Interactive Planning-based Cognitive Assistance on the Edge

Zhiming Hu, Maayan Shvo, Allan Jepson and Iqbal Mohomed
Samsung AI Centre, Toronto

SAMSUNG
AI CENTRE

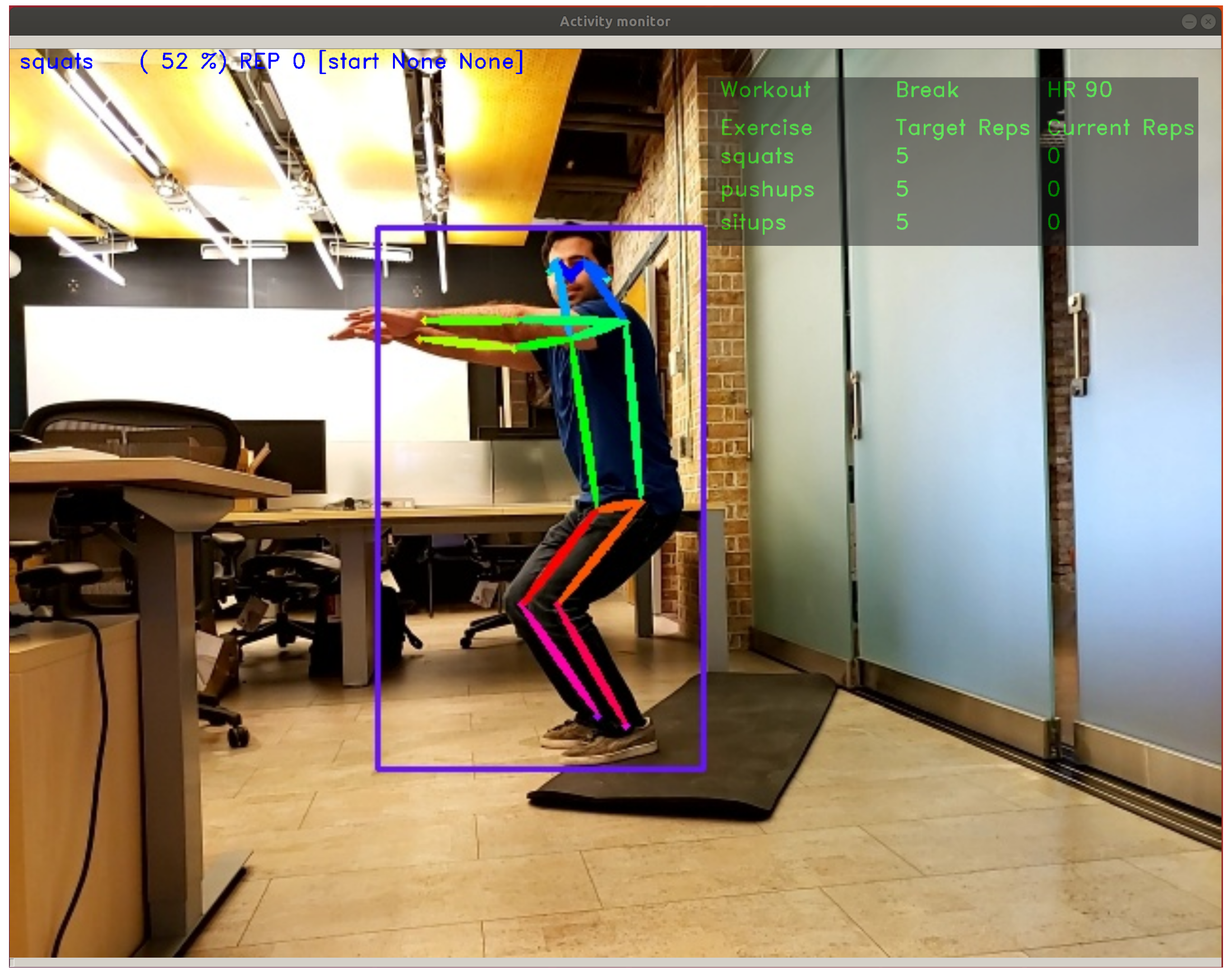
What is cognitive assistance?

- One of the most exciting applications in AR Glasses
  - Google Glass, HoloLens 2
- Helpful in a myriad of tasks
  - Health care education and training
  - Industrial tool for remote support
  - Cooking assistant and fitness coach

Image source for HoloLens 2: https://commons.wikimedia.org/wiki/File:HoloLens_2.jpeg,
https://creativecommons.org/licenses/by/2.0/legalcode, changes are not made on the image.

How to build a cognitive assistant?

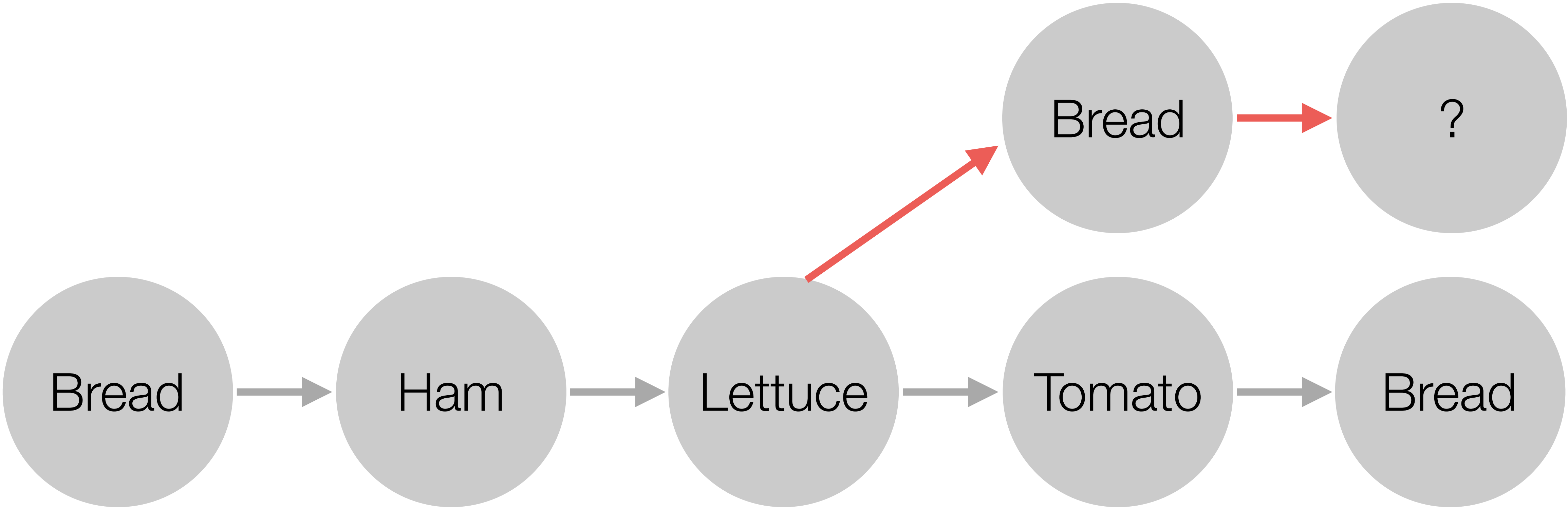
- Lots of existing work on building cognitive assistance [1,2,3,4]
  - Perception module
    - Determine the current task state
  - Cognitive module
    - Generate the next step



[1] VideoPipe: Building Video Stream Processing Pipelines at the Edge, Middleware 2019
[2] https://github.com/cmusatyalab/gabriel-sandwich
[3] Mohan, S., Ramea, K., Price, B., Shreve, M., Eldardiry, H., & Nelson, L. (2019). Building Jarvis-A Learner-Aware Conversational Trainer. In IUI Workshops.
[4] Laird, John E. The Soar cognitive architecture. MIT press, 2012.

The motivation

- While it is simple to build a state machine to guide a user to complete some tasks, there are several issues
  - The state machine needs to be pre-defined
  - It cannot list all the possible user errors, thus cannot recover from such failure cases.



How about a planner?

- Benefits
  - Flexible, can recover from any user errors
- Challenges
  - Need to calculate accurate current task state (CTS)
  - Not as computationally efficient as state machines.

A planning problem

- A planning problem may be encoded in PDDL by defining the domain, initial state, and goal state.
  - $stack(x,y) \in A$
    - $Pre_{stack} = \{clear(x), clear(y), ontable(x)\}$
    - $eff^{+}_{stack} = \{on(x,y)\}$ (note: x is on y)
    - $eff^{-}_{stack} = \{clear(y)\}$
  - $G = \{onTable(\text{bread1}), on(\text{ham}, \text{bread1}), on(\text{lettuce}, \text{ham}), on(\text{bread2}, \text{lettuce}), on(\text{tomato}, \text{bread2}), on(\text{bread3}, \text{tomato})\}$
- If **all of the ingredients are clear and on the table**, one possible solution is π = *stack(ham,bread1),stack(lettuce,ham),stack(bread2,lettuce), stack(tomato,bread2),stack(bread3,tomato)*.

The key to get the correct plan is to obtain accurate current task state

Ambiguity Resolving

- We keep track of the current task state by recognizing the actions taken since the beginning of the interaction.

- However, we may encounter ambiguous cases where we cannot determine which action was performed by the user.

Classifier for the Top Object on the Sandwich

Sequence of Classification Results:
Bread -> Ham -> Bread

↓

Stack Bread on Ham

OR

Unstack Ham from Bread

Dynamic State Tracking

- A planner with state machines
  - The planner will only be called when an unexpected action is detected

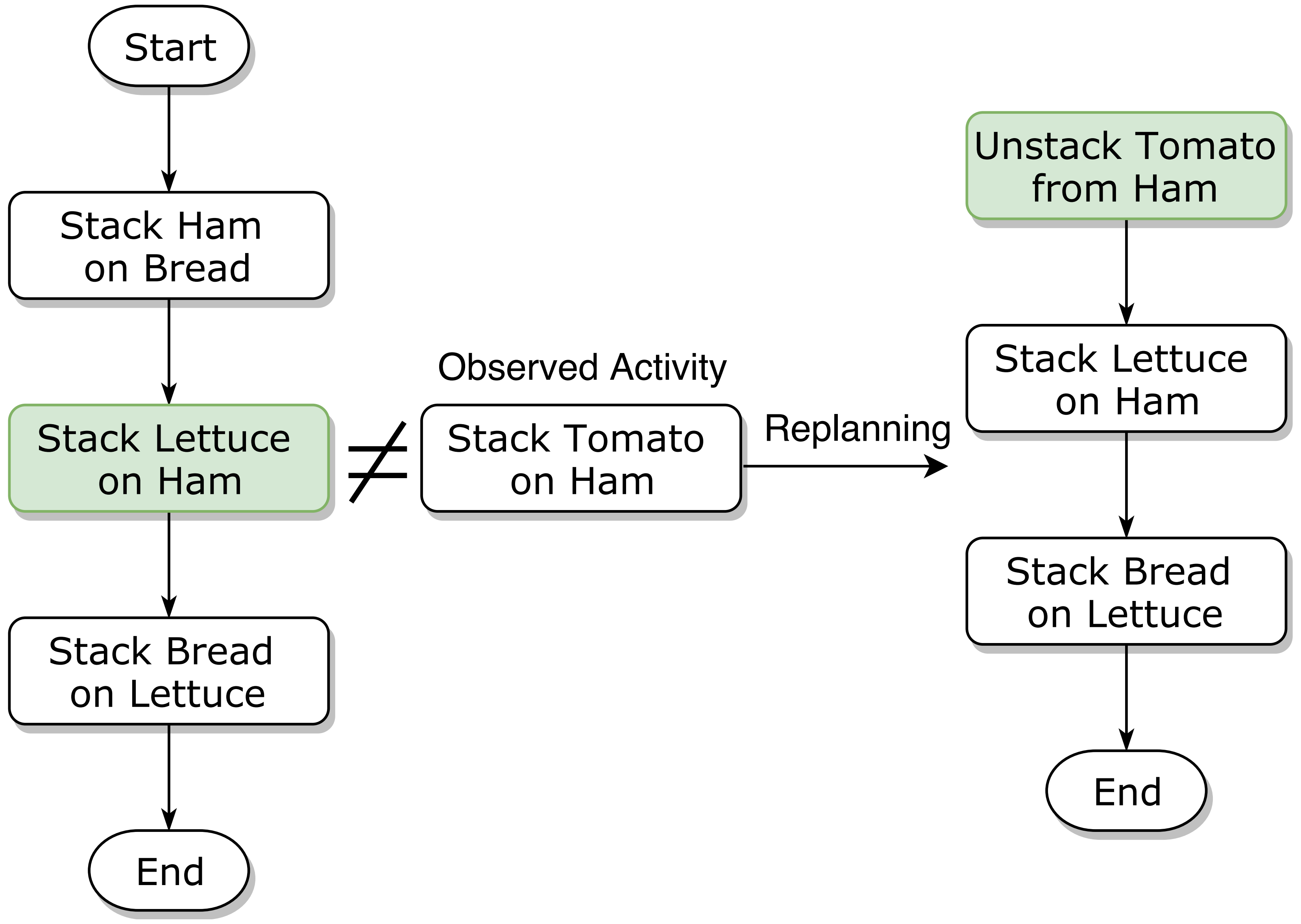
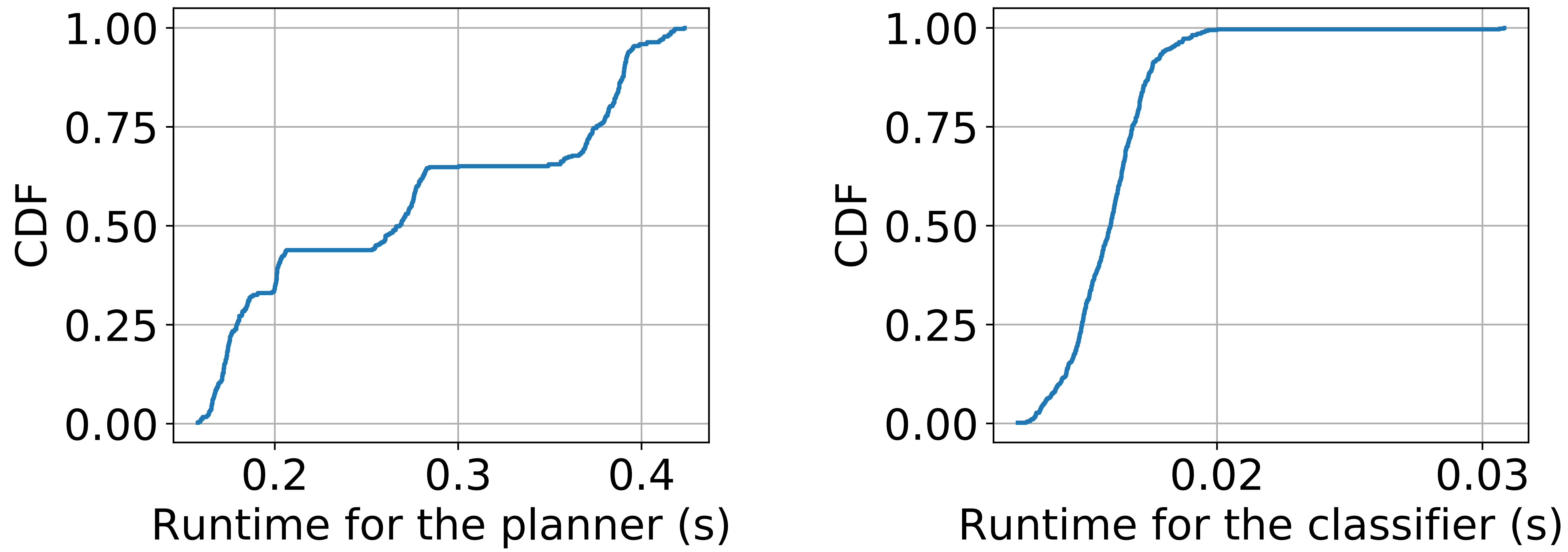


Figure 3: State tracking with a planner and state machines.
The green box shows the current expected action.

Runtime of the planner and classifier



(a) Runtime for the planner. (b) Runtime for the classifier

Figure 4: Runtime for the planner and the classifier.

It is feasible to run both the planner and classifier on the edge.

Demo

- The video for our demo is available here.

Future Work

- Personalized instructions
- Resource management for multiple cognitive assistance agents
- Applications that only need partial order
    - Linear Temporal Logic (LTL)

Summary

- We have proposed an architecture for cognitive assistants on the edge
- Ambiguous task states are prevalent and we need to deal with them
- We should combine the planner with state machines to enjoy both of the benefits.

Thanks!

zhiming.hu@samsung.com