# InterCode: Standardizing & Benchmarking Interactive Coding with Execution Feedback



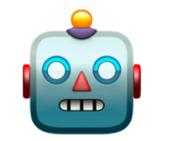




John Yang, Akshara Prabhakar, Karthik Narasimhan, Shunyu Yao

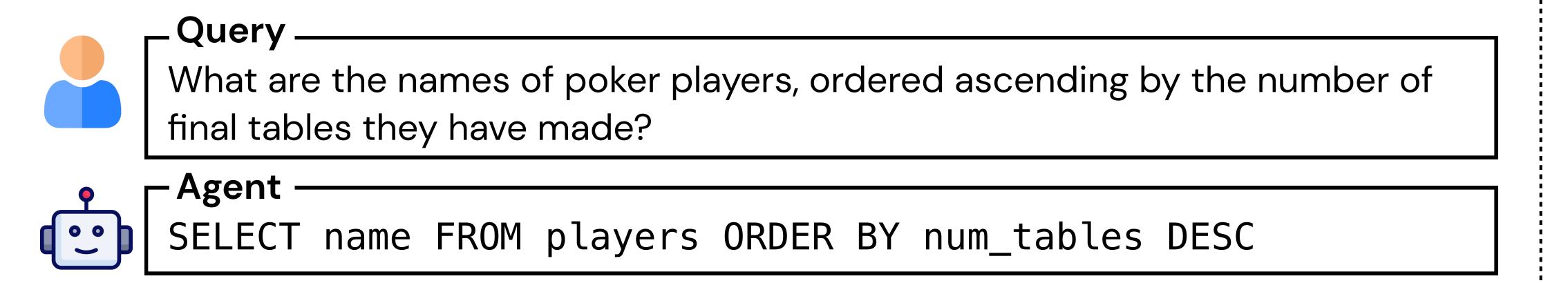
### Seq2Seq Coding

VL Query →



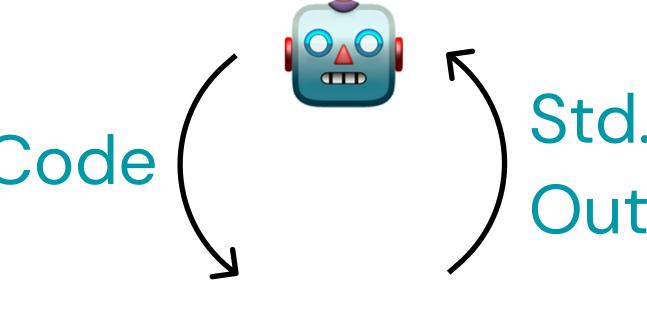


No execution or grounding



### **L** Our Work

### Interactive Coding!





**Execution Environment** 

#### Motivation

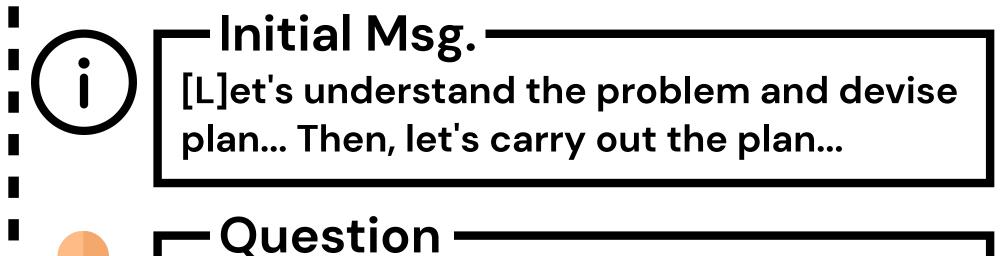
- Interaction & feedback
   necessary for hard tasks
- Standardized
   environment enables
   consistent benchmarking
   of coding agents

### **V** Features

- Abstracted: Handles interaction, execution logic under-the-hood
- Lightweight: Define task env. in <100 lines of code
- Safe: Virtual containers =
   no accidental of

Evaluate Language Models as Agents that write *code* as *actions* to interact with a software system

### Plan & Solve



What is the country code and first name...

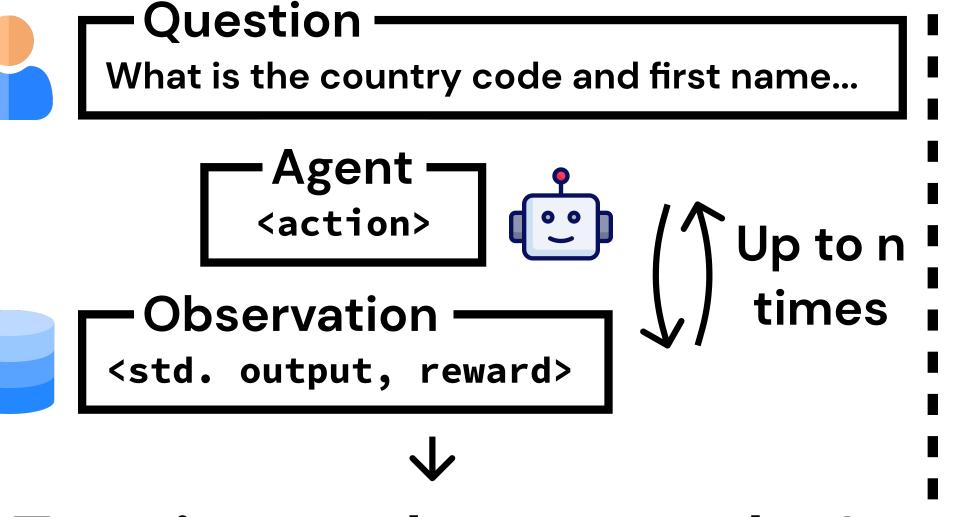
Agent <plan>

Execute Msg.

You will now execute your own plan.

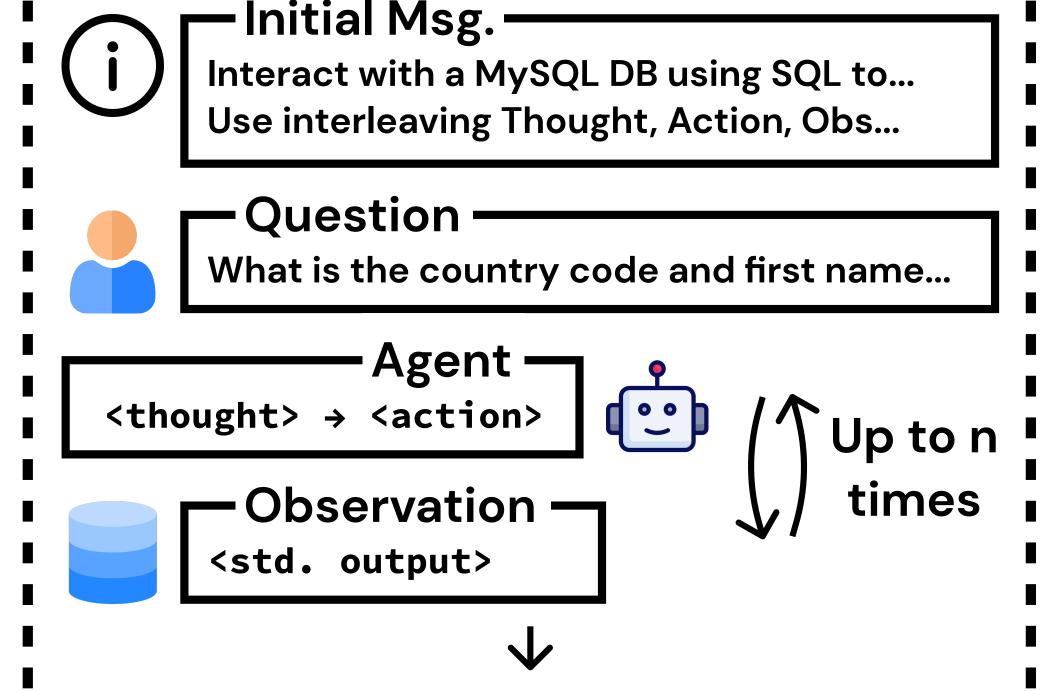
Terminate when plan is completed or n turns exceeded

## "Try Again" Initial Msg. You are a SQL code generator helping... Your task is to interact with a MySQL DB...



Terminate when reward = 1 or exceeds n turns

### ReAct



Terminate when thought chain the finishes or exceeds n turns

### 4 New Interactive Coding Tasks

Action	Setting	Datasets
Bash	Ubuntu Shell	NL2Bash (200)
SQL	MySQL DB	Spider, BIRD (10K)
Python	Interpreter	MBPP, APPS (10K)
CTF	Ubuntu Shell	IC-CTF (100)

### Make Your Own Task is Easy

Provide the following to InterCodeEnv:

- Dockerfile, to define your task setting.
- Dataset of task instances to evaluate.
- Reward Func, to score agent trajectories and determine whether a task is completed.

### **Key Results**

GPT 3.5	Single	Try Again	ReAct	Plan/Solve
Bash	34.5	46.5	20.5	28.0
SQL	10.5	47.3	58.7	49.1

- Multi-turn >> single-turn
- Diff. methods better on diff. tasks
- Large room for improvement

### Check it Out!

intercode-benchmark.github.io

