EVOR Pipeline vs. Static Retrieval Baselines in Code Generation Latency and Throughput

Assignee Research

June 9, 2026

Abstract

This report synthesises findings from 11 peer-reviewed papers addressing the following research question: How does the EVOR pipeline's retrieval efficiency compare to static retrieval baselines on code generation benchmarks like MultiPL-E in terms of latency and throughput metrics. 14 claims were extracted from source literature; 0 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 4.5/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: EVOR: Evolving Retrieval for Code Generation. Research question: How does the EVOR pipeline's retrieval efficiency compare to static retrieval baselines on code generation benchmarks like MultiPL-E in terms of latency and throughput metrics.

2 Methodology

Systematic literature search across multiple databases yielded 11 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 4.5/10.

3 Results

11 papers retrieved. 14 claims extracted; 0 independently verified. Quality review score: 4.5/10.

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
With CodeLlama, the improvements of MPSC, ExeDec, and Reflexion are smaller than 2% on average compared to vanilla gener	×	0.08
The execution accuracy remains 0 in the Ring dataset across MPSC, ExeDec, and Reflexion methods using CodeLlama.	×	0.05
DocPrompting significantly surpasses MPSC, ExeDec, and Reflexion by explicitly using documentation.	×	0.03
EVOR achieves a 16.1% absolute gain over DocPrompting when using ChatGPT.	×	0.06
EVOR achieves a 16.2% absolute gain over DocPrompting when using CodeLlama.	×	0.06
DocPrompting uses documentation as a single retrieval source without evolution in both queries and knowledge.	×	0.12
The default metric used throughout the paper is execution accuracy (pass@1).	×	0.03
The Vanilla baseline generates outputs directly from LLMs based on the coding question without augmenting external knowl	×	0.05
MPSC prompts LLMs to generate diverse outputs from three perspectives: Solution, Specification, and Test case.	×	0.03
MPSC constructs a 3-partite graph and picks the optimal choice of solutions based on confidence scores.	×	0.02
ExeDec employs a subgoal model to predict the subgoal of the desired program state for the next part of the program.	×	0.04
ExeDec uses a synthesizer model to generate the corresponding subprogram to achieve the predicted subgoal.	×	0.02
Retrieval-augmented code generation introduces risks of biased or incorrect information being retrieved.	×	0.11
Retrieval-augmented code generation poses privacy and security concerns if sensitive code snippets are inadvertently inc	×	0.10

References

- http://arxiv.org/abs/2402.12317v2
- http://arxiv.org/abs/2508.21256v1
- http://arxiv.org/abs/2111.13057v3