Scaling Model Size and Syntax Error Reduction in CoT-Generated Code for BigCodeBench

Assignee Research

May 30, 2026

Abstract

This report synthesises findings from 13 peer-reviewed papers addressing the following research question: Does model size scaling (e.g., 7B vs. 13B vs. 30B parameters) correlate with syntax error reduction in CoT-generated code for structured data tasks on BigCodeBench. Large language models (LLMs) have demonstrated impressive performance in code generation, particularly when augmented with chain-of-thought (CoT) prompting techniques. They break down requirements into intermediate reasoning steps, which act as design rationales to guide LLMs in. 6 claims were extracted from source literature; 0 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 3.7/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: Are They All Good? Evaluating the Quality of CoTs in LLM-based Code Generation. Research question: Does model size scaling (e.g., 7B vs. 13B vs. 30B parameters) correlate with syntax error reduction in CoT-generated code for structured data tasks on BigCodeBench?.

2 Methodology

Systematic literature search across multiple databases yielded 13 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 3.7/10.

3 Results

13 papers retrieved. 6 claims extracted; 0 independently verified. Quality review score: 3.7/10.

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
LLMs can significantly outperform a single model in code generation when using Chain of Thought (CoT) prompting.	×	0.13
The effectiveness of LLMs' self-repair capabilities increases with more detailed feedback about errors.	×	0.08
The study proposes a comprehensive taxonomy of CoT errors, including both external and internal factors.	×	0.12
Even correct CoTs can sometimes lead to faulty code, while incorrect CoTs may still generate correct code.	×	0.11
More detailed error information improves LLMs' performance in refining faulty CoTs.	×	0.08
LLMs have gained prominence in Software Engineering for automatic code generation.	×	0.08

References

- http://arxiv.org/abs/2308.08784v2
- http://arxiv.org/abs/2408.11029v2
- http://arxiv.org/abs/2507.06980v1