GNN and LLM Inference Efficiency in Code Generation on BIGCode

Assignee Research

June 2, 2026

Abstract

This report synthesises findings from 8 peer-reviewed papers addressing the following research question: How does the inference efficiency of GNN-based code generation models compare to traditional LLM-based approaches when evaluated on the BIGCode dataset using metrics like latency and tokens per second. Tokens are the basic units of Large Language Models (LLMs). LLMs rely on tokenizers to segment text into these tokens, and tokenization is the primary determinant of computational and inference cost. 8 claims were extracted from source literature; 0 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 3.8/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: Is Sanskrit the most token-efficient language? A quantitative study using GPT, Gemini, and SentencePiece. Research question: How does the inference efficiency of GNN-based code generation models compare to traditional LLM-based approaches when evaluated on the BIGCode dataset using metrics like latency and tokens per second?.

2 Methodology

Systematic literature search across multiple databases yielded 8 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 3.8/10.

3 Results

8 papers retrieved. 8 claims extracted; 0 independently verified. Quality review score: 3.8/10.

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
The study evaluated tokenizers on 701 Bhagavad Gita verses using two different input configurations.	×	0.09
Experiment 1 measured cross-language efficiency for equivalent semantic content using four inputs: Sanskrit (Devanagari)	×	0.09
Experiment 2 measured semantic expansion by tokenizing Sanskrit verses, English translations, English meanings, Hindi tr	×	0.08
Tokenizers were applied without any model inference to ensure results were based on the tokenizer's vocabulary behavior.	×	0.05
Under the SentencePiece (SPM) baseline, Sanskrit demonstrates superior information density (5.07 chars/token) and requir	×	0.10
The cl100k base model fails to capture Sanskrit's density, forcing Sanskrit's CpT down to 1.13.	×	0.09
Table 1 reports Mean Token Count and Characters per Token (CpT) for information density.	×	0.11
Table 2 quantifies the semantic expansion—the additional token count required to translate a compact Sanskrit verse into	×	0.11

References

- http://arxiv.org/abs/2509.21557v2
- http://arxiv.org/abs/2408.06717v3
- http://arxiv.org/abs/2601.06142v1