CodeT5 Zero-Shot Reasoning Performance on Python Vulnerability Detection via Procedural Pretraining

Assignee Research

June 7, 2026

Abstract

This report synthesises findings from 13 peer-reviewed papers addressing the following research question: How does the zero-shot reasoning performance of CodeT5, when pretrained on procedural code data versus natural language data, compare on the CWE-200 benchmark for Python vulnerability detection. 13 claims were extracted from source literature; 3 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 5.8/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: Procedural Pretraining: Warming Up Language Models with Abstract Data. Research question: How does the zero-shot reasoning performance of CodeT5, when pretrained on procedural code data versus natural language data, compare on the CWE-200 benchmark for Python vulnerability detection?.

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 5.8/10.

3 Results

13 papers retrieved. 13 claims extracted; 3 independently verified. Quality review score: 5.8/10.

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
Procedural pretraining improves performance and accelerates language model pretraining compared to standard pretraining	×	0.10
Procedural pretraining consistently improves over standard pretraining using only 0.1% to 0.3% extra procedural tokens.	×	0.08
On the C4 dataset, procedural pretraining enables models to reach the same loss with 55% of the original data volume.	✓	0.16
On the CODEPARROT dataset, procedural pretraining enables models to reach the same loss with 67% of the original data vo	✓	0.16
On the DEEPMIND-MATH dataset, procedural pretraining enables models to reach the same loss with 86% of the original data	✓	0.19
The study validates findings across model sizes up to 1.3 billion parameters.	×	0.06
The study validates findings across data sizes up to 10.5 billion tokens.	×	0.04
Pretrained information from procedural data is localized in specific layers, with MLP and attention layers contributing	×	0.09
Different types of procedural pretraining facilitate learning different algorithmic skills.	×	0.10
Shuffling the sequences of procedural data negates the performance improvements observed with structured procedural pret	×	0.10
The best procedural data type for the Haystack task is 16 DYCK.	×	0.08
The best procedural data type for the Reversed Addition task is ECA.	×	0.04
The best procedural data type for the Sorting task is a combination of UNION and DELETE.	×	0.04

References

- http://arxiv.org/abs/2603.21389v1
- http://arxiv.org/abs/2601.21725v2
- http://arxiv.org/abs/2308.16149v2