LlamaTouch A Faithful and Scalable Testbed for Mobile UI Task Automation

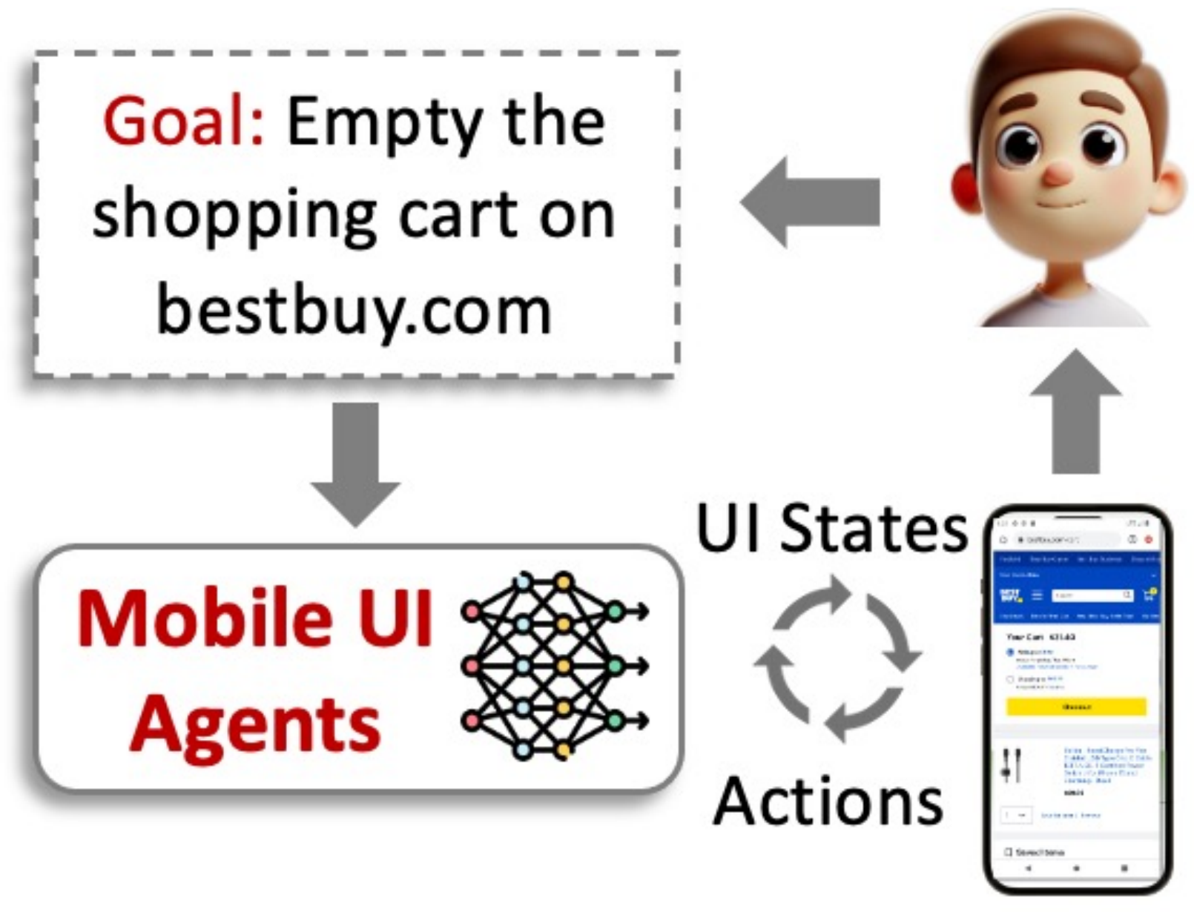
Li Zhang, Shihe Wang, Xianqing Jia, Zhihan Zheng, Yunhe Yan, Longxi Gao, Yuanchun Li[#], Mengwei Xu

Beijing University of Posts and Telecommunications (BUPT)

[#]Institute for AI Industry Research (AIR), Tsinghua University

Background

Mobile UI Task Automation



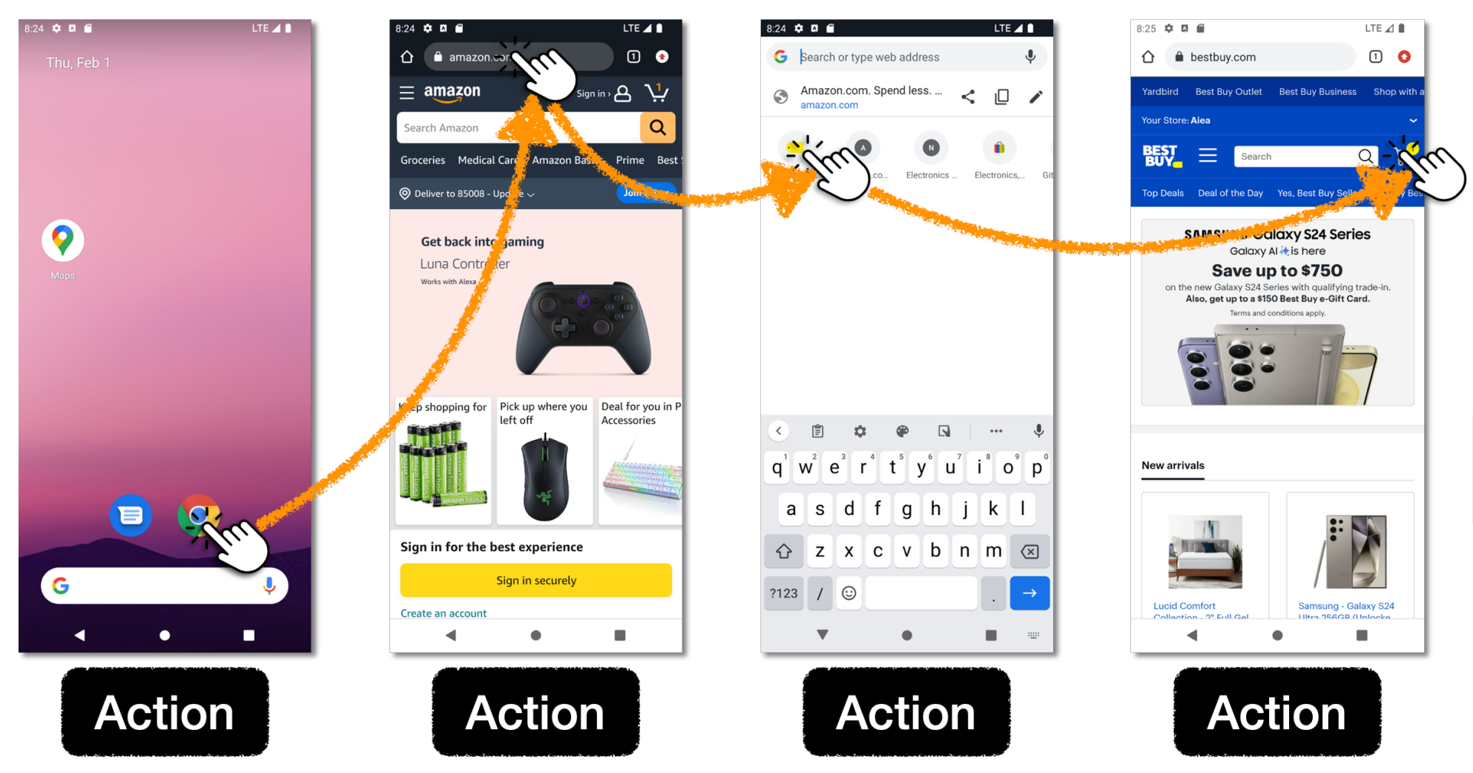
Goal: Empty the shopping cart on bestbuy.com

Mobile UI Agents

UI States

Actions

Task Execution Traces



Logged screenshots, view hierarchies, actions, …

Evaluating Task Completion is Difficult!

Approach #1: *Human validation*

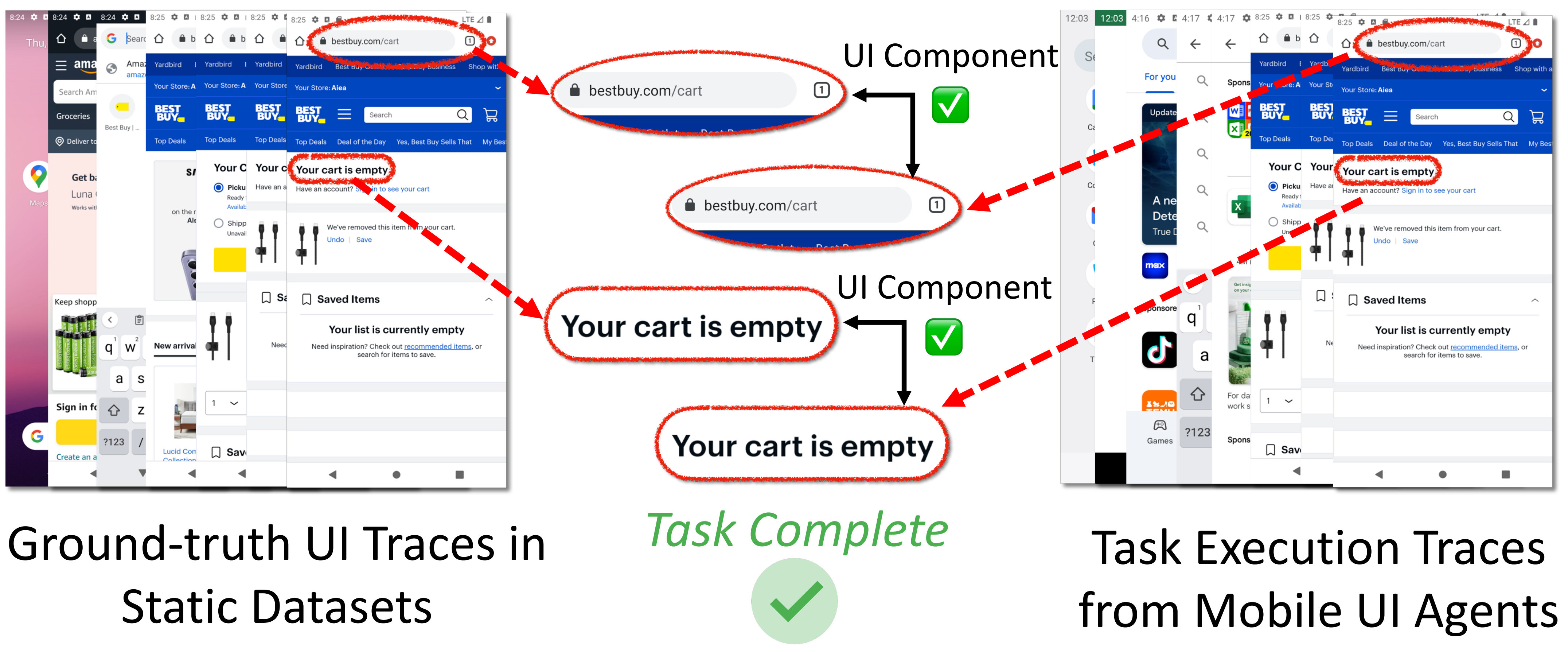
- High accuracy
- Low scalability
- High cost

Approach #2: *Evaluation on static datasets*

- High scalability
- Low accuracy
- Too many false negatives

Target of LlamaTouch: Scalable, faithful, and realistic evaluation in real-world mobile environments.

Key Idea: Extract & Compare Essential States



Ground-truth UI Traces in Static Datasets

Task Complete

Task Execution Traces from Mobile UI Agents

- **Observation**: UI automation tasks transfer app states that may represent on UI.
- **Insights**: Compare app states after task execution with predefined essential states.
- **Define essential app states**: From both the screen level and single UI comp. level.
- Annotation primitives + UI comp. indices!

Exact Fuzzy Activity Exclude Click Type … + <13> <25> …

Dataset & Testbed

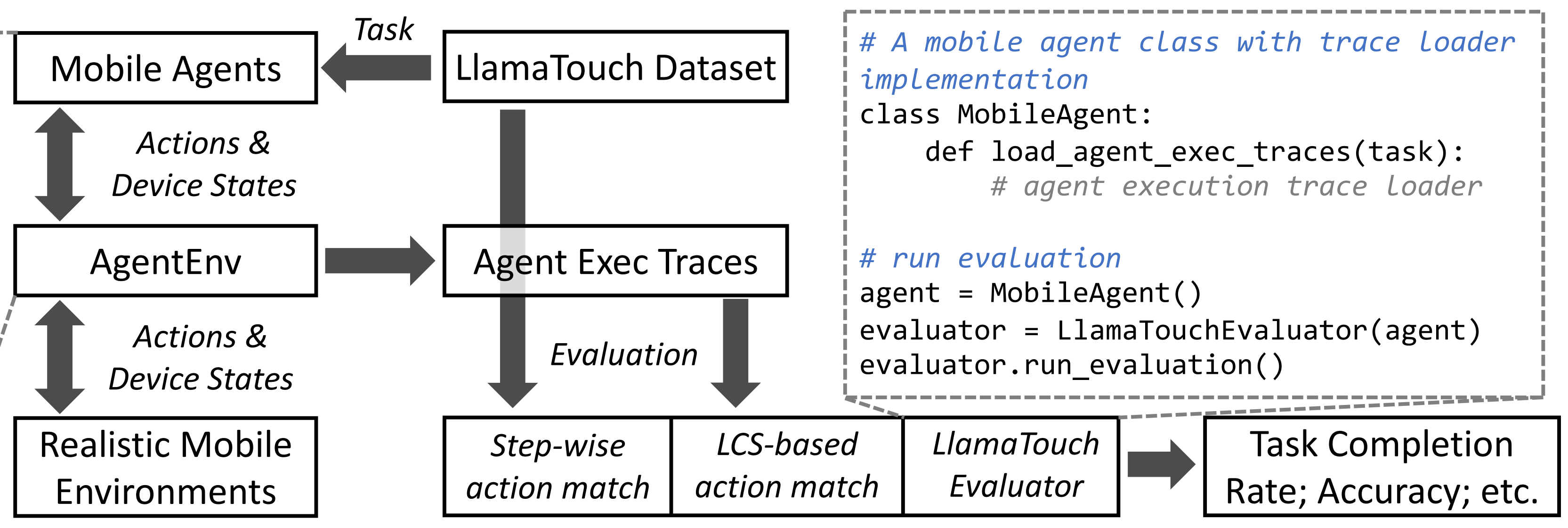
- **Dataset scale**: 496 tasks (102 from previous datasets AITW + 394 self-constructed ones)
- **Data**: Screenshots, view hierarchies, actions, task descriptions, annotated essential states

Code Demo for Mobile UI Task Execution

```
# agent/environment setup
agent = mobile_agent.init() # agent initialization
task = agentenv.task.get()  # get task metadata
agentenv.setup_task(task)   # task-specific setup

# task execution
while not agent.task_complete():
    state = agentenv.get_state()
    act = agent.predict(task, state.screenshot,
state.vh)  # predict action on current UI based on
task instruction and observed UI states
    agentenv.post(act)  # post action to device
```

LlamaTouch Workflow



Mobile Agents

Task

LlamaTouch Dataset

Actions & Device States

AgentEnv

Agent Exec Traces

Actions & Device States

Evaluation

Realistic Mobile Environments

Step-wise action match

LCS-based action match

LlamaTouch Evaluator

Task Completion Rate; Accuracy; etc.

Code Demo for Trace Evaluation

```
# A mobile agent class with trace loader
implementation
class MobileAgent:
    def load_agent_exec_traces(task):
        # agent execution trace loader

# run evaluation
agent = MobileAgent()
evaluator = LlamaTouchEvaluator(agent)
evaluator.run_evaluation()
```

- **Realistic Mobile Environments**: Android emulators and smartphones
- **LlamaTouch AgentEnv**: Bridge between Android envs and mobile UI agents
- **LlamaTouch Evaluator**: For faithful and scalable task evaluation

Experiments & Findings

Experimental setup: 4 mobile UI agents using LLM/MLLMs (ChatGPT-4/4V, LLaVA, etc.)

Table 7: Accuracy (Acc. %) of different evaluation approaches among all successful tasks in human validation.

Mobile Agent	Step-wise action match	LCS action match	LlamaTouch	Human
	Acc.	Acc.	Acc.	# success
AutoUI	0.00	0.00	77.78	9
AutoDroid	0.00	0.00	73.91	69
AppAgent	0.00	3.03	93.94	33
CoCo-Agent	0.00	0.00	70.00	10
Average	0.00	0.76	78.91	30

- ➢ LlamaTouch achieves an 80% success rate in detecting completed tasks; others are around 0%.
- ➢ Existing mobile UI agents struggle to complete end-to-end UI automation tasks in real-world environments.
- ➢ Check out our paper for more results (e.g., effectiveness of anno. primitives).

- ❑ Open-source: https://github.com/LlamaTouch/LlamaTouch
- ❑ Contact: li.zhang@bupt.edu.cn