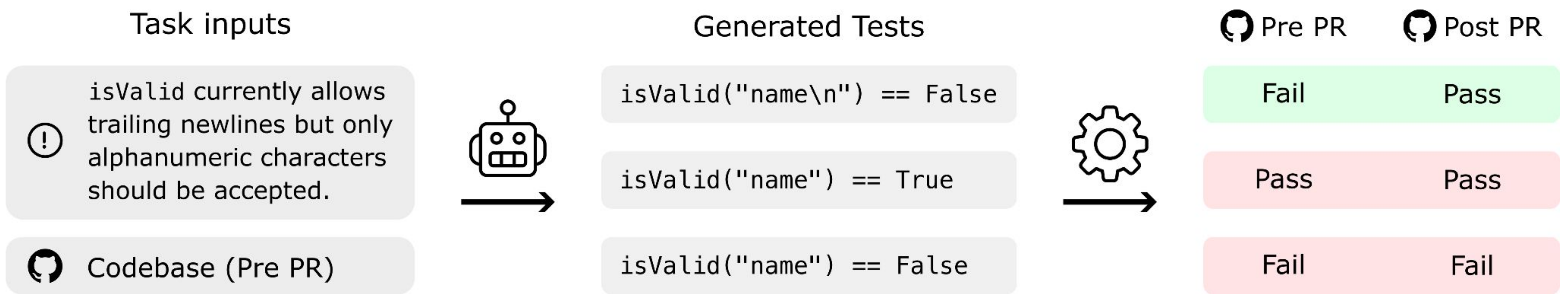
Code Agents are State of The Art Software Testers

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

SRILAB logic* ETH zürich

We propose SWT-Bench, a new test generation benchmark based on issue descriptions and real world code bases. We show that LLM-based Code Agents outperform zero-shot baselines and prior specialized methods at this task.

Benchmark Overview


Task inputs

isValid currently allows trailing newlines but only alphanumeric characters should be accepted.

Codebase (Pre PR)

Generated Tests

`isValid("name\n") == False`

`isValid("name") == True`

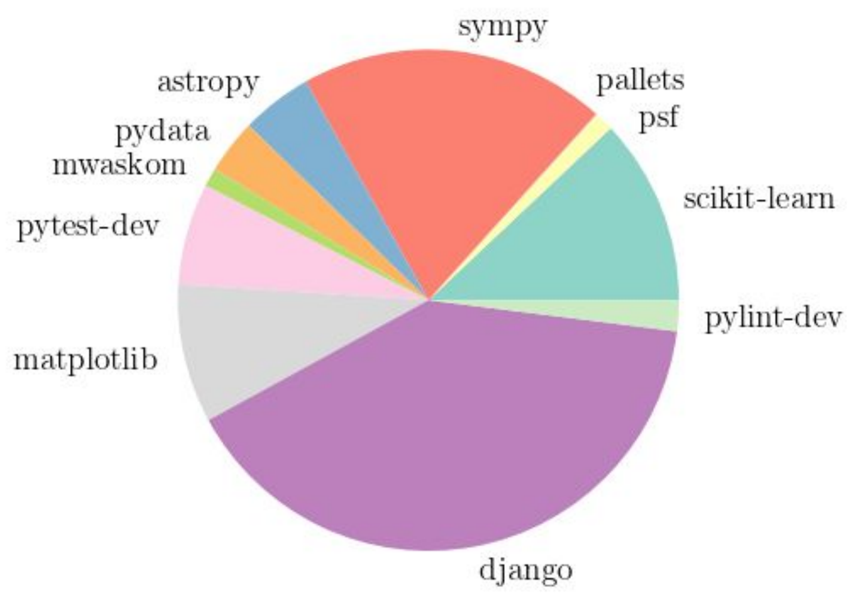
`isValid("name") == False`

	Pre PR	Post PR
`isValid("name\n") == False`	Fail	Pass
`isValid("name") == True`	Pass	Pass
`isValid("name") == False`	Fail	Fail

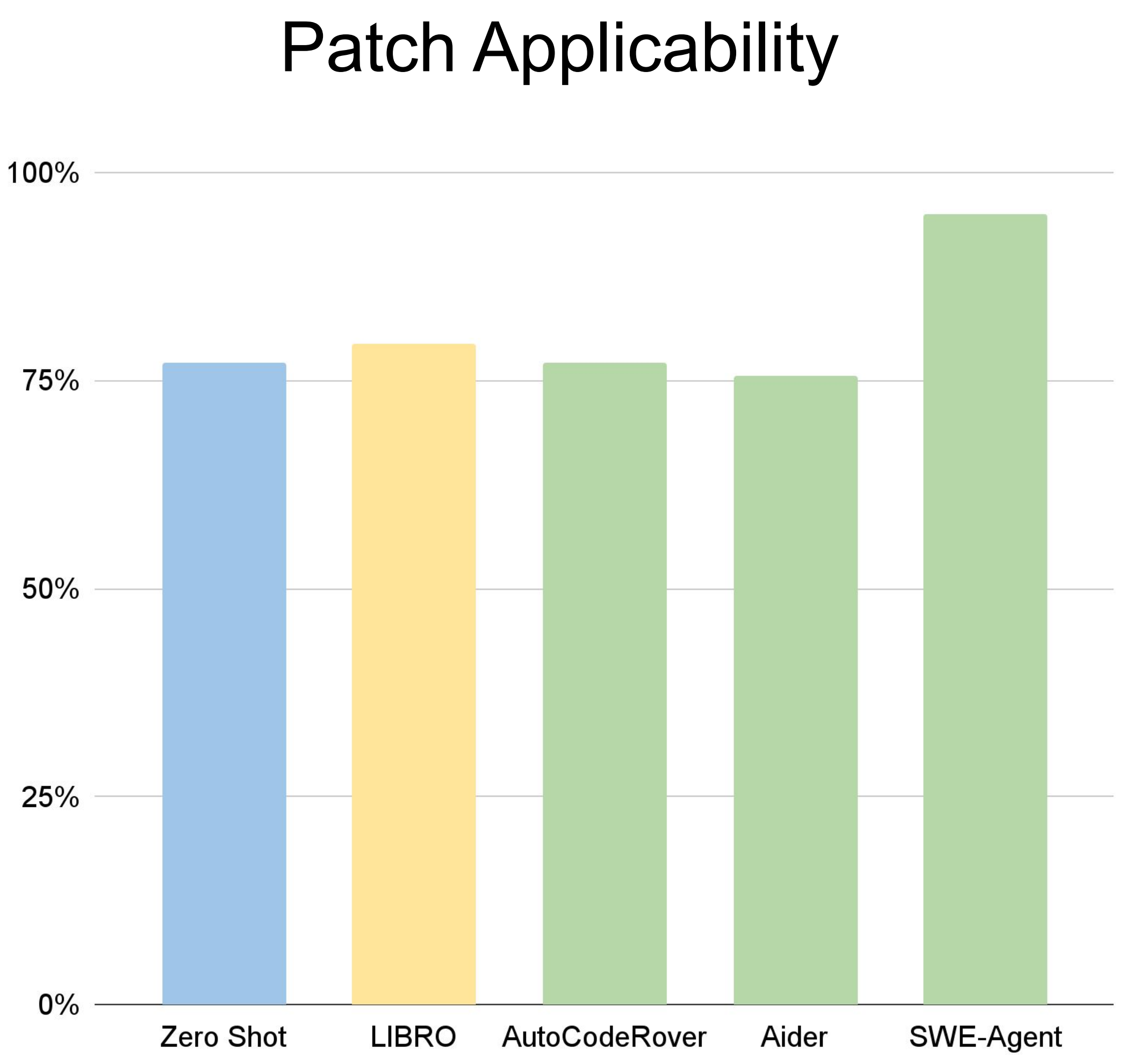
We derive a Software Testing dataset from SWE-Bench [1]. Target: predict a test patch that reproduces a reported user issue.

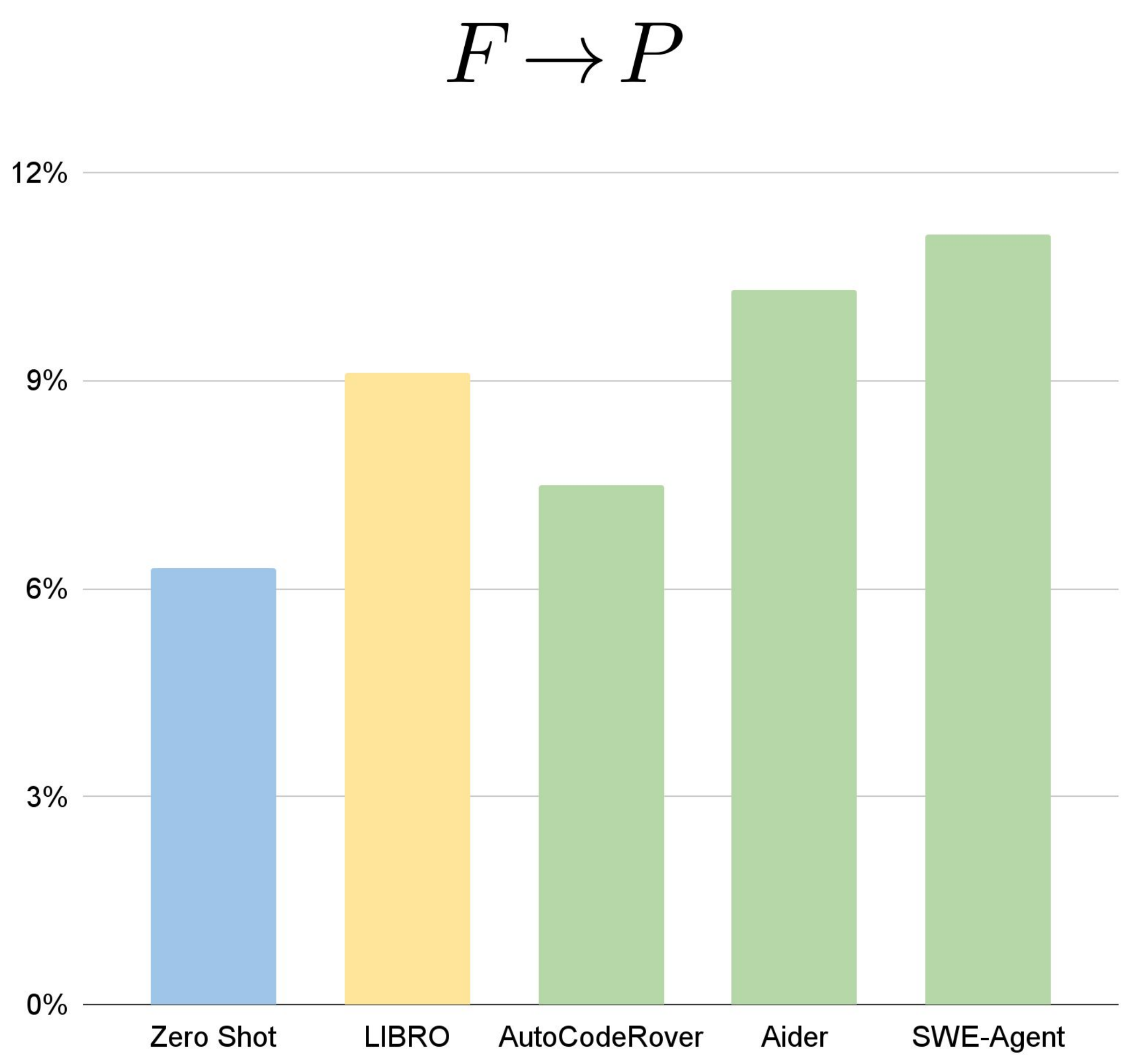
- Patch Applicability: Whether the prediction is a valid patch.
- Fail-to-Pass rate ($F \to P$): Cases where i) at least one test fails before golden code patch and ii) all tests pass after.
- Coverage ($\Delta\mathcal{C}$): Line coverage of the golden code patch.

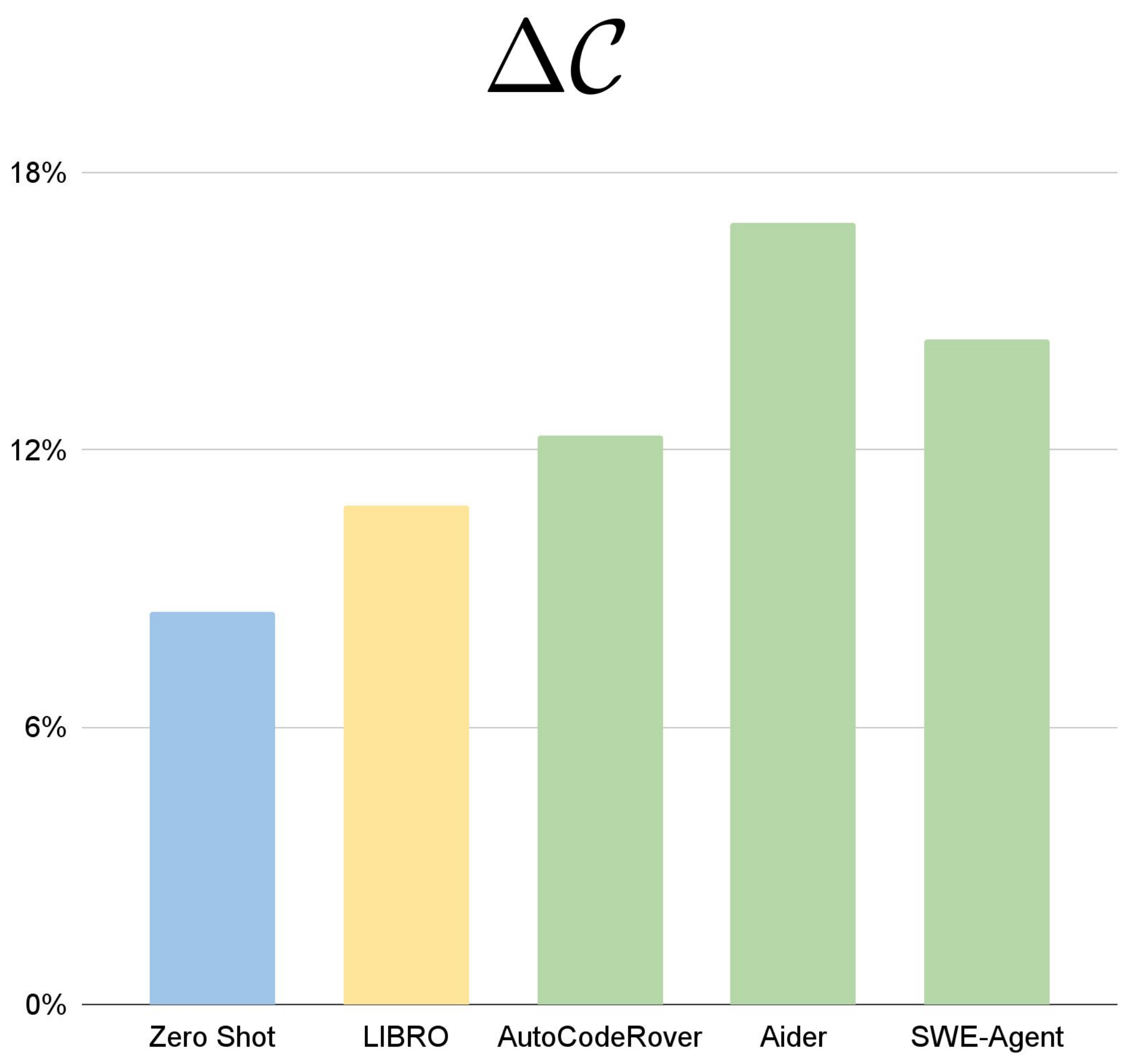
		Mean	Max
Issue Text	# Words	315.1	8756
Codebase	# Files	210.1	384
	# Lines	52330.8	122605
Existing Tests	# $F \to P$	0.05	55
	# $F \to F$	1.5	98
	# $P \to P$	91.4	4837
	# $P \to F$	0.3	40
	# total	105.1	4842
	Coverage	32.3%	67.7%
Golden Tests	# $F \to P$	1.5	952
	# $F \to F$	0.0	5
	# $P \to P$	1.6	766
	# $P \to F$	0.0	0
	# added	2.8	750
	# removed	0.3	104



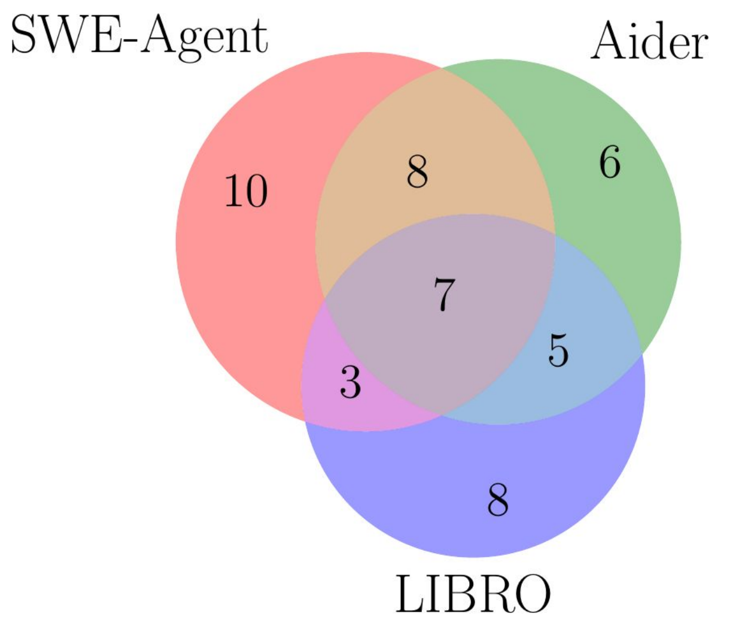
Experimental Results



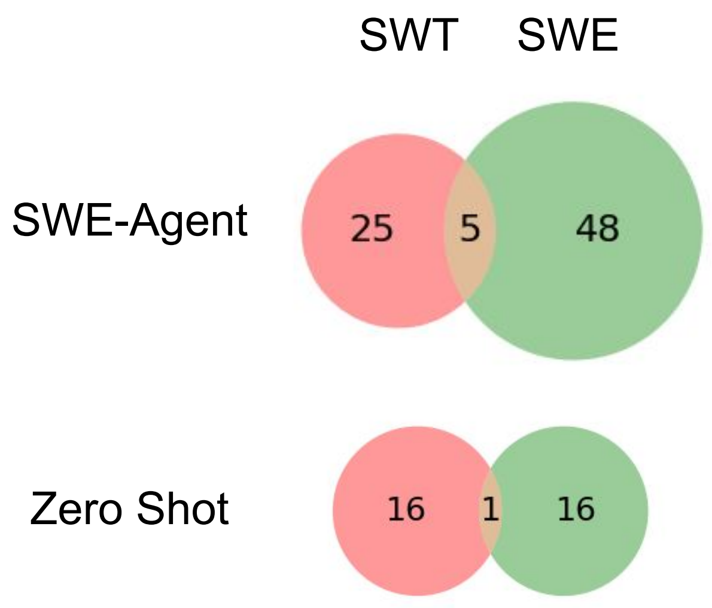




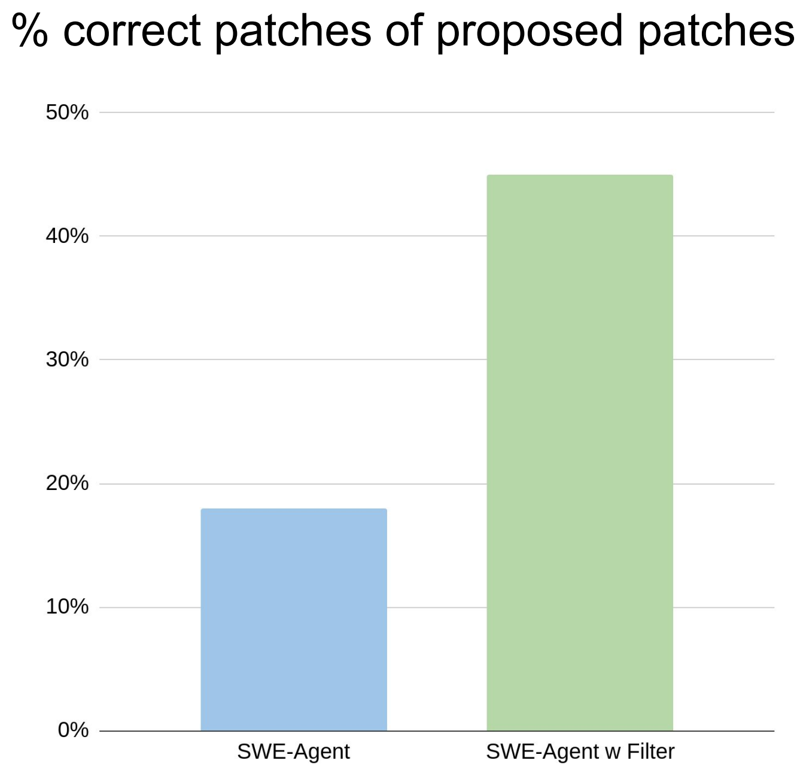
Agentic approaches perform as well or outperform **zero-shot baseline** and **specialized previous methods**



Small overlap in solved instances per agent



Small overlap with instances solved in SWE-Bench



Cross validation boosts patch precision on SWE-Bench

All experiments based on GPT-4. Zero Shot Baseline: GPT-4 + BM25 retrieved files, Previous Method: LIBRO [2], heuristic filtering of proposed tests. Code Agents: Aider [3], AutoCodeRover [4] and SWE-Agent [5] with modified prompts (i.e. "generate a test case", "run the test suite before submission")

[1]: Jimenez et al: Can language models resolve real-world GitHub issues?, 2024 [2] Kang et. al: Large language models are few-shot testers, 2023 [3] https://aider.chat [4] Zhang et al: AutoCodeRover, 2024 [5] Yang et al: SWE-Agent, 2024