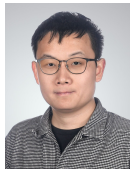




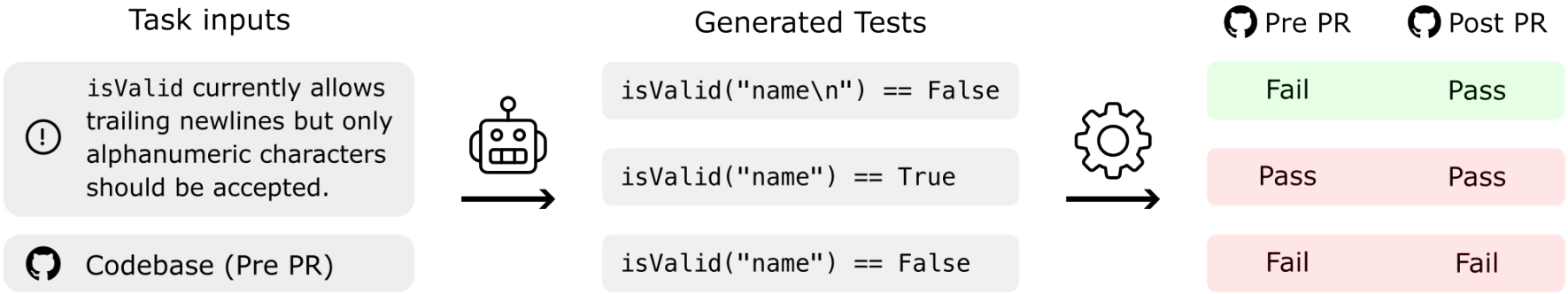

SWT-Bench:

Testing and Validating Real-World Bug-Fixes with Code Agents

Niels Mündler, Mark Niklas Müller, Jingxuan He, Martin Vechev

SRILAB ETH zürich logic*

GitHub based Issue-Reproduction Benchmark



Code Datasets

Function-level benchmarks almost saturated by SOTA models

Repository-level benchmarks gaining traction

Focus on Code Synthesis / Repair

Python Datasets	Code Generation	Test Generation
Single-Function	HumanEval APPS MBPP	TestEval
Repository Level	SWE-Bench RepoBench	SWT-Bench (ours)

Test generation

Metrics:
Codebase Coverage, Crashes (Fuzzing)

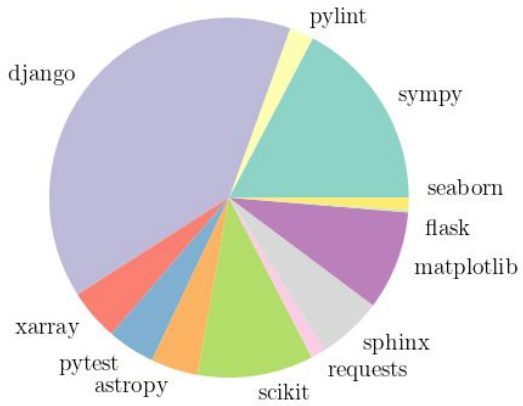
Methods:
Specialized solvers/analyzers, small transformers, bare bone LLMs

SWT-Bench

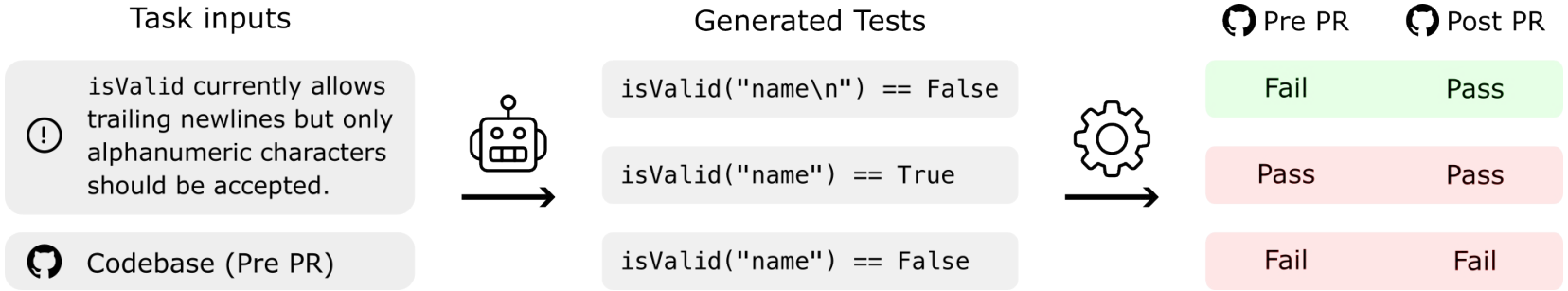
Software Testing dataset based on real-world GitHub repositories

1983 instances (276 in SWT-Bench Lite)

Task: Generate a test that reproduces a reported user issue



Composition



Metrics

Patch Well-Formedness: prediction is a valid patch

Success rate ($\mathcal{S}$): at least one test fails before a ground-truth bug fix is applied and all tests pass after

Coverage Increase ($\Delta\mathcal{C}$): Line coverage of modified code

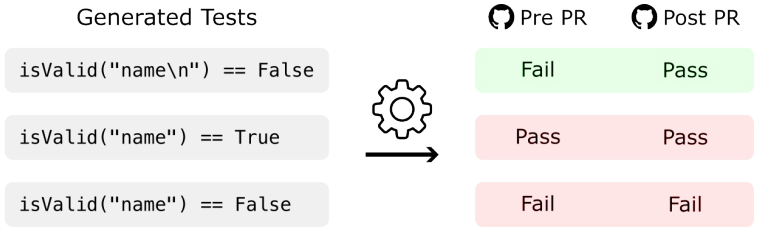
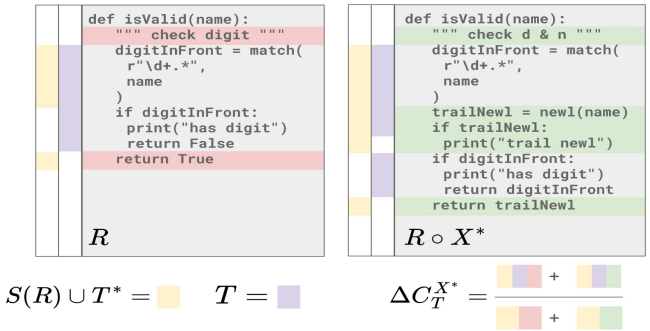


Illustration of Success Rate

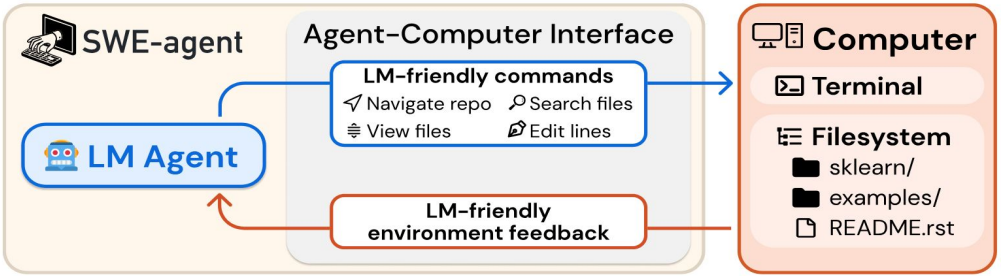


Definition of Coverage Increase $\Delta\mathcal{C}$

Code Agents

LLMs equipped with tooling

Capable of fixing bugs in large code bases



Source: Yang et. al: SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering, Neurips 2024

Can Code Agents write Unit-Tests in complex settings?

Plain LLMs, Specific Methods, Code Agents

ZeroShot

LIBRO

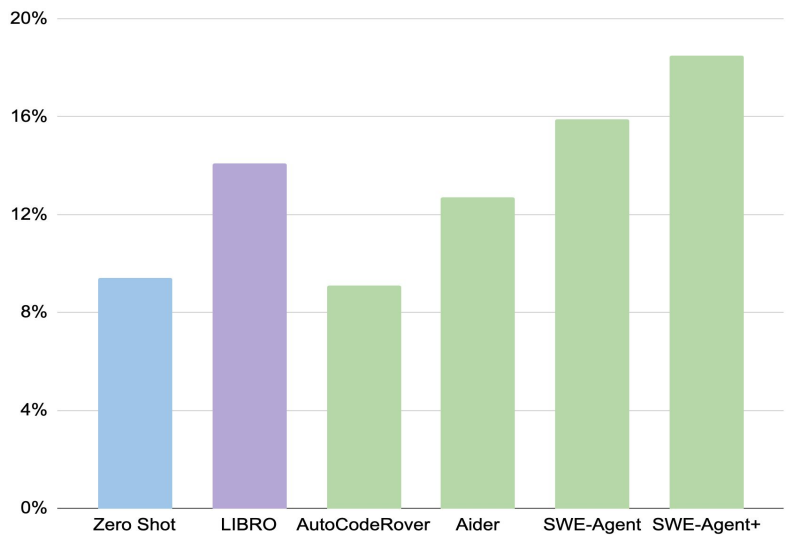
AutoCodeRover

Aider

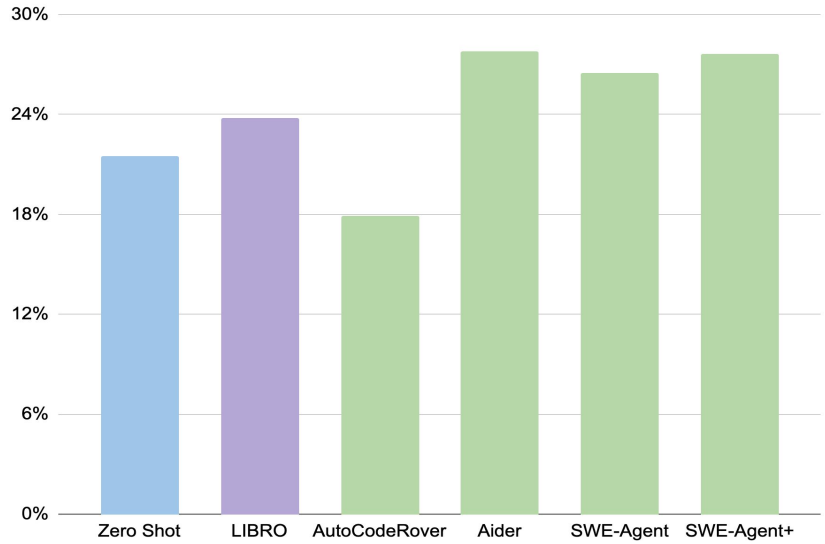
SWE-Agent

SWE-Agent+

Code Agents perform surprisingly well



Success Rate $\mathcal{S}$

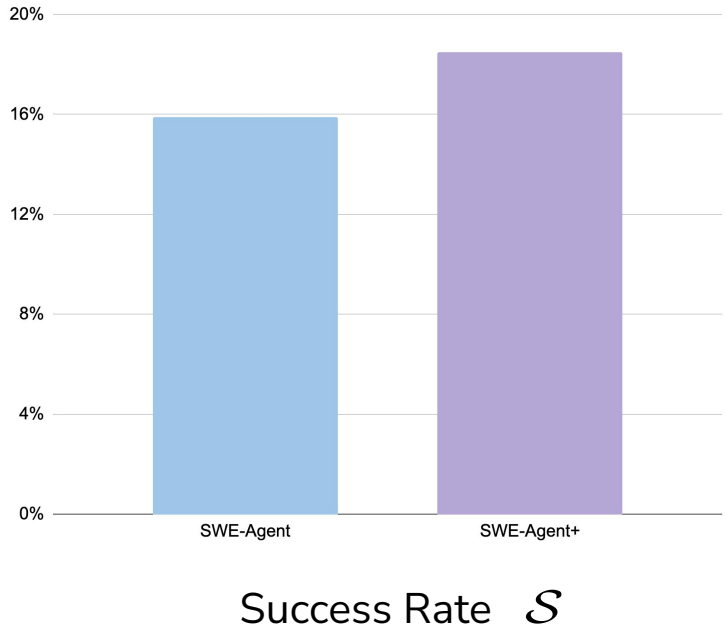


Coverage Increase $\Delta\mathcal{C}$

Execution Feedback helps significantly

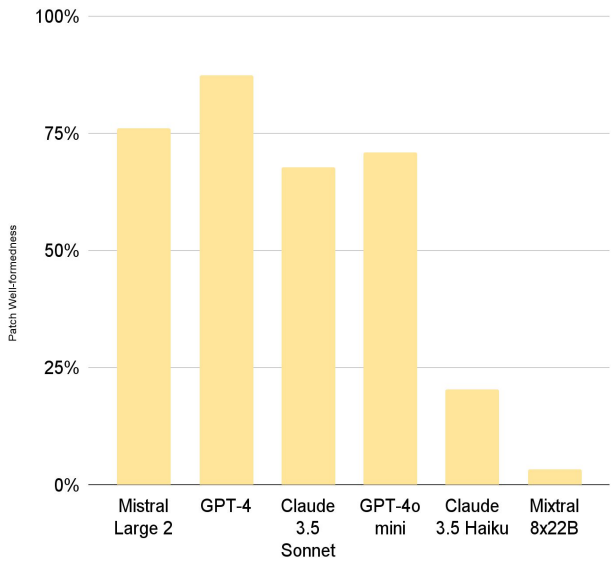
SWE-Agent+ is much stronger than SWE-Agent

Leverages feedback from running the test suite.

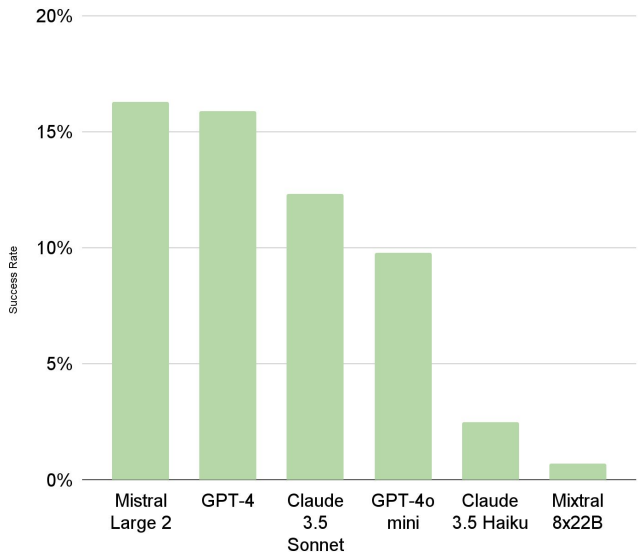


Performance depends on employed model

Smaller models struggle to produce valid patches



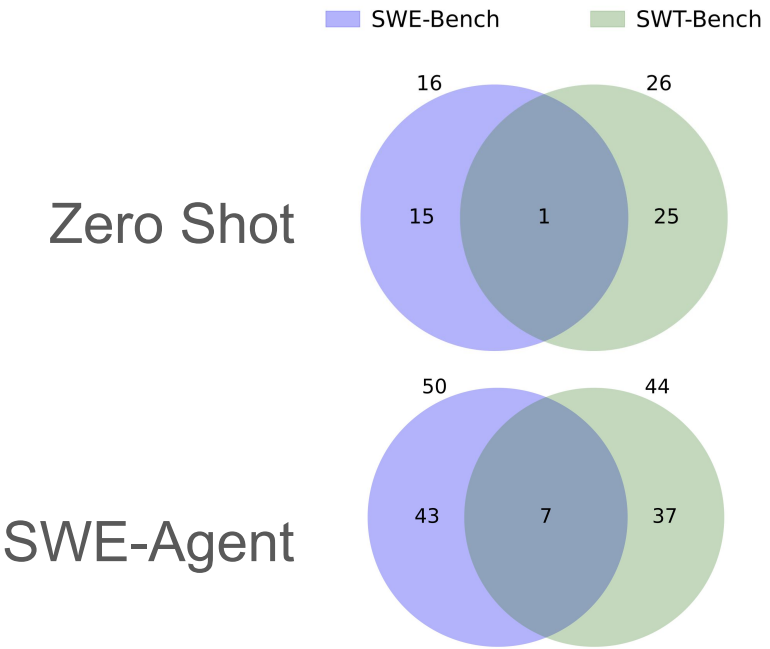
Valid patches of smaller models have lower quality



Small overlap between solved Repair and Testing

No significant correlation between solvability

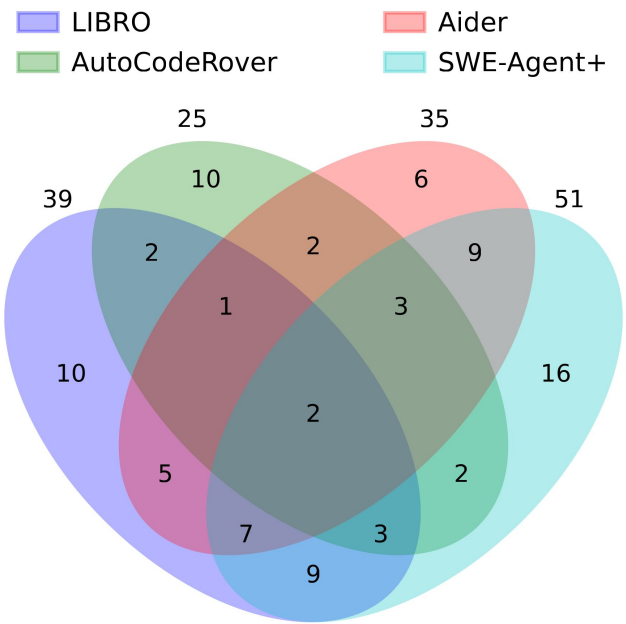
Testing and Repairing are distinct tasks



Different approaches are complementary

Few tasks solved by all approaches

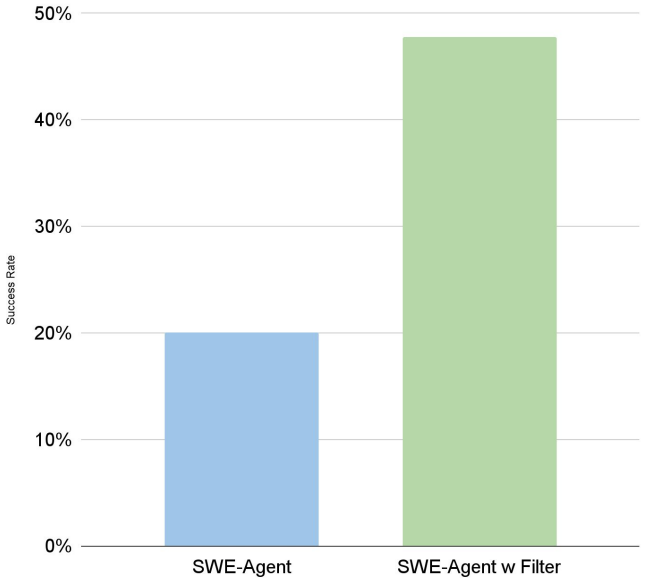
Employing different methods beneficial



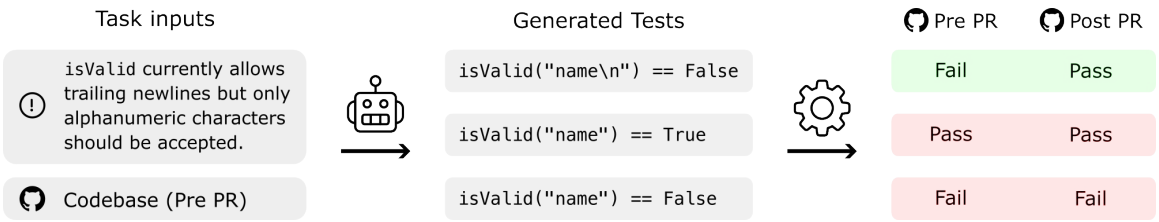
Cross-validation as promising filter for Repair

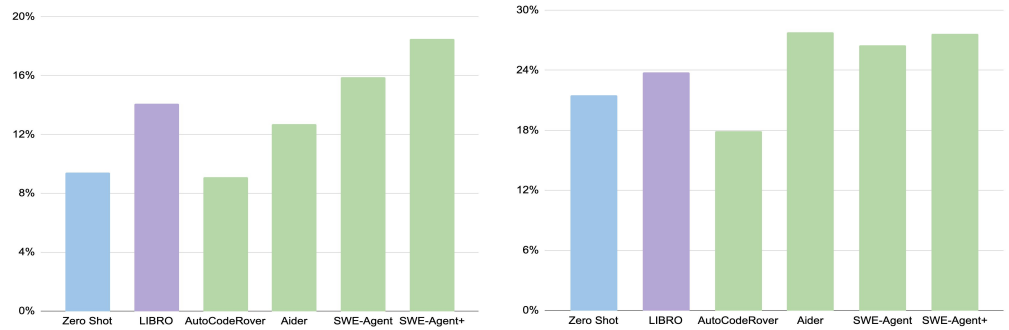
Filter the generated Patches by checking S on self-generated tests

More than doubles precision



More details + Benchmark code





Success Rate $\mathcal{S}$

Coverage Increase $\Delta\mathcal{C}$

https://github.com/logic-star-ai/swt-bench