 0324	0.0%	5.6%	32.4%	984	57.1%	2 712	$0.0030
GPT-4.1	0.0%	0.0%	23.9%	889	64.2%	2 131	$0.0170
GPT-4.5	0.0%	0.0%	26.8%	881	64.8%	968	$0.1452
Qwen-Max	0.0%	0.0%	14.1%	821	69.4%	1 244	$0.0080
Claude 3.7 Sonnet (No Reasoning)	0.0%	1.4%	16.9%	804	70.7%	3 554	$0.0533
Llama 4 Maverick	0.0%	0.0%	15.5%	634	80.4%	1 160	$0.0007
Claude 3.5 Sonnet	0.0%	0.0%	14.1%	617	81.4%	810	$0.0122
Gemma 3 27B	0.0%	0.0%	8.5%	601	82.5%	668	$0.0001
GPT-4o	0.0%	0.0%	9.9%	592	83.1%	1 133	$0.0227
Meta Llama 3.1 405B Instruct	0.0%	0.0%	9.9%	574	84.3%	568	$0.0005
DeepSeek V3	0.0%	0.0%	12.7%	557	84.9%	1 020	$0.0011

Multi-Attempt Somewhat Help, but Not Enough

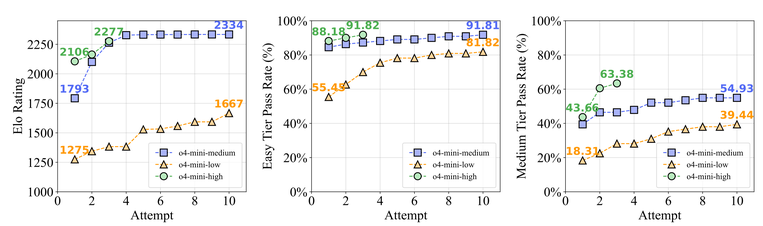

Figure 2. `o4-mini` **performance under pass@k settings.** The plot shows the pass rates for Easy and Medium tier problems as the number of attempts (k) increases. All variants show 0% pass rate on the hard tier in the evaluation.

ICPC and IOI

Models solving ICPC and IOI problems often outperform those on LiveCodeBench Pro (LCP Pro) because these contests include many easier problems, while LCP Pro has more consistently difficult challenges. Additionally, OpenAI and Google employ advanced parallel reasoning and extensive computation time. Finally, ICPC allows multiple solution attempts, whereas LCP Pro restricts submissions to a single pass.

Catch Our Latest Updates

New Feature! We bring LiveCodeBench Pro Verifier to your local machine. So you may training (by reinforcment learning) and evaluating your models or agents locally.

New Feature! We are collaborating with the Terminal-Bench team to support one-line code evaluation.

New Models! We are closely collaborating with frontier labs like OpenAI and xAI to add latest models to the benchmark.