Artisan: Automated Artifact Evaluation with Agents

ANONYMOUS AUTHOR(S)

This is an anonymous submission by the anonymous authors

CCS Concepts: • **Software and its engineering → Software testing and debugging**.

Additional Key Words and Phrases: Artifact Evaluation, Benchmarks, LLM agents

ACM Reference Format:
Anonymous Author(s). 2025. Artisan: Automated Artifact Evaluation with Agents. 1, 1 (September 2025), 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

This is a sample citation [1].

In summary, this paper contributes the following:

- SERE-Bench, a novel benchmark that evaluates the capabilities of AI agents to reproduce Software Engineering papers using script submission.
- Artisan, an LLM-based approach which incorporates several tools to aid the reproduction of software engineering research artifacts.
- Empirical evaluation which shows that (1) SERE-Bench is challenging for the current state of the art LLM agents and (2) Artisan effectively addresses the common pitfalls faced by LLM agents.

2 Benchmark

2.1 Overview

The notable difference between our work and previous work is that we are script-based. Why is it important? It is immediately useful. It acts as a standalone script that is push-button runnable which reproduces the particular table. It is supported by a concrete execution. What's a good word to capture this? Realistic? Execution-based? Execution-proof? Evidence-accompanying? Comparison with existing work in that they formulate reproduction as a prediction problem but we as a code generation problem? It (loosely) implies that this can be used to correct mistakes made in the paper. Previous works have not pointed out any errors in the paper. We don't need separate ground truth. Numbers in the paper is the ground truth. Less prone to data combination? Or it doesn't matter if it's contaminated because it's useful? What's the argument for this?

2.2 Paper Selection

- Top SE (ICSE, FSE, ASE, ISSTA) with Available and Reusable badges 2024: 114 - Find artifact url - Packaged using Docker: 77 - Exclude Commercial LLM API use Github token (non-public): 67 - Exclude GPU use: 46 - ess than 8 hours per task (when they mention it): 40

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM XXXX-XXXX/2025/9-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Table 1. Comparison of prior benchmarks and our benchmark.

	SUPER	CORE-Bench	ReproBench	SERE-Bench (Ours)
Domain	ML, NLP	Computer Science, Social Science, Medicine	Social Science	Software Engineering
Language	Python	Python, R	Python, R, Julia, Stata, MATLAB, Julia, Multiple Languages[1]	Python, R, Julia, C, C++, C#, Go, JS, TS, Perl, Java, Scala, Rust, Ocaml
#Paper	45	90[2]	112	40
#Task	45[3]	270	112	162
Time Limit	30 minutes	45 minutes	2 hours	8 hours
Input	Pre-processed and curated	Pre-processed and curated	Realistic	Realistic
Judging	Accuracy of outputs	Accuracy of outputs	Accuracy of reproducibility scores	Script reproducing the exact numbers
Discovered inconsistensies	0	0	0	3+

Note. [1] As labeled "Multiple Languages" in the source figure. *Note.* [2] 37 for Computer Science *Note.* [3] In terms of Expert set, They additionally provide 152 masked sets and 604 Auto sets.

2.3 Task Selection

We want our benchmark to be - No cherry-picking - Extensive (100 300) but not too expensive and long-running to run. - Manual validation is desired but should not take too much time.

There are few different options: Run full task set without modification (162) Manually validate all task set (162) to filter out unreproducible and keep the partial table Random sample to smaller size (25 / 50)

I argue for the second option (manually validate all task sets). It is not too much work (1 hour per task) to check whether it is sensible to reproduce the table by running artisan. We have about 127 tables to manually validate and we can sensibly validate them by the end of the ISSTA submission Makes our benchmark higher quality. Less underestimation of agent capability If we manually validate all without bias, no cherry-picking. Is using artisan as a time-saving measurement already a bias?

Randomly sample and manually validate and time budget with time budget.

3 Approach

4 Evaluation

5 Related work

References

[1] Doehyun Baek, Jakob Getz, Yusung Sim, Daniel Lehmann, Ben Titzer, Sukyoung Ryu, and Michael Pradel. 2024. Wasm-R3: Record-Reduce-Replay for Realistic and Standalone WebAssembly Benchmarks. In *Proceedings of the ACM on Programming Languages: Object-Oriented Programming, Systems, Languages & Applications (OOPSLA '24)*.