Learning AI-System Capabilities under Stochasticity

Pulkit Verma, Rushang Karia*, Gaurav Vipat, Anmol Gupta, Siddharth Srivastava*

Arizona State University, AZ, USA

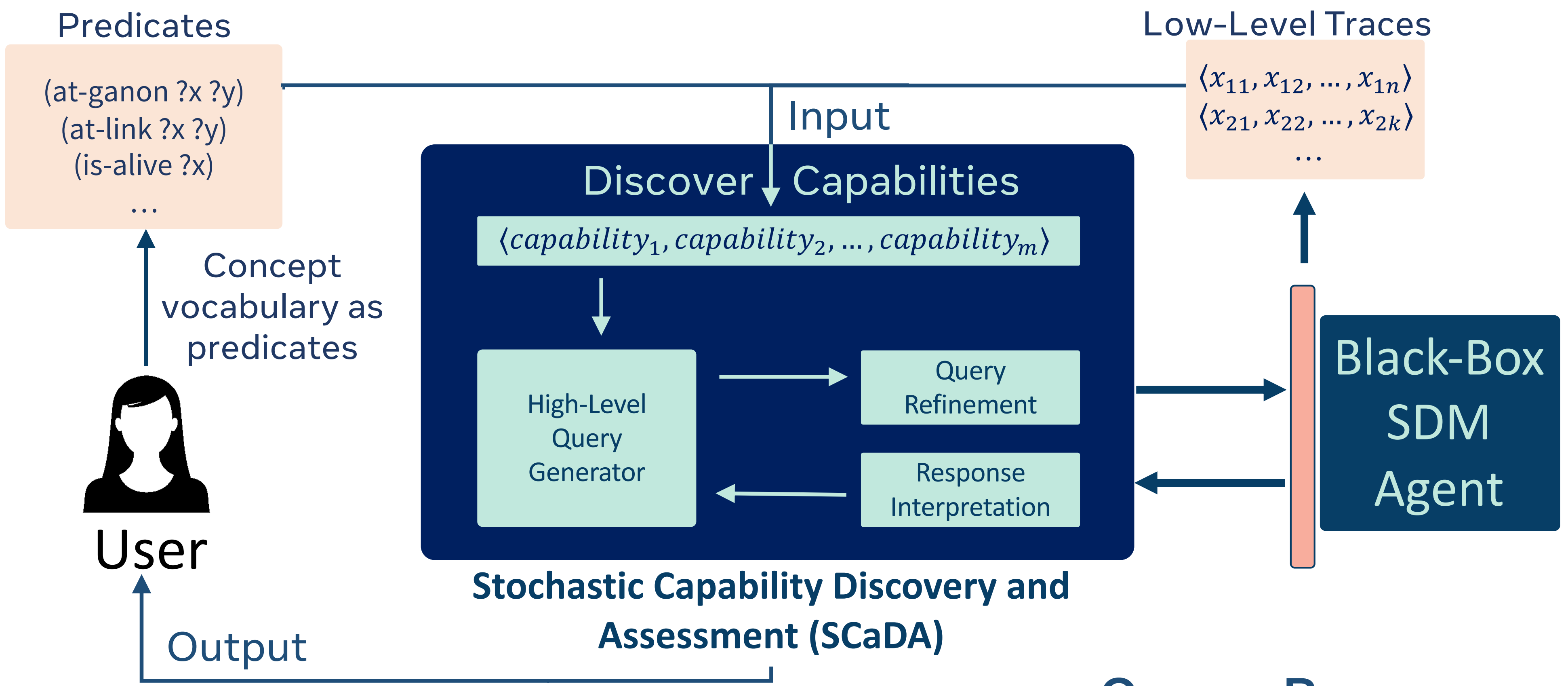
A new approach for discovering and assessing capabilities of AI systems that can plan and learn.

What is a capability?

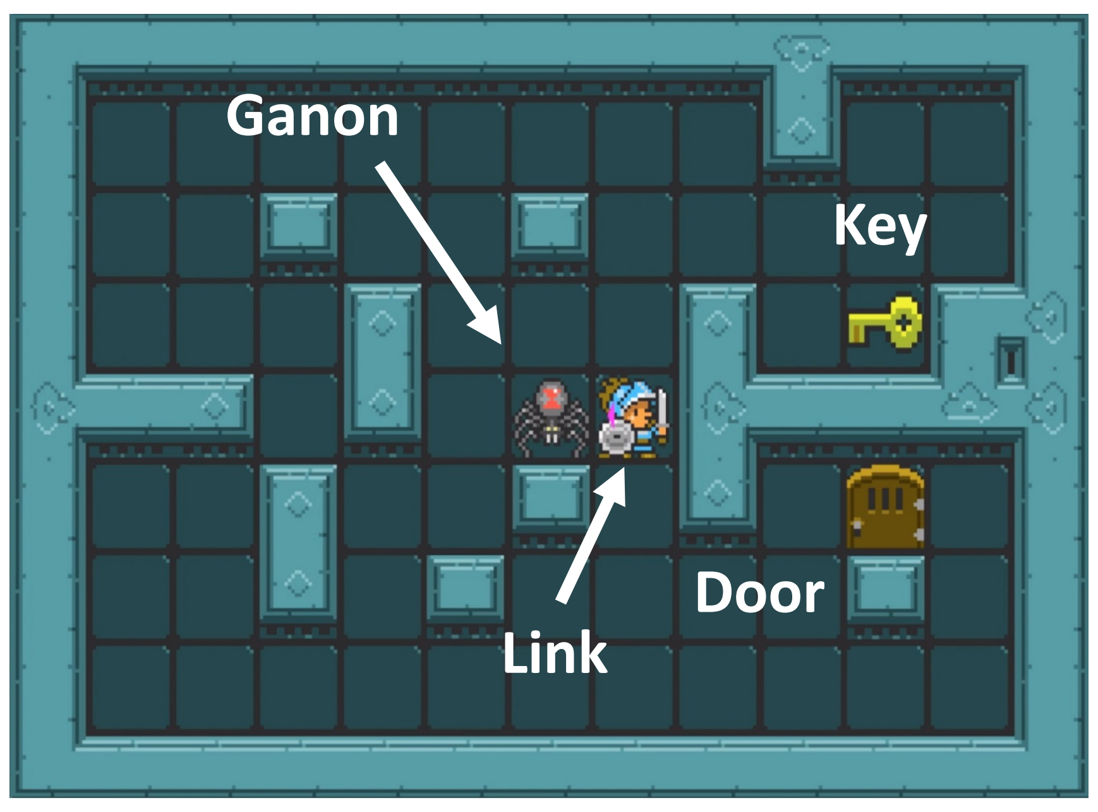
- A high-level task that an SDMA can perform.
- Combination of multiple low-level functionalities of the SDMA.

Why learn capability descriptions?

- Easier to reason about in terms of capabilities than low-level functionalities.

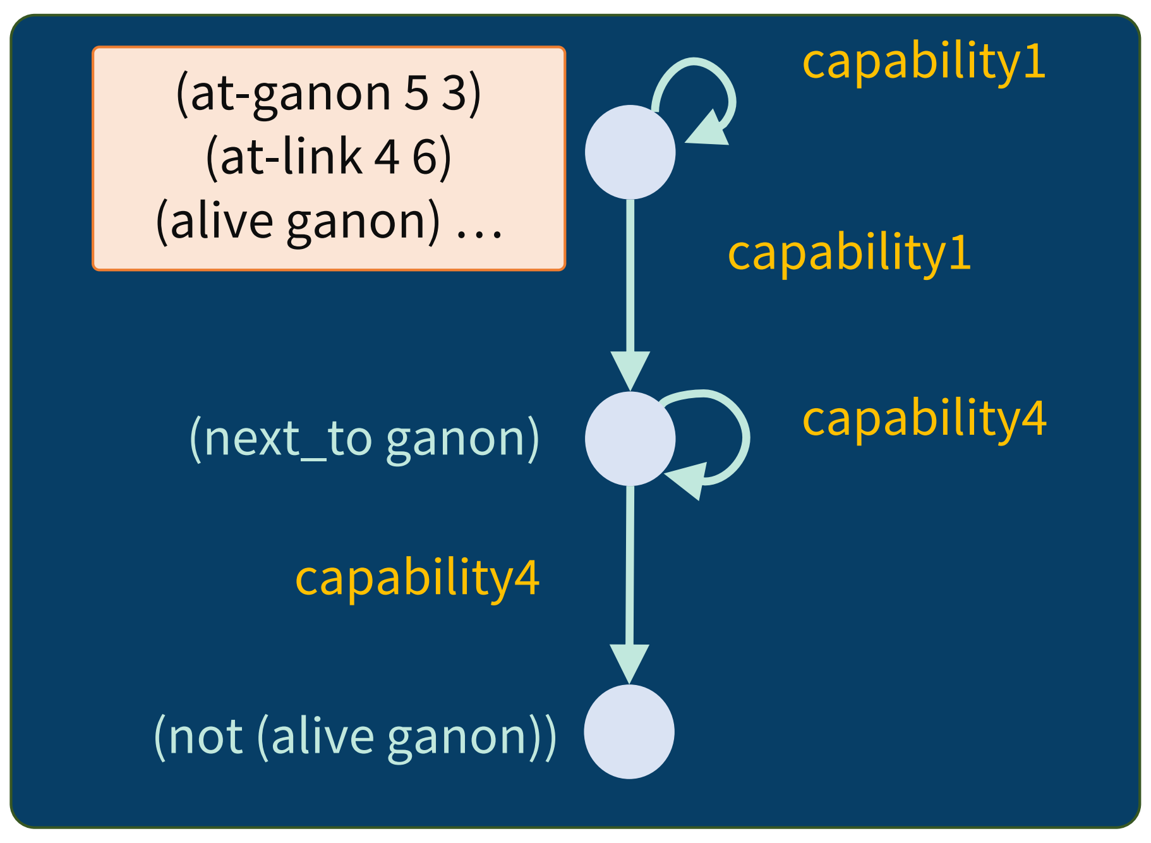

Predicates
(at-ganon ?x ?y)
(at-link ?x ?y)
(is-alive ?x)
…
Concept vocabulary as predicates
User
Input
Output
Discover Capabilities
$\langle capability_1, capability_2, \dots, capability_m \rangle$
High-Level Query Generator
Query Refinement
Response Interpretation
Stochastic Capability Discovery and Assessment (SCaDA)
Low-Level Traces
$\langle x_{11}, x_{12}, \dots, x_{1n} \rangle$
$\langle x_{21}, x_{22}, \dots, x_{2k} \rangle$
…
Black-Box SDM Agent
Query - Response Interface

- A set of capability descriptions.
- For each capability, it describes when it can be executed and what its possible effects are.

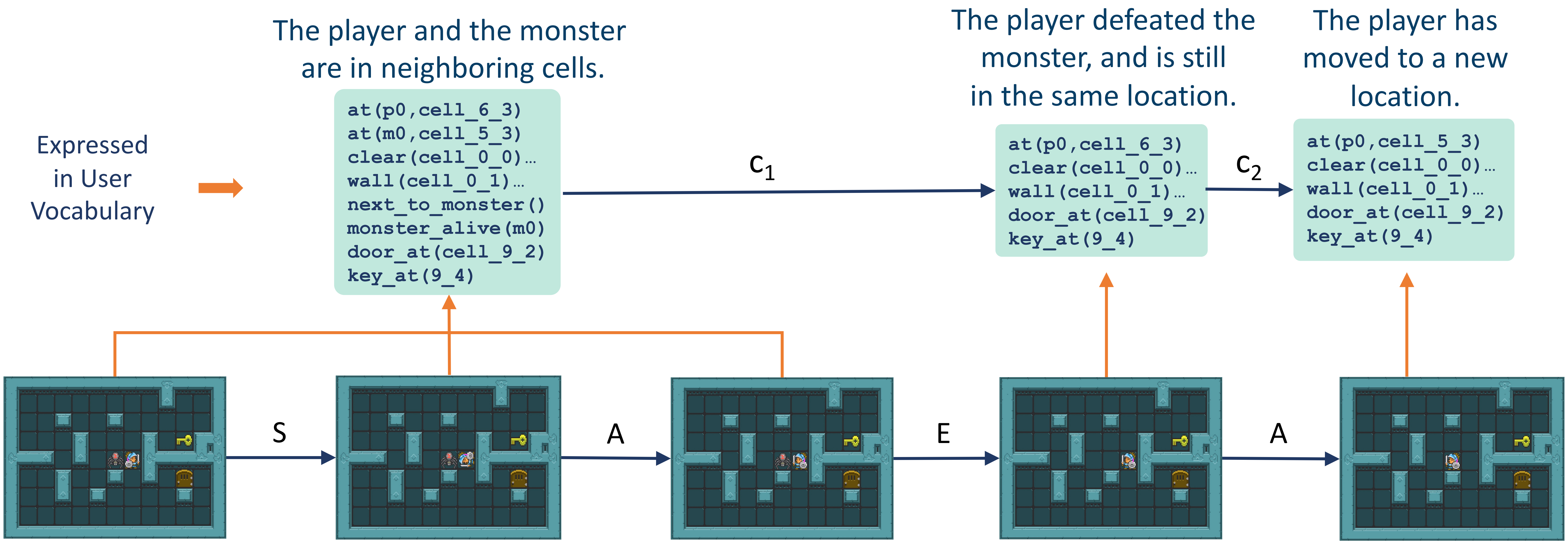


Agent Actions (Keystrokes)	Learned Capabilities
W, A, S, D, E	(defeat ganon), (go to door), (go to key), (go to ganon), (pick key), (open door)
Agent's State Representation	**Interpretable State Representation**
$\langle x_{11}, x_{12}, \dots, x_{1n} \rangle$ $\langle x_{21}, x_{22}, \dots, x_{2k} \rangle$ …	(at ganon 5,3), (at link 6,3) (at key 9,4), (at door 9,2) (alive ganon) (alive link) …..

High-Level Query Example



Discovering Capabilities


The player and the monster are in neighboring cells.
The player defeated the monster, and is still in the same location.
The player has moved to a new location.
Expressed in User Vocabulary
c_1
c_2
S
A
E
A

```
at(p0,cell_6_3)
at(m0,cell_5_3)
clear(cell_0_0)…
wall(cell_0_1)…
next_to_monster()
monster_alive(m0)
door_at(cell_9_2)
key_at(9_4)
```

```
at(p0,cell_6_3)
clear(cell_0_0)…
wall(cell_0_1)…
door_at(cell_9_2)
key_at(9_4)
```

```
at(p0,cell_5_3)
clear(cell_0_0)…
wall(cell_0_1)…
door_at(cell_9_2)
key_at(9_4)
```

Learned Capability Model

```
(:capability c4
  :parameters (?player1 ?cell1
    ?monster1 ?cell2)
  :precondition (and
    (alive ?monster1)
    (at ?player1 ?cell1)
    (at ?monster1 ?cell2)
    (next_to ?monster1))
  :effect (probabilistic
    0.7 (and (clear ?cell2)
      (not (alive ?monster1))
      (not (at ?monster1 ?cell2))
      (not (next_to ?monster1)))
    0.2 (and (game-over)
      (not (at ?player1 ?cell1))
      (not (alive ?player1)))
    0.1 (and )))        #No-change
```

Equivalent to "Defeat Ganon"

Why this representation?

- Easily convertible to natural language
- Supports generalization and transfer

Accuracy verified using Driver Agent (IPPC Tireworld) as the Ground Truth Model available

Results

Environment	# Queries
Escape	592
Zelda	528
Montezuma	849
Driver Agent	34

What Next?

- Plan using learned models
- Expand the scope to Embodied AI Domains
- Expand to Noisy Classifiers