ICLR

# **Robotouille:** An Asynchronous Planning Benchmark for LLM Agents

Gonzalo Gonzalez-Pumariega, Leong Su Yean, Neha Sunkara, Sanjiban Choudhury

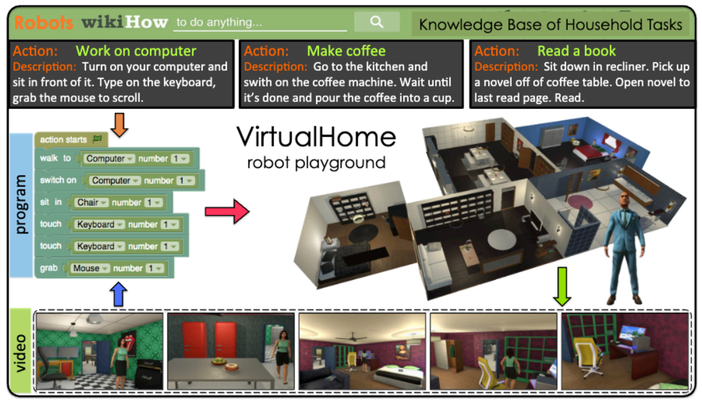



PORTAL

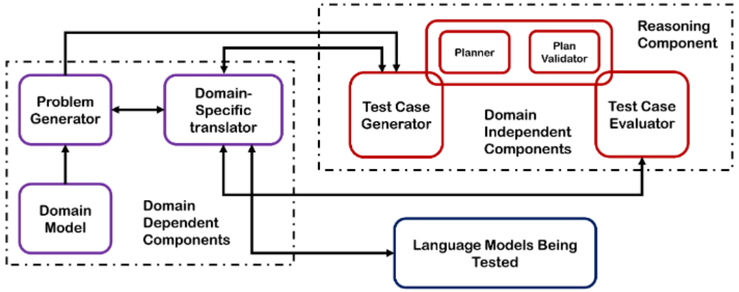
Cornell Bowers CIS
Computer Science

# LLMs x Planning Benchmarks


ALFWorld
[Shridhar, 2021]



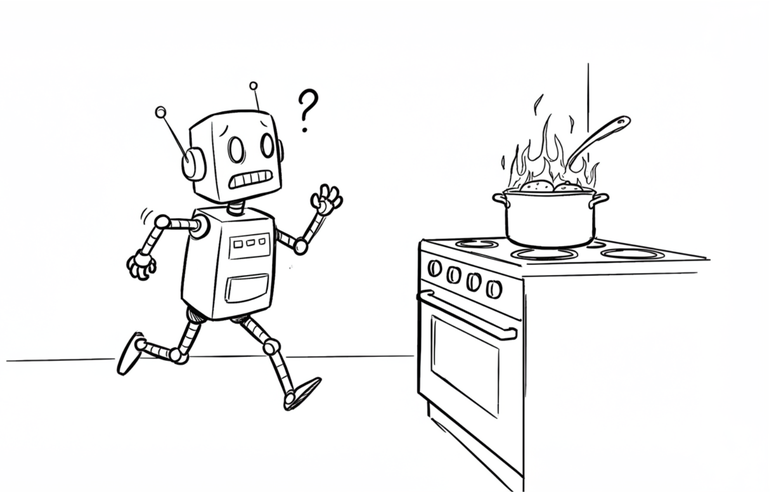

VirtualHome
[Puig, 2018]




PlanBench
[Valmeekam, 2023]

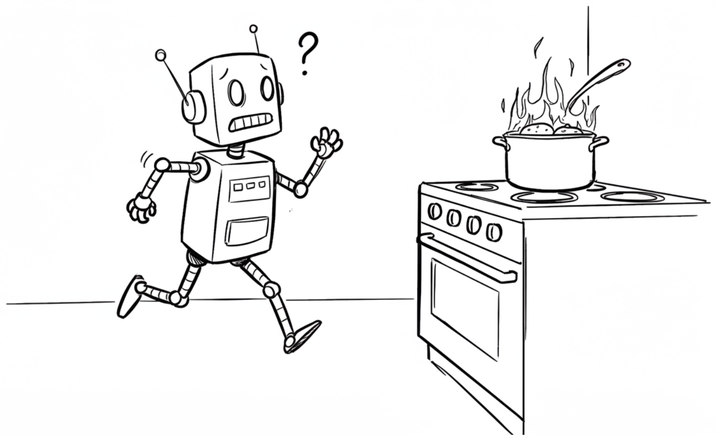
# What are our desiderata for LLM agents?

# What are our desiderata for LLM agents?

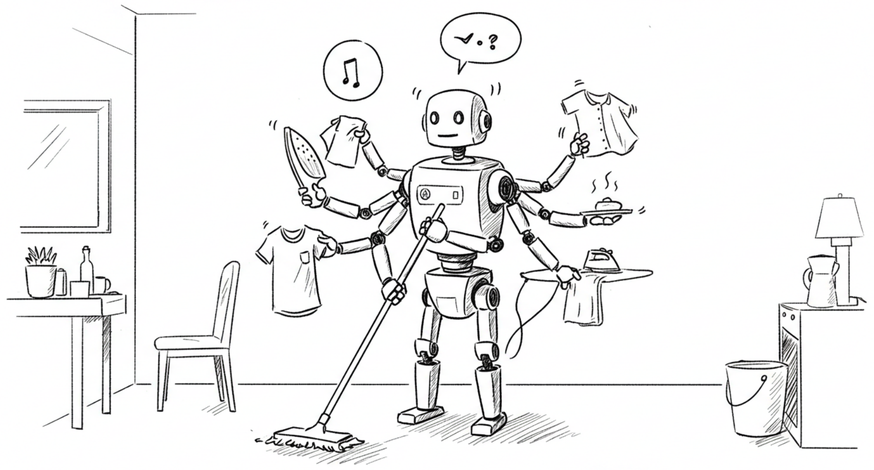


Handle time delays

# What are our desiderata for LLM agents?

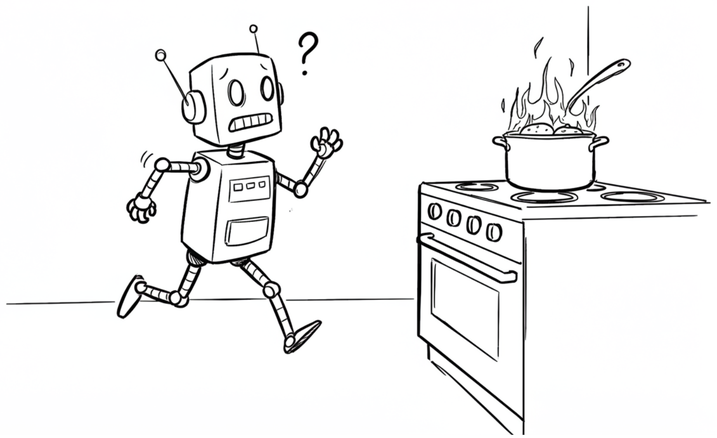


Handle time delays

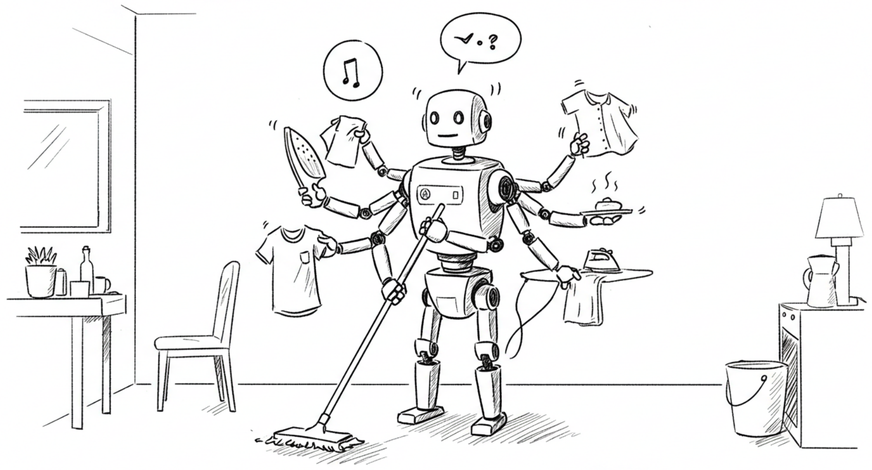


Solve long horizon diverse tasks

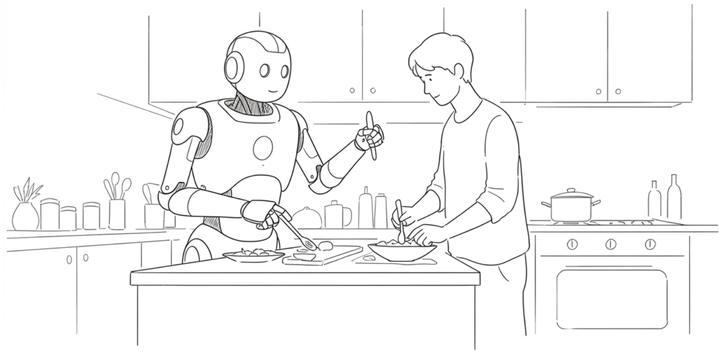
# What are our desiderata for LLM agents?



Handle time delays

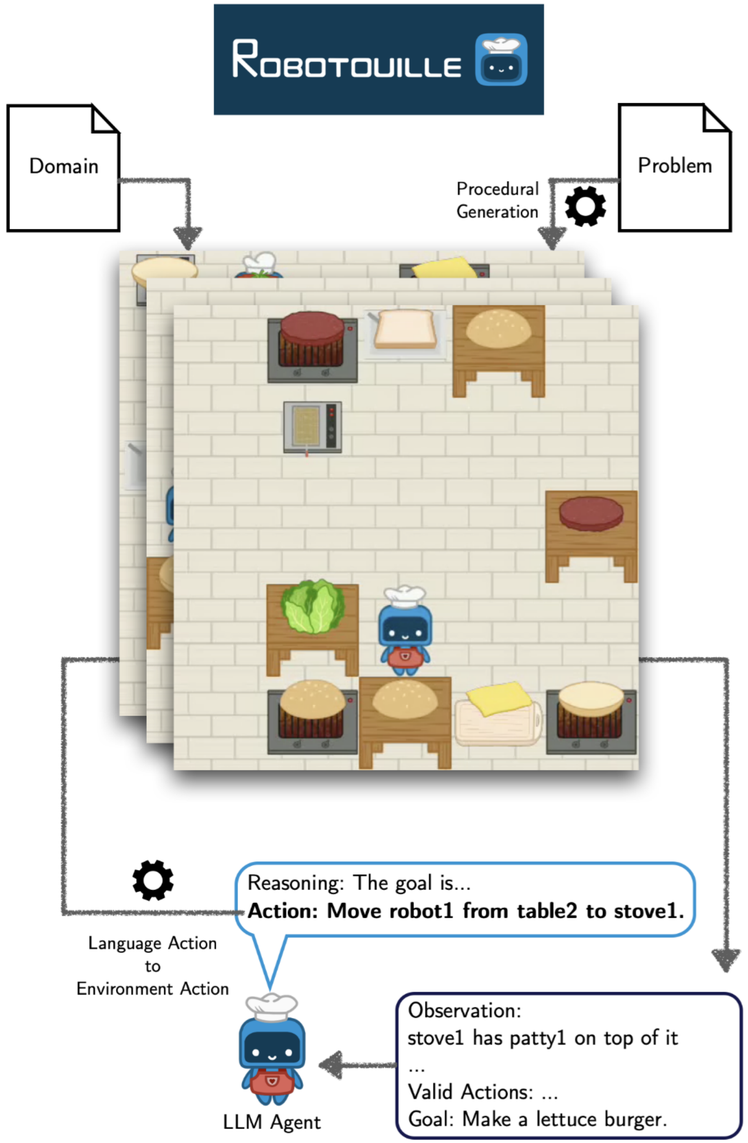


Solve long horizon diverse tasks

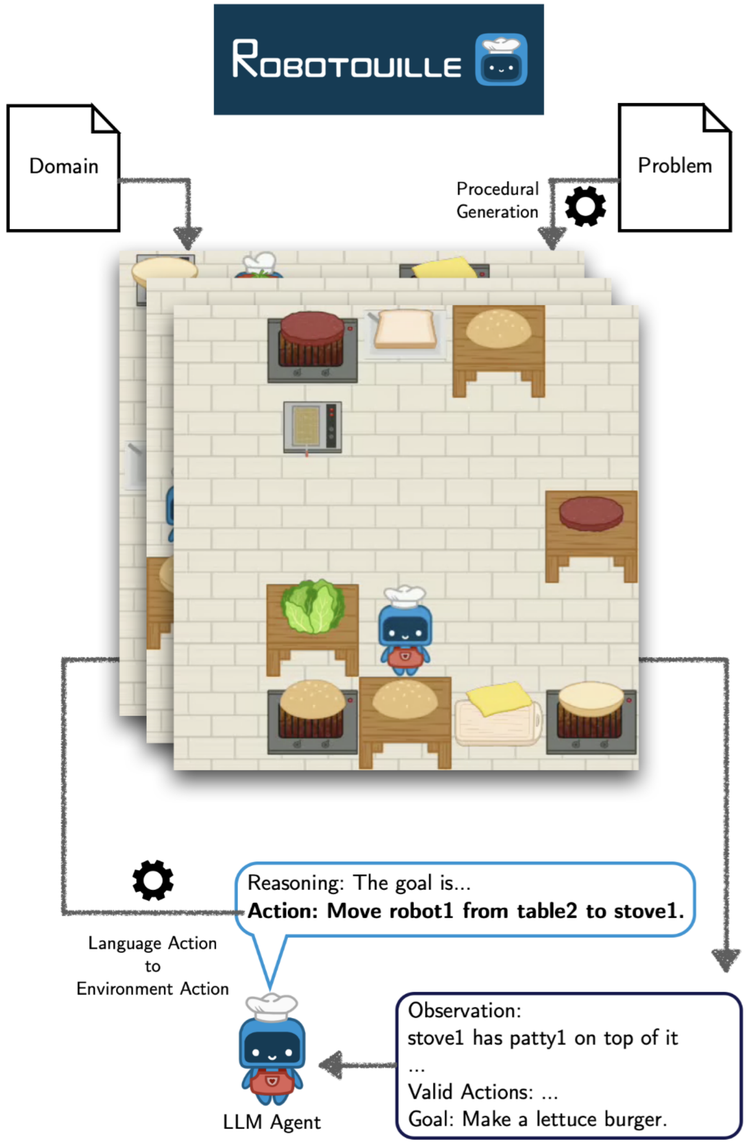


Collaborate with others

# Introducing Robotouille!

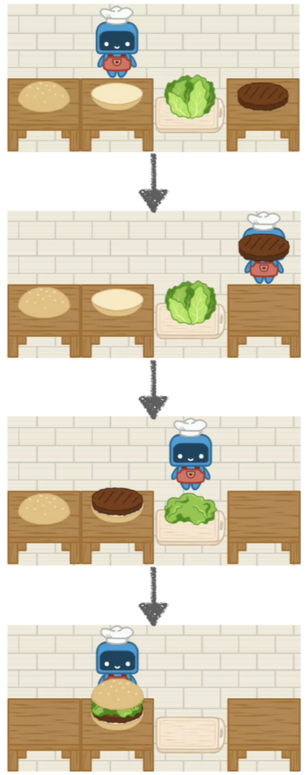


# Introducing Robotouille!

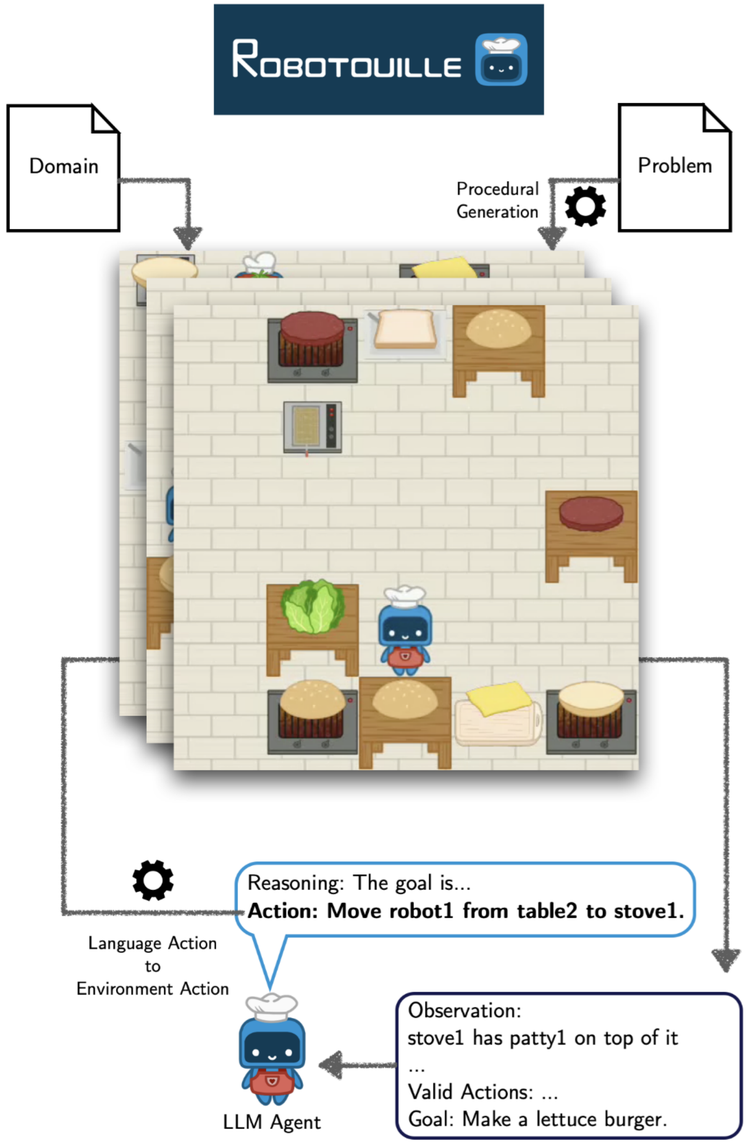



## Synchronous Benchmark

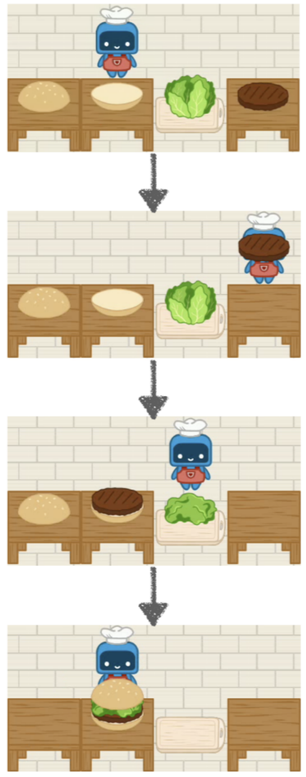
Synchronous Planning



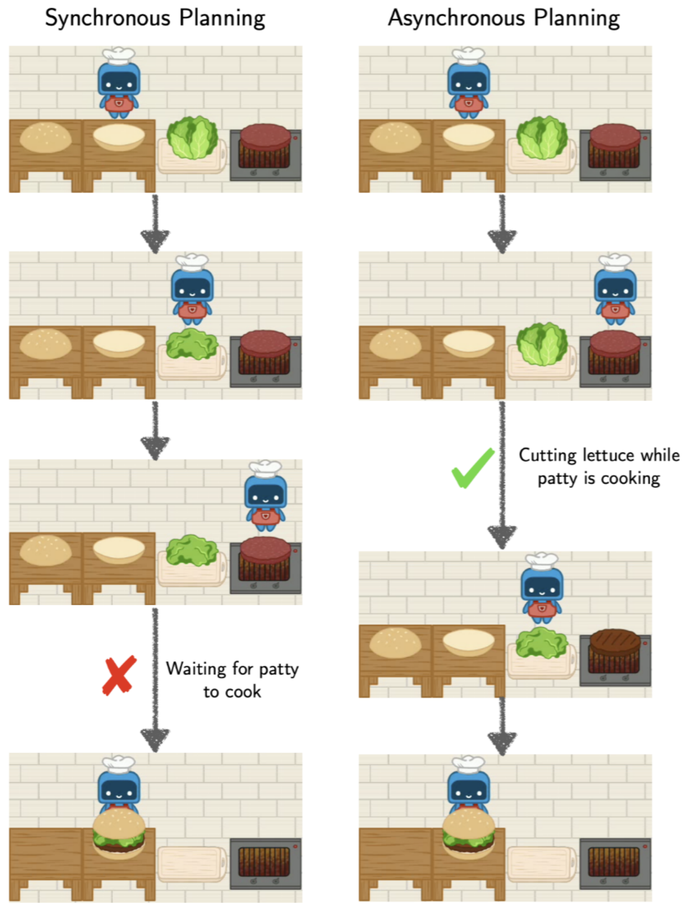
# Introducing Robotouille!



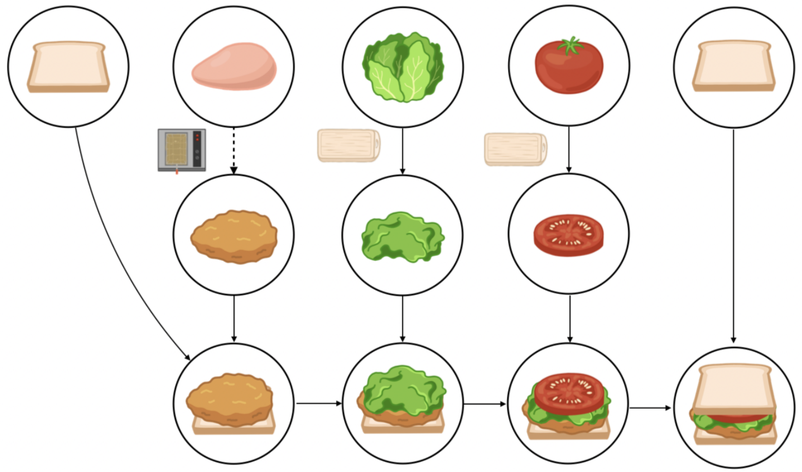

## Synchronous Benchmark




## Asynchronous Benchmark

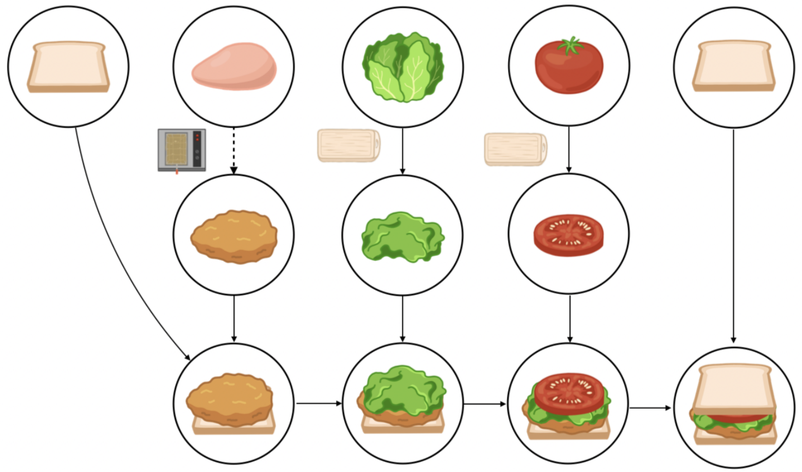


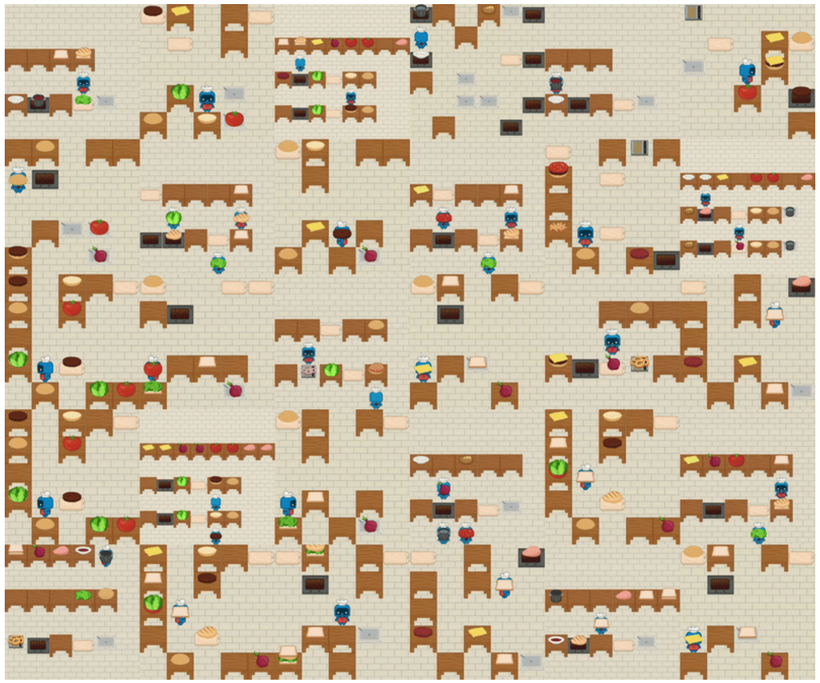
# ✅ Robotouille can evaluate LLM agents on all desiderata



| LLM Agents Should | Handle time delays |
| --- | --- |
| Robotouille | Models subtasks with time delays |

# ✅ Robotouille can evaluate LLM agents on all desiderata





**LLM Agents Should**

Handle time delays

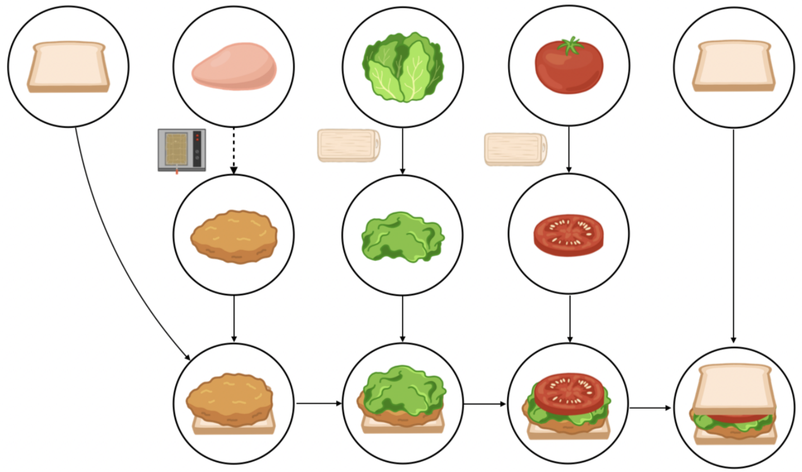
Solve long horizon diverse tasks

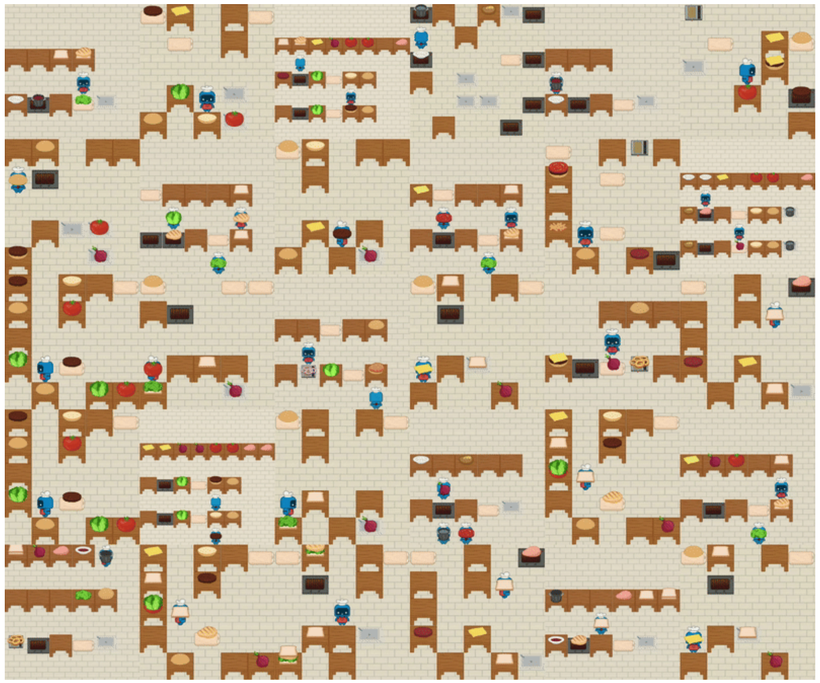
**Robotouille**

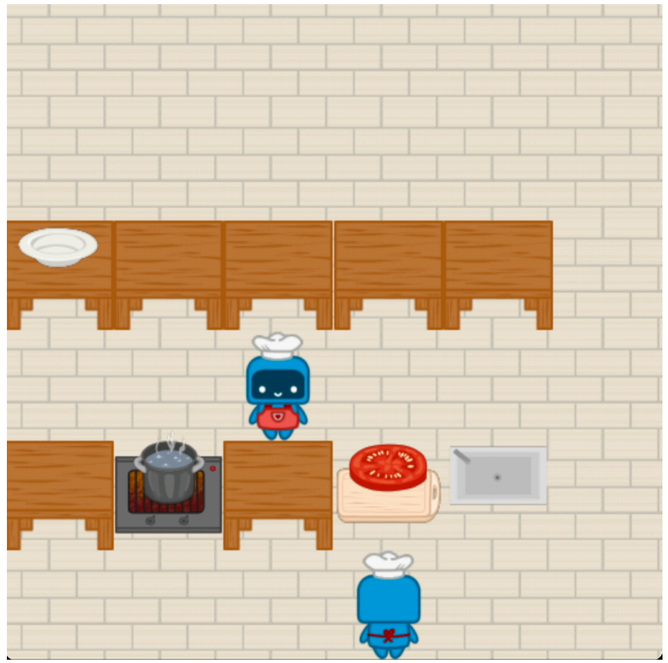
Models subtasks with time delays

30 unique tasks and procedural generation

# ✅ Robotouille can evaluate LLM agents on all desiderata







**LLM Agents Should**

Handle time delays

Solve long horizon diverse tasks

Collaborate with others

**Robotouille**

Models subtasks with time delays

30 unique tasks and procedural generation

Supports multiple agents locally and over network

# Experiments

We evaluate the following baselines on synchronous and asynchronous settings

1. **Input/Output (I/O):** Act
2. **I/O CoT**: Think, Act
3. **ReAct**: Sense, Think, Act

**Metrics**

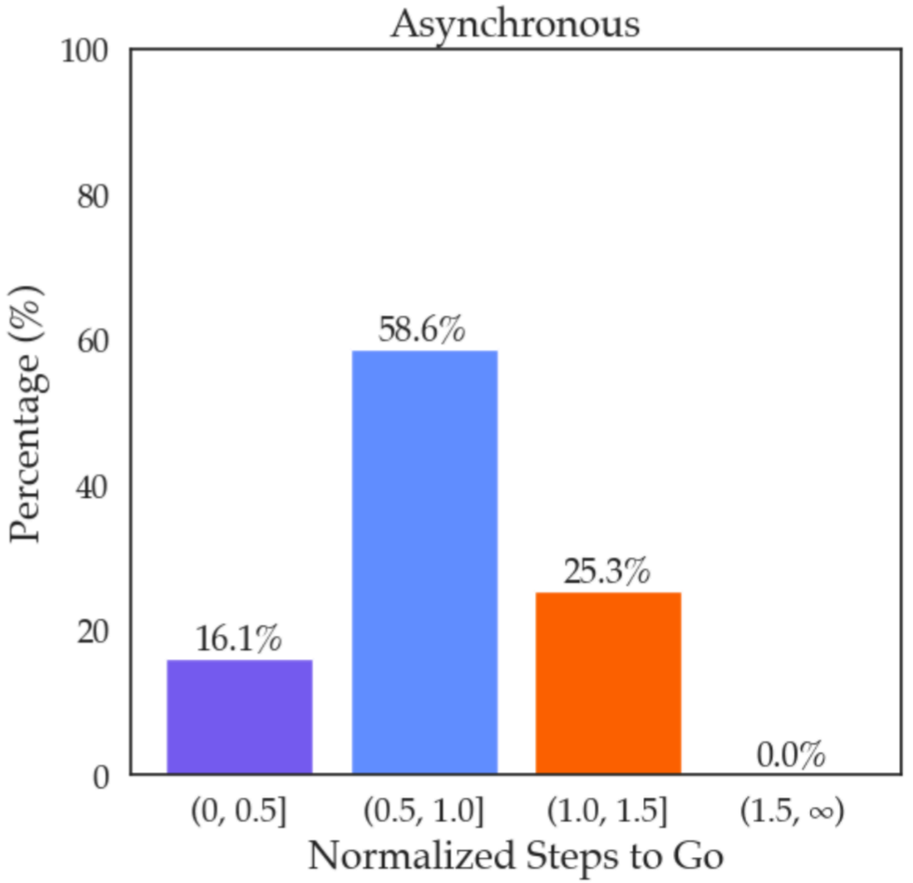
- Success rate: Average # goals reached in 10 procedurally generated levels
- Steps to go: Optimal number of steps to reach the goal from failure state

# Takeaway 1: Closed-loop agents are superior

| | Synchronous (%) | | | Asynchronous (%) | | |
|---|---|---|---|---|---|---|
| | **I/O** | **I/O CoT** | **ReAct** | **I/O** | **I/O CoT** | **ReAct** |
| `gpt4-o` | 4.00 | 14.0 | **47.0** | 1.00 | 1.00 | **11.0** |
| `gpt-4o-mini` | 4.00 | 10.0 | 11.0 | 0.00 | 1.00 | 0.00 |
| `gemini-1.5-flash` | 0.00 | 13.0 | 0.00 | 0.00 | 0.00 | 0.00 |
| `claude-3-haiku` | 1.00 | 2.00 | 2.00 | 0.00 | 0.00 | 0.00 |

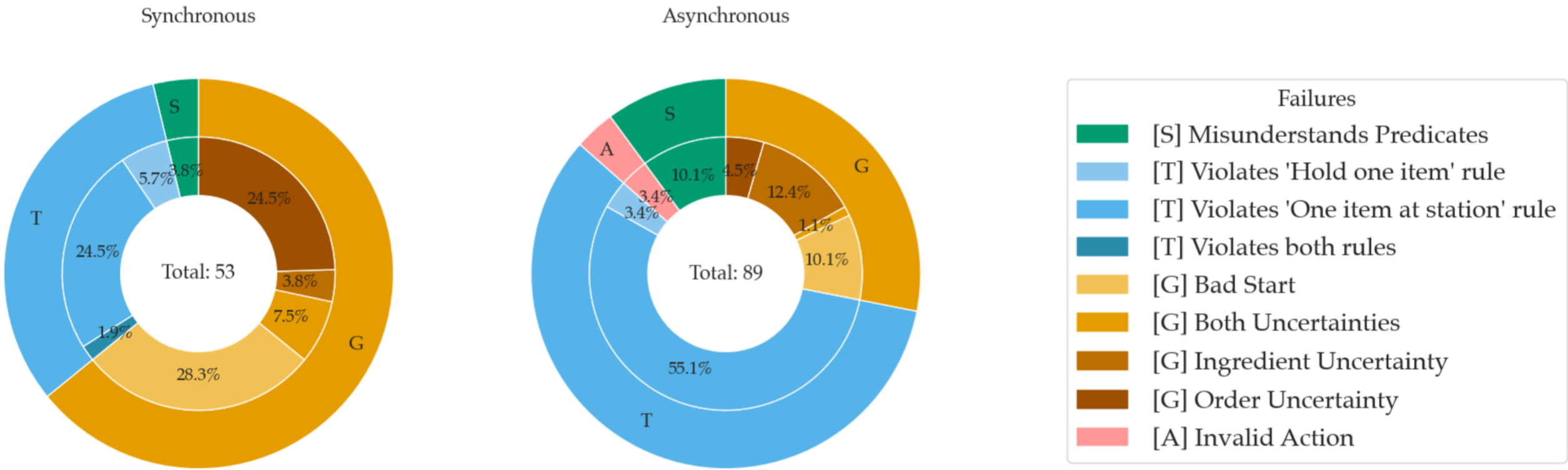
Agents should make use of feedback from interactive environments to recover from mistakes and reach the goal

# Takeaway 2: Asynchronous failures make little progress towards the goal




A majority of asynchronous failures fail more than halfway from the goal.

# Takeaway 3: Synchronous and asynchronous failures are closely related




Agents struggle at recovering from violating environment constraints and self-correcting from small mistakes towards the goal

ICLR

# **Robotouille:** An Asynchronous Planning Benchmark for LLM Agents

Gonzalo Gonzalez-Pumariega, Leong Su Yean, Neha Sunkara, Sanjiban Choudhury

https://github.com/portal-cornell/robotouille

PORTAL

Cornell Bowers CIS
Computer Science