AgentLens

Revealing the Lucky Pass Problem in SWE-Agent Evaluation

Priyam Sahoo[1,2*], Gaurav Mittal[2†], Xiaomin Li[2†], Shengjie Ma[2], Benjamin Steenhoek[2], Pingping Lin[2], Yu Hu[2]

[1] UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN [2] Microsoft

*Work done as a student researcher at Microsoft. †Co-second authors.

NeurIPS 2026 · Evaluations & Datasets Track

NEURAL INFORMATION PROCESSING SYSTEMS

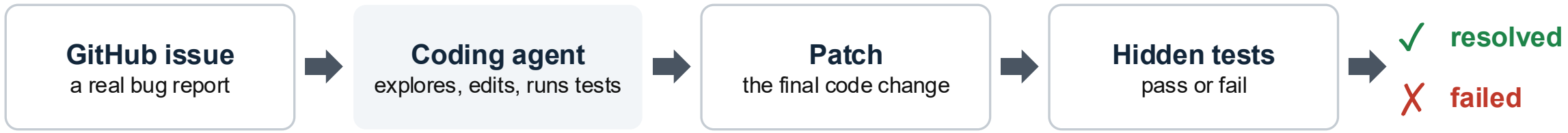
Outline

1. **Motivation**
   Why pass/fail isn't enough to judge a coding agent

2. **How AgentLens works**
   Turning agent logs into a quality score for each run

3. **Results**
   What we found across 8 models on SWE-bench Verified

Coding agents are judged on one bit: pass or fail



That one bit decides a lot:

Leaderboards

Models are ranked by how often they pass.

Training data

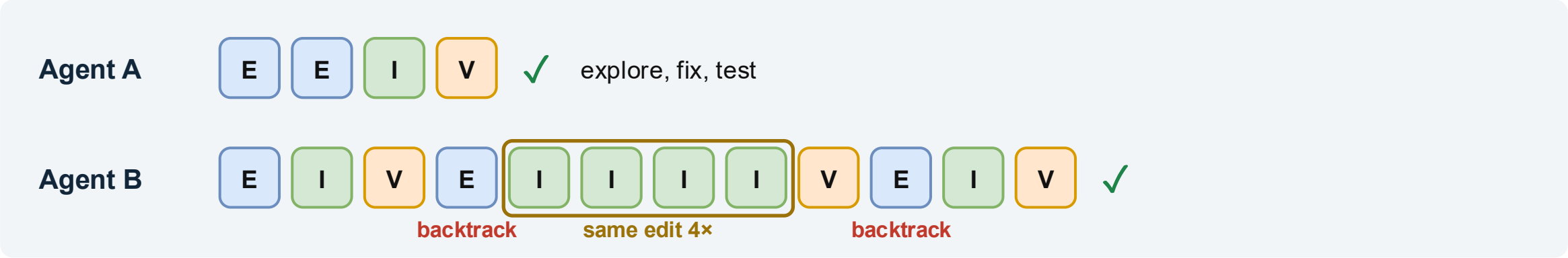
SWE-Gym, R2E-Gym and SWE-smith keep only the runs that pass.

Deployment

Teams pick agents based on pass rate.

SWE-bench: Jimenez et al., ICLR 2024

But two runs can pass in very different ways



E explore **I** implement **V** verify (run tests)

Same check mark, very different work. We call a run that passes with a weak process a **Lucky Pass**.

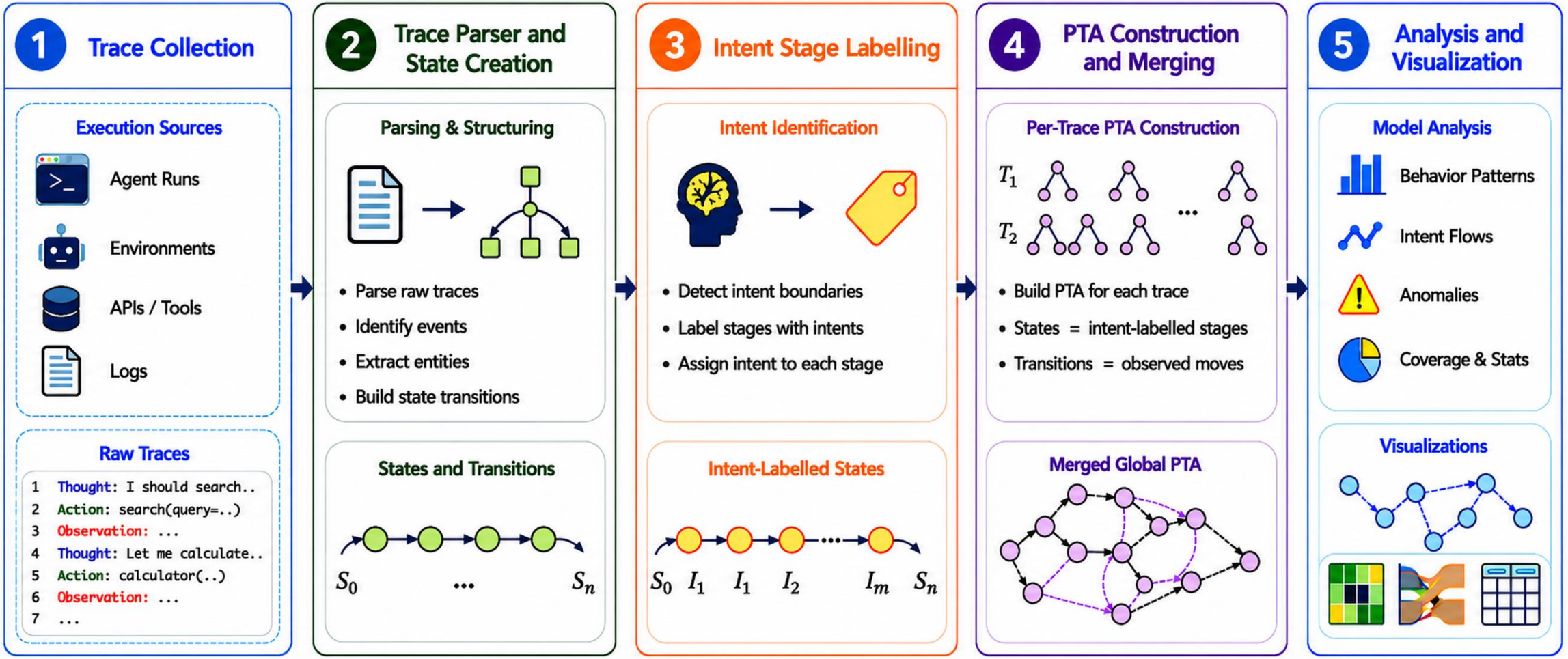
This happens on real tasks

Five passing runs on psf/requests-1724. SWE-bench marks all five as resolved.

	Passed	AgentLens score	What the run did
GPT-4o	✓	**88 Ideal**	11 steps: fixed it and added a test
Opus 4.5	✓	**67 Solid**	28 steps: looked around a lot
Opus 4.6	✓	**33 Lucky**	4 steps: never ran a test
GPT-5.3-Codex	✓	**32 Lucky**	6 steps: only part of the fix
GPT-4.1	✓	**22 Lucky**	34 steps: mostly grep and re-reads

Same result for SWE-bench. AgentLens scores them from **88** down to **22**.

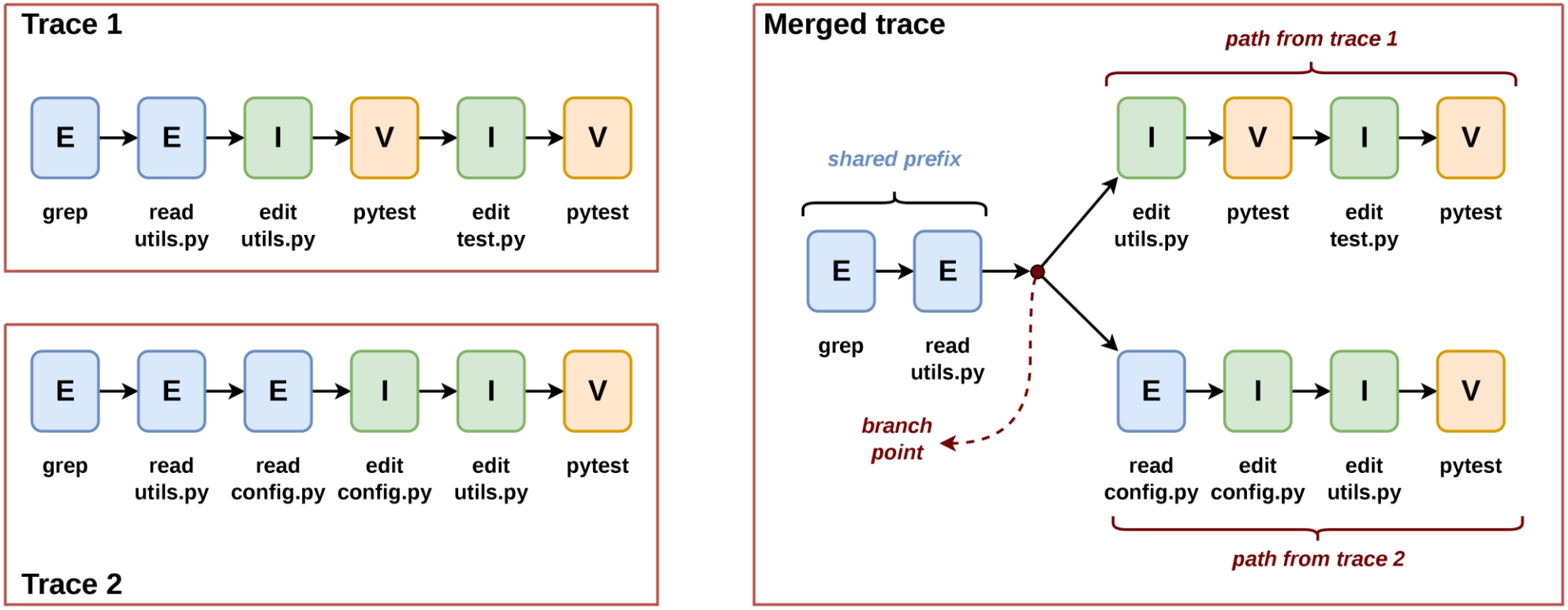
How AgentLens works



Logs become tagged states, passing runs merge into one reference graph per task, and new runs are scored against it.

Merge passing runs into one reference graph per task

A prefix tree acceptor (PTA): shared steps share a node, and different approaches branch off



5 passing runs per task. The run being scored is never part of its own graph.

PTA: Oncina & García, 1992

Score each run: six metrics in four categories

Completeness	Focus	Process	Outcome
Did it do the right work?	Did it avoid extra work?	Did it work in a sensible order?	Did the tests pass?

Coverage
share of the path's steps it did
w 0.25

Stages
explored, fixed and tested?
w 0.18

F1
needed steps vs. extra steps
w 0.10

Coherence
moves forward, few loops
w 0.25

Workflow
same order of work as the path
w 0.12

Outcome
100 if tests pass
w 0.10

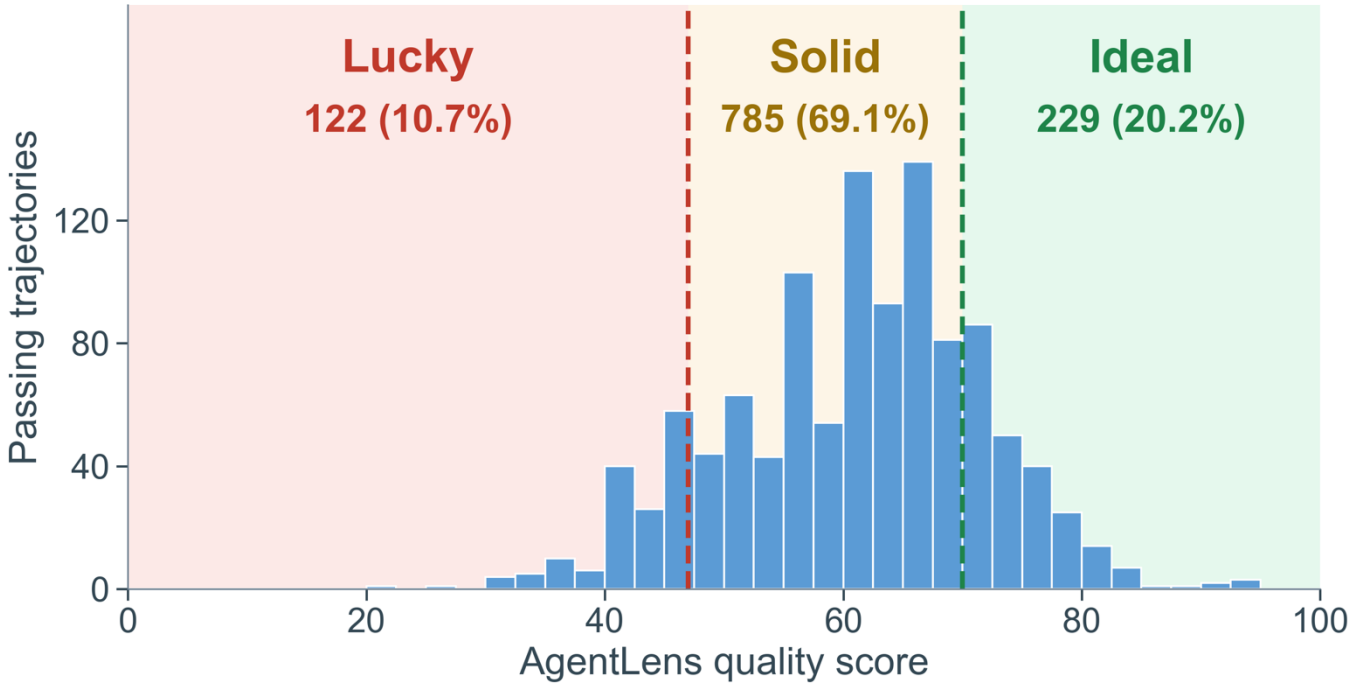
Score = 0.25 Coverage + 0.18 Stages + 0.10 F1 + 0.25 Coherence + 0.12 Workflow + 0.10 Outcome



Tier for passing runs, from the 0-100 score

About 1 in 10 passing runs is a Lucky Pass

1,815 OpenHands runs from 8 models on 47 SWE-bench Verified tasks



Quality scores for all 1,136 passing runs.

10.7%

of passing runs are Lucky Passes (122)

0.5-23.2%

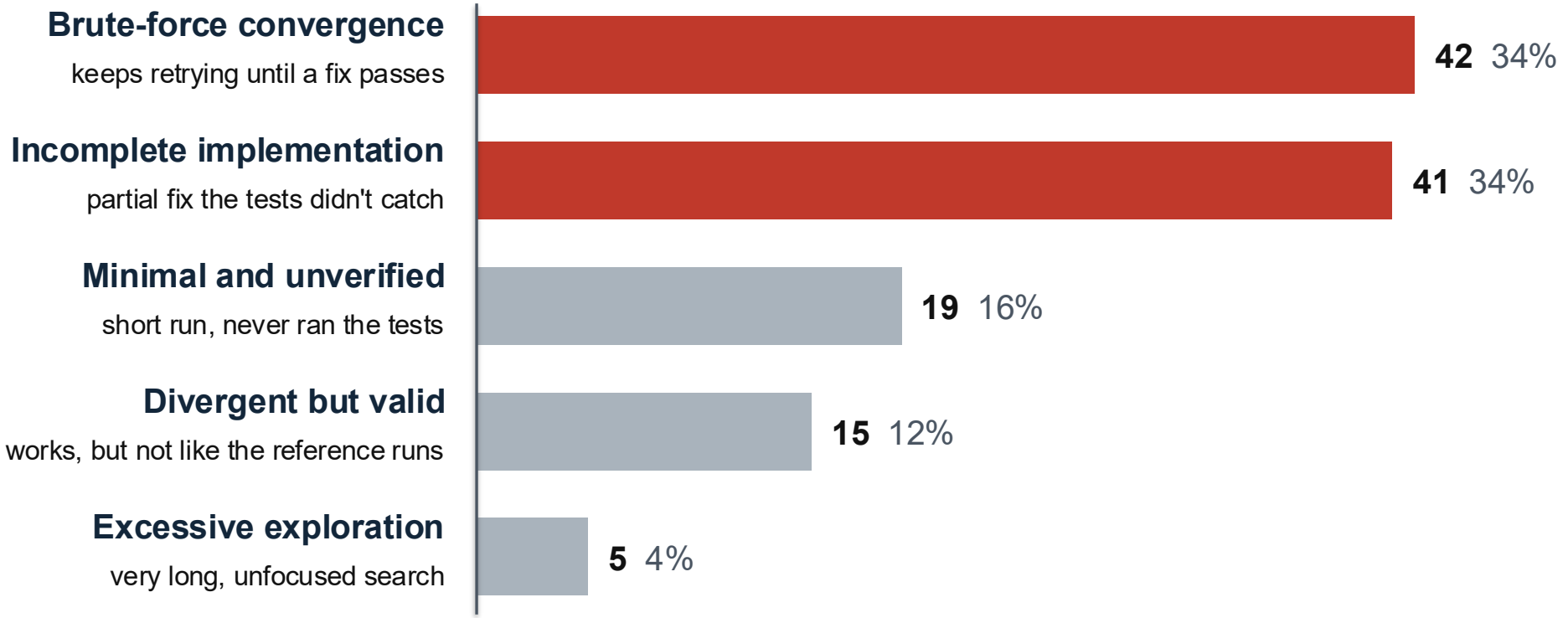
Lucky Pass rate, from the best model to the worst

All

8 models change rank when ranked by quality

Why Lucky Passes happen

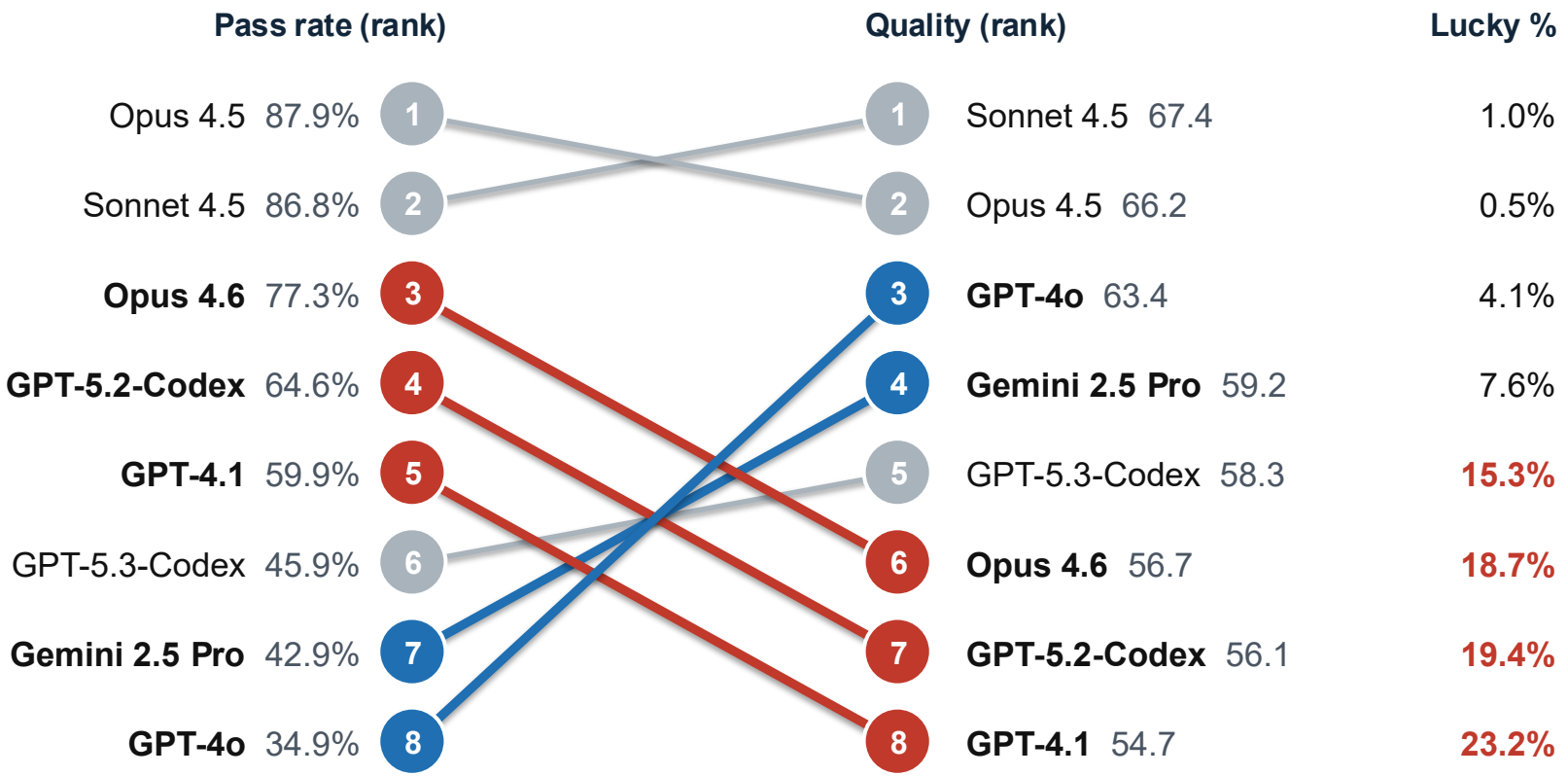
Five patterns cover all 122



The top two make up **68%** of Lucky Passes.

Ranking by quality reorders all eight models

Left: rank by pass rate. Right: rank by average quality of passing runs.



GPT-4o goes from 8th to 3rd. **Opus 4.6** drops from 3rd to 6th: almost 1 in 5 of its passes is Lucky.

Takeaways

1. Passing the tests is not the same as solving the task well. About **1 in 10** passing runs is a **Lucky Pass**.

2. AgentLens scores how the agent worked, using a reference graph built from passing runs.

3. Ranking by quality changes which model looks best. All 8 models move.



Code, dataset, SDK and leaderboard

agentlens-framework.github.io/agentlens

1,815 runs · 47 tasks · 8 models