# Demo2Code:

## From Summarizing Demonstrations to Synthesizing Code via Extended Chain-of-Thought

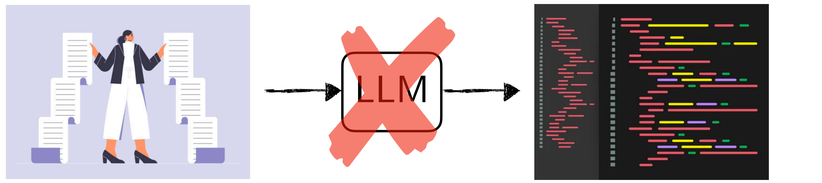
*Huaxiaoyue Wang, Gonzalo Gonzalez-Pumariega, Yash Sharma, Sanjiban Choudhury*
Cornell University

PORTAL

NEURAL INFORMATION PROCESSING SYSTEMS

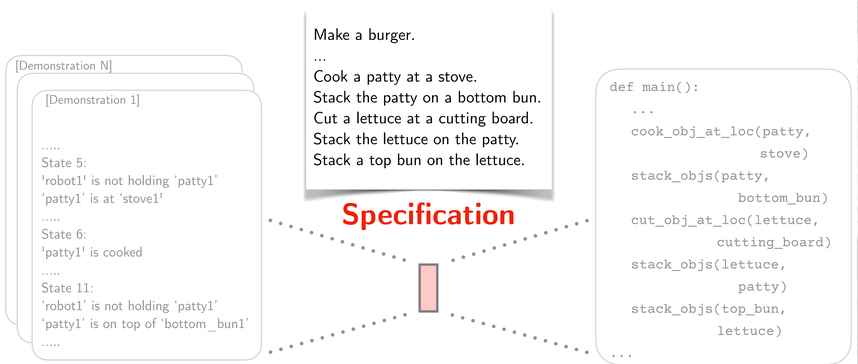
## Generate Robot Code From Demonstration


## Challenges



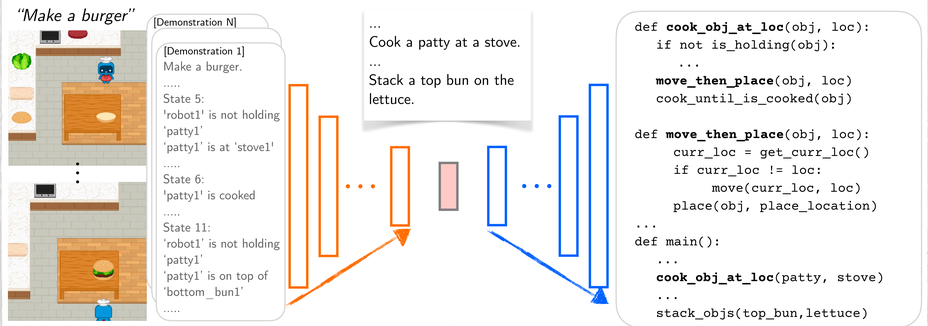

Directly generating code from demonstrations is intractable!

## Key Insight




Both demonstrations and code are connected by a latent **task specification**.

## Our Approach: Demo2Code




### Stage 1

**Recursively summarize**

demo —> specification

Recursively summarizes each demo, then concatenates all summaries to generate a task specification

### Stage 2

**Recursively expand**

specification —> task code

From the task specification, generates high-level code, then recursively defines helper functions

## Tabletop Manipulation

when the language **lacks specificity**

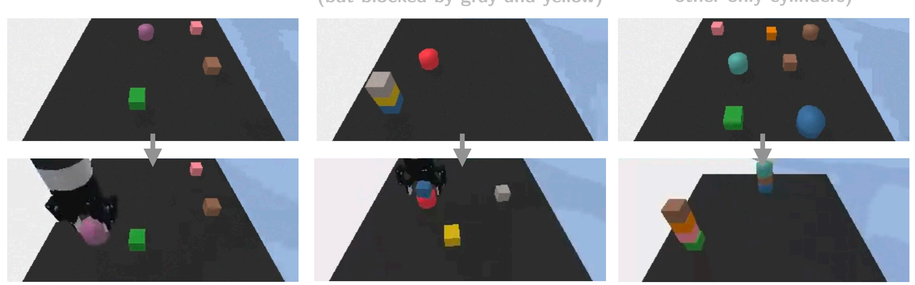
"Place the purple cylinder to (the left) of the green block."

when the world has **hidden constraints**

"Place the blue block on top of red cylinder." (but blocked by gray and yellow)

when the user has **personal preferences**

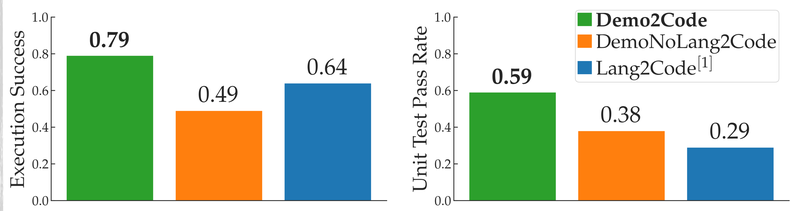
"Stack all objects into two stacks." (one stack has only blocks, other only cylinders)



## Novel Kitchen Game: *Robotouille*

- **Open-source game!**
- Procedurally generated environments
- Easy to customize (new assets, tasks, actions, etc.)


Demo2Code generalizes to unseen, more complex tasks




[1] Jacky Liang, et al. Code as policies: Language model programs for embodied control. arXiv preprint arXiv:2209.07753, 2022.

## Real World Dataset: EPIC-Kitchens


User 22:
Prefers soaping all first then rinsing all objects

```
objs = get_all_objs()

for obj in objs:
    soap(obj)
    place(obj,
        "sink_2")
turn_on("tap_1")
for obj in objs:
    rinse(obj)
    place(obj,
        "dishrack_1")
turn_off("tap_1")
```

User 30:
Prefers soaping then rinsing each object

```
objs = get_all_objs()

for obj in objs:
    pick_up(obj)
    go_to("sink_1")
    turn_on("tap_1")
    soap(obj)
    rinse(obj)
    turn_off("tap_1")
    place(obj,
        "counter_1")
```