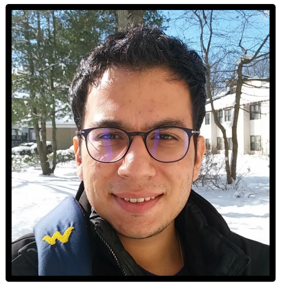
AppWorld

A Controllable World of Apps and People for Benchmarking Interactive Coding Agents

🏆 ACL’24 Best Resource Paper


Harsh Trivedi

Stony Brook University

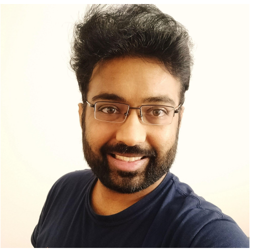
Tushar Khot

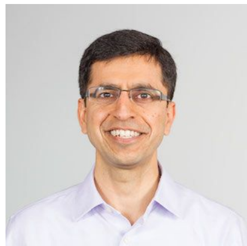
Mareike Hartmann

Ruskin Manku

Vinty Dong

Edward Li


Shashank Gupta


Ashish Sabharwal

Niranjan Balasubramanian

Stony Brook University

Ai2

SAARLAND UNIVERSITY

Agents for Day-to-Day Tasks

Return my last Amazon ordered shirt & buy it in one size larger.

I owe money to friends on Splitwise. Pay them on Venmo.

Agents for Day-to-Day Tasks

Return my last Amazon ordered shirt & buy it in one size larger.

I owe money to friends on Splitwise. Pay them on Venmo.

Hey AI! Here are my app accounts. Do this task for me:

Can AI agents to do such day-to-day tasks for us?

Day-to-Day Tasks Are Actually Quite Complex

Play my **Spotify** playlist with enough songs for the workout today.
My workout plan is in **SimpleNote**.

Joe

Coding Agent Solution using APIs

Day-to-Day Tasks Are Actually Quite Complex

Play my **Spotify** playlist with enough songs for the workout today.
My workout plan is in **SimpleNote**.

Joe

Coding Agent Solution using APIs

1

Code has to be written ***interactively***

Day-to-Day Tasks Are Actually Quite Complex

Play my **Spotify** playlist with enough songs for the workout today.
My workout plan is in **SimpleNote**.

Joe

Coding Agent Solution using APIs

Let me find Joe's workout plan.

```
token = simplenote.login(...)["token"]
note = simplenote.search_notes("workout", token)[0]
print(note) # have to read it to decide the next step
```

... Monday: ... 25 mins ... Tuesday ... 45 mins ...

1

Code has to be written ***interactively***

Day-to-Day Tasks Are Actually Quite Complex

Play my Spotify playlist with enough songs for the workout today.
My workout plan is in SimpleNote.

Joe

Coding Agent Solution using APIs

Let me find Joe's workout plan.

```
token = simplenote.login(…)[“token”]
note = simplenote.search_notes(“workout”, token)[0]
print(note) # have to read it to decide the next step
```

```
... Monday: ... 25 mins ... Tuesday ... 45 mins ...
```

Plan is day-wise, let’s see what day is today.

```
print(datetime.now().strftime(“%A”))
```

```
Tuesday
```

1

Code has to be written ***interactively***

Day-to-Day Tasks Are Actually Quite Complex

Play my **Spotify** playlist with enough songs for the workout today.
My workout plan is in **SimpleNote**.

Joe

Coding Agent Solution using APIs

Let me find Joe's workout plan.

```
token = simplenote.login(…)[“token”]
note = simplenote.search_notes(“workout”, token)[0]
                                              step
```

Plan is day-wise, let's see what day is today.

```
print(datetime.now().strftime(“%A”))
```

Tuesday

2

Code has to be ***rich*** with various programming constructs

Today is Tuesday, so it’s 45 mins. Now, let me play Joe’s playlist with enough songs for this duration.

```
workout_mins = 45
token = spotify.login(...)["token"]
playlists = spotify.playlist_library(token)
for playlist in playlists:
    duration = 0
    for id in playlist["song_ids"]:
        duration += spotify.song(id)["duration"]
    if total_duration >= workout_mins:
        spotify.play(playlist["id"], token)
        break
```

Day-to-Day Tasks Are Actually Quite Complex

Play my **Spotify** playlist with enough songs for the workout today.
My workout plan is in **SimpleNote**.

Joe

Key Question

How can we **develop & benchmark** such coding **agents for** complex day-to-day **tasks** in a **rigorous** & **reproducible** manner?

```
print(datetime.now().strftime(“%A”))
```

Tuesday

```
for id in playlist["song_ids"]:
    duration += spotify.song(id)["duration"]
if total_duration >= workout_mins:
    spotify.play(playlist["id"], token)
    break
```

Our Contribution

AppWorld

Engine

A **rich & reproducible execution environment** of many API-operable apps

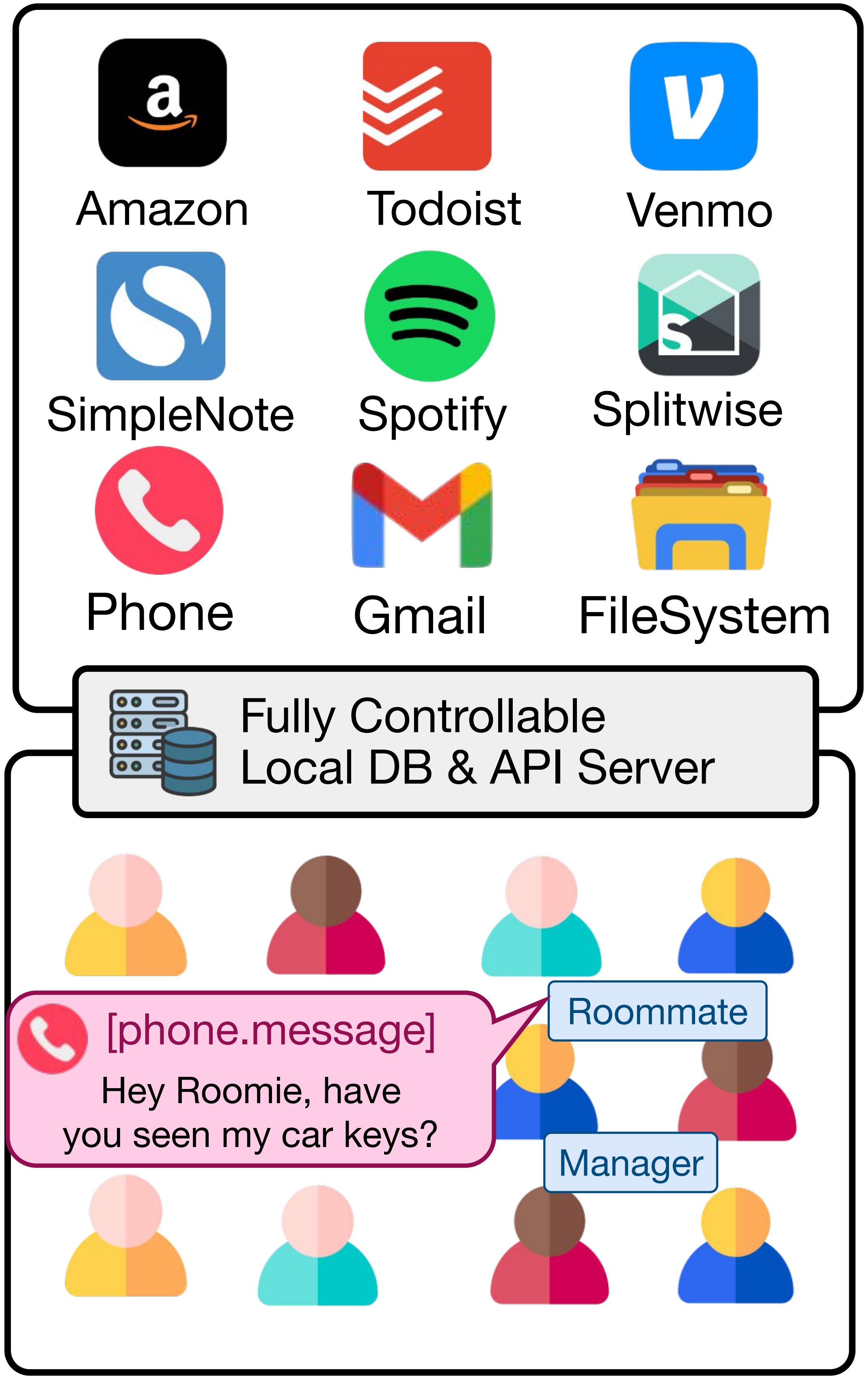
Benchmark

A set of **complex tasks** needing API calls with **rich & interactive coding**

Evaluation

A **robust & programmatic** evaluation framework for checking goal completion

Engine | Benchmark | Evaluation



[phone.message]

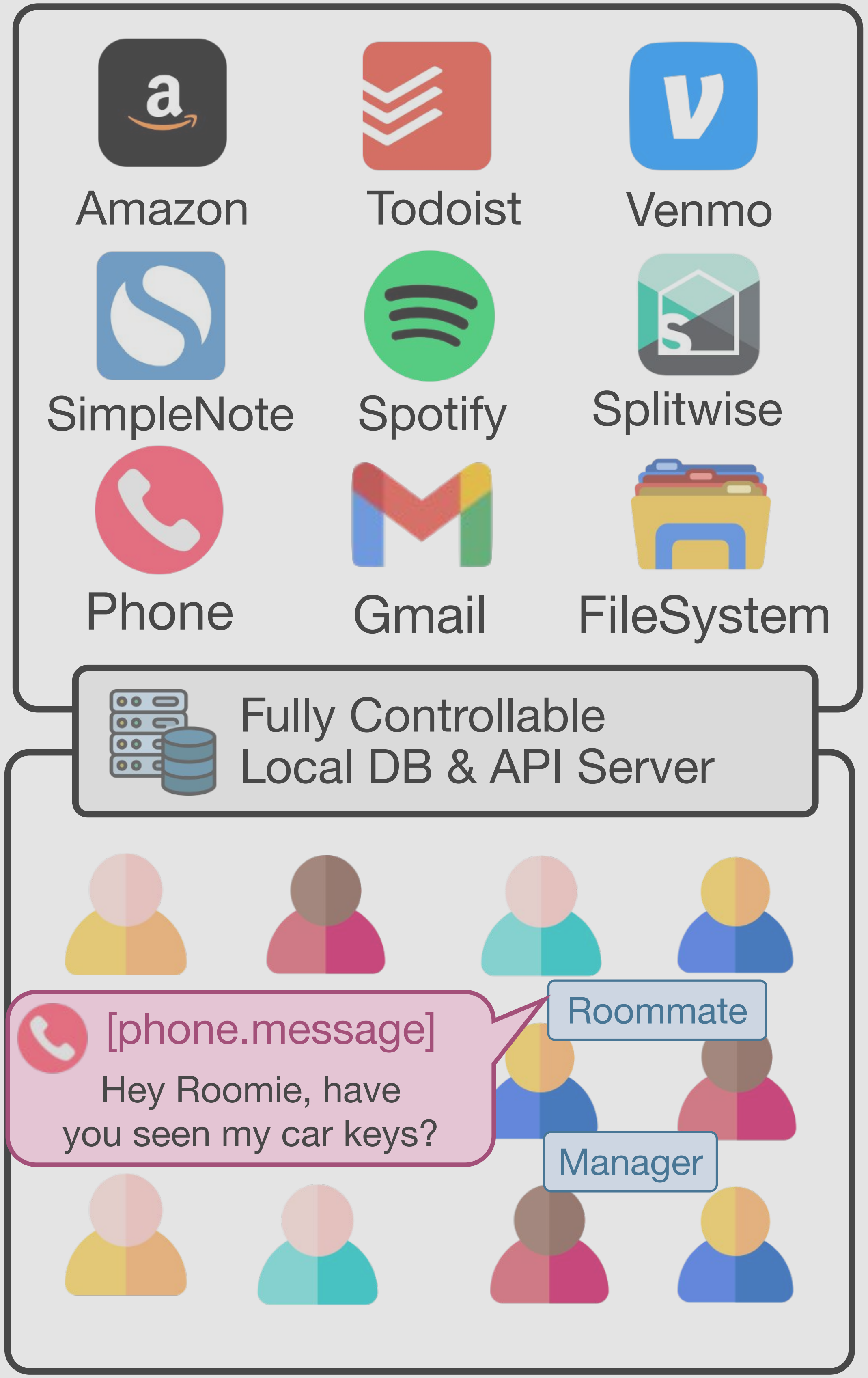
Hey Roomie, have you seen my car keys?

Apps & People

- Local implementations of 9 apps ★
  - API-operable
  - Fully controllable
- Data simulating digital activities of 100+ people with relationships

★ The simulated apps are *our* implementations do not imply any affiliation, endorsement, or sponsorship by the trademark owners.

Engine | Benchmark | Evaluation



[phone.message]
Hey Roomie, have you seen my car keys?

Apps & People

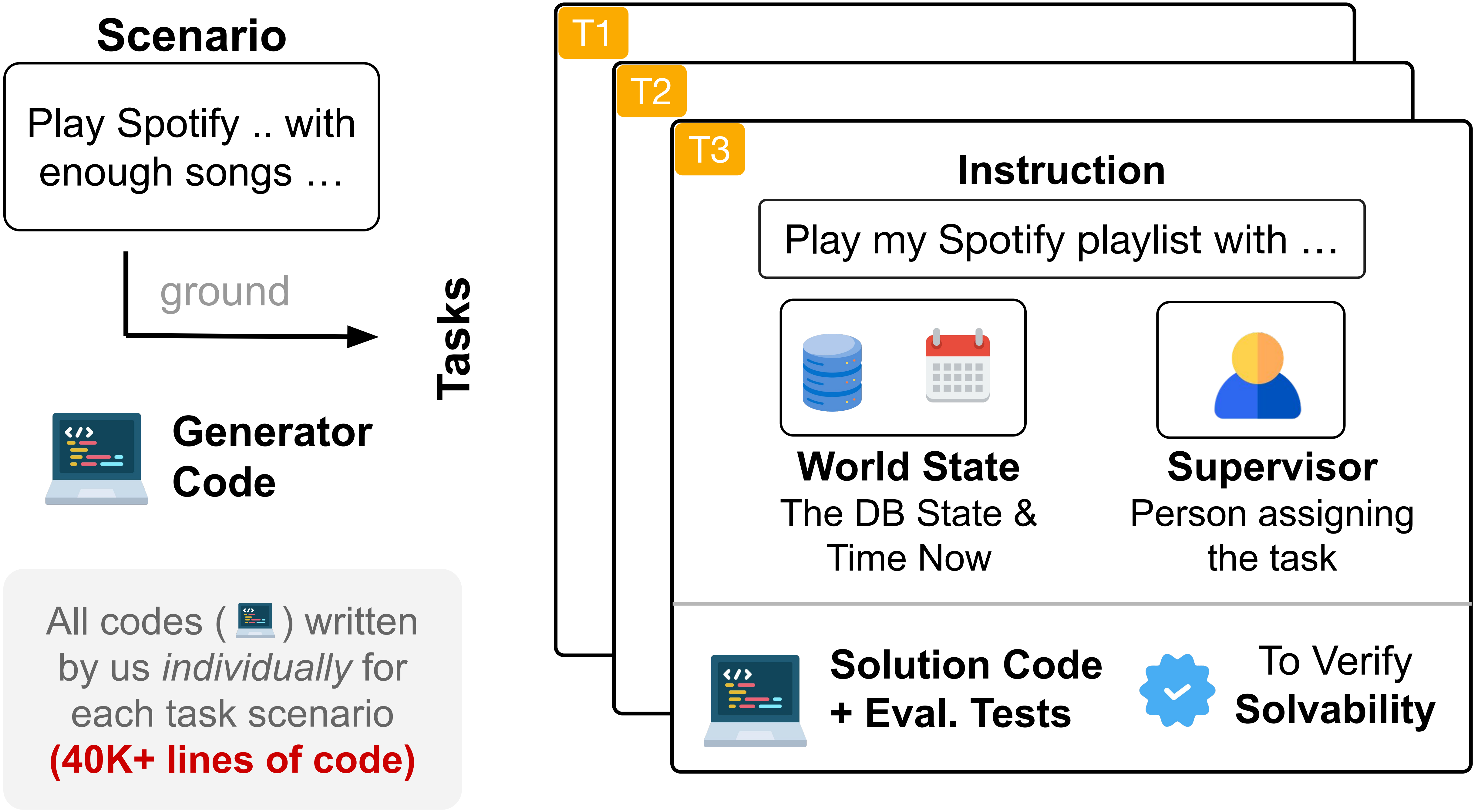
- Local implementations of 9 apps
  - API-operable
  - Fully controllable
- Data simulating digital activities of 100+ people with relationships

High-Fidelity & Reliable Digital World

- 60K+ lines of code
- 457 APIs w/ detailed docs
- 100+ DB tables
- 1700+ unit tests (98% code coverage)

Engine | **Benchmark** | Evaluation

Day-to-day (rich & interactive coding) tasks developed using Engine



Scenario

Play Spotify .. with enough songs …

ground

Tasks

Generator Code

All codes (💻) written by us *individually* for each task scenario **(40K+ lines of code)**

T1

T2

T3

Instruction

Play my Spotify playlist with …

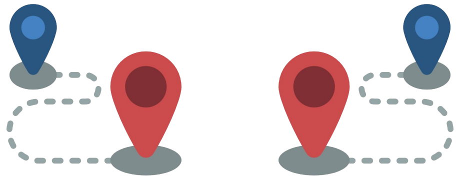
World State
The DB State & Time Now

Supervisor
Person assigning the task

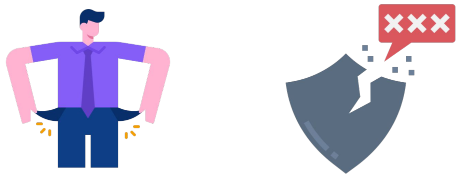
Solution Code + Eval. Tests

To Verify **Solvability**

How to *robustly* evaluate agents on such tasks?



Many ways of **completing the goal**

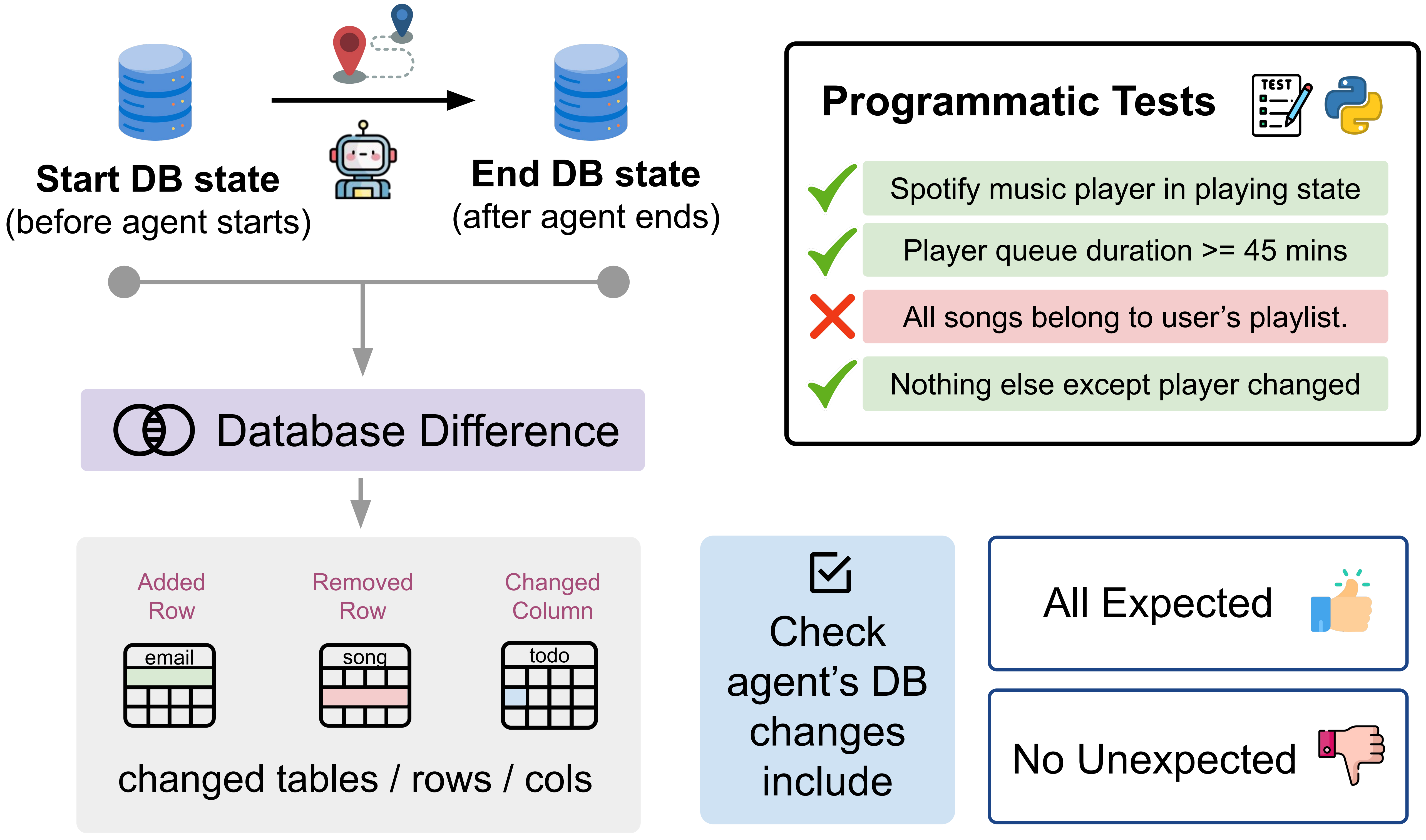


Many ways of **causing collateral damage**

✗ **Comparison to a reference** code/API calls **isn't suitable**

✓ AppWorld uses **State-based** & **Execution-based** approach.

Engine | Benchmark | **Evaluation**



Start DB state
(before agent starts)

End DB state
(after agent ends)

Database Difference

Added Row | Removed Row | Changed Column

email | song | todo

changed tables / rows / cols

Programmatic Tests

✓ Spotify music player in playing state

✓ Player queue duration >= 45 mins

✗ All songs belong to user's playlist.

✓ Nothing else except player changed

☑ Check agent's DB changes include

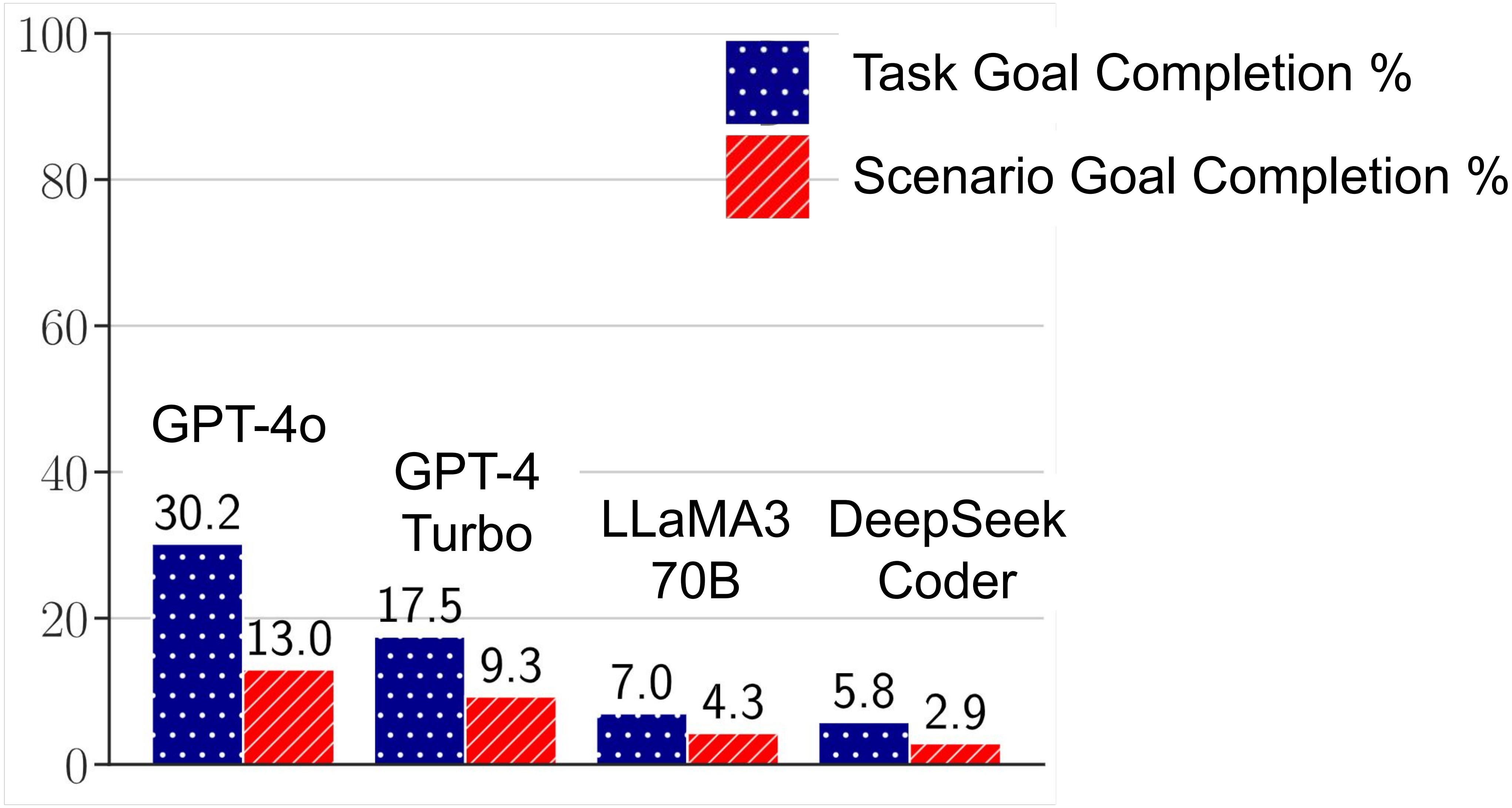
All Expected

No Unexpected

Cumulative DB Changes

How do Agents perform on AppWorld?

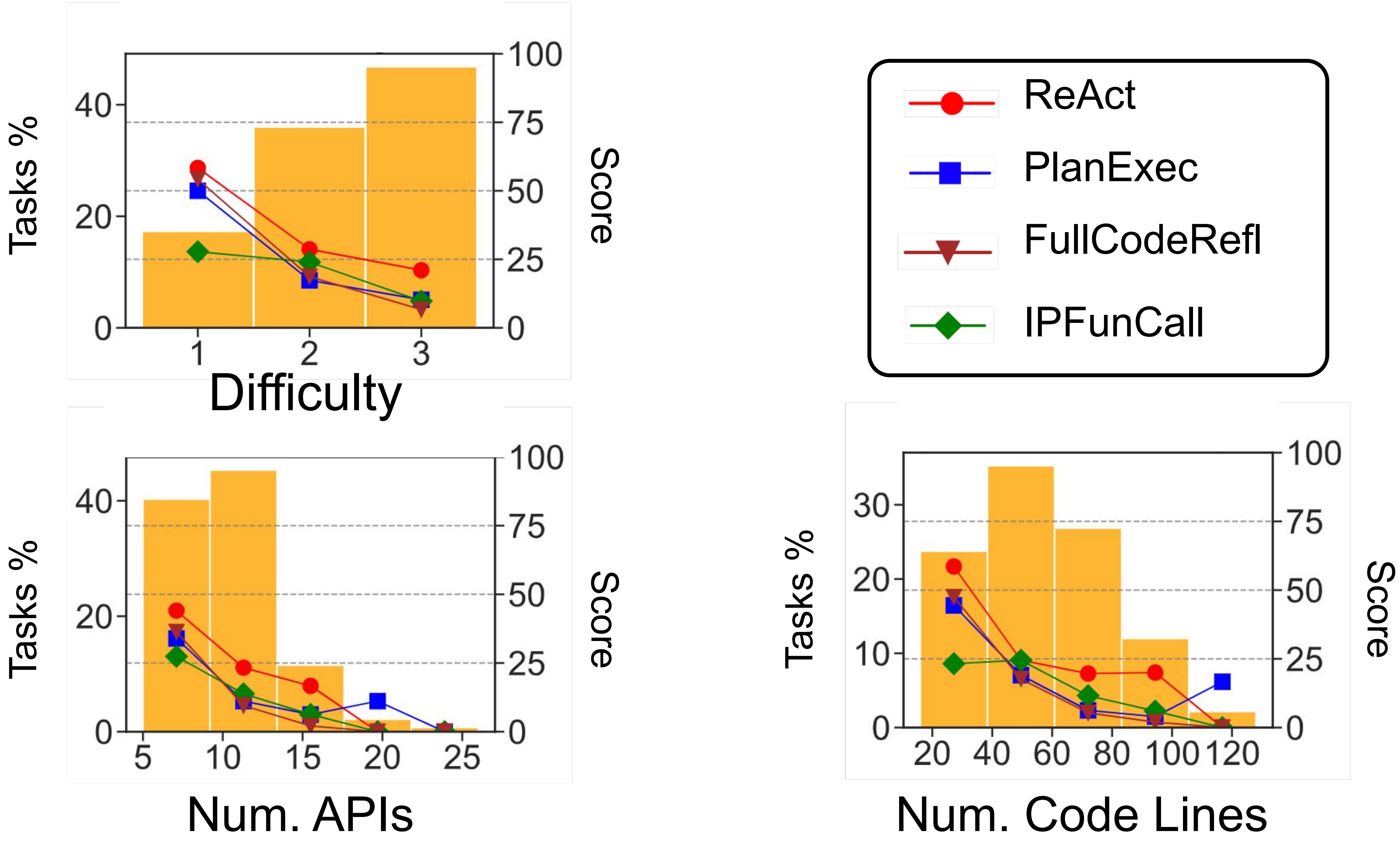
State-of-the-art LLM agents struggle on AppWorld.



For each LM, **max score** across **4 few-shot methods**:
ReAct, PlanExec, FullCodeRefl, IPFunCall

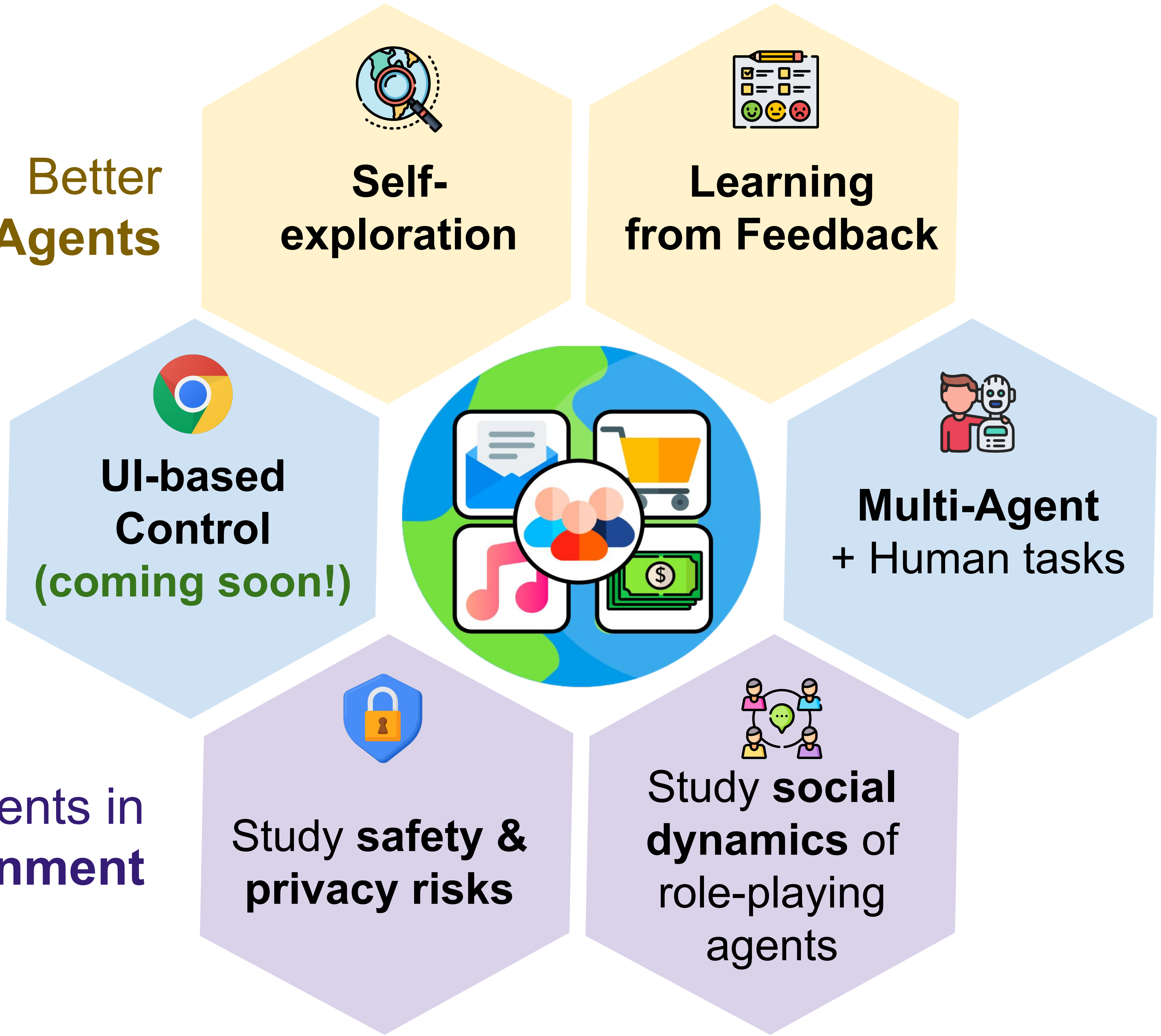
How do Agents perform on AppWorld?

Benchmark enables analysis across difficulty levels.



GPT-4o Task Goal Completion %

Future Possibilities



Better **Agents**

Self-exploration

Learning from Feedback

New Agent **Benchmarks**

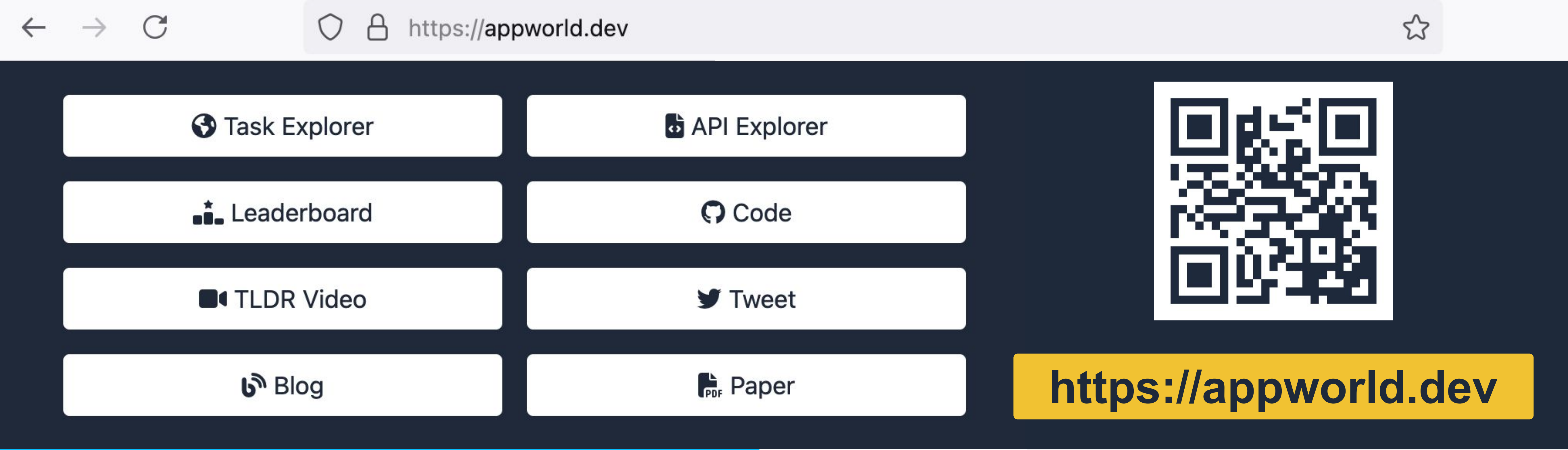
UI-based Control (coming soon!)

Multi-Agent + Human tasks

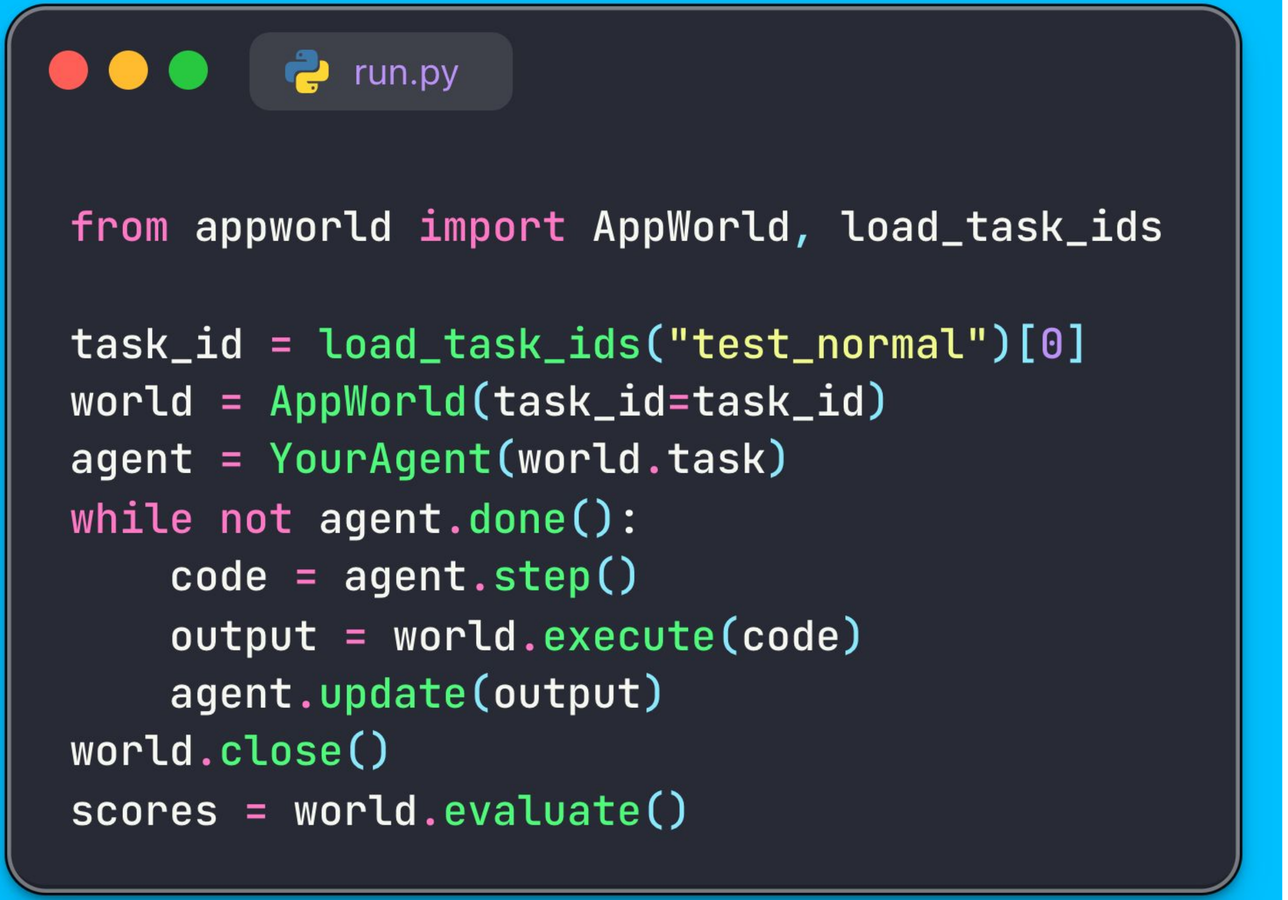
Study Agents in **Environment**

Study **safety & privacy risks**

Study **social dynamics** of role-playing agents



https://appworld.dev



```python
from appworld import AppWorld, load_task_ids

task_id = load_task_ids("test_normal")[0]
world = AppWorld(task_id=task_id)
agent = YourAgent(world.task)
while not agent.done():
    code = agent.step()
    output = world.execute(code)
    agent.update(output)
world.close()
scores = world.evaluate()
```

Build & Test your Agent

🍰 **AppWorld is Easy-to-Use**!
Just ‘pip install appworld’ & start.
No docker / server necessary,
Comes with Jupyter-styled shell.

⚡ **And it is Fast**!
Tasks load in < 0.5s,
evaluate in < 0.6s,
& APIs respond in << 30ms.