LiveCodeBench Robustness and Generalization Across Peer-Reviewed Evaluations

Assignee Research

June 3, 2026

Abstract

This report synthesises findings from 15 peer-reviewed papers addressing the following research question: LiveCodeBench benchmark: robustness and generalization analysis — rotation 0. 16 claims were extracted from source literature; 0 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 4.3/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. Research question: LiveCodeBench benchmark: robustness and generalization analysis — rotation 0.

2 Methodology

Systematic literature search across multiple databases yielded 15 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 4.3/10.

3 Results

15 papers retrieved. 16 claims extracted; 0 independently verified. Quality review score: 4.3/10.

4 Limitations

This report is a machine-generated literature synthesis and does not constitute original research. Automated retrieval and verification may introduce

errors or omissions. Review scores reflect automated assessment, not human peer review. Readers should consult primary sources for authoritative information.

5 Extracted Claims

Claim	Verified	Confidence
LiveCodeBench curates problems from three coding competition websites: LeetCode, AtCoder, and CodeForces.	×	0.13
The problems consist of a natural language problem statement along with example input-output examples.	×	0.12
The goal is to write a program that passes a set of hidden tests.	×	0.03
Thousands of participants participate, solving these problems thus ensuring that the problems are vetted for clarity and	×	0.02
HTML scrapers are used to collect problems and corresponding metadata from the websites.	×	0.01
Problems with images are excluded to ensure quality and consistency.	×	0.03
Problems that are not suitable for grading by input-output examples are excluded.	×	0.04
Ground truth solutions and test cases are collected whenever directly available.	×	0.04
Each problem is associated with a contest date D to mark the release date.	×	0.01
The release date allows measuring the performance of LLMs over different time windows by filtering problems based on the	×	0.05
A UI is developed to compare models on problems released during different time windows.	×	0.04
Tests are crucial for assessing the correctness of the generated outputs and are used in all four scenarios.	×	0.03
Tests available on platform websites are collected whenever possible.	×	0.02
GPT-4-Turbo is used to generate tests for problems when they are not available on the platform.	×	0.02
Input generators are constructed to sample inputs based on problem specifications using in-context learning.	×	0.04
A small fraction of failing tests are collected from the platform for more recent problems to allow more directed advers	×	0.03

References

- http://arxiv.org/abs/2506.11928v1
- http://arxiv.org/abs/2403.07974v2
- http://arxiv.org/abs/2507.16200v1