

JaCoText Robustness to Adversarial Noise on QuixBugs Java Benchmarks

Assignee Research

June 8, 2026

Abstract

This report synthesises findings from 13 peer-reviewed papers addressing the following research question: How does the JaCoText model's ability to maintain functional correctness under adversarial noise on QuixBugs differ from its baseline performance on unperturbed Java bugs. 9 claims were extracted from source literature; 0 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 2.8/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: A Deep Dive into Adversarial Robustness in Zero-Shot Learning. Research question: How does the JaCoText model's ability to maintain functional correctness under adversarial noise on QuixBugs differ from its baseline performance on unperturbed Java bugs?.

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 2.8/10.

3 Results

13 papers retrieved. 9 claims extracted; 0 independently verified. Quality review score: 2.8/10.

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 CUB dataset has 312 attributes, 200 classes, and 11788 images.	×	0.02
The SUN dataset has 102 attributes, 717 classes, and 14340 images.	×	0.02
The AWA2 dataset has 85 attributes, 50 classes, and 37322 images.	×	0.02
The standard per-class top-1 accuracy is used for ZSL evaluation.	×	0.05
For GZSL, per-class top-1 accuracy values for seen and unseen classes are used to compute harmonic-scores.	×	0.04
The ResNet-101 feature extractor is merged with the ALE model to make the computational graph end-to-end differentiable.	×	0.01
PyTorch is used for the experiments.	×	0.04
The ALE model is formulated as $F(x, y; W) = \theta(x)W^T \varphi(y)$, where $\theta(x)$ is the visual and $\varphi(y)$ is the class embeddings.	×	0.02
The ALE model is one of the earlier studies that showed direct mapping by exploiting data and auxiliary information is m	×	0.02

References

- <http://arxiv.org/abs/2008.07651v1>
- <http://arxiv.org/abs/2303.12869v1>
- <http://arxiv.org/abs/1805.03454v4>