RefactorBench: Evaluating Stateful Reasoning in Language Agents

Dhruv Gautam*, Spandan Garg, Jinu Jang, Neel Sundaresan, Roshanak Zilouchian Moghaddam

Berkeley UNIVERSITY OF CALIFORNIA

Microsoft

Definitions

Language Agent: POMDP interleaving action and feedback. The core action taker is the language model.

Stateful Reasoning: Reasoning dependent on the accumulation of actions and feedback. Partially observable and changes dynamically.

Background & Goals

Current Coding Agent Evaluations:
Solely focused on ***bug-fixing.***

There are many difficulties in evaluating code that doesn't have ***specified before and after*** behavior for solutions.

Our Goals:

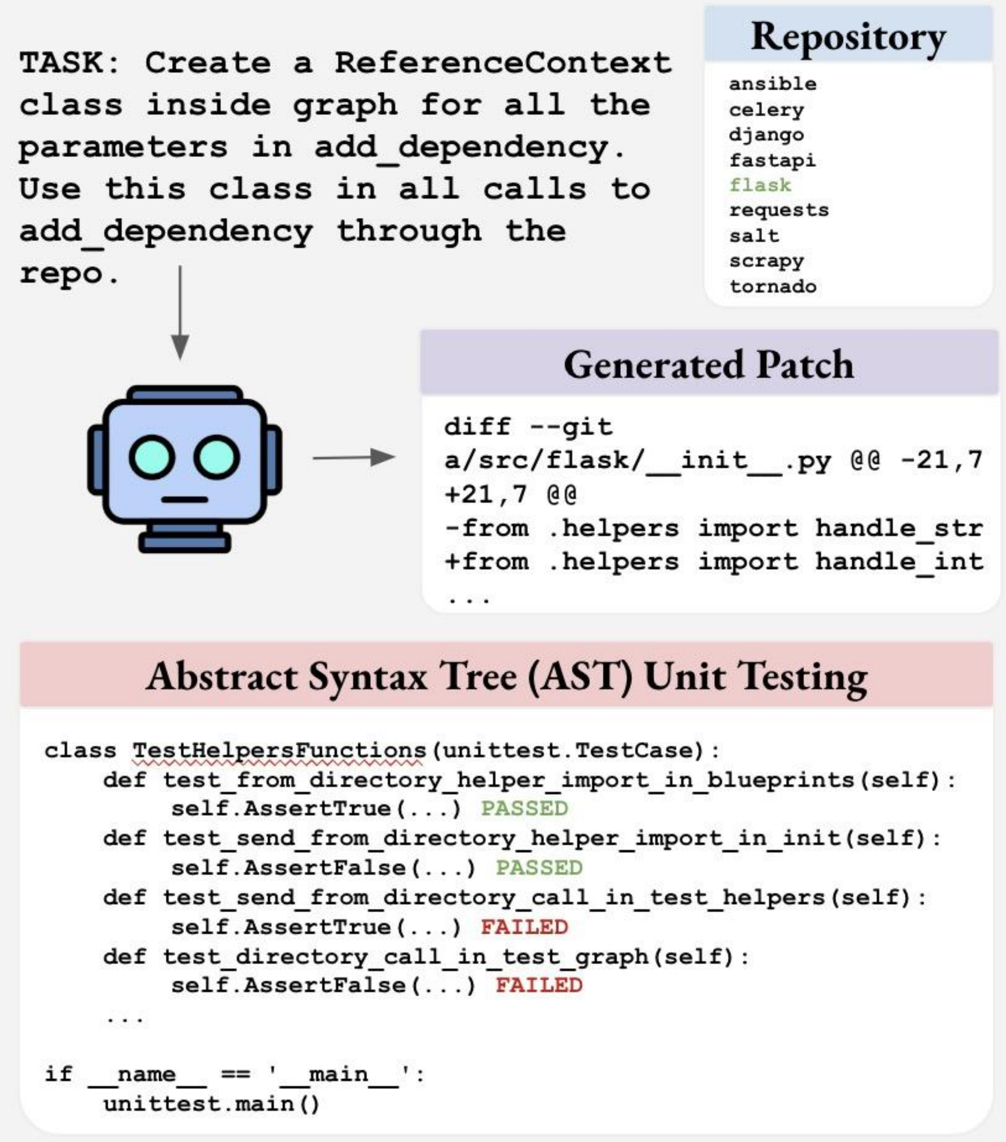
1. Bridge the gap in end-to-end code generation evaluations.
2. Evaluate on simple tasks that require stateful reasoning.
3. Better understand new failure modes for language agents.
4. Improve on those failure modes and hypothesize about fruitful future directions :)

Dataset and code available at:
https://github.com/microsoft/RefactorBench

RefactorBench Task & Test Construction Pipeline

Main Pipeline:

1. Localization and Filtering
2. Construction of Reference Solutions
3. Development of Testing Files
4. Generation of Relevant Task Instructions (lazy, base, descriptive)



RefactorBench tasks are distributed into 9 public Python repositories: Django, Ansible, Scrapy, Tornado, Requests, Flask, Salt, FastAPI, and Celery.

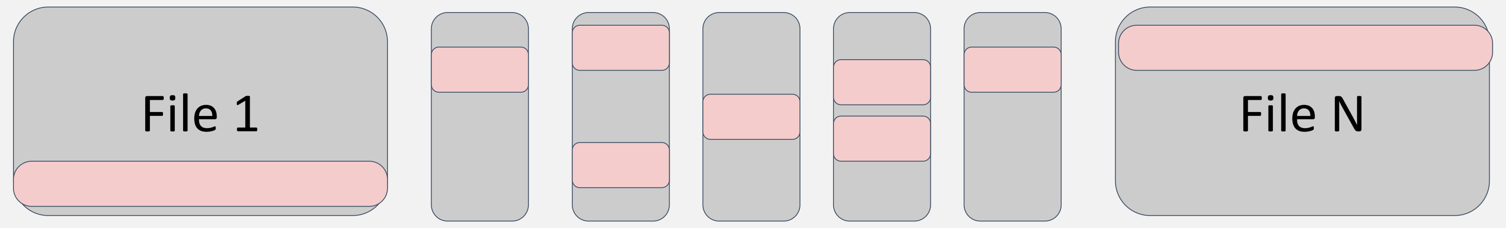
Reference solutions: Mean, 4.3 files | Max, 31 files
AST Tests: Mean, 6.5 Tests | Max, 27 tests

Results & New Failure Modes

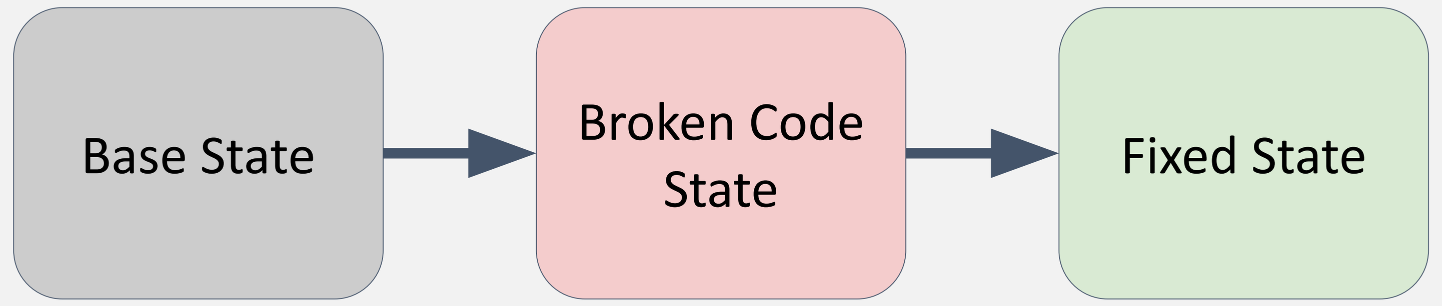
Prompt Adjusted *SWE-Agent* Performance: 12%, 22%, 27%

Failure Modes

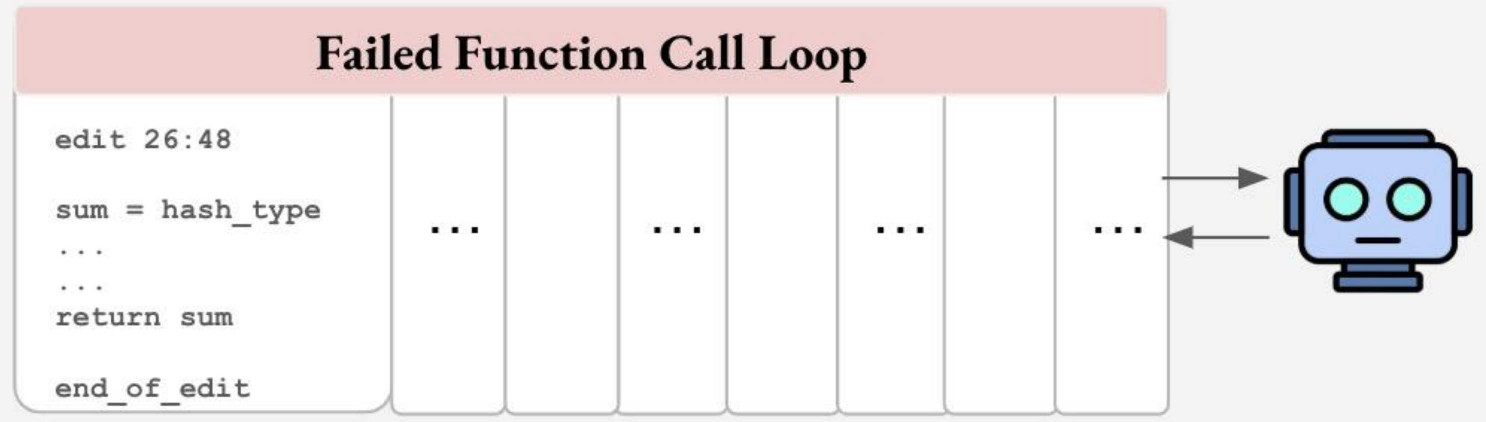
1. Multi-File Localization



2. Erroneous States



3. In-Context Reward Hacking



4. Agents haven't improved much, except for reasoning models.

New SWE-Agent 1.0: 31%
Custom Agent With Reasoning Oversight: 76%

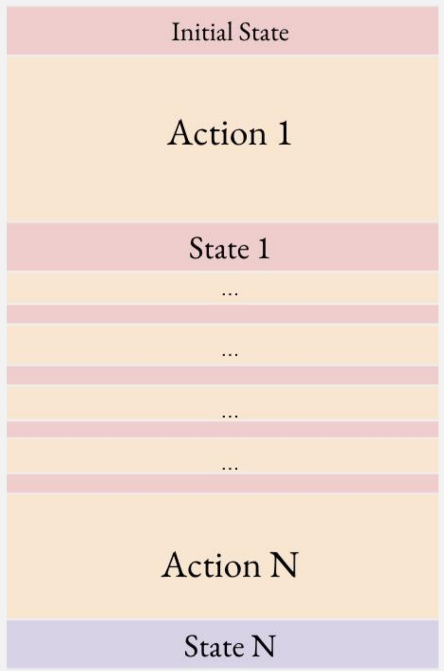
State-Aware Interfaces & Policies

State-Aware Agent: POMDP interleaving action, feedback, and state.

State Update Policy: Generalize the state update environment-wide, allows for communication between concurrent agents making changes.

43.9% increase in performance across all instruction sets!

State Interleaved Trajectory



Takeaways

1. Generalist evaluations are difficult to get right.
2. Language Agents have innate weaknesses from the model side (entity tracking), and the scaffold side (interface design).
3. State aware interfaces help scaffolded agents reason better in long, stateful tasks.

Future Directions

1. Leverage AST setup for create more "verifiable" coding tasks for reasoning models.
2. Train reasoning models to recover state as reward to create better long-horizon agents.
3. Create new evaluations that truly find the failure modes of coding agents in the wild.