Talk @ ISI Kolkata | September 18, 2026

Beyond the Lab: User-Aligned Assessment of Taskable AI Systems

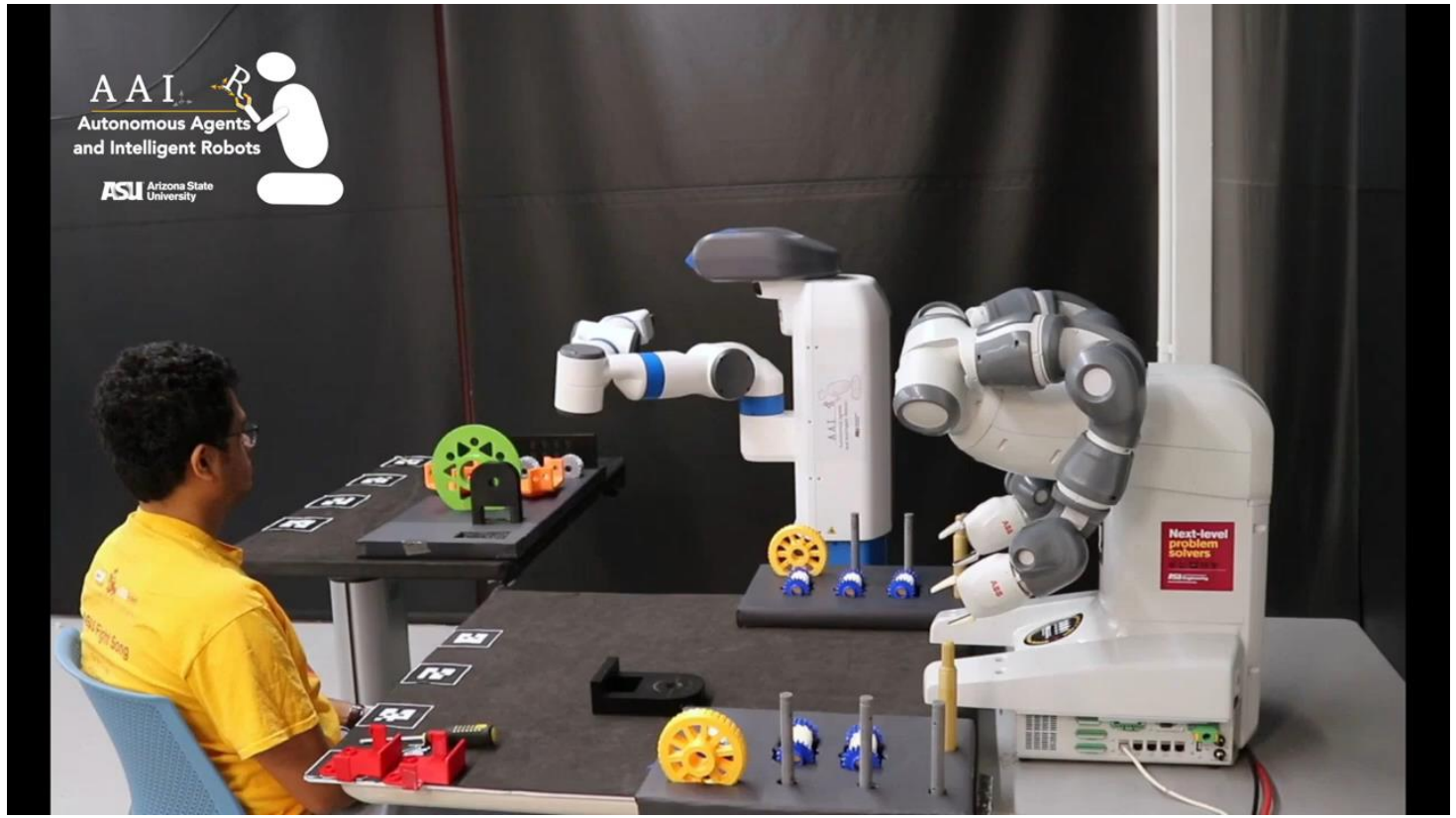
Pulkit Verma

pulkitv@cse.iitm.ac.in

pulkitverma.net

CeRAI
Centre for Responsible AI

Taskable AI Systems: Systems that can Learn and Plan



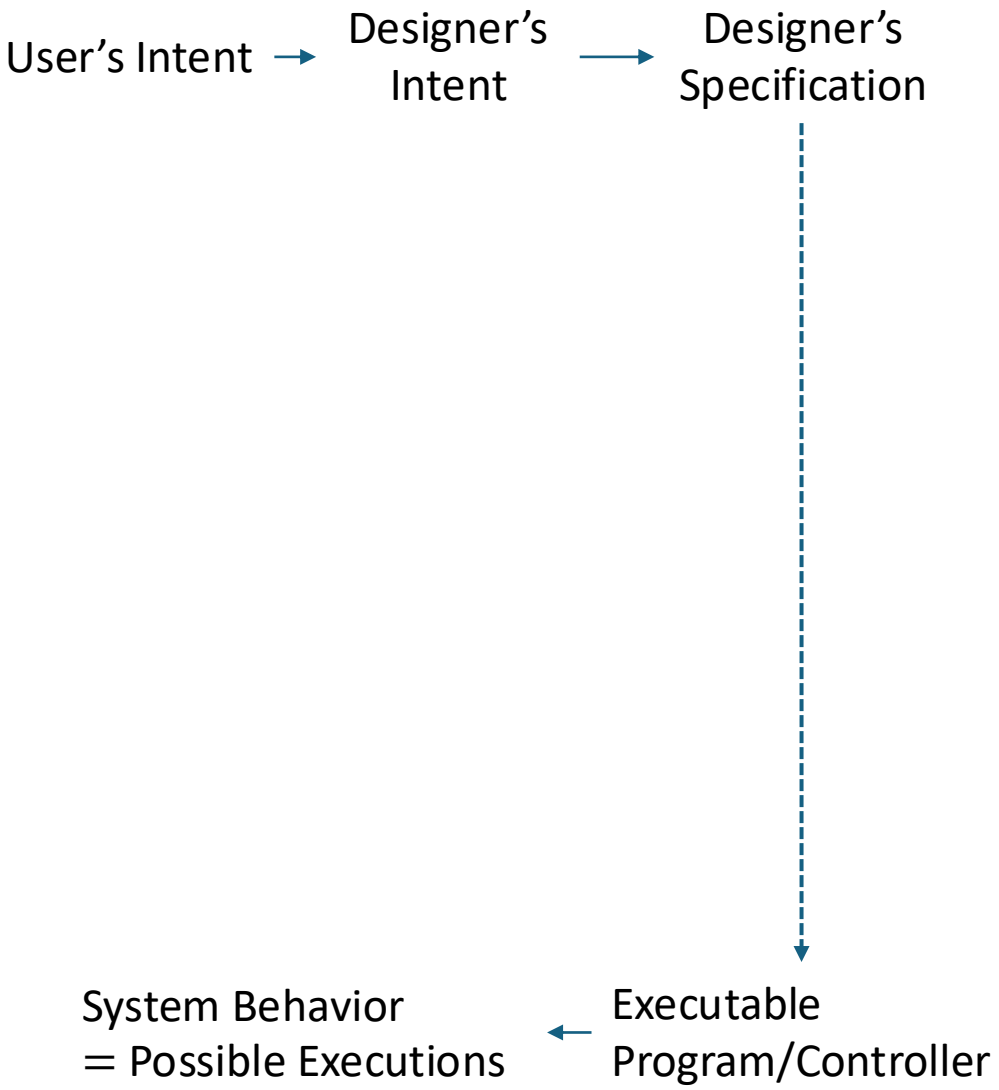
User gives a task and Robots have to complete it

- Sequential Decision-Making Systems.
- Systems designed to be able to help user.
- User has a task in mind and expects the AI system to help in that task.

Personalized Assessment of AI Systems that can Learn and Plan

- Users would like to give AI systems multiple tasks.
  - How would users know what the AI systems can do?

Classical Notion of Verification

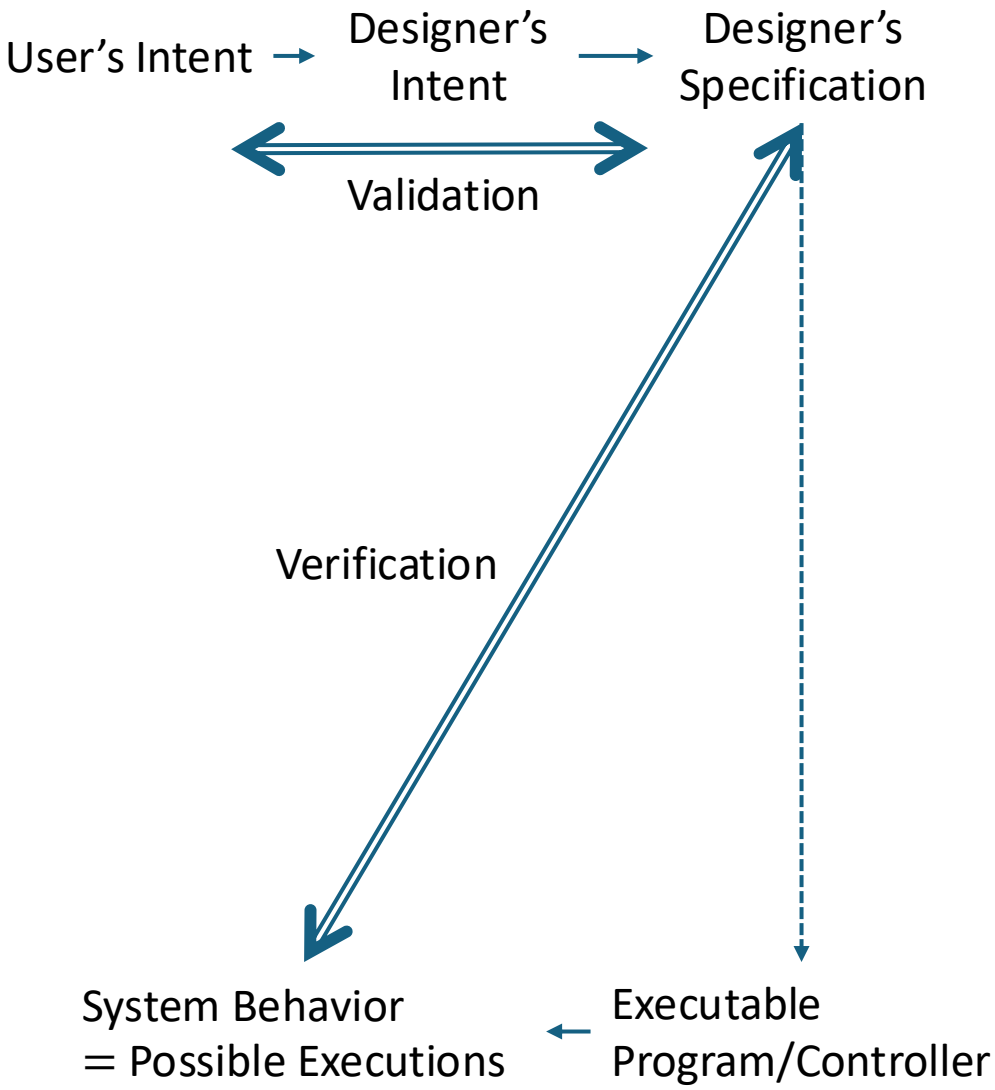


[Srivastava, Fainekos, **Verma**, Bramblett; AIA @ IJCAI '25]

Classical Notion of Verification

Input for Verification: Executable Program/Controller
(includes task spec)
+ Model/assumptions on env
+ Safety property

The designer plays a central role

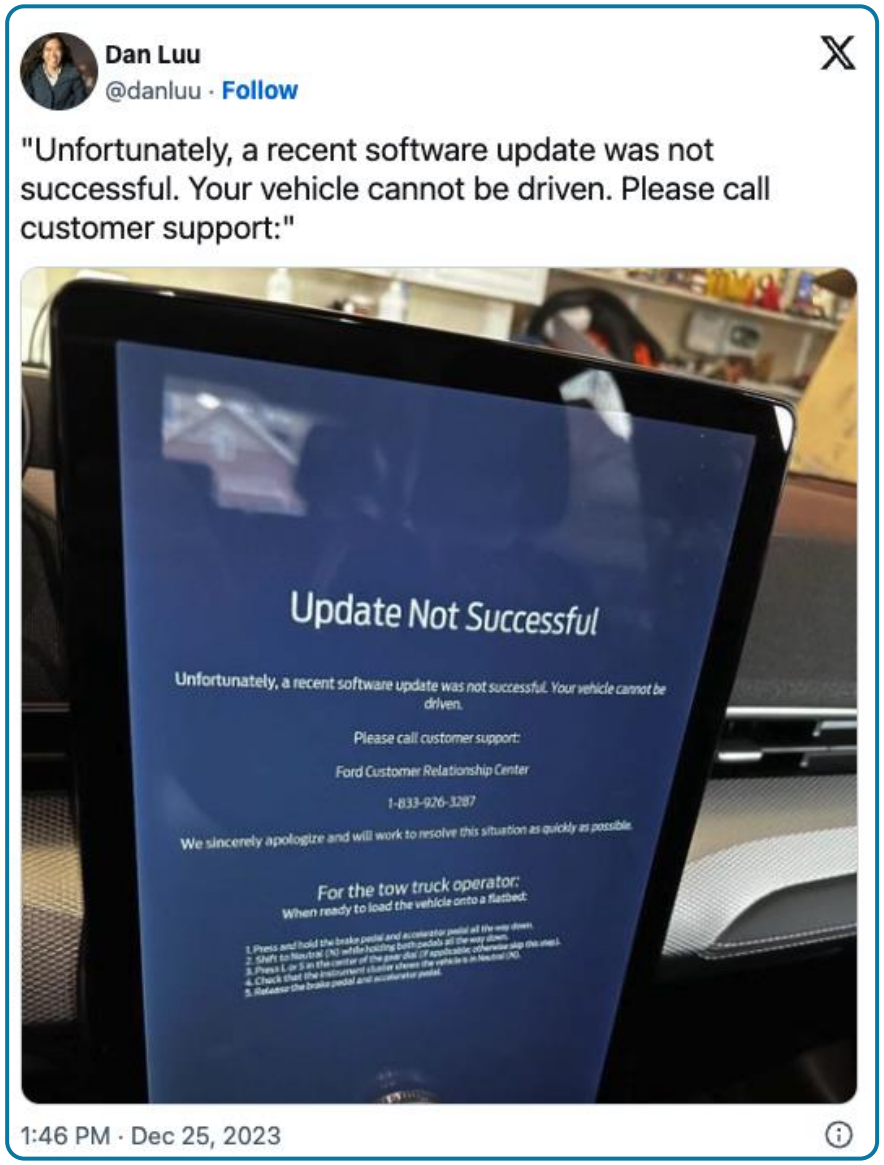


[Srivastava, Fainekos, **Verma**, Bramblett; AIA @ IJCAI '25]

Personalized Assessment of AI Systems that can Learn and Plan

- Users would like to give AI systems multiple tasks.
  - How would users know what the AI systems can do?
- AI systems should support third-party assessment.
- The assessment should work with:
  - Adaptive AI systems.
  - Black-Box AI systems.

Adaptive Taskable AI Systems



USA TODAY

Tesla self-driving software update begins rollout though company says to use with caution

Charisse Jones, USA TODAY
July 12, 2021 · 2 min read

Lucid Owners Facing Software Glitches That Brick EVs Or Drive the Wrong Direction

1 Owen Bellwood
November 10, 2022 · 2 min read

AEG combi microwave thinks it is a steam oven and no longer works after an incorrect update

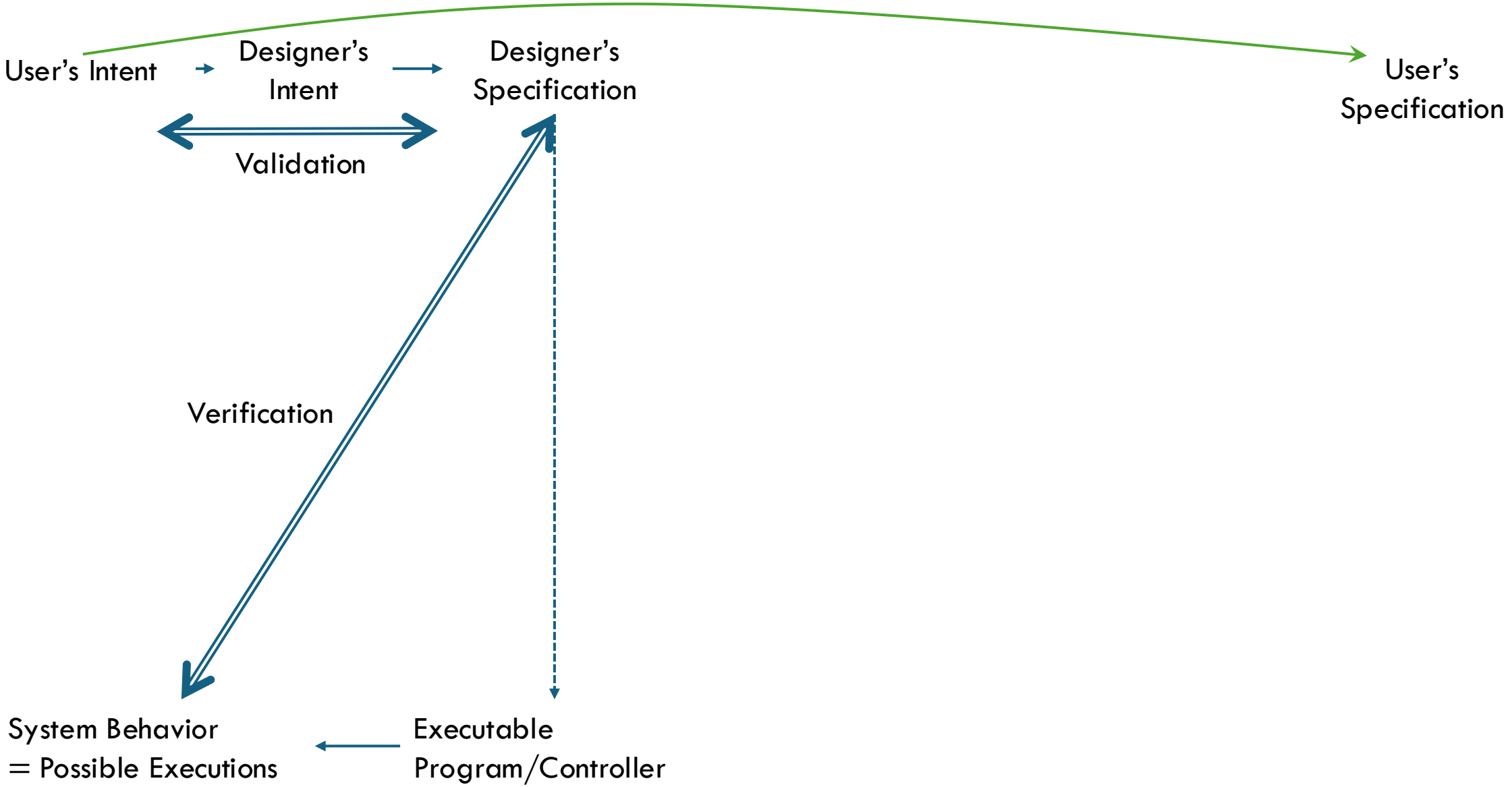
By **Julian Huijbregts** 04-03-2022 • 10:48
News editor

NEWS

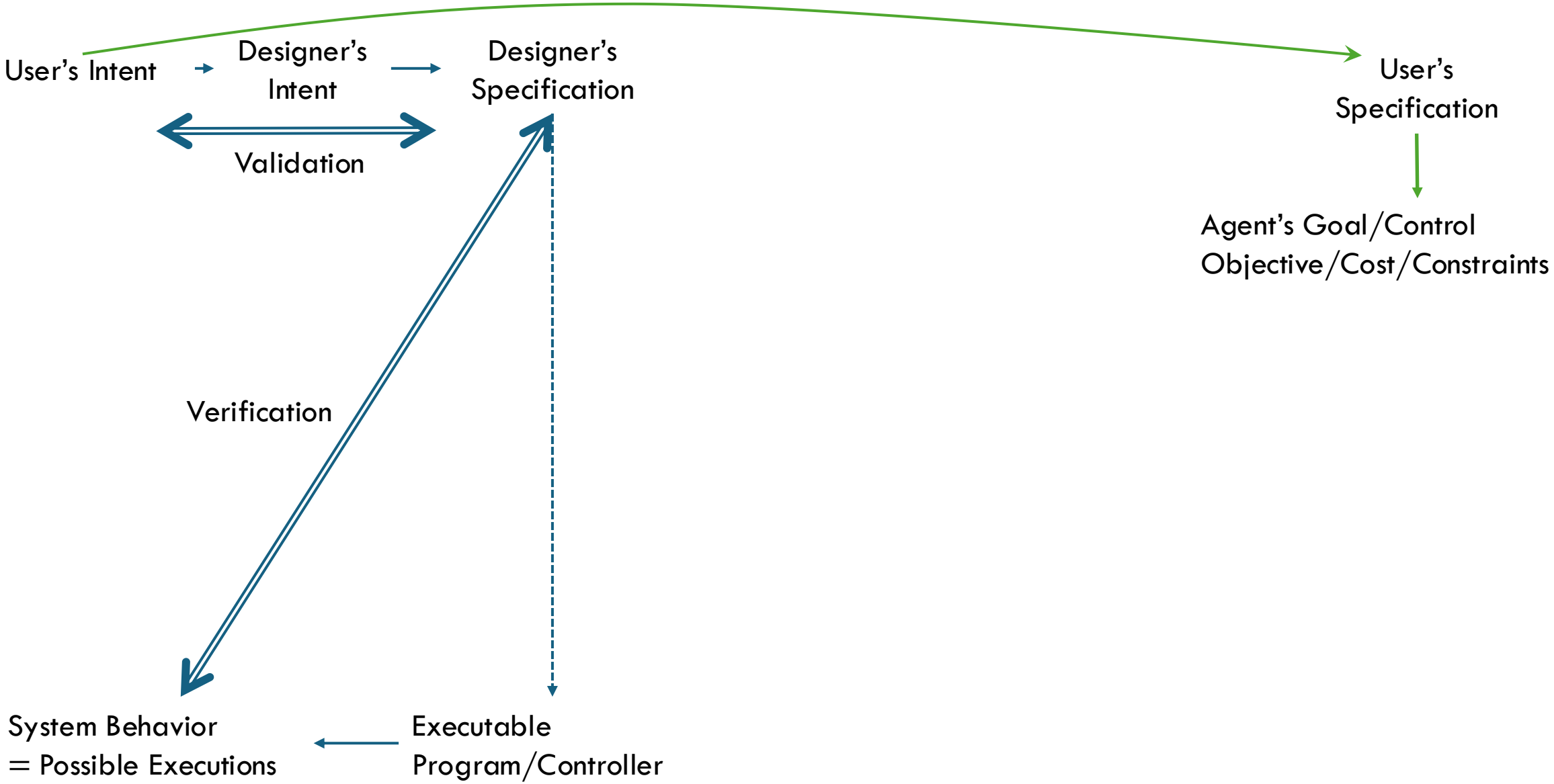
Nest thermostat software bug chills users once again

A faulty software update for the smart thermostat made batteries drain and home temperatures drop.

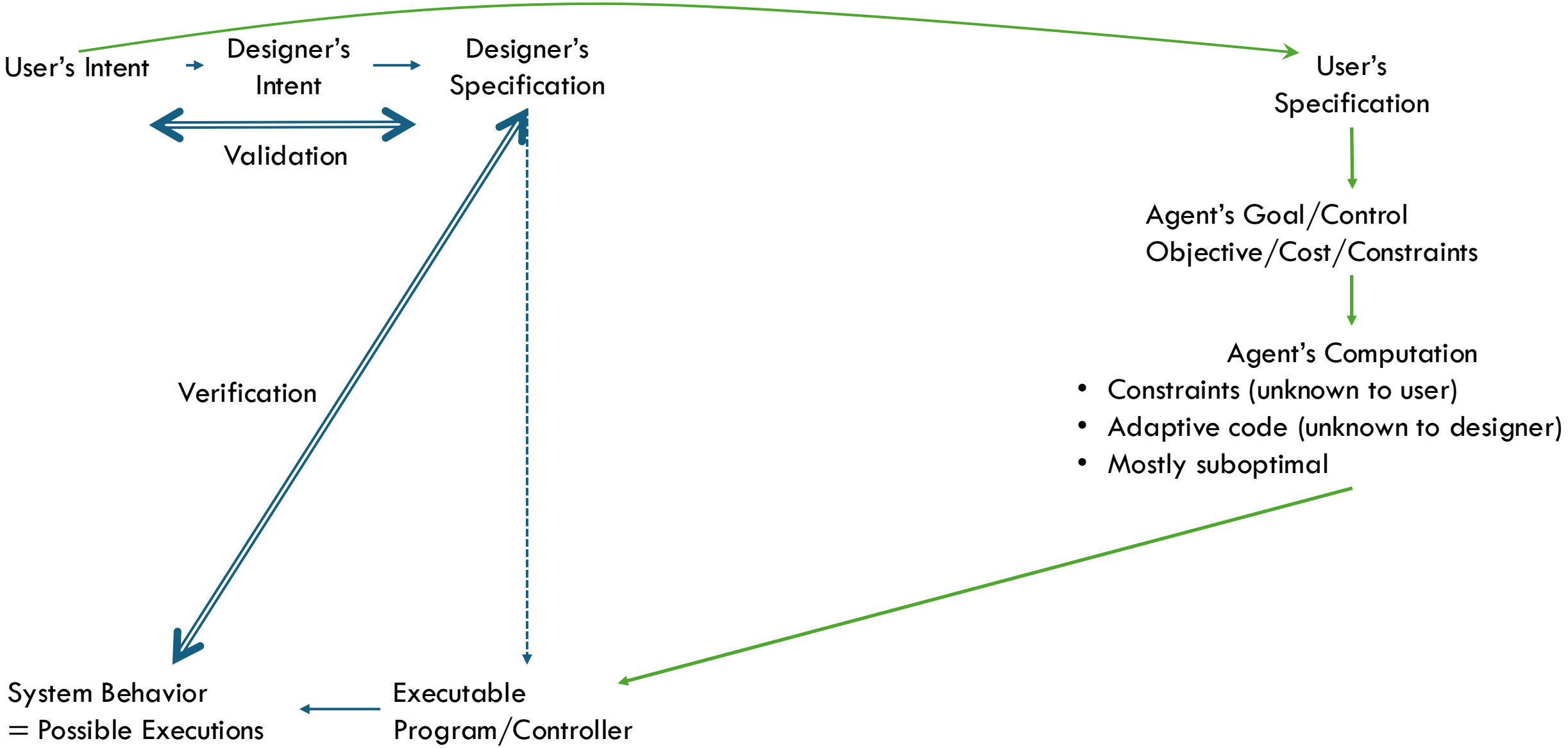
By Jared Newman
TechHive | JAN 14, 2016 8:38 AM PST



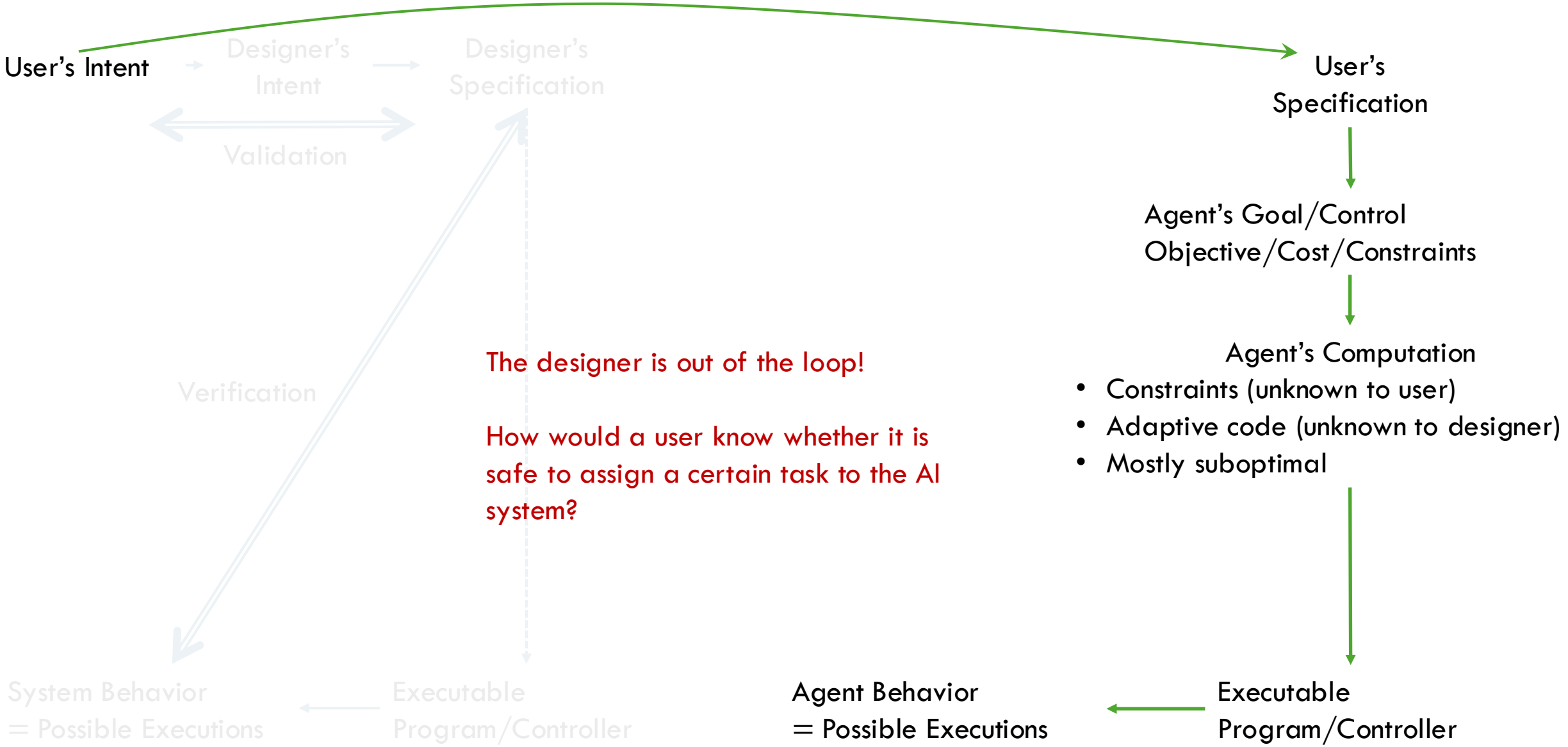
[Srivastava, Fainekos, **Verma**, Bramblett; AIA @ IJCAI '25]



[Srivastava, Fainekos, **Verma**, Bramblett; AIA @ IJCAI '25]



[Srivastava, Fainekos, **Verma**, Bramblett; AIA @ IJCAI '25]

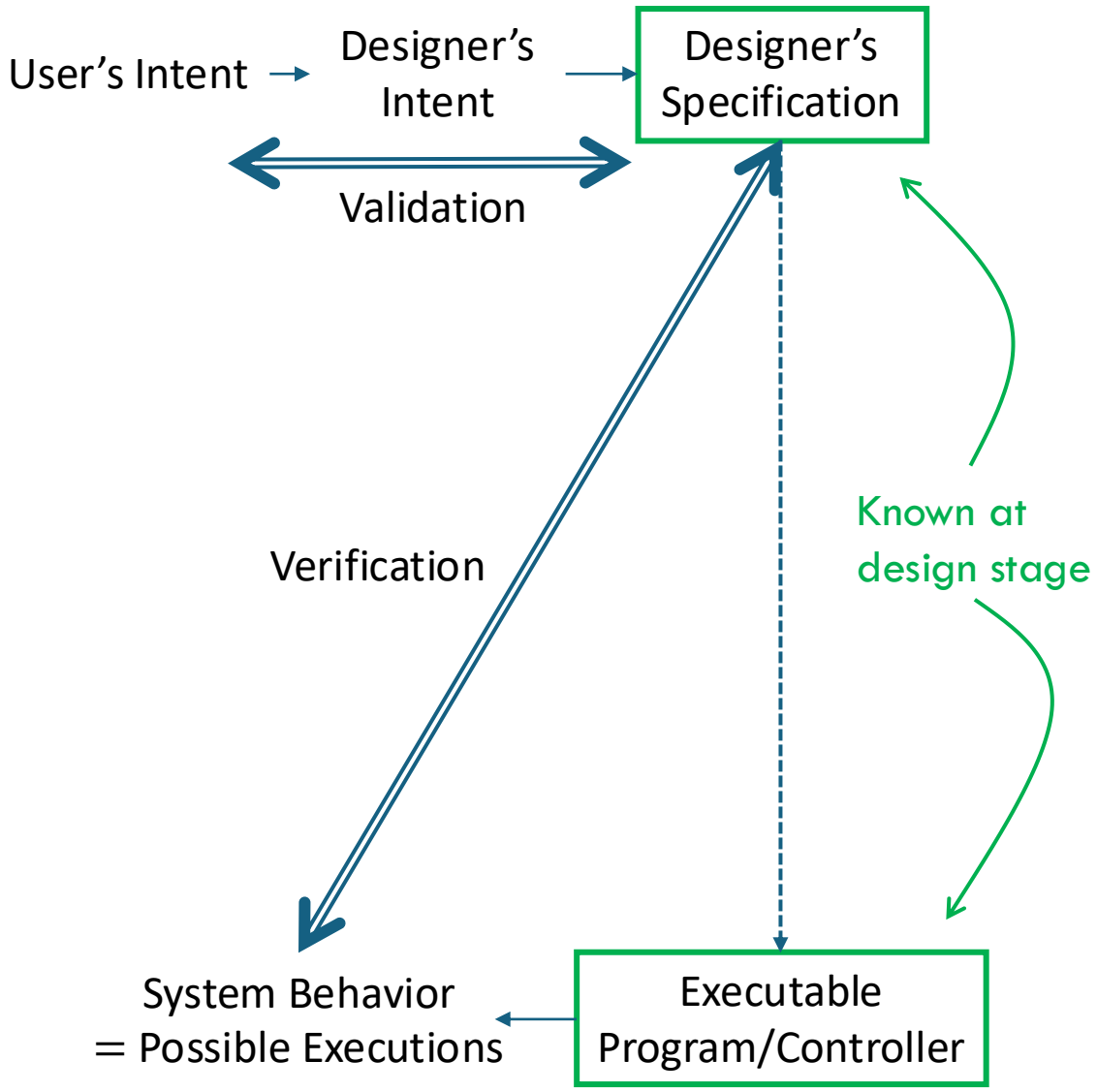


The designer is out of the loop!

How would a user know whether it is safe to assign a certain task to the AI system?

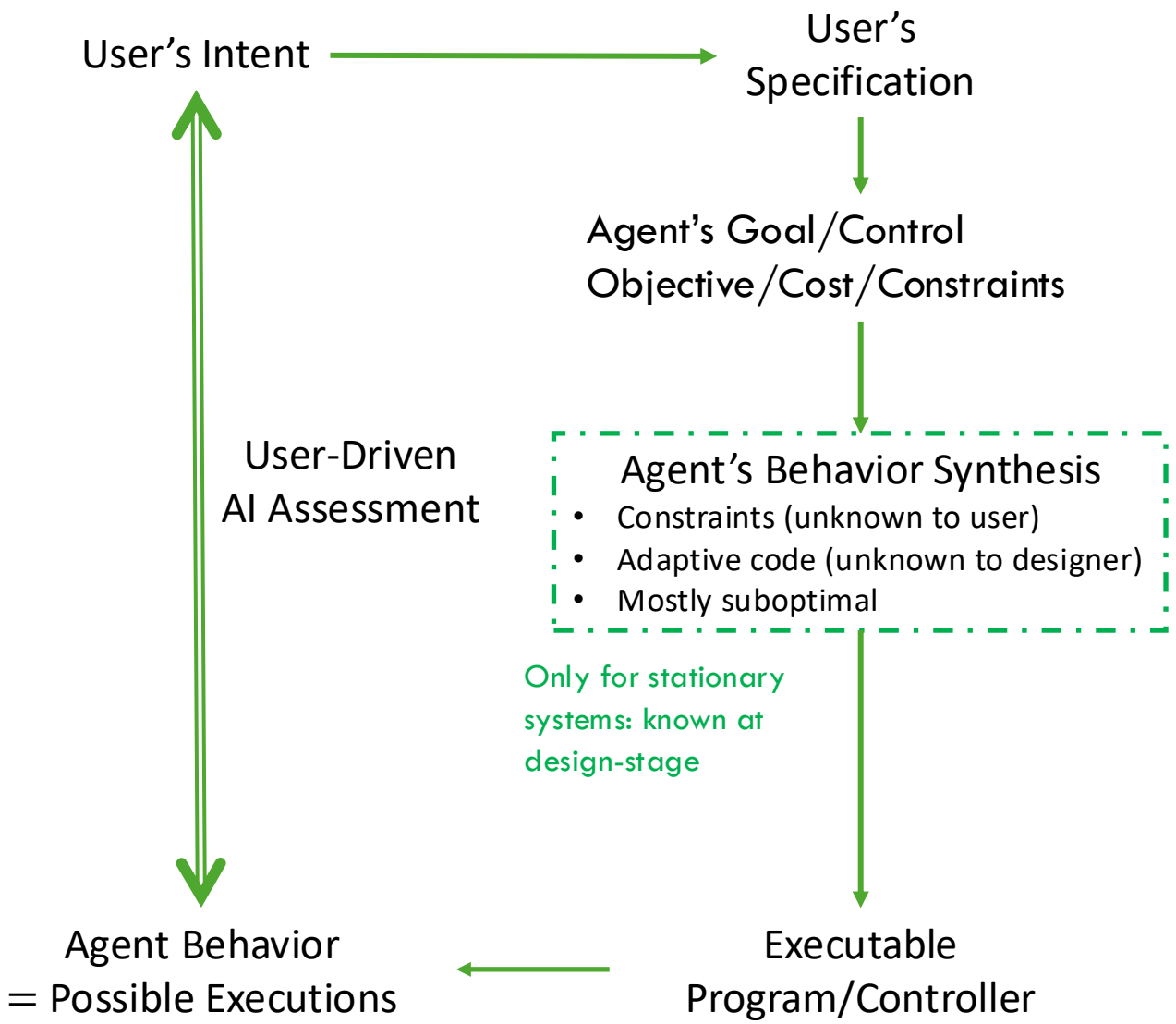
[Srivastava, Fainekos, **Verma**, Bramblett; AIA @ IJCAI '25]

Conventional Verification

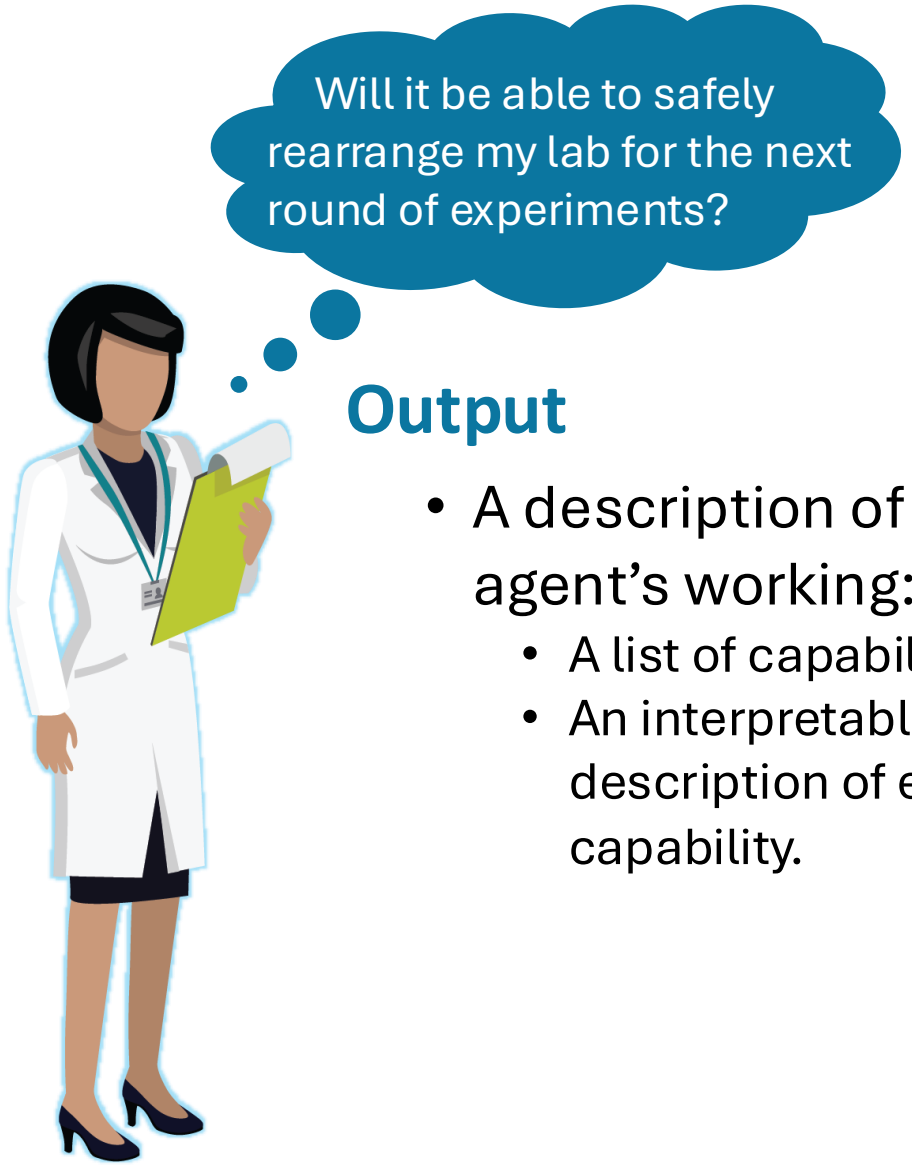


[Srivastava, Fainekos, **Verma**, Bramblett; AIA @ IJCAI '25]

Needed for AI Systems

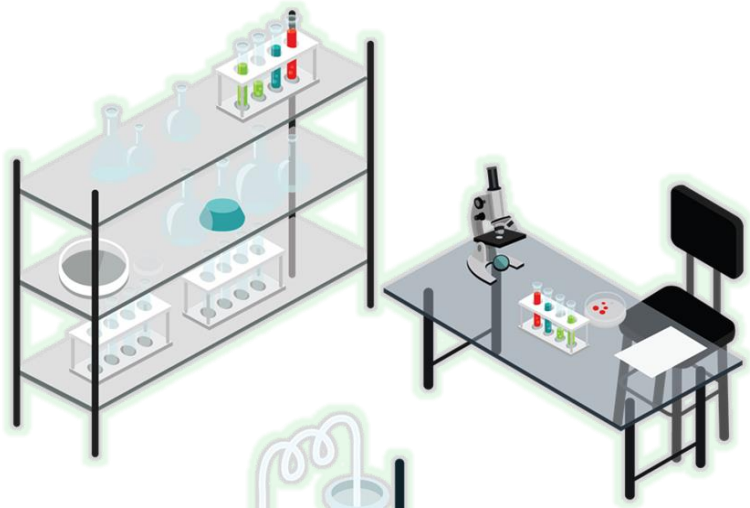


The Assessment Problem



Output

- A description of agent's working:
  - A list of capabilities.
  - An interpretable description of each capability.



Black-Box AI

- Arbitrary internal implementation
- Comes with a Trained Policy

```
at(p0,cell_6_3)
clear(cell_0_0)
door_at(cell_9_2)
next_to(m0)
alive(m0)
key_at(9_4)
```

[Input]
Concepts that the user understands

Personalized AI-Assessment Module (AAM)

Black-Box AI

Arbitrary internal implementation

Doesn't know user's vocabulary

```
at(p0,cell_6_3)
clear(cell_0_0)
door_at(cell_9_2)
next_to(m0)
alive(m0)
key_at(9_4)
```

[Input]
Concepts that the user understands

Personalized AI-Assessment Module (AAM)

Query

Response

simulator

Black-Box AI

Arbitrary internal implementation

Doesn't know user's vocabulary

[Input]
Concepts that the user understands

[Output]
User-Interpretable description of Black-Box AI's capabilities

Personalized AI-Assessment Module (AAM)

This Work

Query

Response

simulator

Black-Box AI

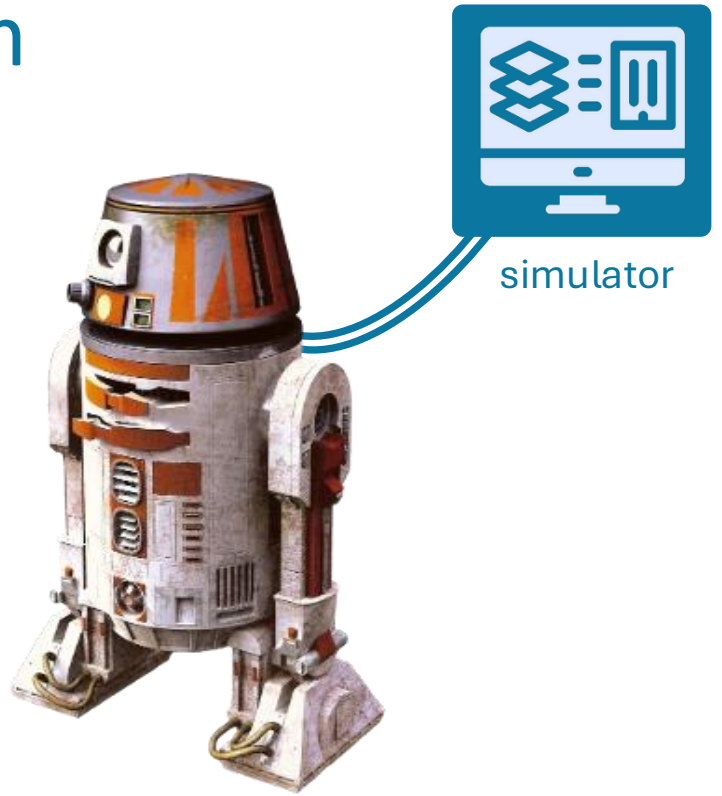
Arbitrary internal implementation

Doesn't know user's vocabulary

The Assessment Problem

Black-Box (Taskable) AI

- Can be connected to a simulator.
- Can have arbitrary internal implementation.
- Does not know input vocabulary.



Black-Box AI

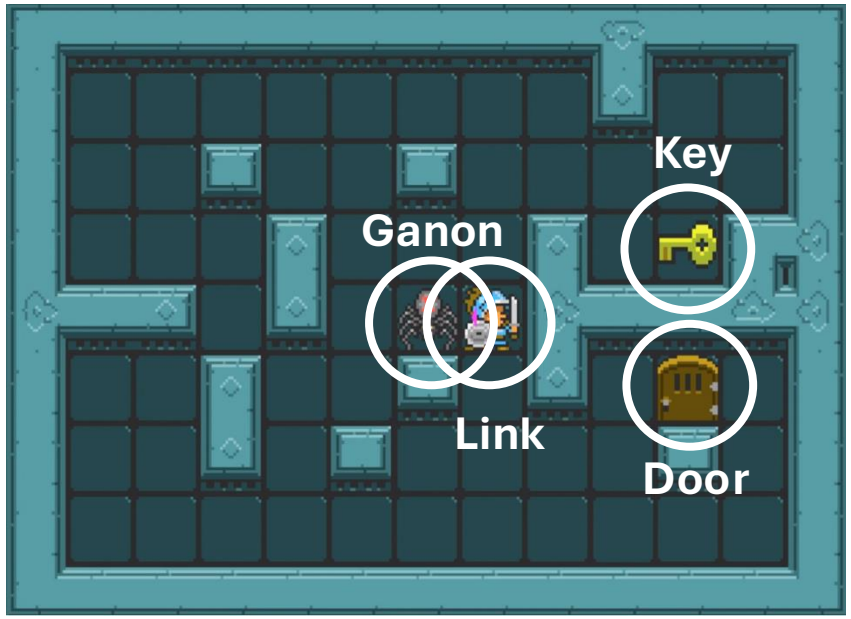
Input

- Predicates (User vocabulary)
  - With their evaluation functions

Output

- A description of agent's working:
  - A list of capabilities.
  - An interpretable description of each capability.

Probabilistic Planning Domain Definition Language




```
(:objects
    loc1 loc2 loc3 - location
    key1 - key
    door1 - door
)

(:init
    (player_at loc1)
    (has_key key1)
    (door_adjacent loc2 door1)
    (not (door_open door1))
)

(:goal
    (and (door_open door1)
         (not (game-over))))
```

Probabilistic Planning Domain Definition Language

```
(:action open-door
   :parameters (?l1)
   :precondition (and
      (has_key)
      (player_at ?l1)
      (door_adjacent ?l1))
   :effect (probabilistic
      0.95 (and (door_open))
      0.05 (and (not (has_key))
                (game-over))
)
```

Precondition: This condition must be true for this action to execute

Effect: This is a set of conditions, one of which becomes true when this action is executed

Probabilities: Each set of effect has an associated probability with which that effect set is executed

Interpretable: Easily Convertible to Natural Language

```
(:action open-door
  :parameters (?l1)
  :precondition (and
     (has_key)
     (player_at ?l1)
     (door_adjacent ?l1))
  :effect (probabilistic
     0.95 (and (door_open))
     0.05 (and (not(has_key))
               (game-over))
)
```

The player can open the door when in location ?l1 if:

- It has the key
- The player is at location ?l1
- The door is adjacent to location ?l1

After executing that capability:

- With 95% probability, the door will open
- With 5% probability, the player will not have the key and the game will be over

Deterministic and Stationary Setting

Input

- Predicates (User vocabulary)
  - With their evaluation functions
- List of capabilities.

Output

- PDDL-like description of each capability.

Siddharth Srivastava

Shashank R Marpally

Asking the Right Questions: Learning Interpretable Action Models Through Query Answering

Pulkit Verma, Shashank Rao Marpally, Siddharth Srivastava

In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, 2021 (CORE Ranking: A*)

Assumptions

- User's vocabulary matches simulator's vocabulary.
- Black-Box AI provides a list of capabilities.
- Stationary agent model.
- Deterministic environment.
- Fully observable setting.

Exponential Search for Learning Correct Description

- Consider the following 4 predicates/concepts:
  - `(has_key)`
  - `(door_open)`
  - `(door_adjacent ?x)`
  - `(player_at ?x)`
- Consider just one capability: `(open-door ?x)`
- $9^{|C|\times|P|}$ = $9^{1\times4}$=6561 possible models (Assuming deterministic models/ descriptions, i.e., no probabilities).

```
(:action open-door
  :parameters (?l1)
  :precondition (and
    (+/-/∅)(has_key)
    (+/-/∅)(door_open)
    (+/-/∅)(door_adjacent ?l1)
    (+/-/∅)(player_at ?l1))
  :effect (and
    (+/-/∅)(has_key)
    (+/-/∅)(door_open)
    (+/-/∅)(door_adjacent ?l1)
    (+/-/∅)(player_at ?l1))
```

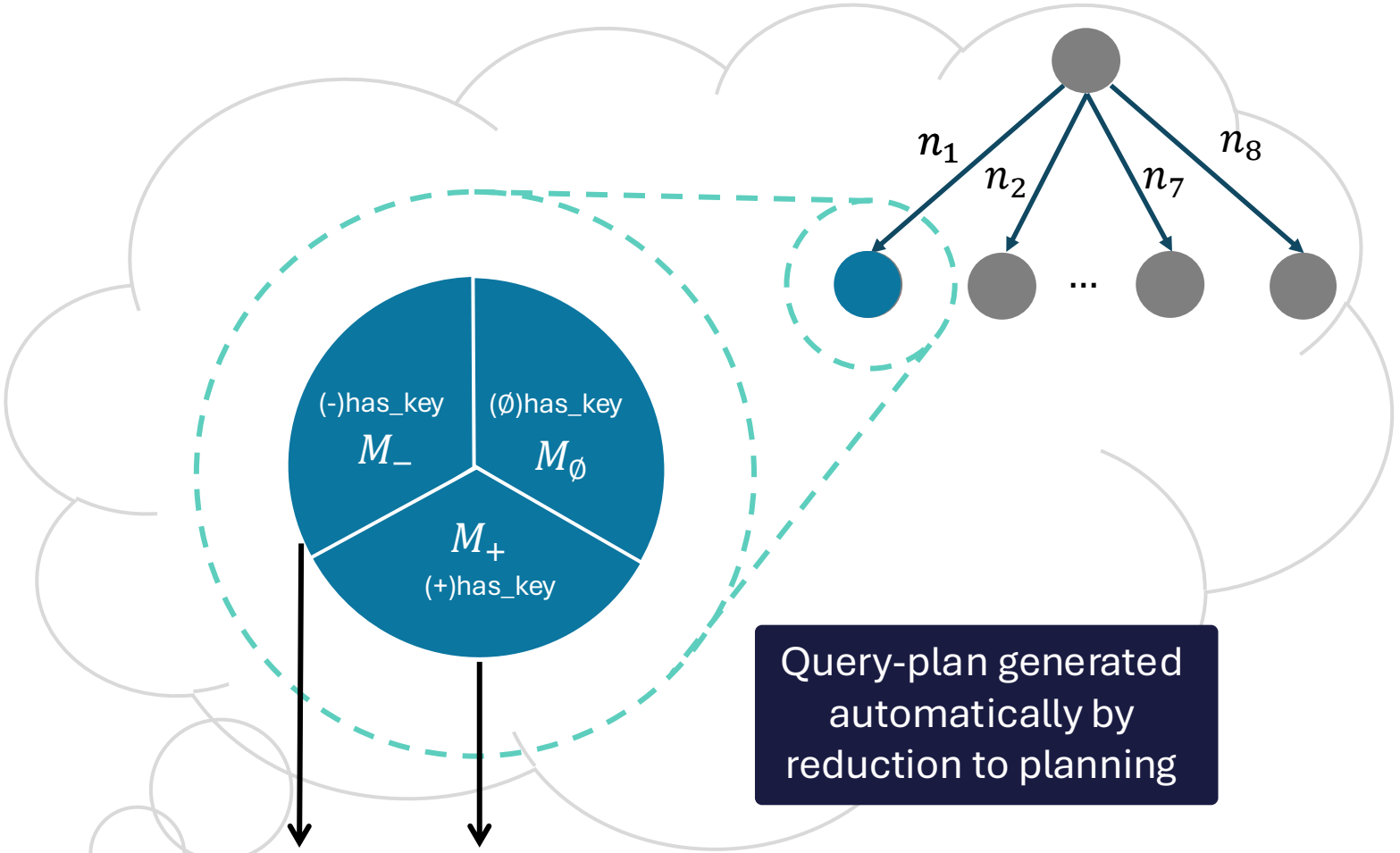
Simple Queries

	Plan Outcome Queries	State Reachability Query
Query	In state s_I, what will happen if you execute the plan $\pi = \langle c_1, \ldots, c_n \rangle$?	Can you go from state s_I to state s_F?
Response	I can execute first ℓ steps of the plan, ending up in state s_F.	Yes / No.

- How to generate the queries?
- How to use the responses to generate models?

We have a reduction that converts this to a planning problem, so it automatically generates queries.

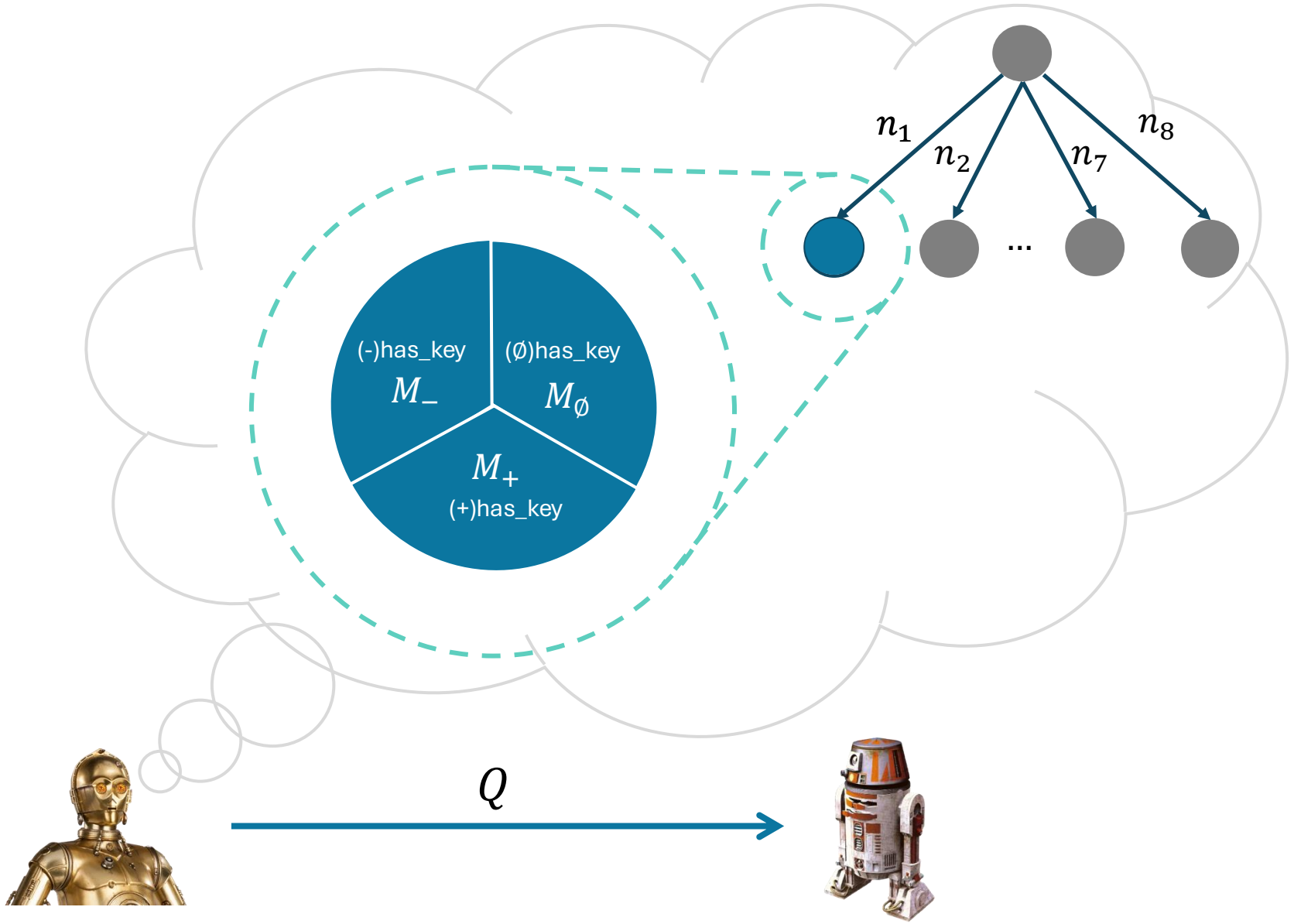
Hierarchical Query Synthesis



```
(:action open-door
  :parameters (?l1)
  :precondition (and
n1 (+/-/∅)(has_key)
n2 (+/-/∅)(door_open)
n3 (+/-/∅)(door_adjacent ?l1)
n4 (+/-/∅)(player_at ?l1))
  :effect (and
n5 (+/-/∅)(has_key)
n6 (+/-/∅)(door_open)
n7 (+/-/∅)(door_adjacent ?l1)
n8 (+/-/∅)(player_at ?l1))
```

[**Verma**, Marpally, Srivastava; AAAI '21]

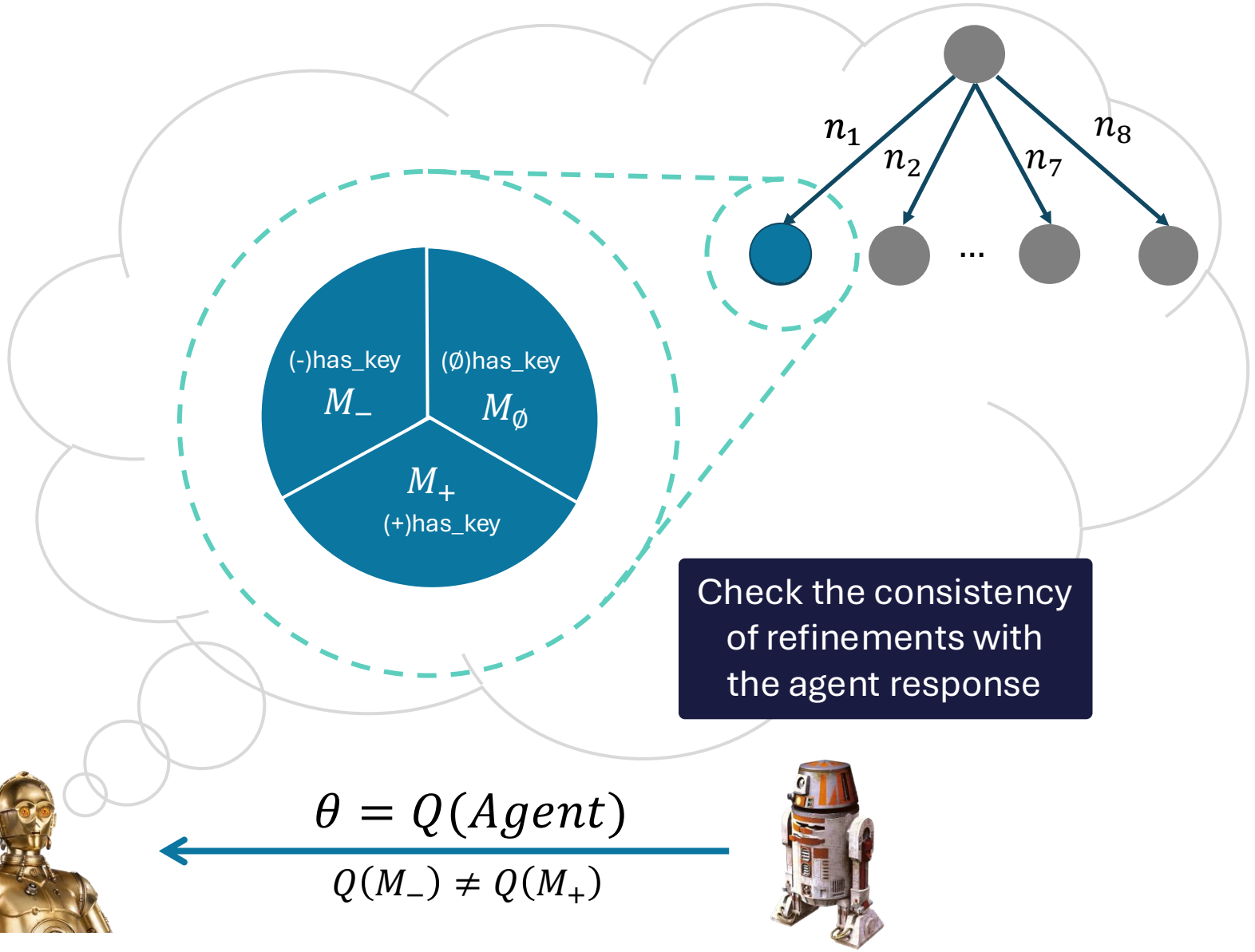
Hierarchical Query Synthesis




```
(:action open-door
  :parameters (?l1)
  :precondition (and
n1 (+/-/∅)(has_key)
n2 (+/-/∅)(door_open)
n3 (+/-/∅)(door_adjacent ?l1)
n4 (+/-/∅)(player_at ?l1))
  :effect (and
n5 (+/-/∅)(has_key)
n6 (+/-/∅)(door_open)
n7 (+/-/∅)(door_adjacent ?l1)
n8 (+/-/∅)(player_at ?l1))
```

[**Verma**, Marpally, Srivastava; AAAI '21]

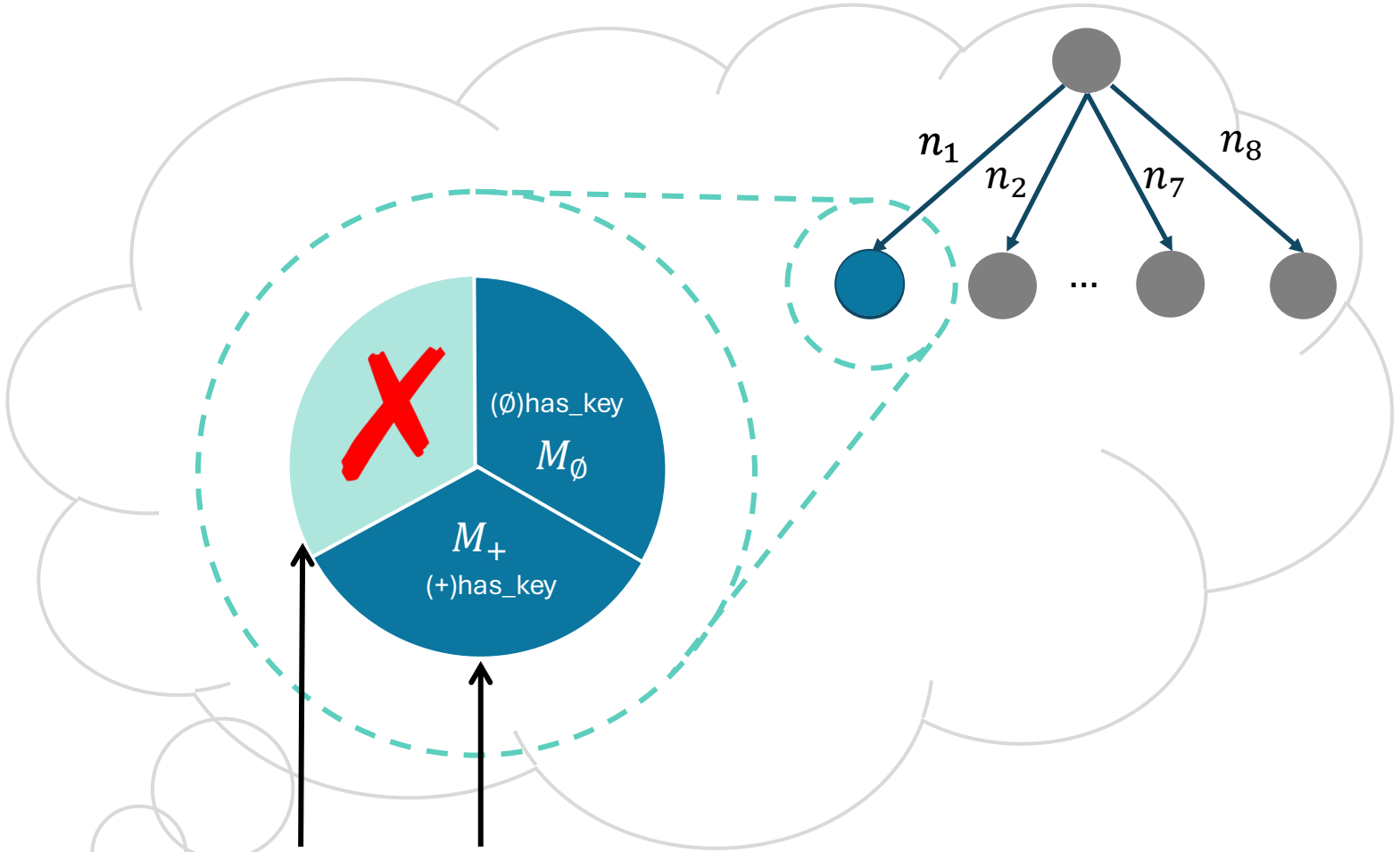
Hierarchical Query Synthesis




```
(:action open-door
  :parameters (?l1)
  :precondition (and
n1 (+/-/Ø)(has_key)
n2 (+/-/Ø)(door_open)
n3 (+/-/Ø)(door_adjacent ?l1)
n4 (+/-/Ø)(player_at ?l1))
  :effect (and
n5 (+/-/Ø)(has_key)
n6 (+/-/Ø)(door_open)
n7 (+/-/Ø)(door_adjacent ?l1)
n8 (+/-/Ø)(player_at ?l1))
```

[**Verma**, Marpally, Srivastava; AAAI '21]

Hierarchical Query Synthesis

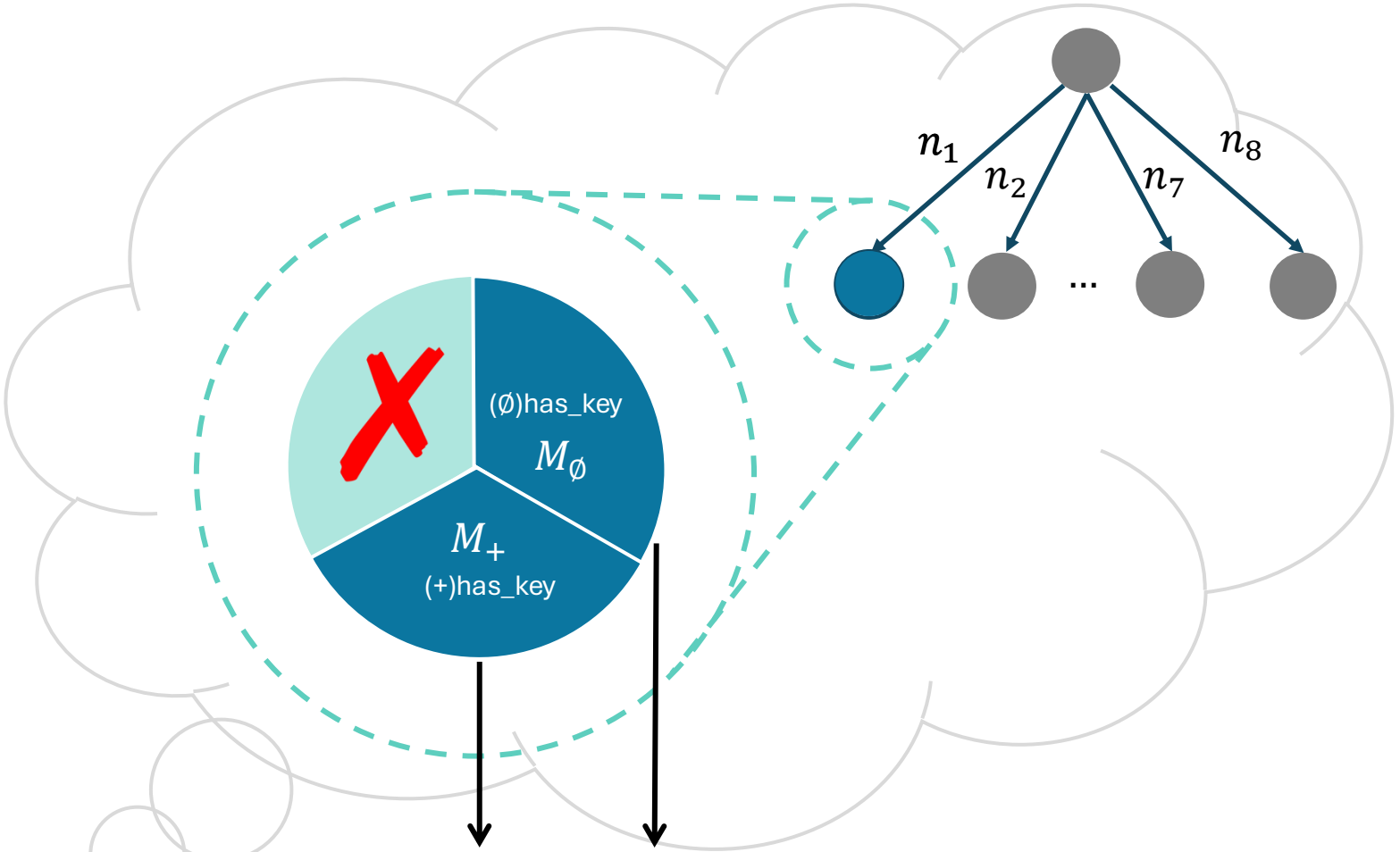



```
(:action open-door
  :parameters (?l1)
  :precondition (and
n1  (+/Ø)(has_key)
n2  (+/-/Ø)(door_open)
n3  (+/-/Ø)(door_adjacent ?l1)
n4  (+/-/Ø)(player_at ?l1))
  :effect (and
n5  (+/-/Ø)(has_key)
n6  (+/-/Ø)(door_open)
n7  (+/-/Ø)(door_adjacent ?l1)
n8  (+/-/Ø)(player_at ?l1))
```

Reject abstract model(s) that are not consistent with the agent

[**Verma**, Marpally, Srivastava; AAAI '21]

Hierarchical Query Synthesis

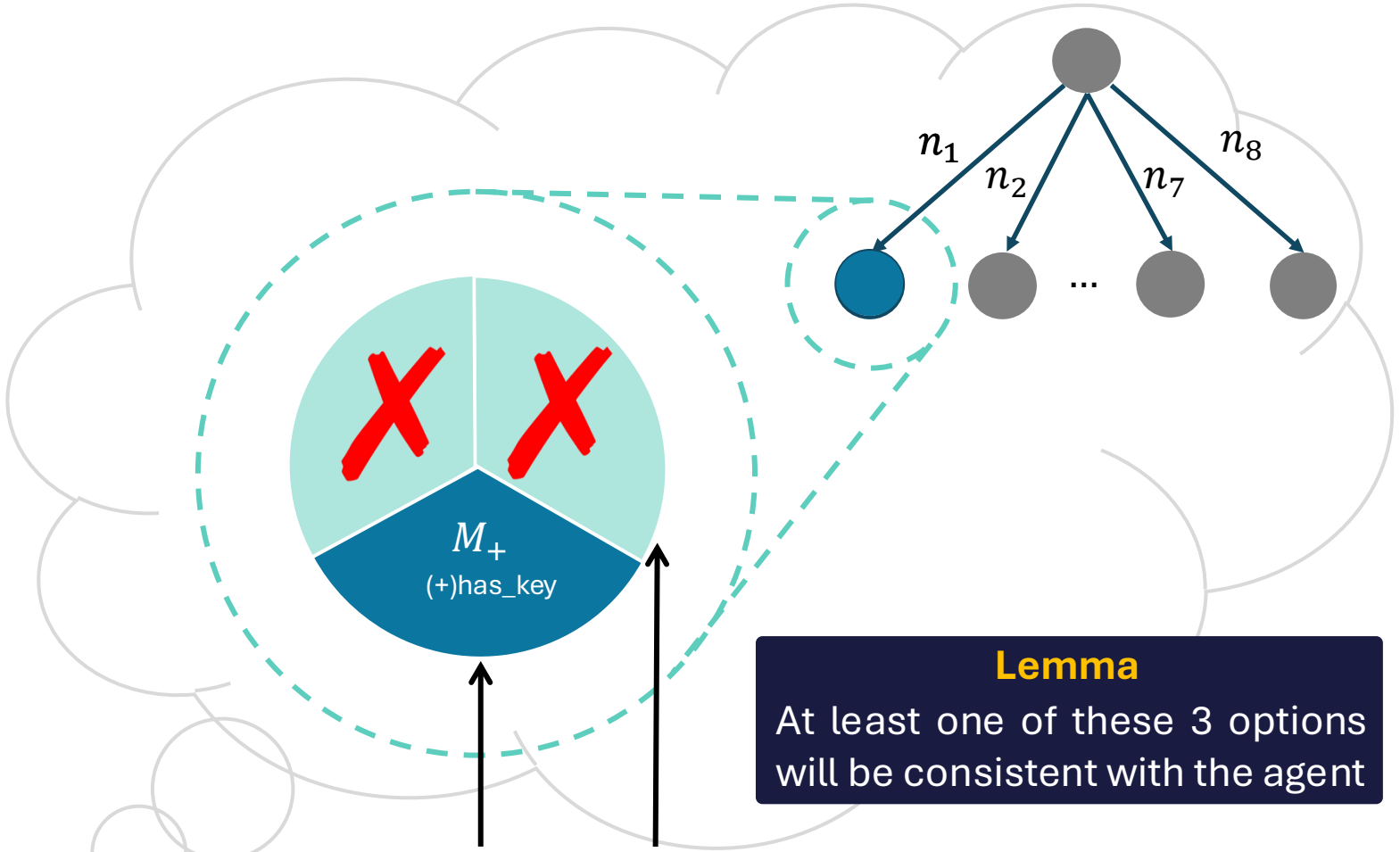


Generate a distinguishing query for these two abstract models

```
(:action open-door
  :parameters (?l1)
  :precondition (and
n1  (+/Ø)(has_key)
n2  (+/-/Ø)(door_open)
n3  (+/-/Ø)(door_adjacent ?l1)
n4  (+/-/Ø)(player_at ?l1))
  :effect (and
n5  (+/-/Ø)(has_key)
n6  (+/-/Ø)(door_open)
n7  (+/-/Ø)(door_adjacent ?l1)
n8  (+/-/Ø)(player_at ?l1))
```

[**Verma**, Marpally, Srivastava; AAAI '21]

Hierarchical Query Synthesis



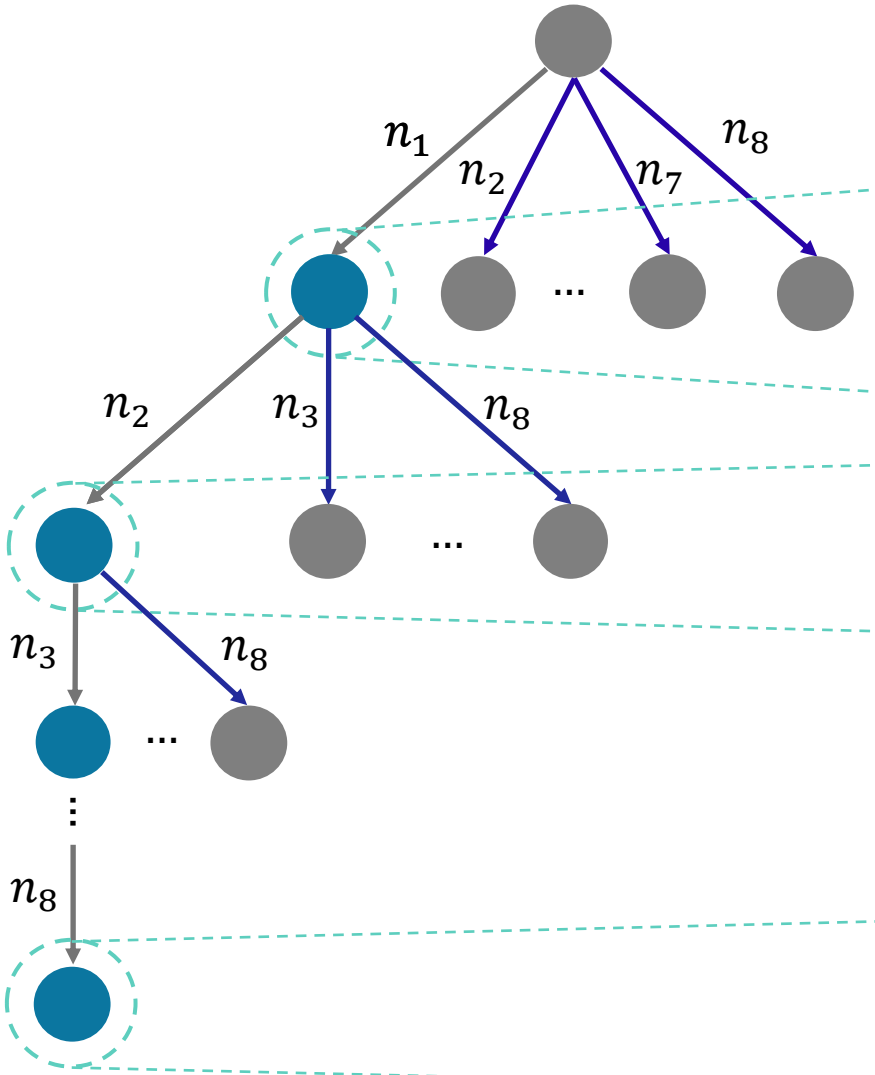
Lemma
At least one of these 3 options will be consistent with the agent

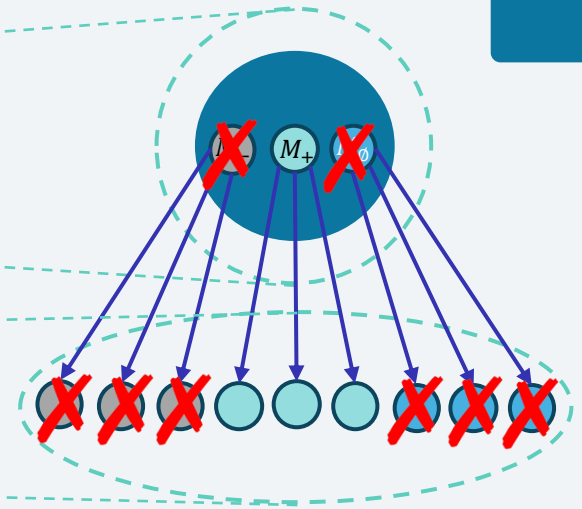
```
(:action open-door
  :parameters (?l1)
  :precondition (and
 n1      (+)(has_key)
 n2   (+/-/∅)(door_open)
 n3   (+/-/∅)(door_adjacent ?l1)
 n4   (+/-/∅)(player_at ?l1))
  :effect (and
 n5   (+/-/∅)(has_key)
 n6   (+/-/∅)(door_open)
 n7   (+/-/∅)(door_adjacent ?l1)
 n8   (+/-/∅)(player_at ?l1))
```

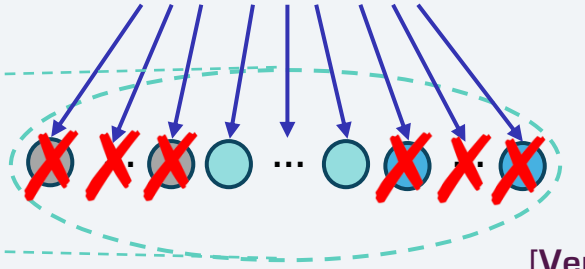
Reject the abstract model
that is not consistent with the agent

[**Verma**, Marpally, Srivastava; AAAI '21]

Hierarchical Query Synthesis







Key feature of the algorithm

Whenever we prune an abstract model, we prune a large number of concrete models.

Active Learning

[**Verma**, Marpally, Srivastava; AAAI '21]

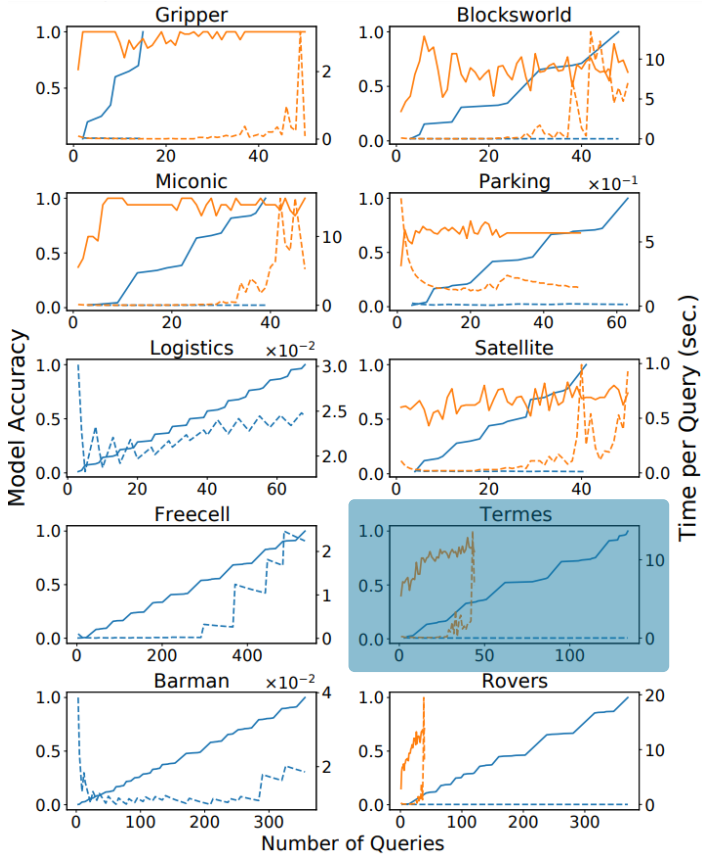
AAM learns Accurate Model with fewer Queries

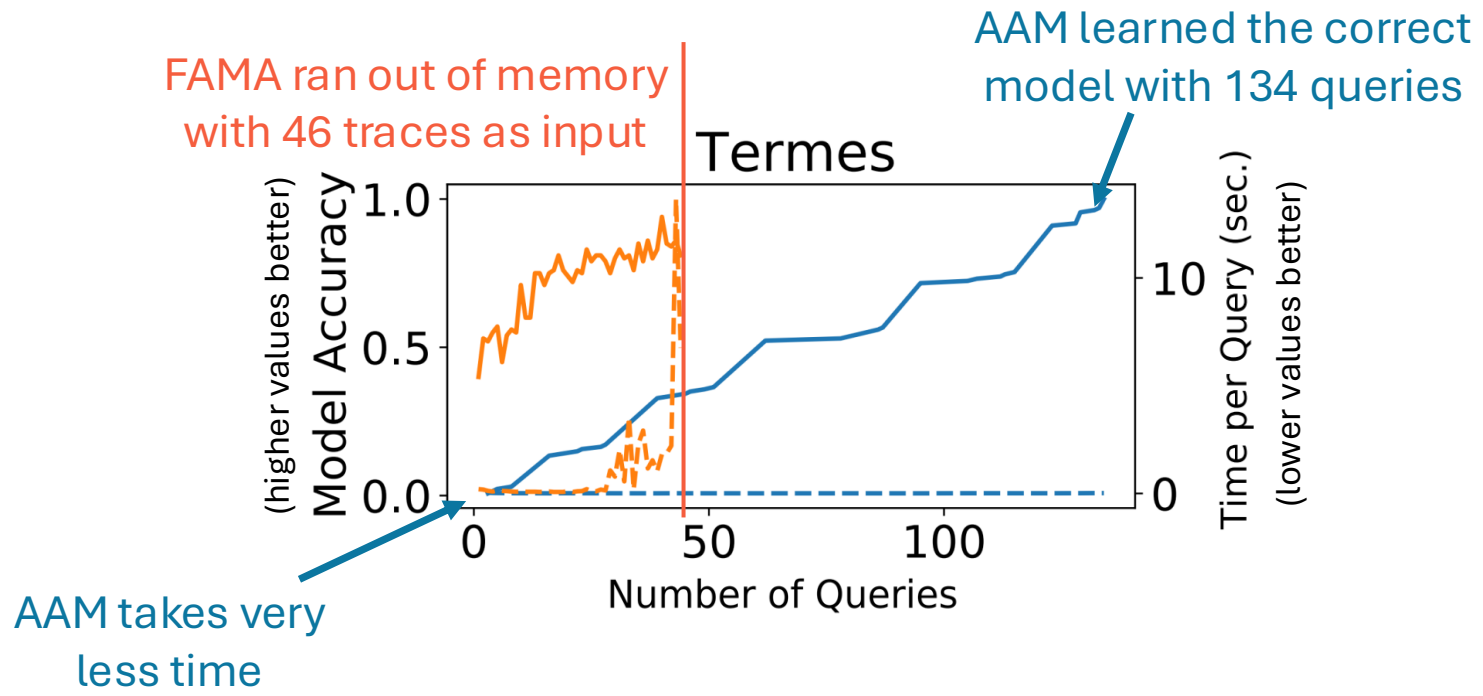
- Asses by learning the model and compare with ground truth.
- Baseline†: A passive learner (FAMA) that observes agent behavior

Random deterministic planning agent from IPC

Accuracy: AAM FAMA

Time: AAM FAMA





[**Verma**, Marpally, Srivastava; AAAI '21]

†Aineto, D.; Celorrio, S. J.; and Onaindia, E. 2019. *Learning Action Models With Minimal Observability*. Artificial Intelligence 275: 104–137.

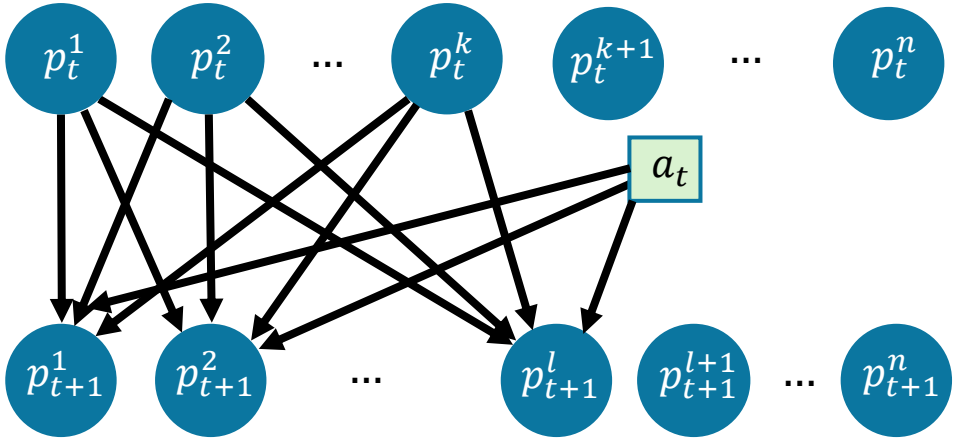
AAM learns Accurate Deterministic Models

- Theorem (*termination*) : The algorithm terminates after a finite number of iterations.

- Theorem (*soundness*): The resulting (set of) model(s) is(are) functionally equivalent to the ground truth model.

Causal Accuracy Analysis

- Use the framework for *Actual Causality*† to define the causal accuracy of the models that we learn.
- Explain theoretically why models learned using passive learners may not be causally accurate.
- Show that the models AAM learns are causally accurate†.



[**Verma**, Srivastava; 2024]

†According to AC2 definition of Actual Cause. Joseph Halpern. *Actual Causality*. MIT Press. 2016.

Stochastic and Stationary Setting

Input

- Predicates (User vocabulary)
  - With their evaluation functions
- List of capabilities.

Output

- PPDDL-like description of each capability.

Siddharth Srivastava

Rushang Karia

Autonomous Capability Assessment of Sequential Decision-Making Systems in Stochastic Settings

Pulkit Verma, Rushang Karia, Siddharth Srivastava

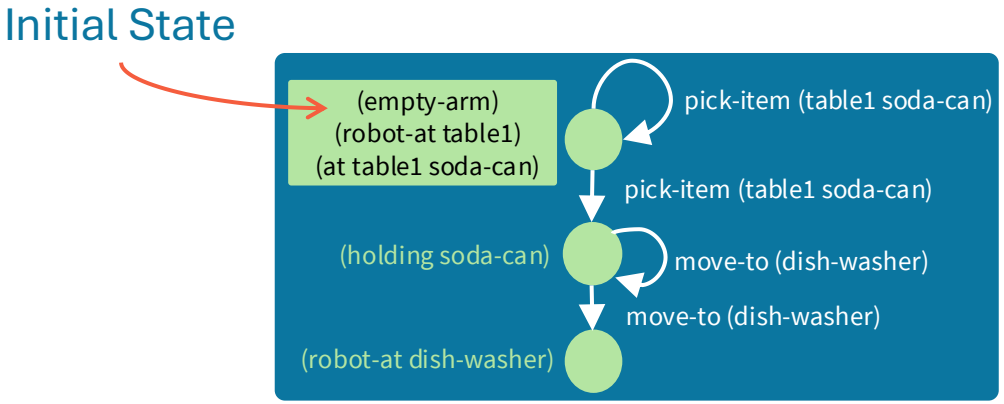
In Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems,, 2023

Assumptions

- User's vocabulary matches simulator's vocabulary.
- Black-Box AI provides a list of capabilities.
- Stationary agent model.
- ~~Deterministic~~ Stochastic environment.
- Fully observable setting.

Changes for Stochastic Settings

New Queries



Policy: Generated Autonomously by Reduction to Non-Deterministic Planning

What happens if you start in the given initial state and follow this partial policy?

Assumptions

- User's vocabulary matches simulator's vocabulary.
- Black-Box AI provides a list of capabilities.
- Stationary agent model.
- ~~Deterministic~~ Stochastic environment.
- Fully observable setting.

Changes for Stochastic Settings

Step 1: Learn a Non-Deterministic Model

```
(:action open-door
  :parameters (?l1)
  :precondition (and
    (+/-/∅)(has_key)
    (+/-/∅)(door_open)
    (+/-/∅)(door_adjacent ?l1)
    (+/-/∅)(player_at ?l1))
  :effect (oneof
    (and
      (+/-/∅)(has_key)
      (+/-/∅)(door_open)
      (+/-/∅)(door_adjacent ?l1)
      (+/-/∅)(player_at ?l1))
    (and
      (+/-/∅)(has_key)
      (+/-/∅)(door_open)
      (+/-/∅)(door_adjacent ?l1)
      (+/-/∅)(player_at ?l1))))
```

Apply Maximum Likelihood Estimation on the observed data (query responses)

Step 2: Convert to Probabilistic Model

```
(:action open-door
  :parameters (?l1)
  :precondition (and
    (+/-/∅)(has_key)
    (+/-/∅)(door_open)
    (+/-/∅)(door_adjacent ?l1)
    (+/-/∅)(player_at ?l1))
  :effect (probabilistic
    0.xx (and
      (+/-/∅)(has_key)
      (+/-/∅)(door_open)
      (+/-/∅)(door_adjacent ?l1)
      (+/-/∅)(player_at ?l1))
    0.yy (and
      (+/-/∅)(has_key)
      (+/-/∅)(door_open)
      (+/-/∅)(door_adjacent ?l1)
      (+/-/∅)(player_at ?l1))))
```

[**Verma**, Karia, Srivastava; NeurIPS '23]

Stochastic and Stationary Setting

Input

- Predicates (User vocabulary)
  - With their evaluation functions
- List of capabilities.

Output

- PPDDL-like description of each capability.

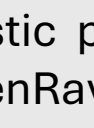
[**Verma**, Karia, Srivastava; NeurIPS '23]

Assumptions

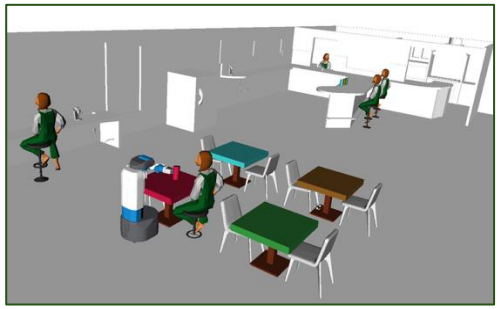
- User's vocabulary matches simulator's vocabulary.
- Black-Box AI provides a list of capabilities.
- Stationary agent model.
- ~~Deterministic~~ Stochastic environment.
- Fully observable setting.

AAM learns accurate models for Continuous Domains

- Use Task and Motion Planning (TMP) to convert actions into motion plans.
- Increase the time taken to learn the model.

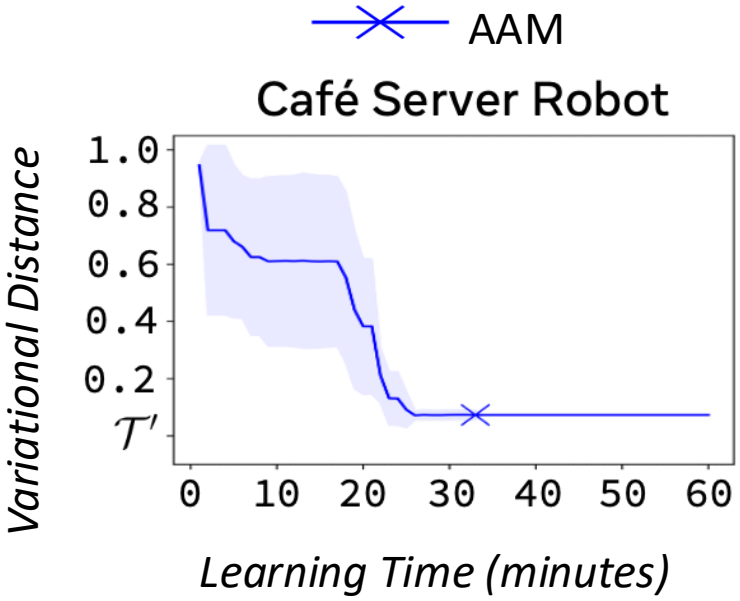


Probabilistic planning agent using OpenRave and TMP



	x	y	z	θ	φ	ψ
robot-base	1.0	-3.2	4.7	0.9	1.3	3.1
soda-can1	6.0	-2.8	3.5	8.3	6.7	9.2
⋮					⋮	
table4	-2.1	4.1	1.9	3.7	9.5	4.8

(empty-arm)
(robot-at table1)
(at table1 soda-can)



[**Verma**, Karia, Srivastava; NeurIPS '23]

AAM learns Accurate Probabilistic Models

- Theorem (*soundness and completeness*):
  The intermediate non-deterministic model (after step 1) is sound and complete w.r.t. the ground truth model.

- Theorem (*probabilistic correctness*):
  The resulting probabilistic model is correct w.r.t. the ground truth model.

Discovering Capabilities

Input

- Predicates (User vocabulary)
  - With their evaluation functions
- Samplers: high-level state to low-level state.
- Low-level state transitions.

Output

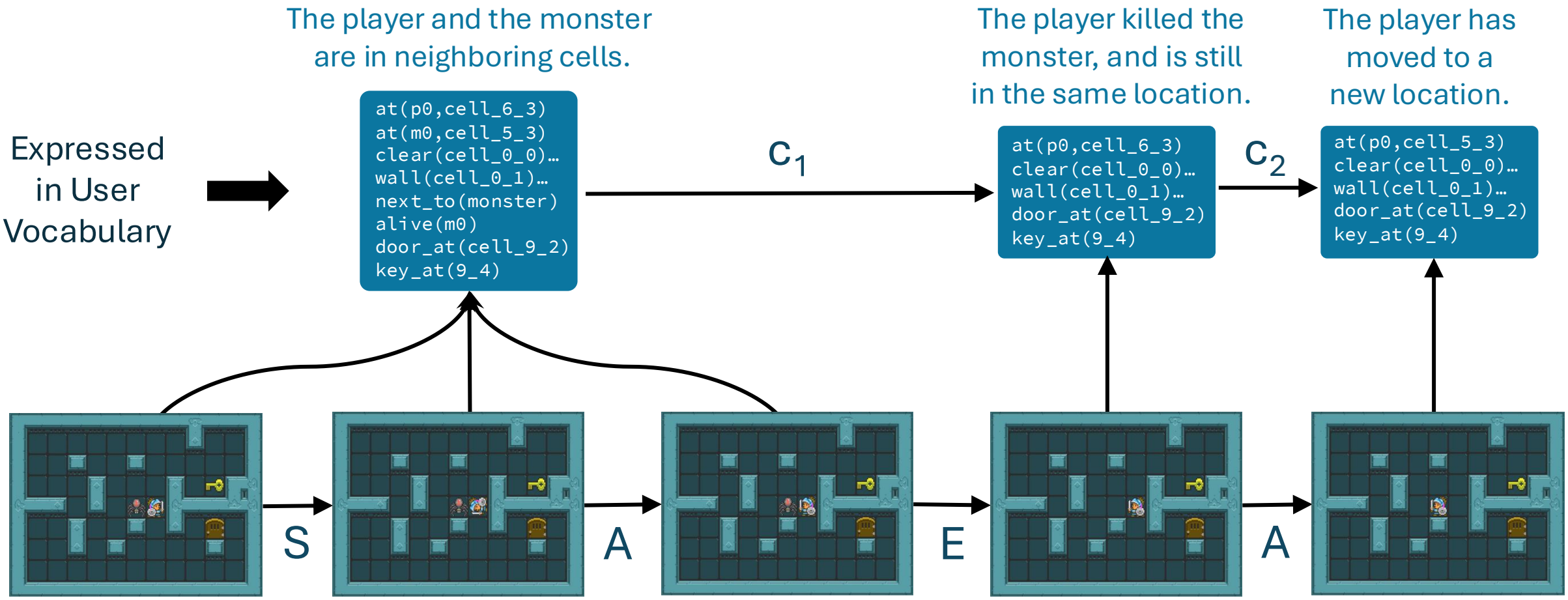
- List of capabilities.
- PDDL-like description of each capability.

[**Verma**, Marpally, Srivastava; KR '22]

Assumptions

- ~~User's vocabulary matches simulator's vocabulary~~.
- Black-Box AI provides a list of ~~capabilities~~. transitions.
- Stationary agent model.
- Deterministic environment.
- Fully observable setting.

Discovering Capabilities using Input Predicates as Abstractions



[**Verma**, Marpally, Srivastava; KR '22]

Example of a Learned Capability Description

```
(:capability c4
 :parameters (?player1 ?cell1
   ?monster1 ?cell2)
 :precondition
  (and (alive ?monster1)
    (at ?player1 ?cell1)
    (at ?monster1 ?cell2)
    (next_to ?monster1))
 :effect
  (and (clear ?cell2)
   (not(alive ?monster1))
   (not(at ?monster1 ?cell2))
   (not(next_to ?monster1))))
```

Position of Link has not changed

Ganon is not at its previous location

Ganon is not alive anymore

Link is not next to Ganon

This capability is: “Defeat Ganon”

[**Verma**, Marpally, Srivastava; KR ‘22]

User Study Setup to Verify Interpretability

Preconditions

Effects

4. **Capability C4**:
The *player* can execute this capability when:

- The *monster* is not defeated.
- The *player* is in *cell1*.
- The monster is in *cell2*.
- The player is in a cell adjacent to the *monster*.

After the *player* executes this capability:

- *Cell2* is empty.
- The *monster* is defeated.
- The *monster* is not in *cell2*.
- The player is not in a cell adjacent to the *monster*.

Question 4 of 12:
Select the phrase that best summarizes the capability **C4**? We will use your response while referring to this capability **C4** later in the survey.

Go next to Door
Go next to Ganon
Go next to Key
Go next to Wall
Defeat Ganon
Break Key
Pick Key
Open Door

Possible options to choose from

[Capability Description Example]

Keystroke Description

W: Pressing this key does the following:

- If Link is facing up and there is no wall, door, or key in the cell above, then Link moves to the cell above.
- If there is a wall, door, or key in the cell above Link, then Link stays in the same cell.
- If Link is facing Left, Right, or Down before pressing W, then Link faces up but stays in the same cell.

Question 1 of 11:
Select the phrase that best summarizes pressing **W**? We will use your response while referring to this key **W** later in the survey.

Up
Down
Left
Right
Interact

Possible options to choose from

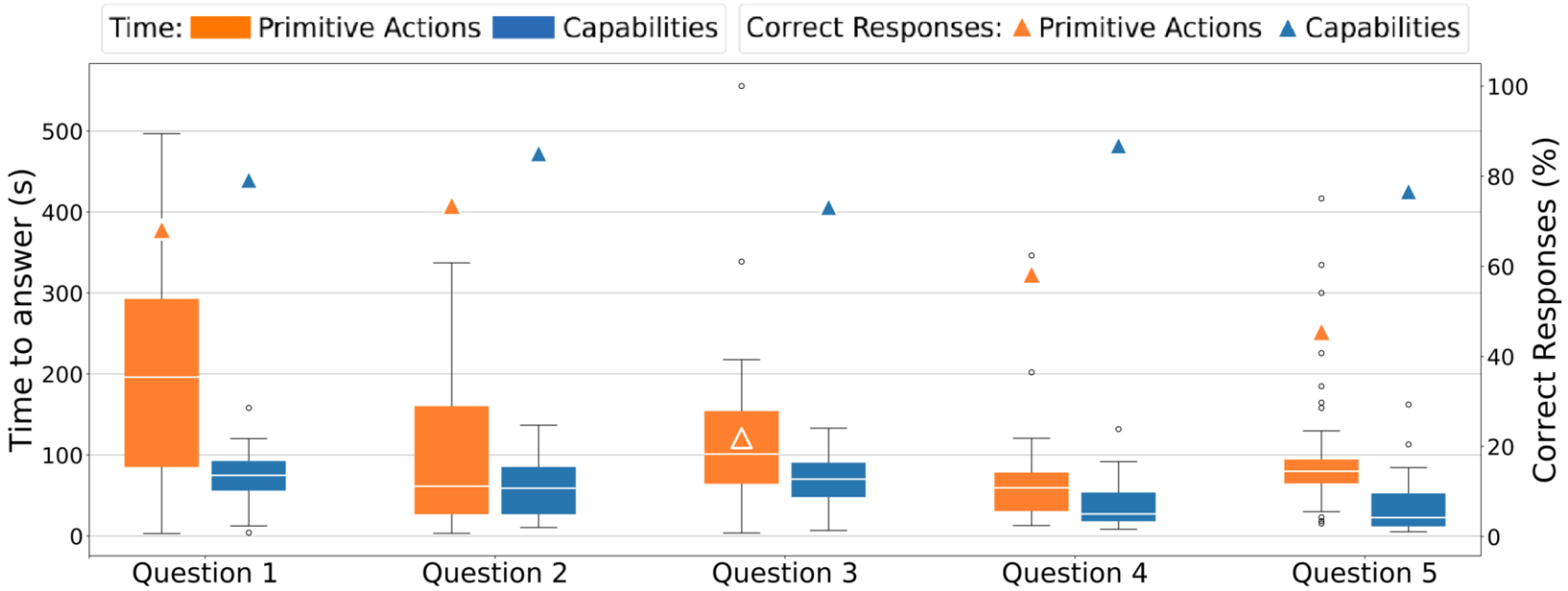
[Functionality Description Example]

[**Verma**, Marpally, Srivastava; KR '22]

Utility of Discovered Capability Descriptions

If Link starts in the state shown below:

Which sequence of actions can Link take to reach the state shown below:



[**Verma**, Marpally, Srivastava; KR '22]

Differential Assessment

Input

- Initial model of the AI system.
  - Predicates (User vocabulary)
    - With their evaluation functions
  - List of capabilities.
- Observations collected from AI system.

Output

- Updated PDDL-like description of each capability.

Siddharth Srivastava

Rashmeet K Nayyar

Differential Assessment of Black-Box AI Agents

Rashmeet K Nayyar*, *Pulkit Verma**, *Siddharth Srivastava*

In Proceedings of Thirty-Sixth AAAI Conference on Artificial Intelligence, 2022

Assumptions

- User's vocabulary matches simulator's vocabulary.
- Black-Box AI provides a list of capabilities.
- Adaptive
- ~~Stationary~~ agent model.
- Deterministic environment.
- Fully observable setting.

Discovering Capabilities in Stochastic Settings

Input

- Predicates (User vocabulary)
  - With their evaluation functions
- Samplers: high-level state to low-level state.
- Low-level state transitions.

Output

- List of capabilities.
- PDDL-like description of each capability.


Daniel Bramblett


Rushang Karia


Adrian Ciotinga


YooJung Choi


Siddharth Srivastava

Autonomous Assessment of Generalizability of AI Agent Capabilities

D Bramblett, R Karia, A Ciotinga, P Verma, Y Choi, & S Srivastava

In ICAPS GenPlan Workskhop, 2026

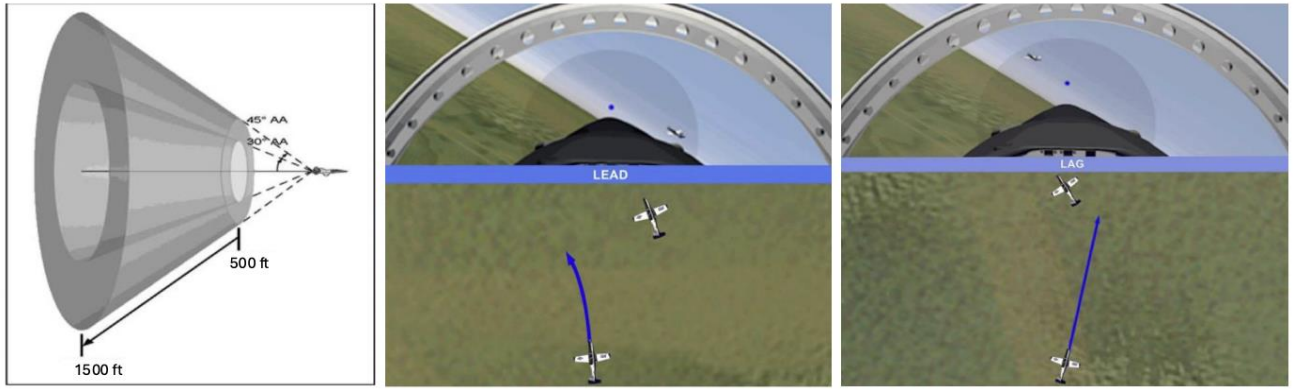
Assumptions

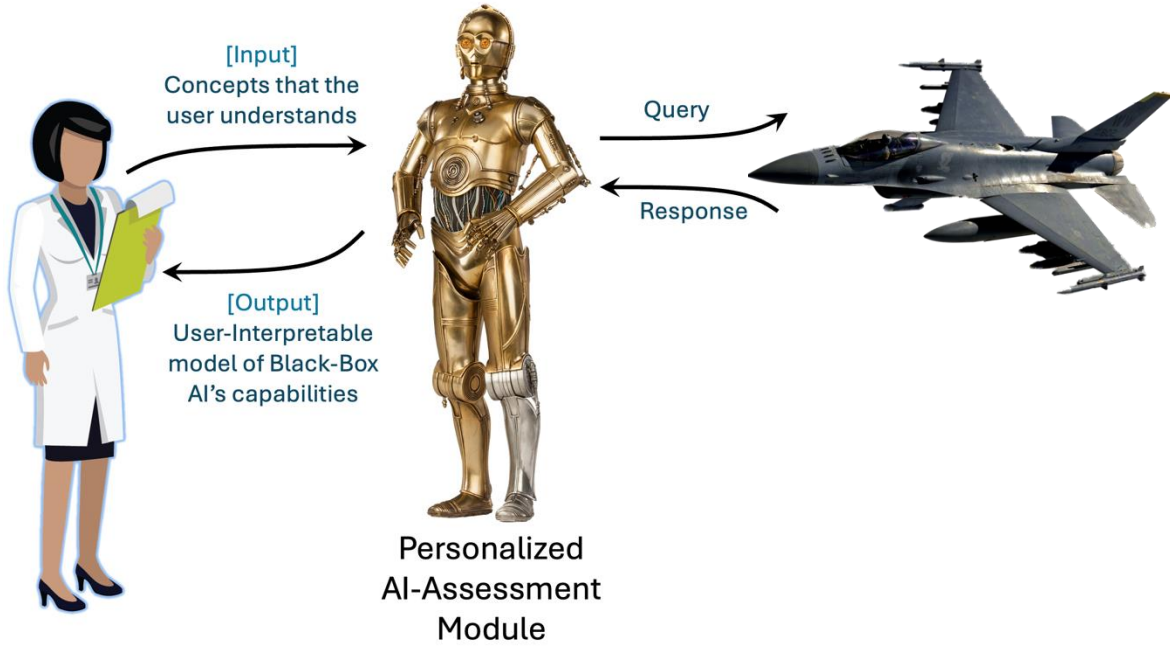
- ~~User's vocabulary matches simulator's vocabulary~~.
- Black-Box AI provides a list of ~~capabilities~~. transitions.
- Adaptive ~~Stationary~~ agent model.
- Stochastic ~~Deterministic~~ environment.
- Fully observable setting.

Real World Setting

- Flying Extended Trail
- Human must collaborate with the AI-powered aircraft.

- How can we explain the aircraft's policy to the human in an interpretable way?





Josh Rountree | Oswin So | Chuchu Fan | Julie A. Shah

[Rountree, **Verma**, So, Fan, Shah; openreview.net/forum?id=uEyJmixFiA]

Human Understanding of Deep Neural Policies

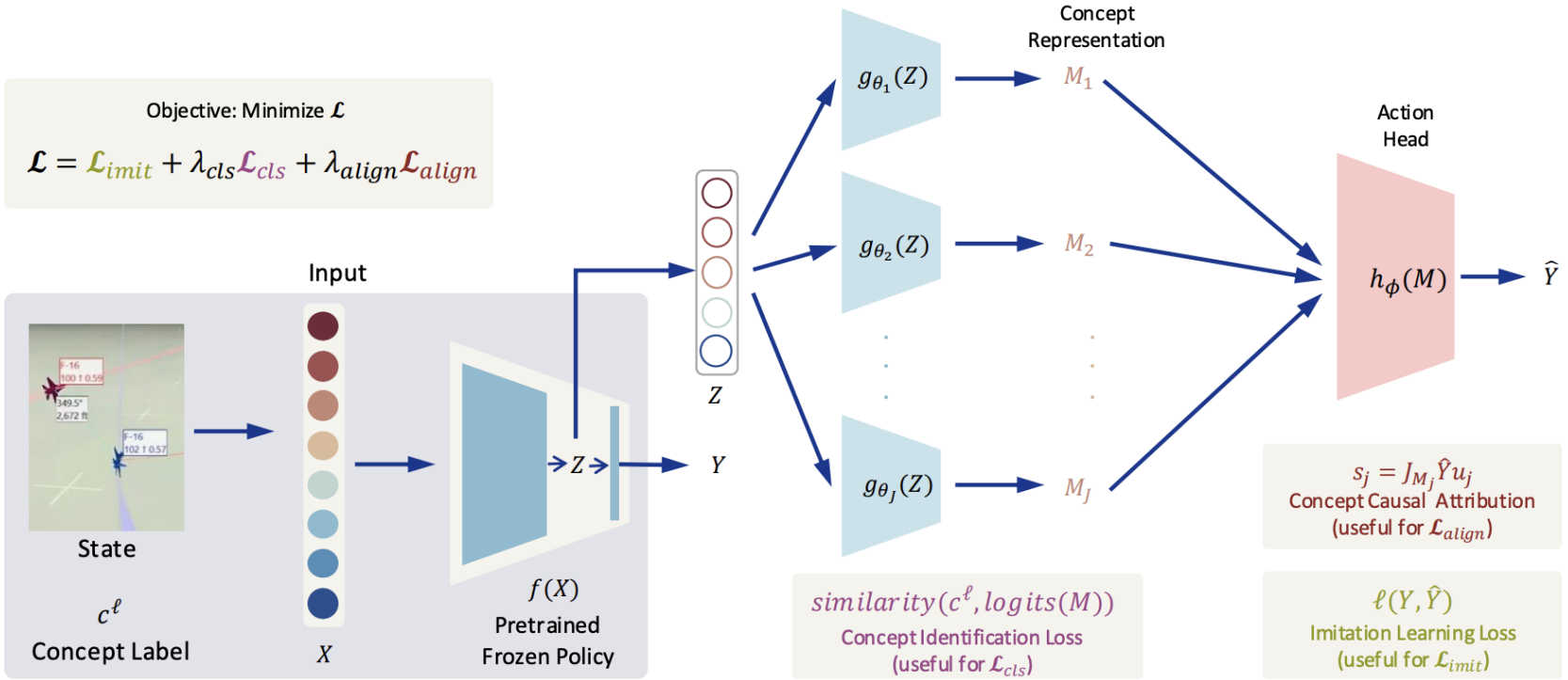
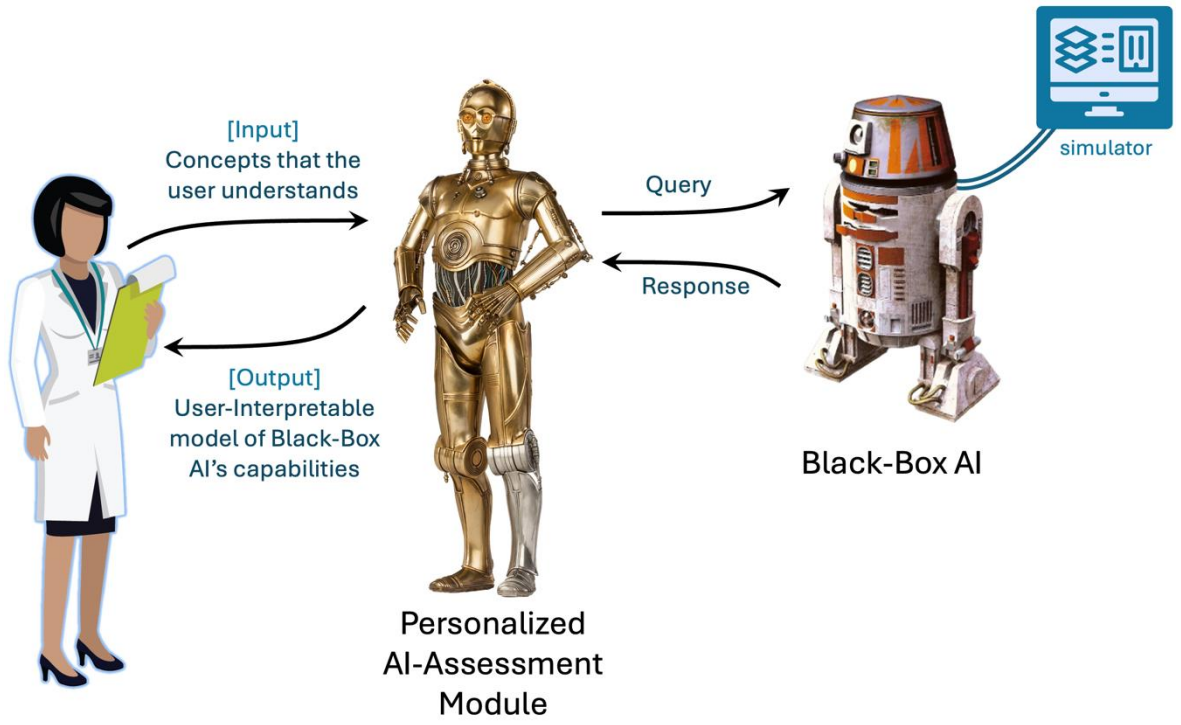


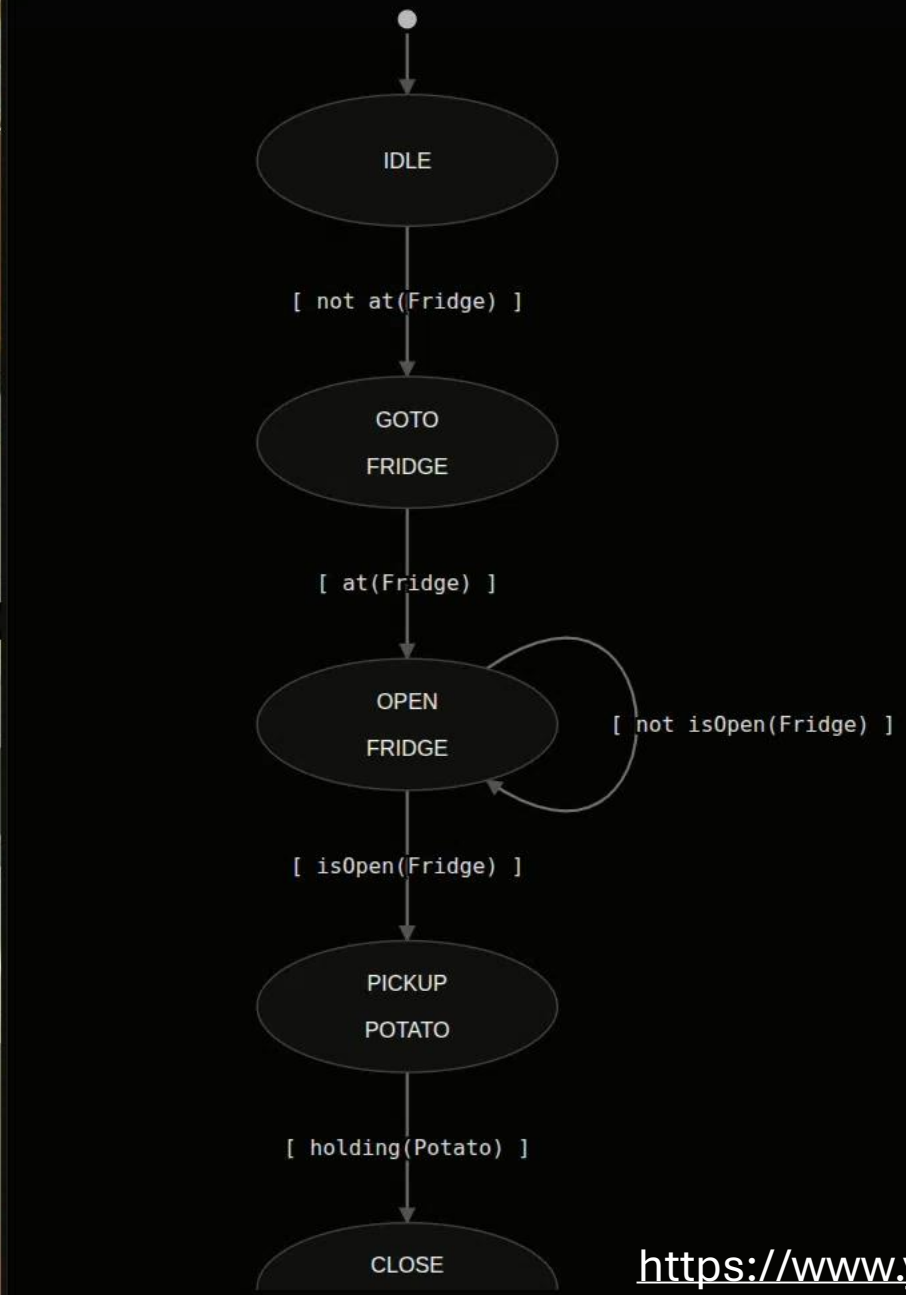
Figure 1: **CCW-Net Architecture.** CCW-Net wraps a frozen backbone f with a concept encoder g_θ that transforms latents Z into human-interpretable vector concepts M, and a differentiable head h_ϕ that maps concepts to wrapper actions $\hat{Y}$. CCW-Net estimates per-concept causal targets IE_j from expert trajectories, computes local concept-to-action sensitivities $s_j = J_{M_j}\hat{Y}\,u_j$, and aligns them with a masked cosine loss. Vector concepts encode both the strengths and context-dependent expression of each concept, enabling faithful, human-interpretable explanations while preserving task performance.

[Rountree, **Verma**, So, Fan, Shah; openreview.net/forum?id=uEyJmixFiA]

Alternate Interpretable Representations

- Representations beyond PDDL.
  - LTL – Linear Temporal Logic
  - Decision Trees
  - Finite State Machines
- How interpretable each representation is?



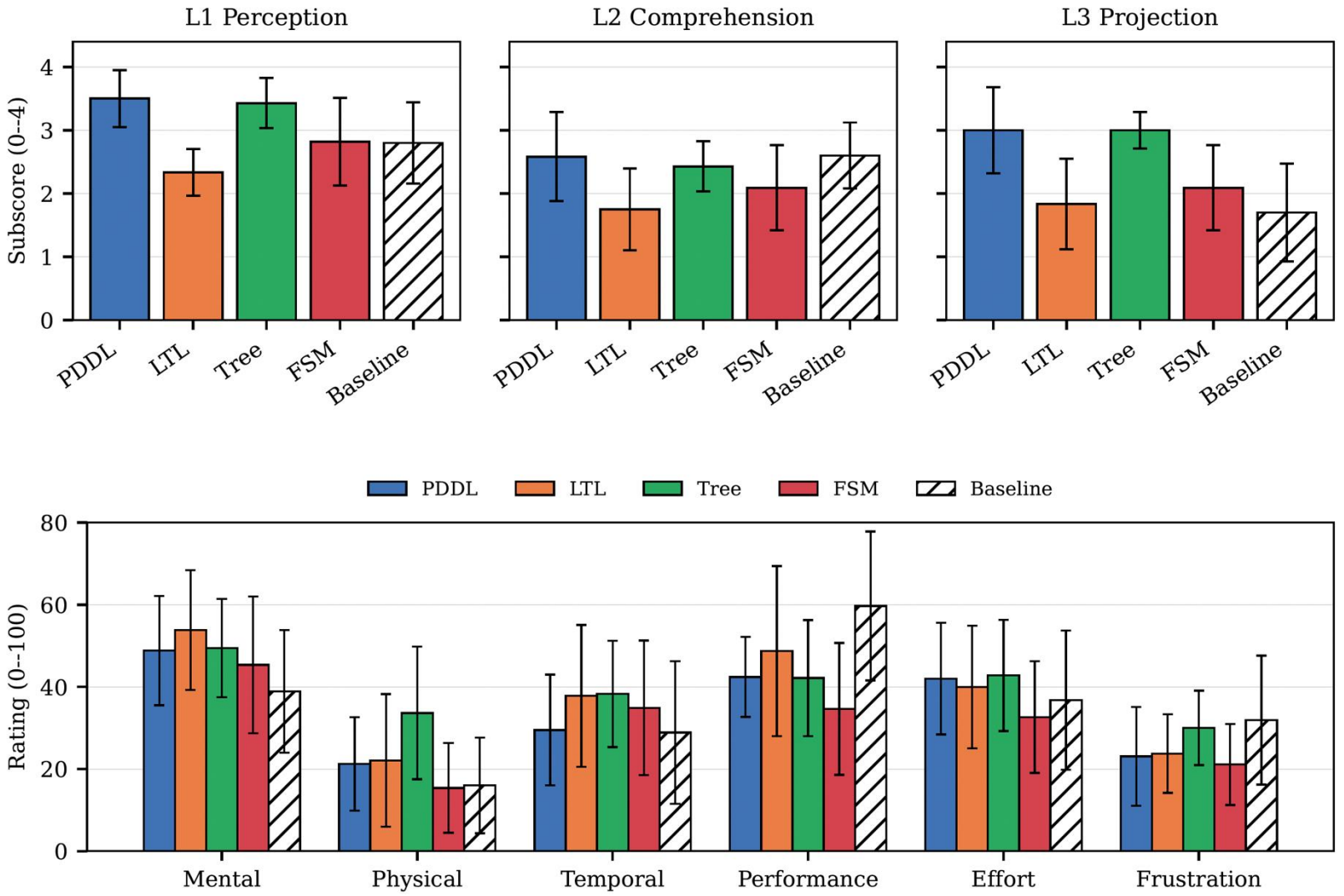


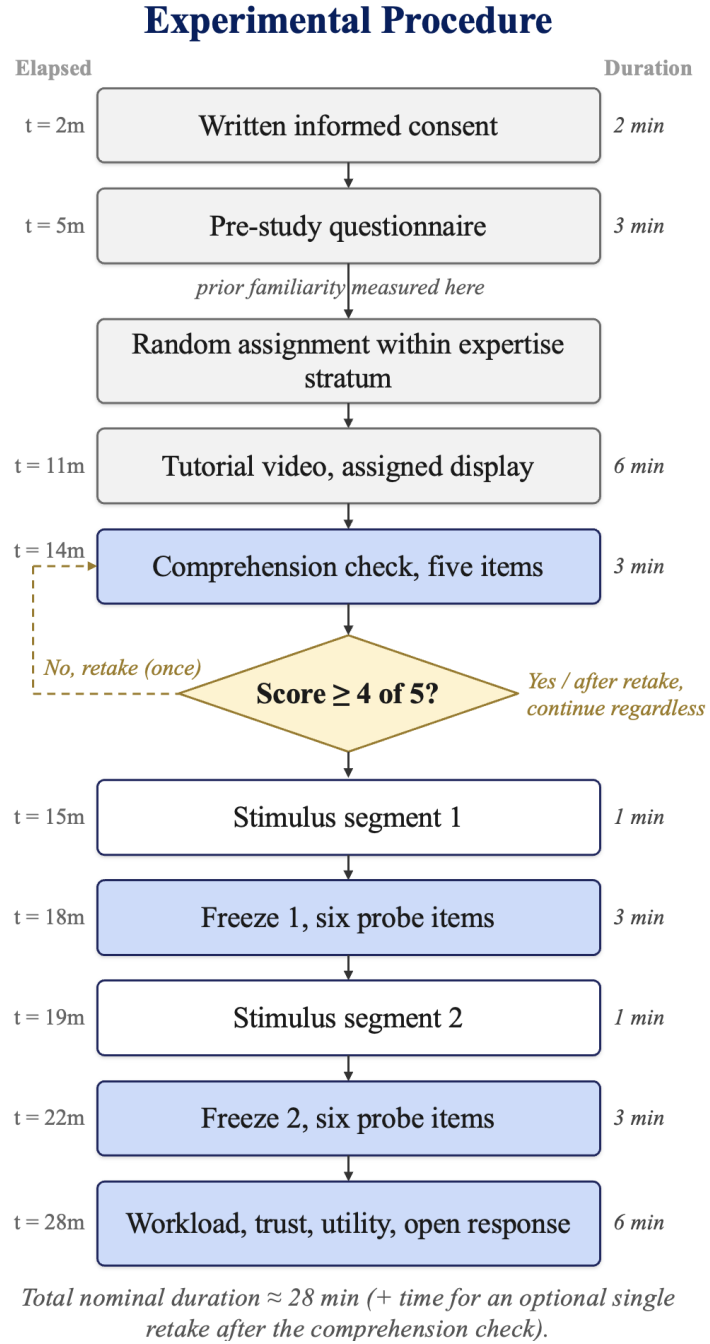
World State

- at(CounterTop)
- hand-empty
- in(Potato, Fridge)
- on(Pan, CounterTop)

https://www.youtube.com/watch?v=liek8bEj1-E

Comparing Interpretable Representations





Lifelong Learning with Human-in-the Loop

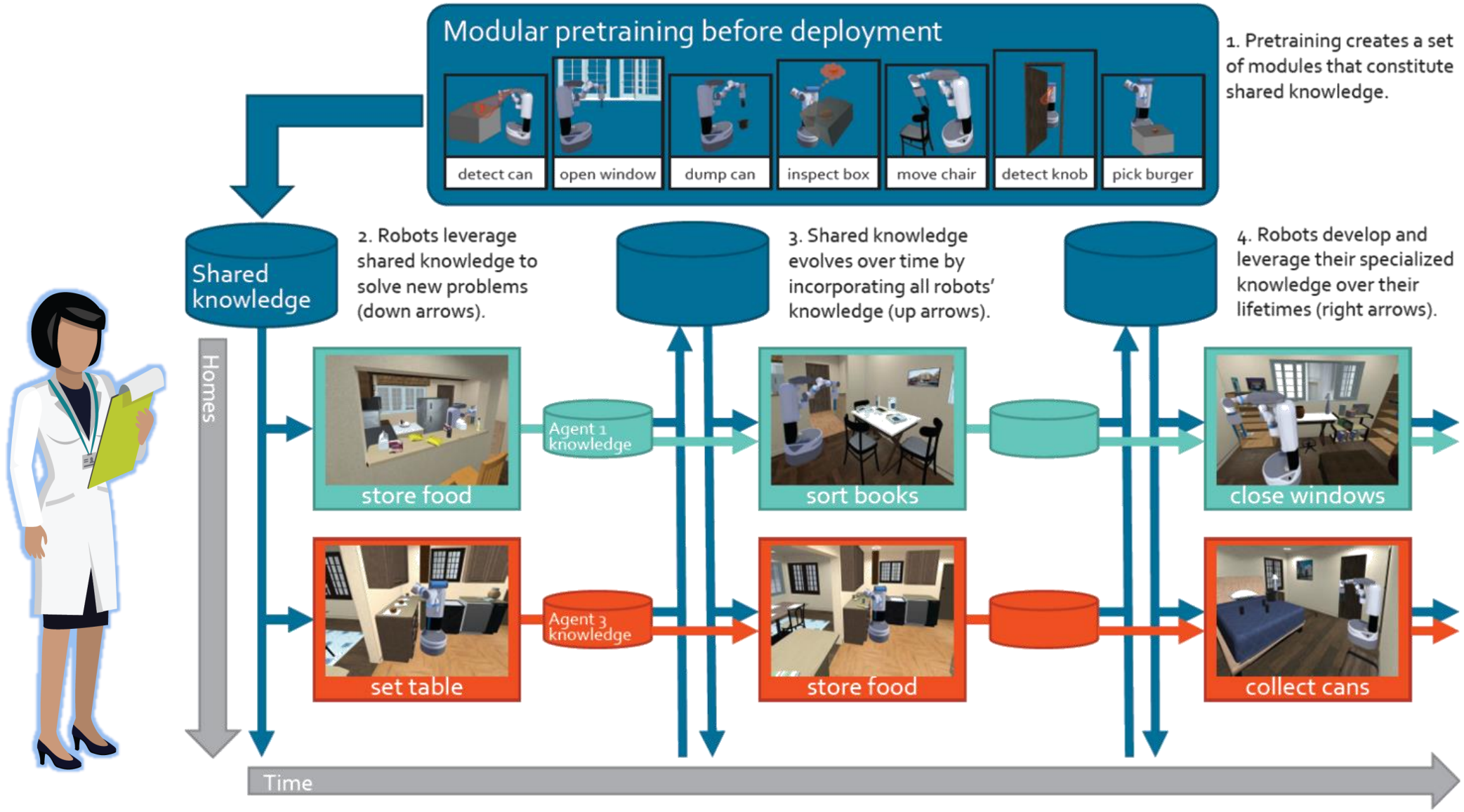
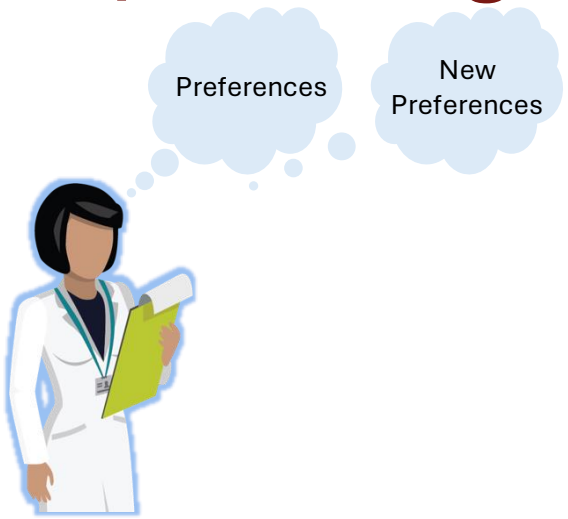
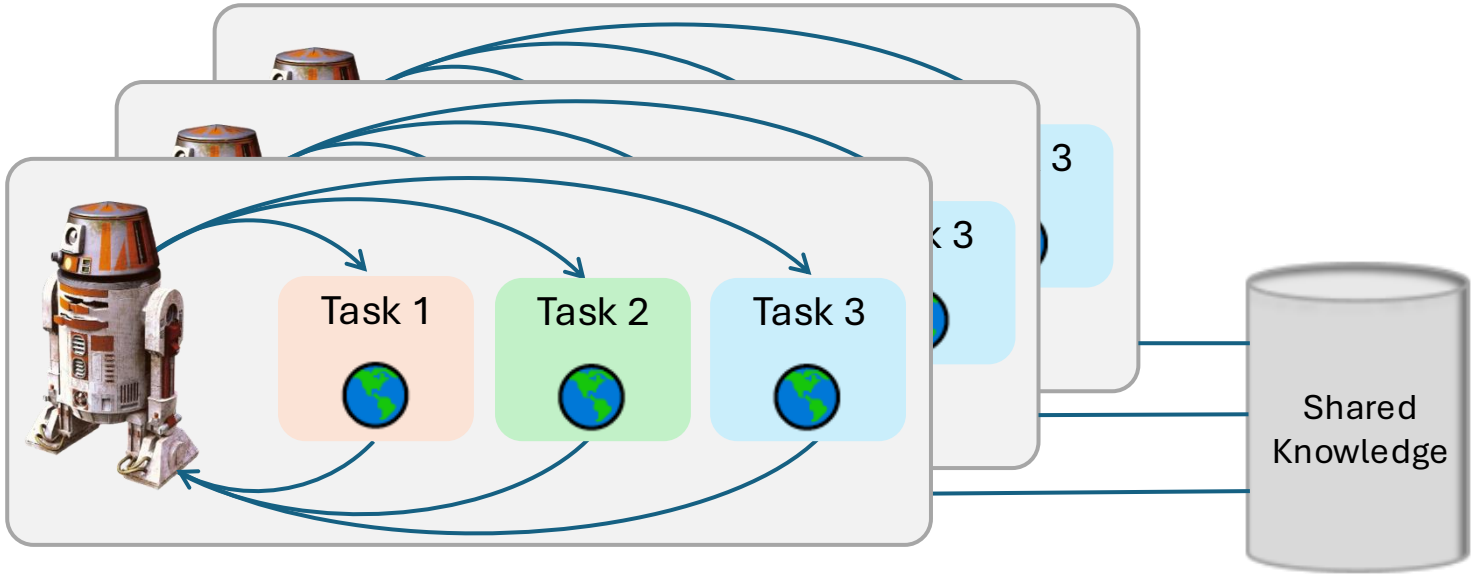


Image Courtesy: Jorge Mendez-Mendez, IRL Lab, MIT

Human-in-the Loop Lifelong Learning (HITL3)

Assessment Controller

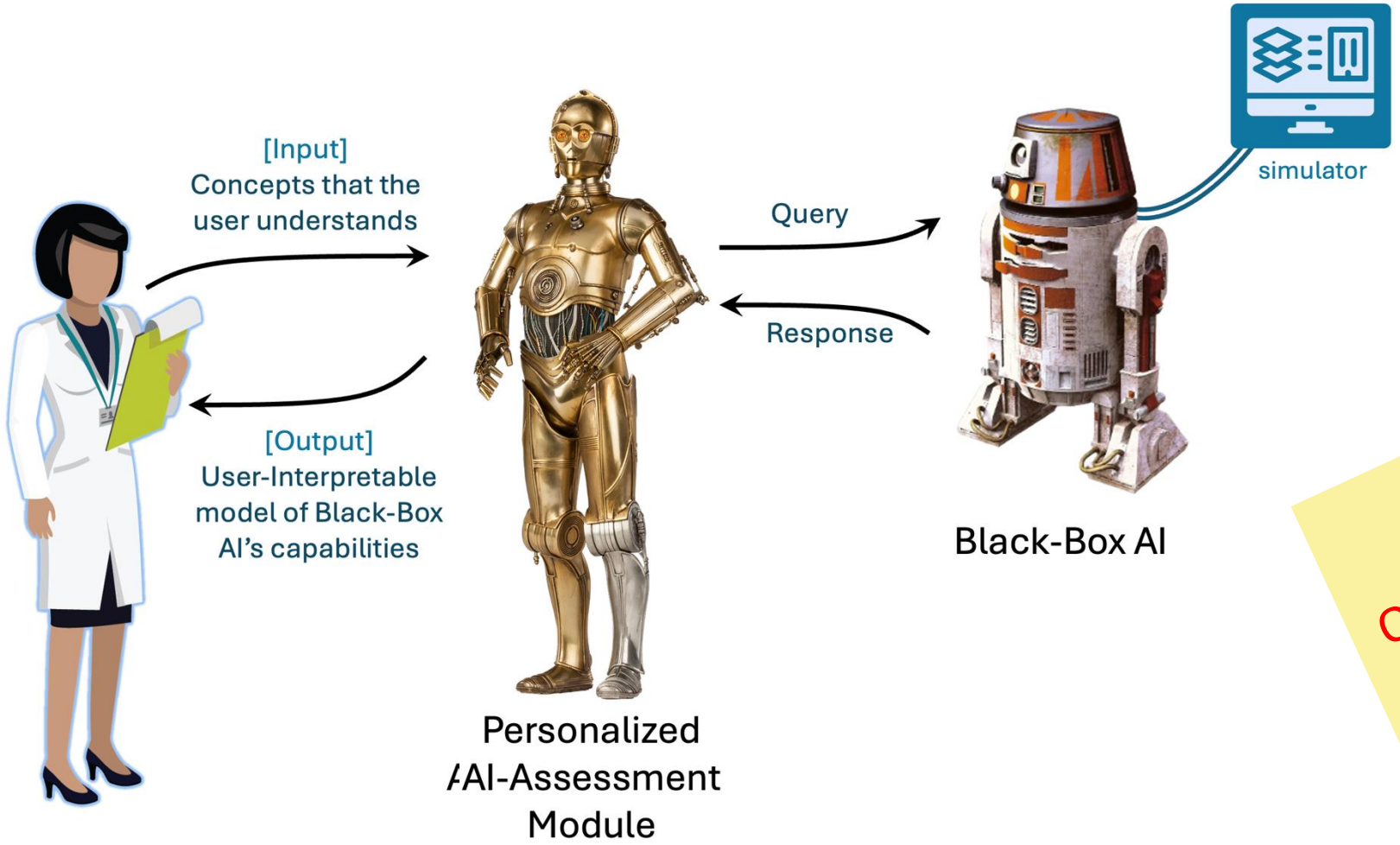




Beyond the Lab: User-Aligned Assessment of Taskable AI Systems

Pulkit Verma

pulkitv@cse.iitm.ac.in



Questions?