Computational Reproducibility (CR) is poor: one study suggests only 54% of published papers in CS have code that can be built (Collberg et al.).

CR is hard: no standardized approach to build and run code, documentation often poor, difficult to install dependencies, code contains bugs, unclear what commands to run to reproduce given experiments, hard to extract information from results, etc.

Studying CR is labor intensive: it requires manual effort to reproduce and re-run code at scale.

Building agents would allow us to study CR at greater scale, make it easier to reproduce work, and improve reproducibility norms across science.

Construct Validity: benchmark solves a real-world problem, and agents can be immediately deployed to study CR.

We propose (1) building an agent that evaluates papers’ CR, (2) creating a benchmark to evaluate agent ability, and (3) using agent to assess state of CR.

**Baseline Agent**
- Create a baseline-agent to test benchmark.
- Allows gpt-4-turbo and to execute arbitrary Bash commands in environment.
- Prompt agent with examples of running Bash commands and makes it check its work before submitting.

**Benchmark Task Setup**
- Task Prompt
- Project Repository
- Hint
- Environment
- Evaluation
  - Report JSON
  - Code Output

**Preliminary Results**
- We run agent on 5 tasks, each with varying degrees of difficulty (i.e. hints provided).
- Even on most simple task, where agent is provided exact Docker command to run to replicate all results, only succeeds 40% of time (due to information extraction failures).

**Future Plans**
- Create a benchmark of ~ 100 tasks.
- Improve baseline agent to make it multimodal for extracting results from images.
- Publish benchmark on platform to make it easy for others to contribute own agents.
- Apply the agent to analyze how state of reproducibility degrades over time.