# Multi-SWE-bench: A Multilingual Benchmark for Issue Resolving

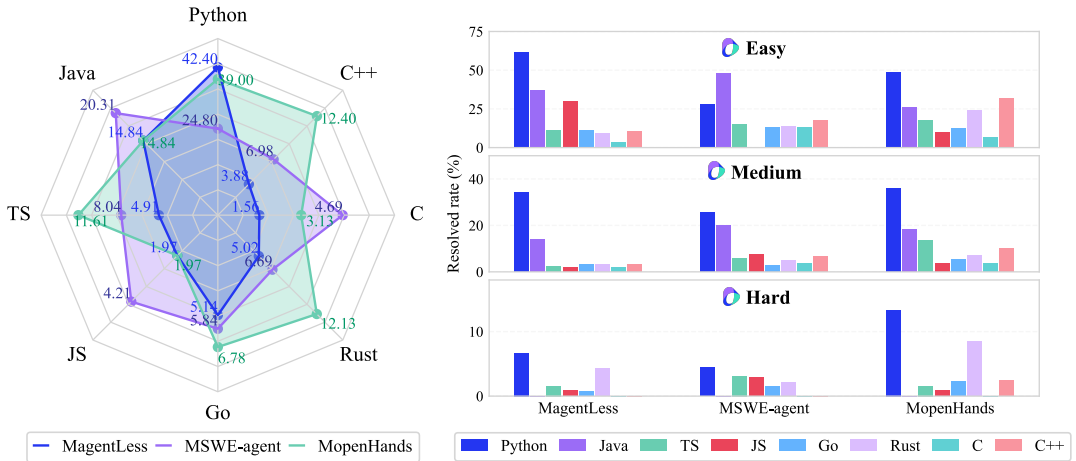
**ByteDance Seed**

Leaderboard Benchmark RL Community GitHub Repo




## Abstract

The task of issue resolving is to modify a codebase to generate a patch that addresses a given issue. However, existing benchmarks, such as SWE-bench, focus almost exclusively on Python, making them insufficient for evaluating Large Language Models (LLMs) across diverse software ecosystems. To address this, we introduce a multilingual issue-resolving benchmark, called Multi-SWE-bench, covering Java, TypeScript, JavaScript, Go, Rust, C, and C++. It includes a total of 1,632 high-quality instances, which were carefully annotated from 2,456 candidates by 68 expert annotators, ensuring that the benchmark can provide an accurate and reliable evaluation. Based on Multi-SWE-bench, we evaluate a series of state-of-the-art models using three representative methods (Agentless, SWE-agent, and OpenHands) and present a comprehensive analysis with key empirical insights. In addition, we launch a Multi-SWE-RL open-source community, aimed at building large-scale reinforcement learning (RL) training datasets for issue-resolving tasks. As an initial contribution, we release a set of 4,723 well-structured instances spanning seven programming languages, laying a solid foundation for RL research in this domain. More importantly, we open-source our entire data production pipeline, along with detailed tutorials, encouraging the open-source community to continuously contribute and expand the dataset. We envision our Multi-SWE-bench and the ever-growing Multi-SWE-RL community as catalysts for advancing RL toward its full potential, bringing us one step closer to the dawn of AGI.


(a) Overall resolved rate (%)

(b) Resolved rate (%) across various difficulty levels

Figure 1. Resolved rate (%) on Multi-SWE-bench (Claude-3.5-Sonnet).

Author contributions listed at end of paper. Correspondence to: {zandaoguang, shen.kai}@bytedance.com

## Contents


**1 Introduction** **3**

**2 Related Work** **4**

**3 Multi-SWE-bench** **5**

3.1 Benchmark Construction . . . . . 5

3.1.1 Phase 1: Repository Selection . . . . . 6

3.1.2 Phase 2: Pull Request Crawling . . . . . 6

3.1.3 Phase 3: Environment Determination . . . . . 6

3.1.4 Phase 4: Pull Request Filtering . . . . . 7

3.1.5 Phase 5: Manual Verification . . . . . 7

3.2 Features of Multi-SWE-bench . . . . . 9

**4 Multi-SWE-RL Open-Source Community** **10**

**5 Experimental Setups** **11**

5.1 Evaluated LLMs and Methods . . . . . 11

5.2 Evaluation Metrics . . . . . 12

**6 Experimental Results** **13**

6.1 Performance on Multi-SWE-bench . . . . . 13

6.1.1 Performance across Programming Languages . . . . . 13

6.1.2 Performance across Various Methods and LLMs . . . . . 15

6.1.3 Performance across Different Repositories . . . . . 17

6.2 Influencing Factors of Performance . . . . . 19

6.2.1 Issue Type . . . . . 20

6.2.2 Characteristics of Issue Description . . . . . 21

6.2.3 Characteristics of Fix Patches . . . . . 22

6.3 Case Study . . . . . 23

6.4 Resource Consumption . . . . . 25

6.5 Troubleshooting . . . . . 26

**7 Conclusions and Future Works** **27**