Inference Efficiency Trade-offs Across Qwen3-235B Model Scales on SWE-Bench Verified Tasks

Assignee Research

May 30, 2026

Abstract

This report synthesises findings from 10 peer-reviewed papers addressing the following research question: How does inference efficiency (latency and throughput) vary across Qwen3-235B model sizes when processing SWE-bench Verified tasks, and does training data contamination exacerbate or mitigate. The issue-resolving task, where a model generates patches to fix real-world bugs, has emerged as a critical benchmark for evaluating the capabilities of large language models (LLMs). While SWE-bench and its variants have become standard in this domain, they suffer from key. 16 claims were extracted from source literature; 0 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 3.5/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: SWE-bench Goes Live!. Research question: How does inference efficiency (latency and throughput) vary across Qwen3-235B model sizes when processing SWE-bench Verified tasks, and does training data contamination exacerbate or mitigate efficiency trade-offs?.

2 Methodology

Systematic literature search across multiple databases yielded 10 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 3.5/10.

3 Results

10 papers retrieved. 16 claims extracted; 0 independently verified. Quality review score: 3.5/10.

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
SWE-bench-Live was evaluated using three agent frameworks: OpenHands, SWE-Agent, and Agentless.	×	0.11
OpenHands was set to a maximum of 60 iterations per instance.	×	0.03
SWE-Agent was limited to 100 LLM calls per instance.	×	0.04
Agentless was evaluated without the reranking stage based on regression testing.	×	0.04
Four LLMs were used for testing: GPT-4o, GPT-4.1, Claude 3.7 Sonnet, and DeepSeek V3.	×	0.02
The primary evaluation metric is Resolved Rate (%).	×	0.02
Patch Apply Rate (%) measures the percentage of generated patches that are syntactically correct and can be applied with	×	0.03
Localization Success Rate (%) reflects whether the set of files modified by the generated patch matches the gold patch.	×	0.02
The highest resolved rate on SWE-bench-Live is 19.25%.	×	0.08
Recent state-of-the-art agents and models report a resolved rate exceeding 60% on the SWE-bench Verified subset.	×	0.13
SWE-bench-Live includes 93 repositories with an average of 85k lines of Python code and 423 files.	×	0.09
SWE-bench-Live includes 1319 instances with an average of 3.3 files, 9.0 hunks, and 102.6 lines per gold patch.	×	0.06
SWE-bench-Live includes an average of 5.4 F2P test cases and 2953.4 P2P test cases per instance.	×	0.06
OpenHands with GPT-4o achieved a resolved rate of 7.00%, a patch apply rate of 72.00%, and a localization success rate o	×	0.02
SWE-Agent with GPT-4o achieved a resolved rate of 10.00%, a patch apply rate of 93.33%, and a localization success rate	×	0.03
Agentless with GPT-4o achieved a resolved rate of 11.67%, a patch apply rate of 91.67%, and a localization success rate	×	0.01

References

- http://arxiv.org/abs/2603.00520v1
- http://arxiv.org/abs/2412.21139v2
- http://arxiv.org/abs/2505.23419v2