AAAI's Workshop on Artificial Intelligence Safety, 2022

Differential Assessment of Black-Box AI Systems

Rashmeet Kaur Nayyar*, Pulkit Verma*, Siddharth Srivastava

Arizona State University

How Would an End-User Assess an Adaptive AI System?

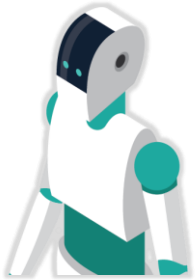
- AI systems should make it easy for its operators to learn how to use them safely.†
- How would we assess an AI system that can adapt its behavior?
- Much more challenging when the AI system is a black-box.

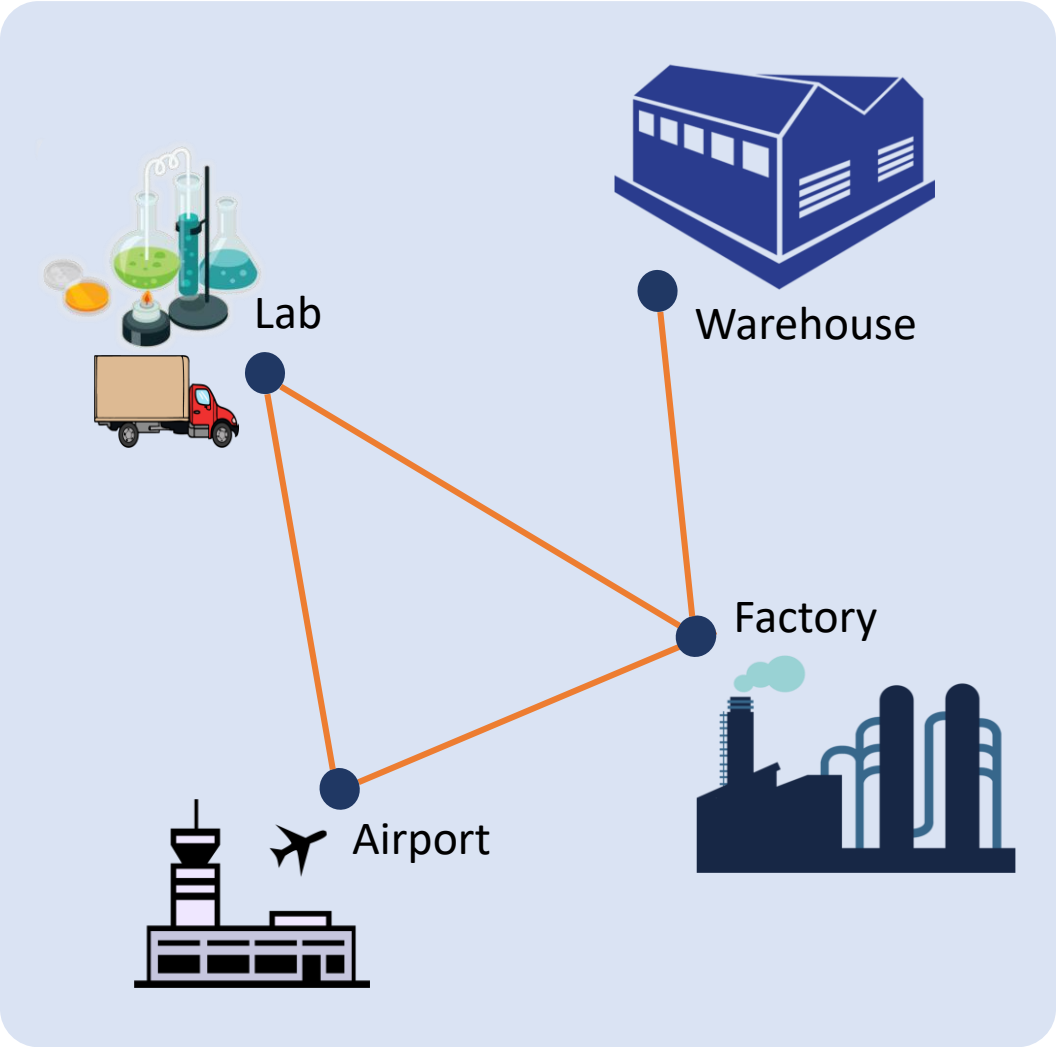
†Srivastava S. *Unifying Principles and Metrics for Safe and Assistive AI.* In Proc. AAAI 2021.

Image Source: Google Images

Will it be able to safely take my samples from the lab to the warehouse?

I did not expect the truck to go from the lab to the warehouse via factory. What has changed?





Related Work

Related Approaches

Some approaches closely related to this work include:

- Learning models of agent behavior.
  - From passive observations. E.g., ARMS – Yang et al. (AIJ 2007), LOCM - Cresswell et al. (ICAPS 2009), FAMA - Aineto et al. (AIJ 2019).
  - From active querying. E.g., AIA – Verma et al. (AAAI 2021).
  - These approaches assume the model is stationary.

- Model maintenance and Model Reconciliation.
  - E.g., Marshal - Bryce et al. (IJCAI 2016), MRP – Chakraborti et al. (ICAPS)
  - They assume availability of updated model in STRIPS-like form.

STRIPS-like Modeling Language

Advantages

- Support interventions, assessment of causality
- Easy to convert into natural language text

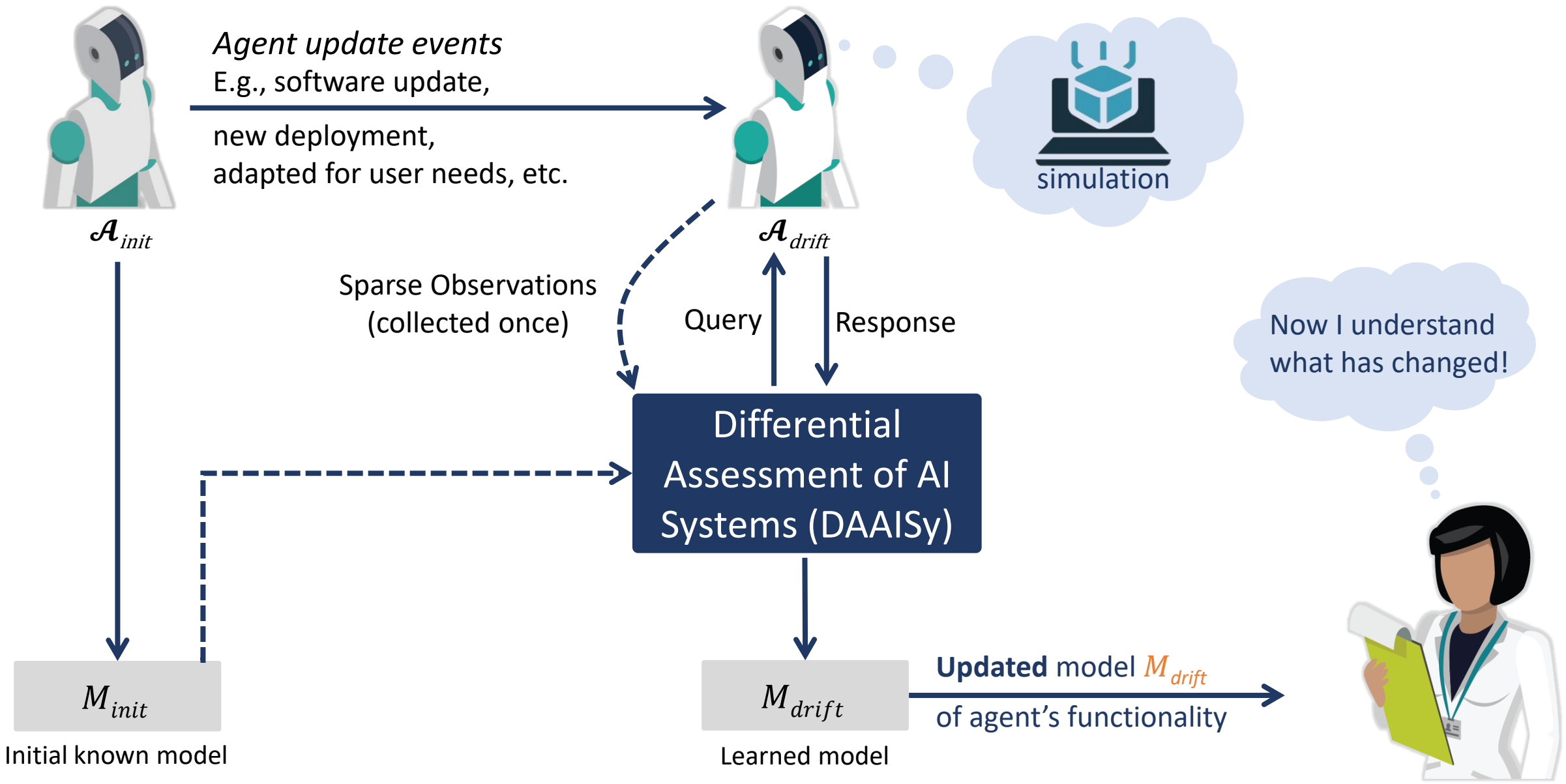
Challenges

- Large space of possible models ($9^{|\mathbb{P}|\times|\mathbb{A}|}$)
- $\mathbb{P}$: variable-instantiated predicates
- $\mathbb{A}$: parameterized actions

```
(:action pick-sample
  :parameters (?s)
  :precondition (and (handempty)
                     (onshelf ?s))
  :effect (and (not (handempty))
               (not (onshelf ?s))
               (holding ?s)))
```

[Fully Observable, Deterministic]

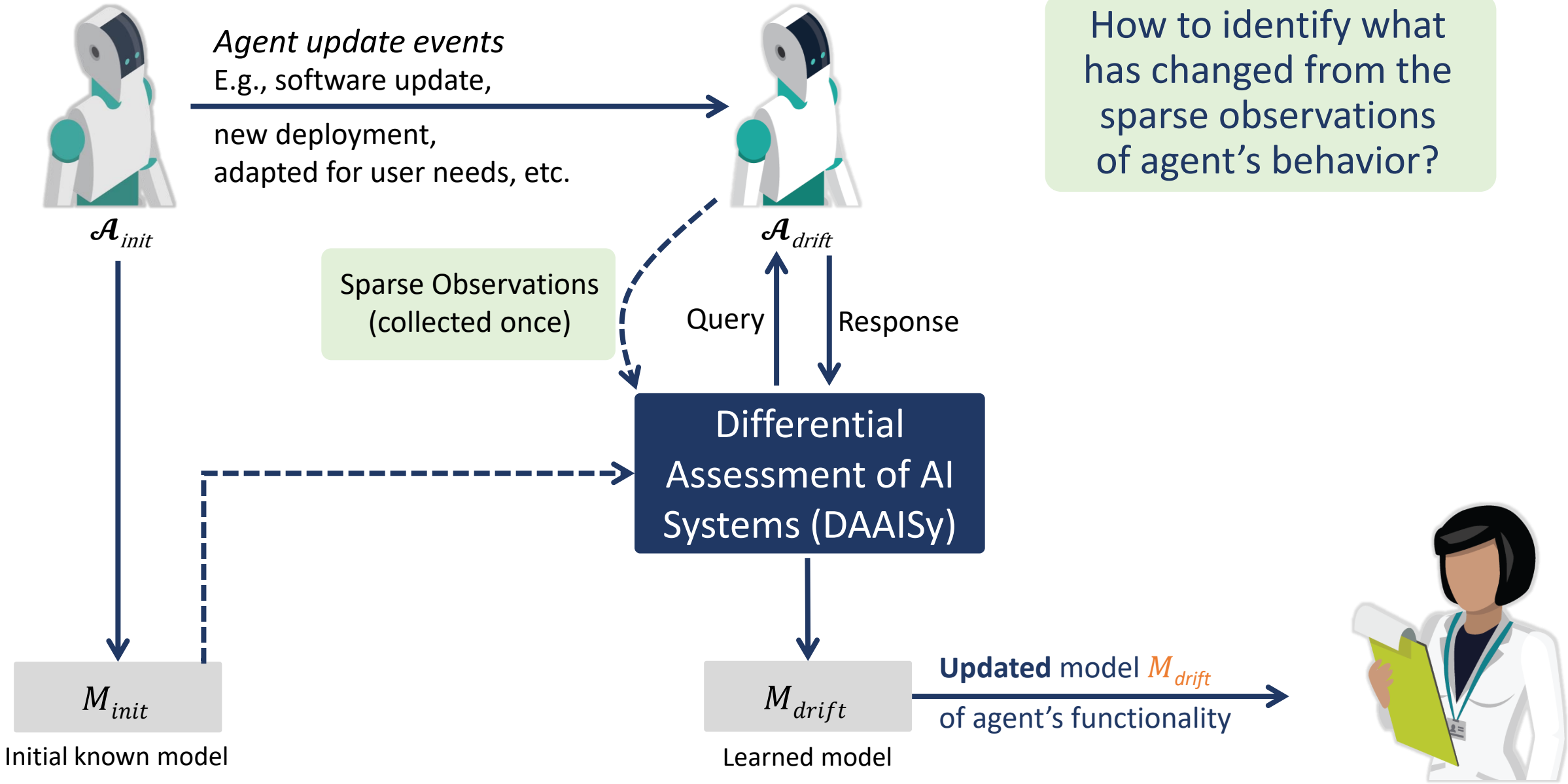
Problem Overview

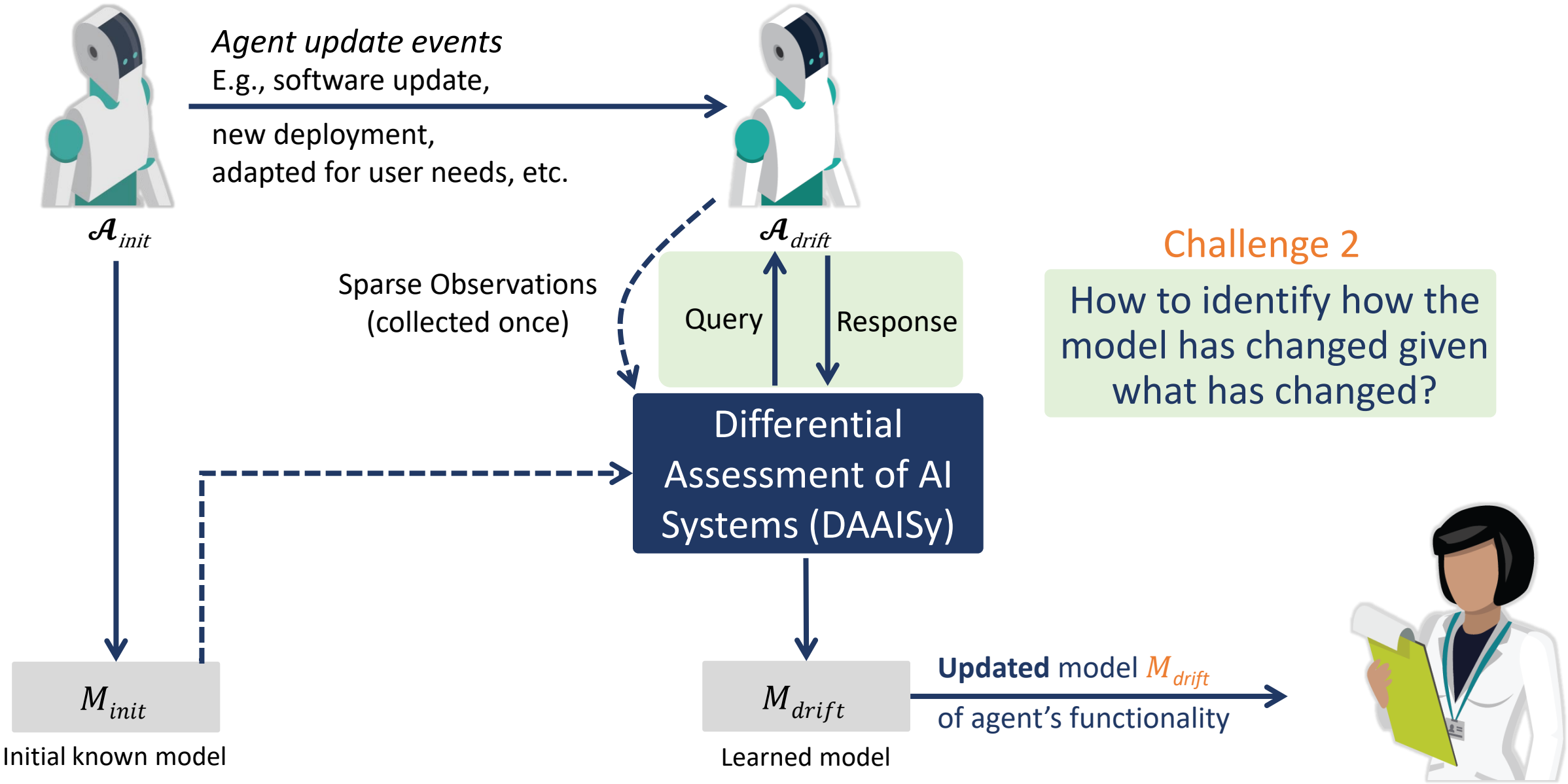

Agent update events
E.g., software update,
new deployment,
adapted for user needs, etc.
$\mathcal{A}_{init}$
$\mathcal{A}_{drift}$
simulation
Sparse Observations
(collected once)
Query
Response
Differential Assessment of AI Systems (DAAISy)
Now I understand what has changed!
M_{init}
Initial known model
M_{drift}
Learned model
Updated model M_{drift}
of agent's functionality

Key Challenges

Challenge 1

How to identify what has changed from the sparse observations of agent's behavior?





Challenge 2

How to identify how the model has changed given what has changed?

Challenge 1

Identifying what has changed given the observations

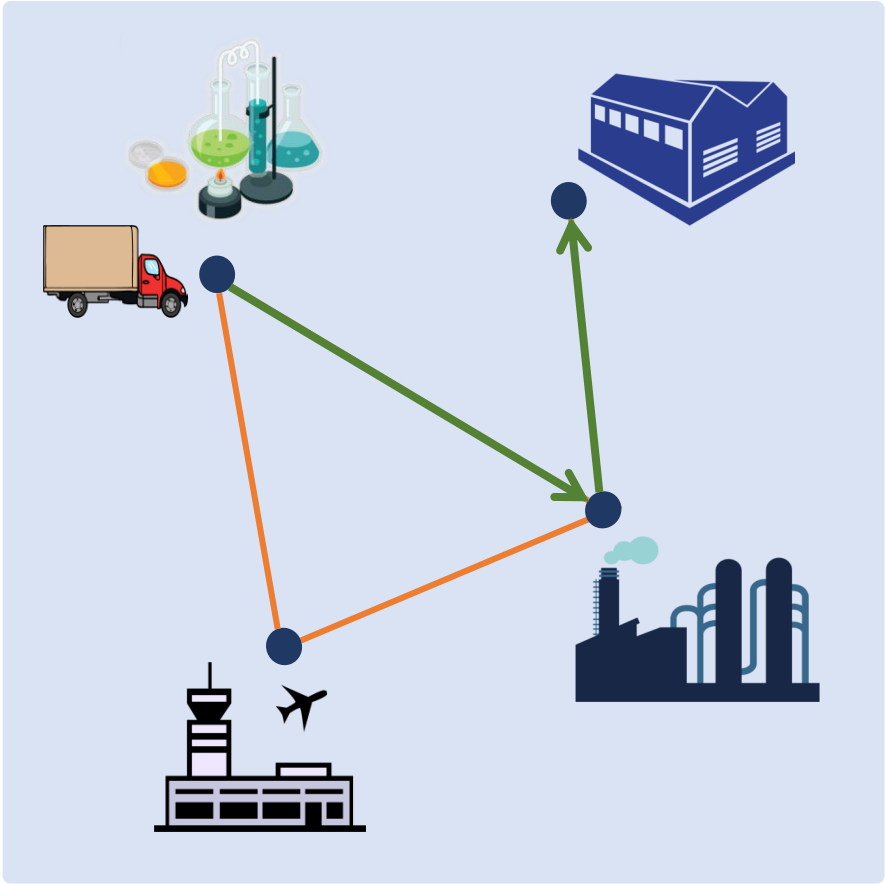
What can Change?

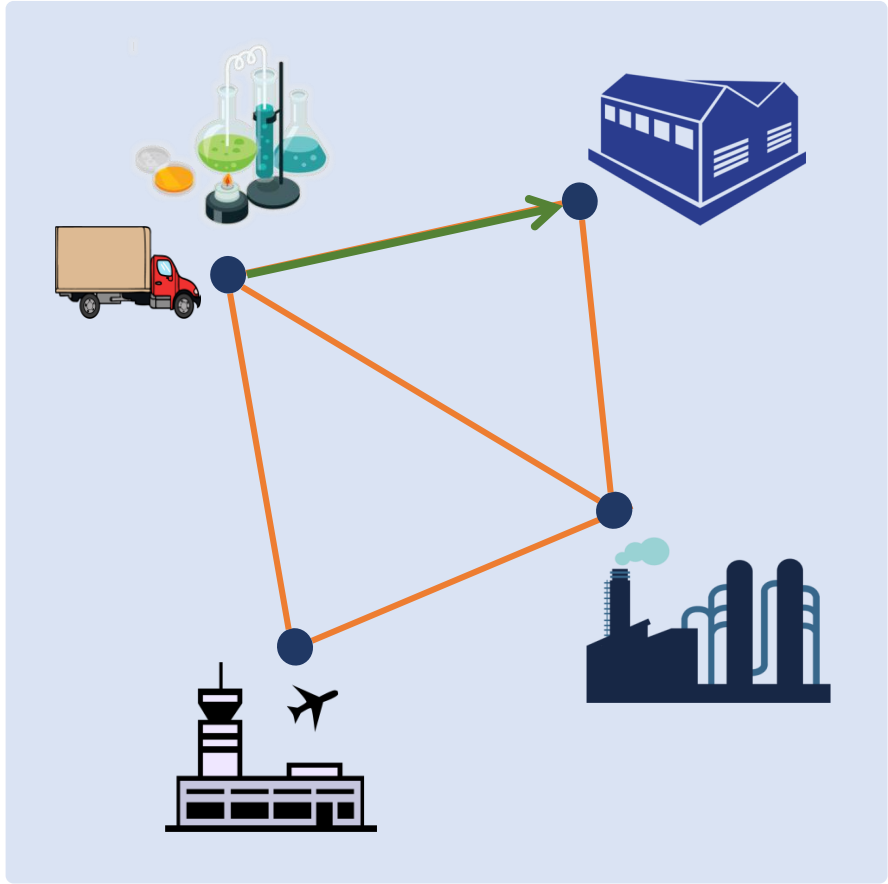
```
(:action pick-sample
  :parameters (?s)
  :precondition (and (handempty)
                     (onshelf ?s))
  :effect (and (not (handempty))
               (not (onshelf ?s))
               (holding ?s)))
```

- Any of these tuples could change their form:
  - From + to –, e.g., (handempty) to (not(handempty))
  - From – to +, e.g., (not(handempty)) to (handempty)
- Can get dropped from precondition or effect.
- Another literal can get added as a precondition or effect.

Two broad categories

Increased Functionality

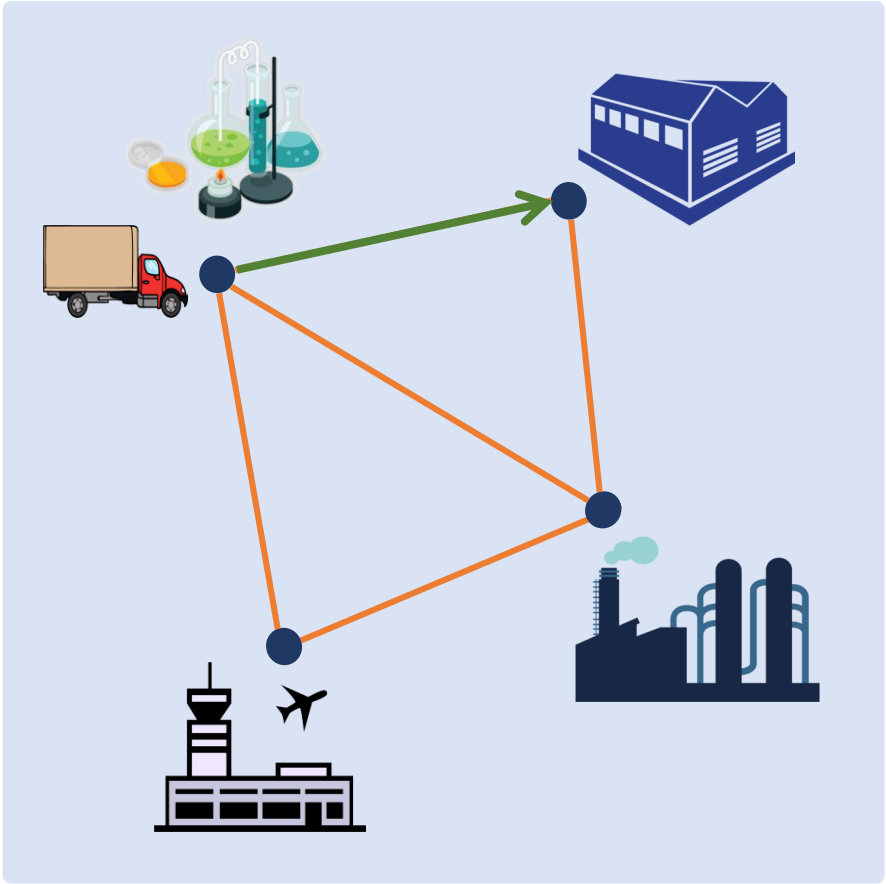


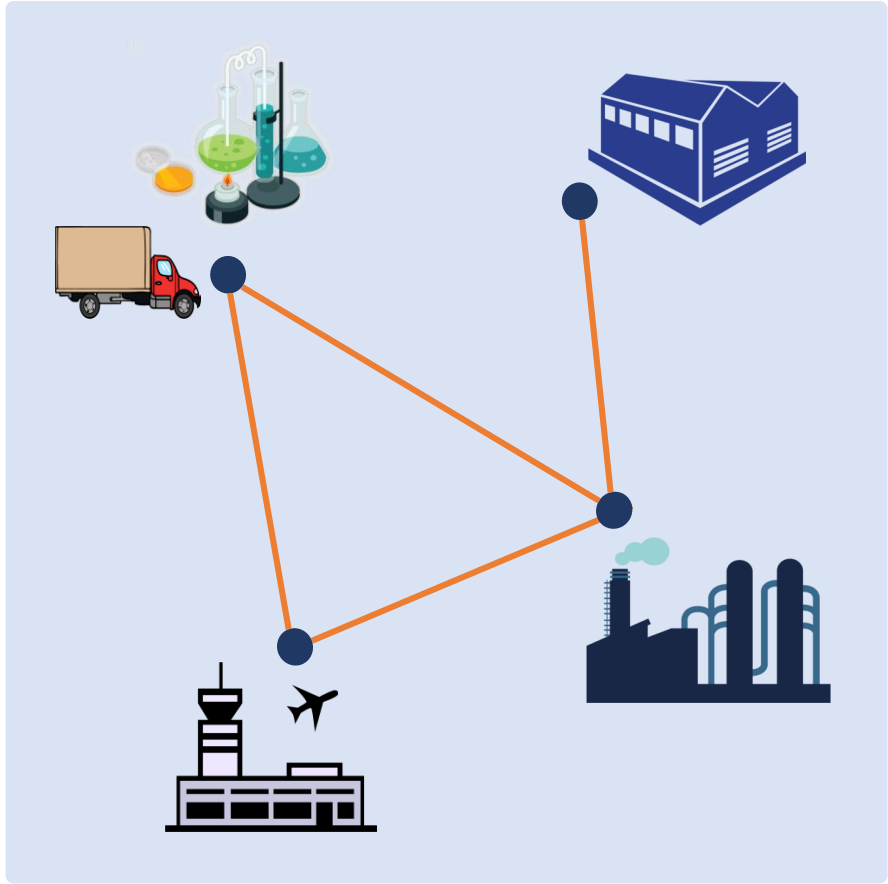


Observed that the agent executed

`move(lab,warehouse)`

Reduced Functionality





No observation corresponding to this reduction

Challenge 2

Identifying how the model has changed given what has changed.

Query-Based Interaction

- We use a query-response mode of interaction with the agent.
- Helps identify how each of the tuple has changed.
- **Query:** $\langle \boldsymbol{s_I}, \boldsymbol{\pi} \rangle$

  Initial State, Plan
- **Response:** $\langle \ell, \boldsymbol{s_F} \rangle$

  Length of plan that can be executed successfully and the final state.

Tuples

```
(:action pick-sample
  :parameters (?s)
  :precondition (and (handempty)
                     (+/-/∅) (onshelf ?s))
  :effect (and (+/-/∅) (handempty)
               ¬(onshelf ?s)))
```

n_1
n_2
n_3
n_4

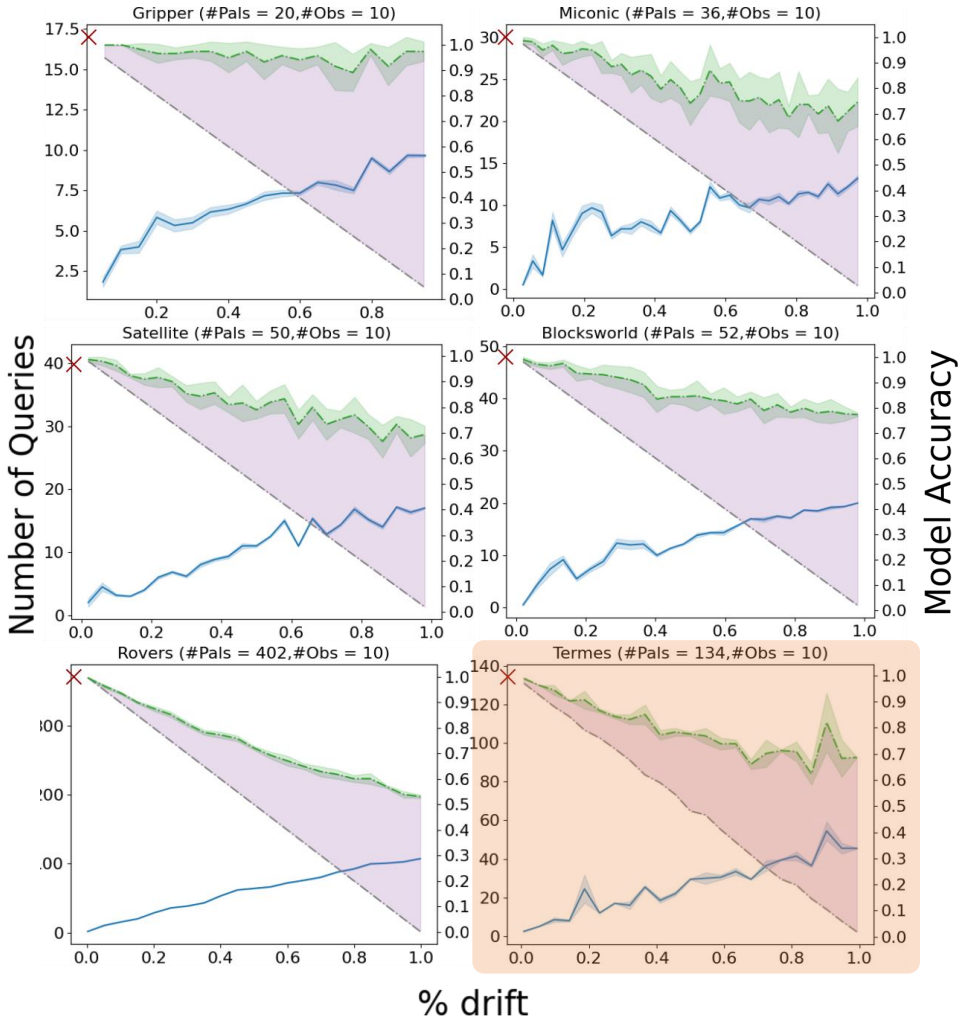
Example Setting

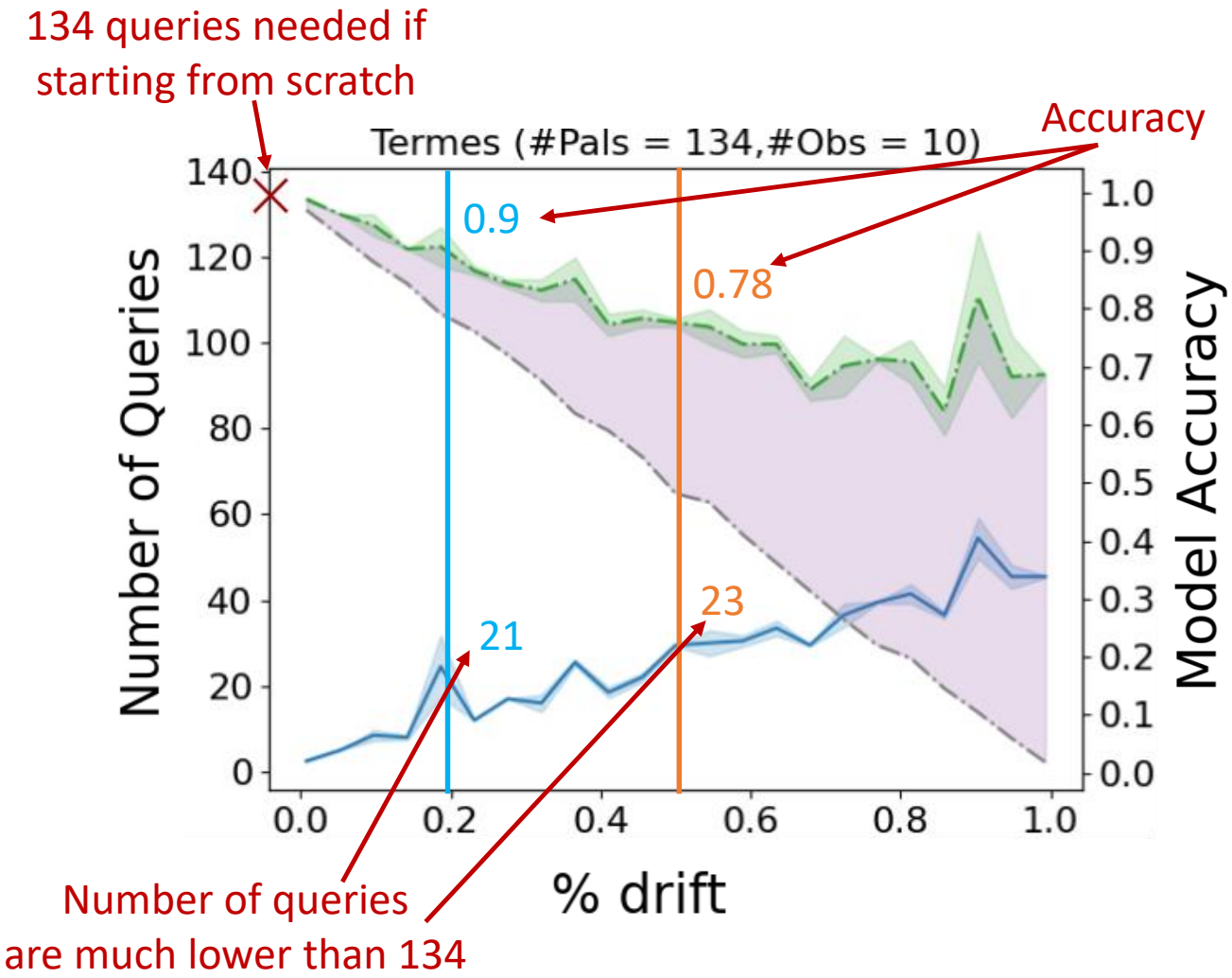
Empirical Evaluation

Experimental Setup

- Randomly generate initially known agents using IPC benchmark suite.
- Use 10 initial observations (state-action-state pairs) for unknown drifted IPC agent generated using IPC problems.
- Using previous model and available observations, predict what may have changed.
- Learn the updated model by querying for changed portions of the model.
- Evaluate performance of the assessment module and compare it with the vanilla active querying approach of assessing model from scratch.

Results


Gripper (#Pals = 20,#Obs = 10)
Miconic (#Pals = 36,#Obs = 10)
Satellite (#Pals = 50,#Obs = 10)
Blocksworld (#Pals = 52,#Obs = 10)
Rovers (#Pals = 402,#Obs = 10)
Termes (#Pals = 134,#Obs = 10)
Number of Queries
Model Accuracy
% drift


Number of queries
Accuracy gained
Accuracy of initial model
Accuracy of model computed by DAAISy
Number of queries by AIA
134 queries needed if starting from scratch
Accuracy
Termes (#Pals = 134,#Obs = 10)
0.9
0.78
21
23
Number of Queries
Model Accuracy
% drift
Number of queries are much lower than 134

Results: Number of Queries

Domain	Max Tuples#	AIA	DAAISy
Gripper	20	15.0	6.5
Miconic	36	32.0	7.7
Satellite	50	34.0	9.0
Blocksworld	52	40.0	11.4
Termes	134	115.0	27.0
Rovers	402	316.0	61.0

#Max Tuples is the upper bound on number of tuples in the domain.

The average number of queries to achieve same level of accuracy for 50% drifted models

- Results with FD planner with LM-Cut.
- Our approach, DAAISy, takes up to five times lesser queries than reassessment from scratch using AIA.
- Reassessment from scratch using other passive learning methods would take even longer.

Key Takeaways

- A novel method for differential assessment of black-box AI systems that have drifted from their previously known functionality.
- Able to learn highly accurate models of functionality of agents issuing a significantly lower number of queries as opposed to relearning from scratch.
- Plan to extend the framework to more general classes, stochastic settings, and models.

rmnayyar@asu.edu
verma.pulkit@asu.edu
siddharths@asu.edu

Paper: bit.ly/3so0nrx