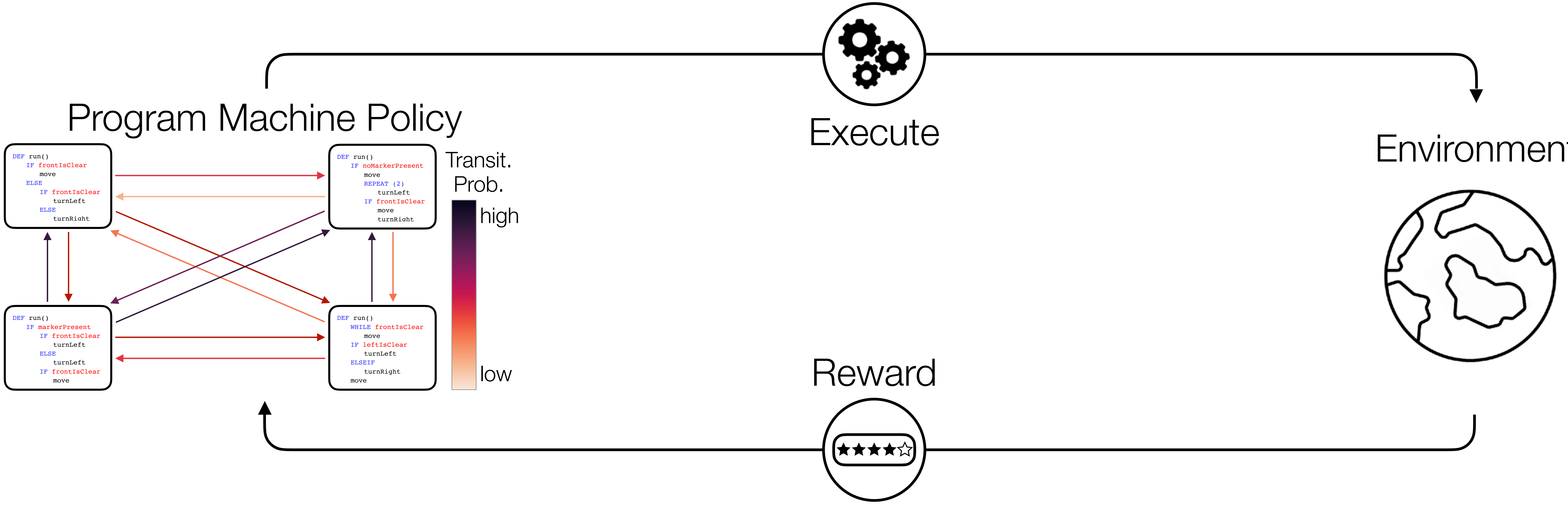
Addressing Long-Horizon Tasks by Integrating Program Synthesis and State Machines

NeurIPS 2023 Workshop on Generalization in Planning




Yu-An Lin*


Chen-Tao Lee*

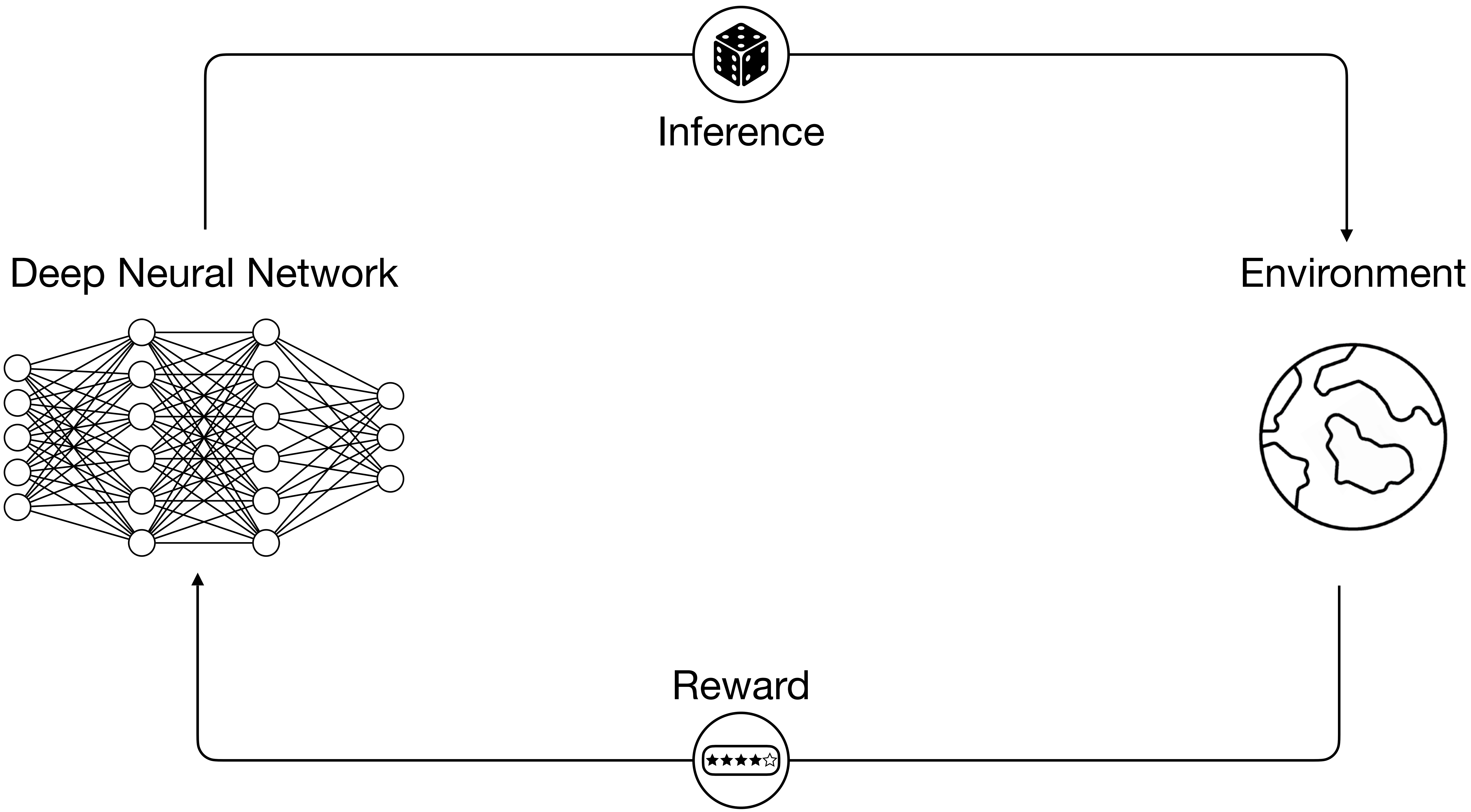

Guan-Ting Liu*


Pu-Jen Cheng


Shao-Hua Sun

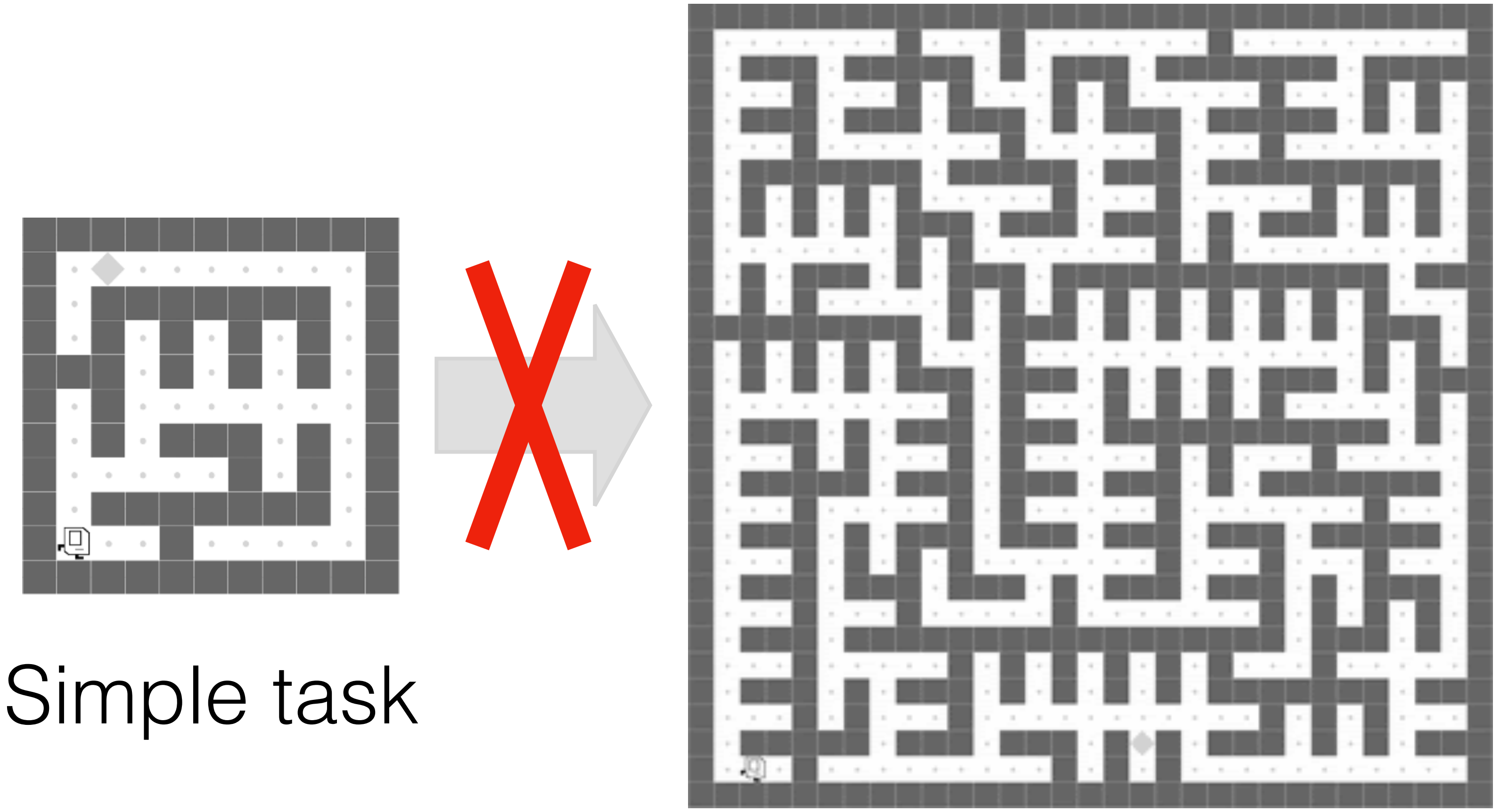
National Taiwan University

Deep Reinforcement Learning



Deep Reinforcement Learning - Issues

Generalization

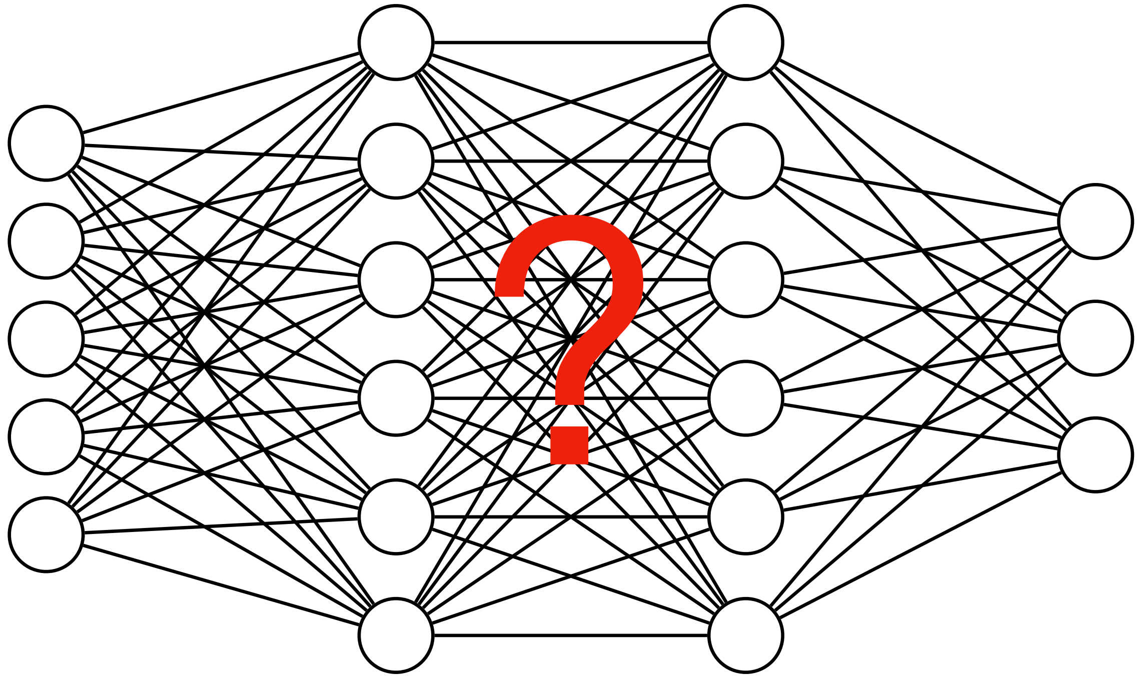


Simple task

Complex task

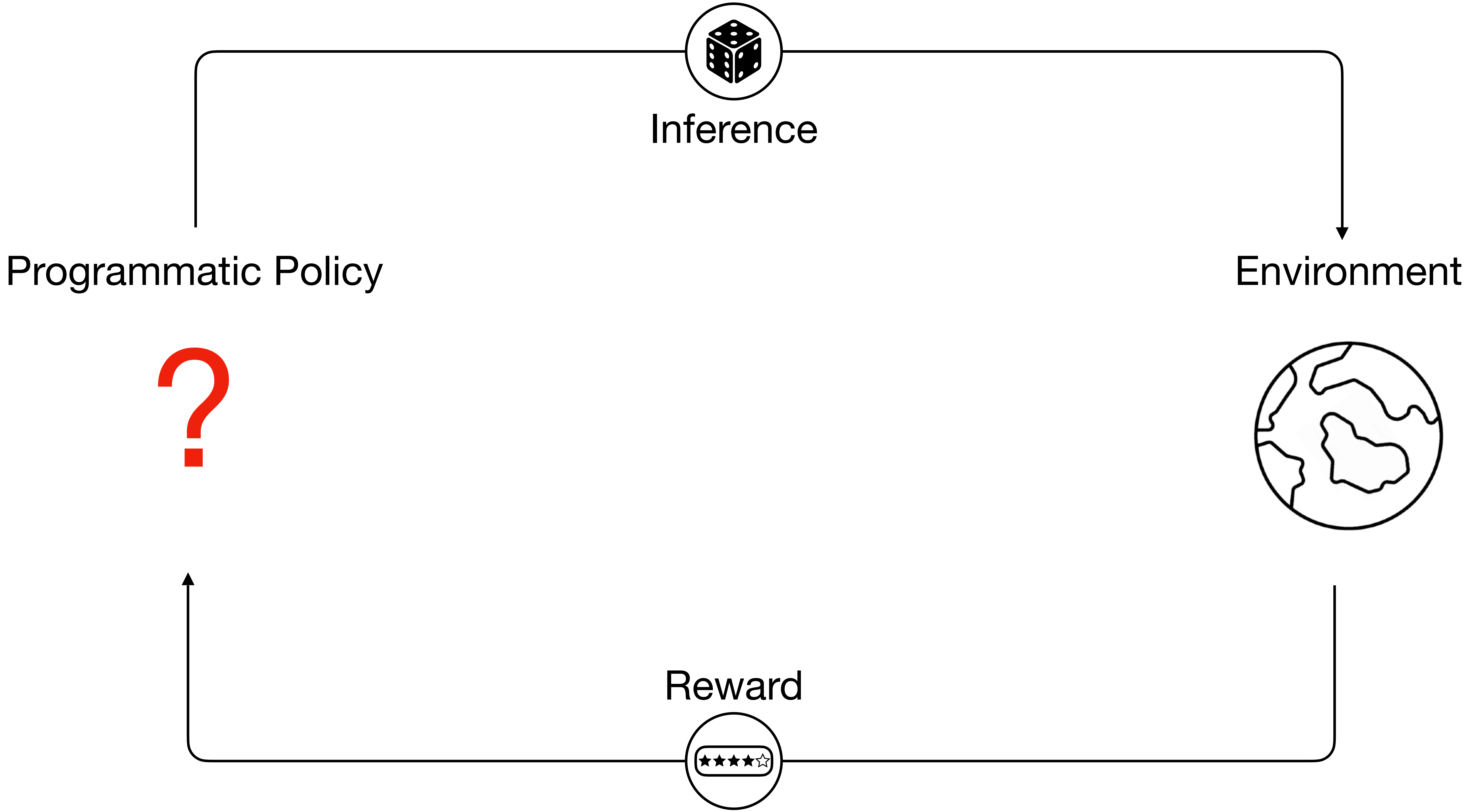
Interpretability

Trust, Safety, and Contestability

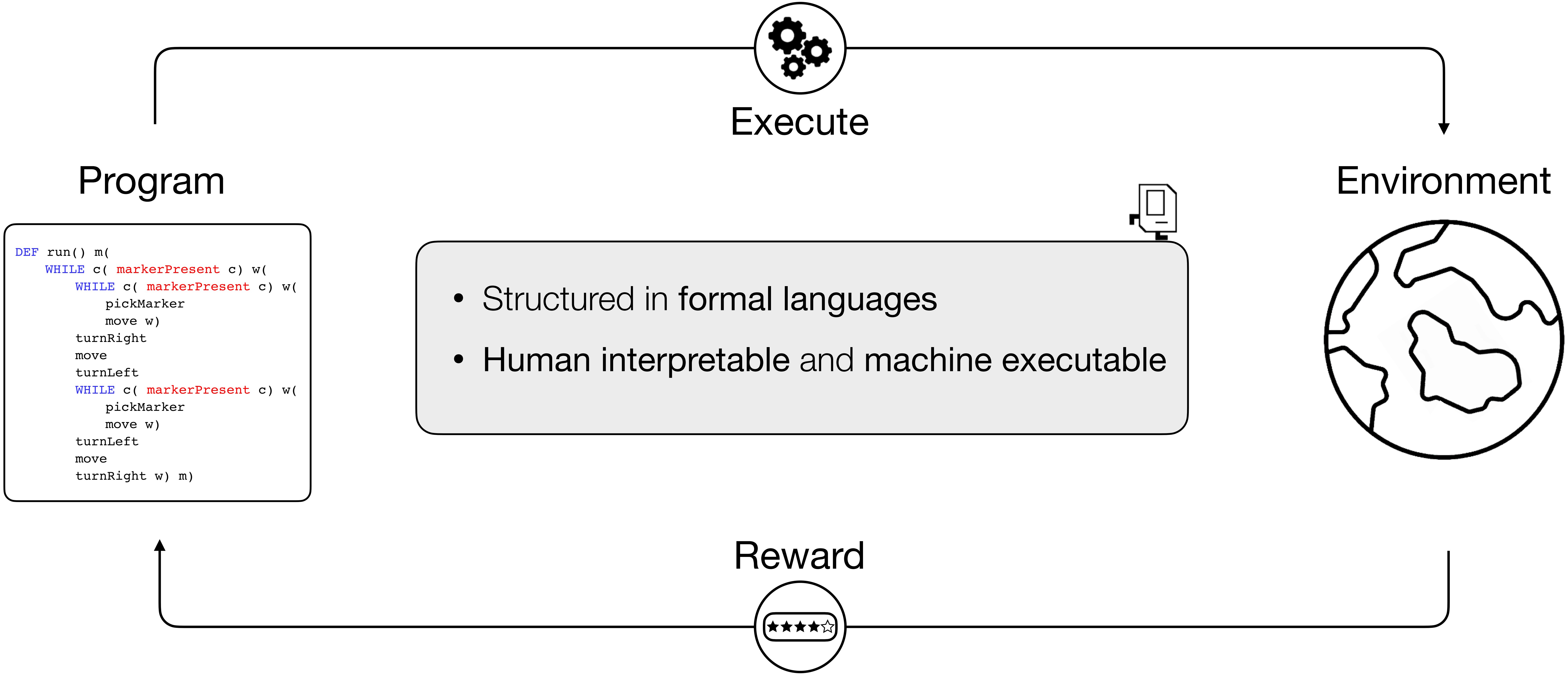


Deep neural network policy

Programmatic Reinforcement Learning



Programmatic Reinforcement Learning - Program Synthesis

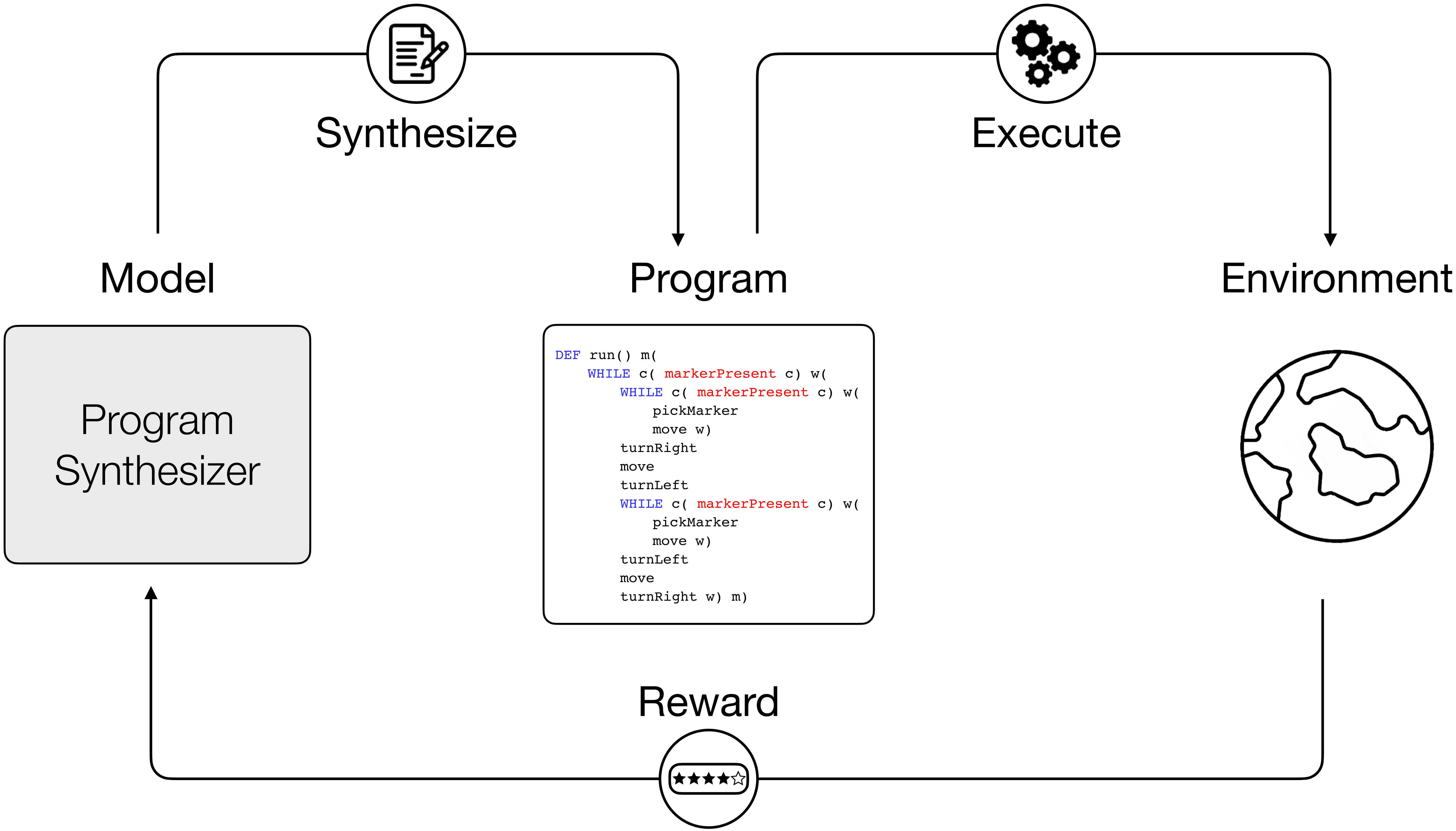



```
DEF run() m(
    WHILE c( markerPresent c) w(
        WHILE c( markerPresent c) w(
            pickMarker
            move w)
        turnRight
        move
        turnLeft
        WHILE c( markerPresent c) w(
            pickMarker
            move w)
        turnLeft
        move
        turnRight w) m)
```

- Structured in **formal languages**
- **Human interpretable** and **machine executable**

Sun et al. "Neural program synthesis from diverse demonstration videos." ICML 2018.
Sun et al. "Program Guided Agent." ICLR 2020.
Dang-Nhu "PLANS: Neuro-symbolic program learning from videos." NeurIPS 2020.
Trivedi et al. "Learning to Synthesize Programs as Interpretable and Generalizable Policies." NeurIPS 2021.
Liu et al. "Hierarchical Programmatic Reinforcement Learning via Learning to Compose Programs." ICML 2023.

LEAPS: Learning Embeddings for lAtent Program Synthesis



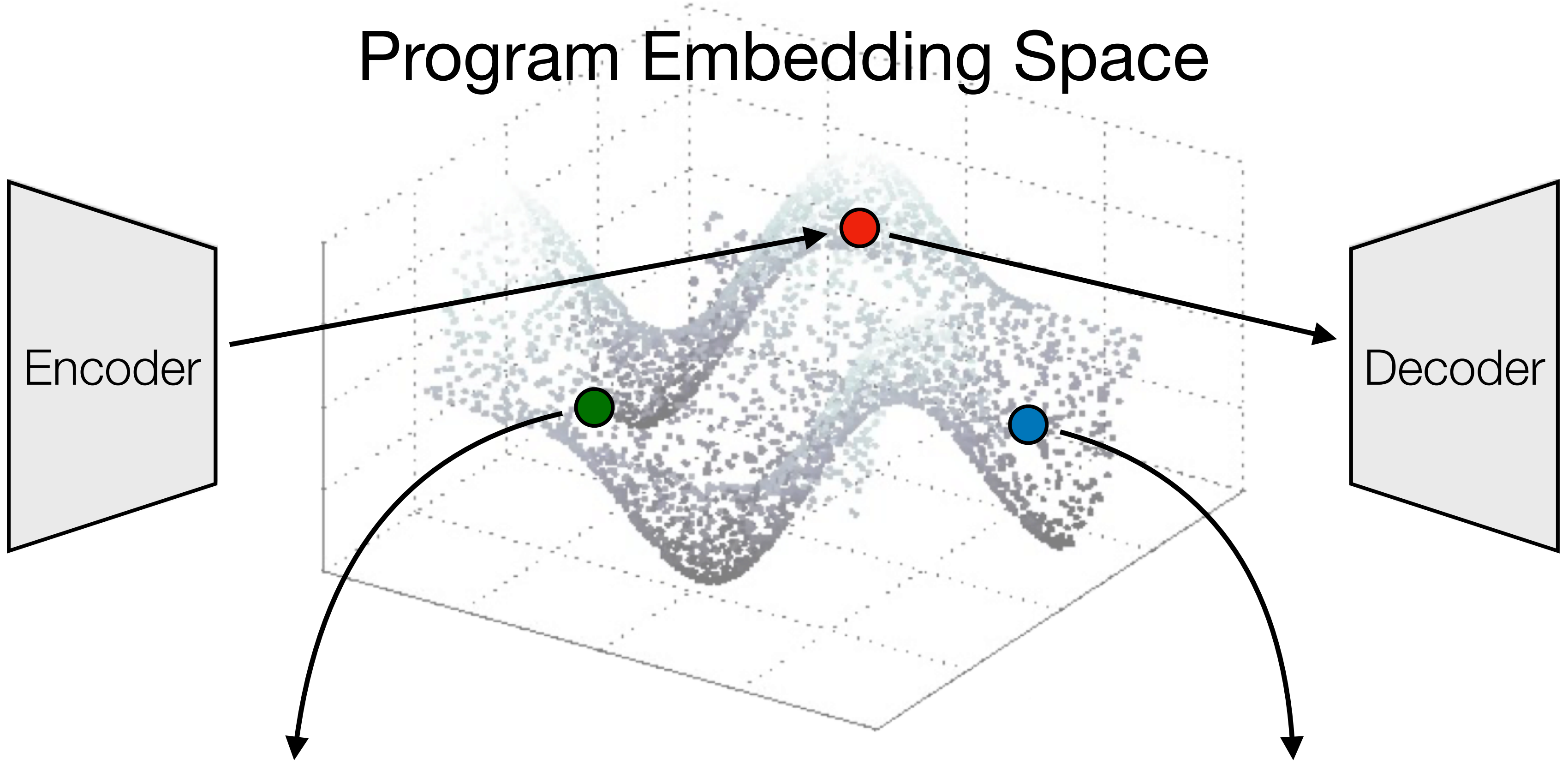
Trivedi et al. "[Learning to Synthesize Programs as Interpretable and Generalizable Policies](#)." NeurIPS 2021.

LEAPS - Learning a Program Embedding Space

Goal: Learn the **grammar** and the **environment dynamics**

Program

```
DEF run()
   IF frontIsClear
      move
   ELSE
      IF markerPresent
         pickMarker
      ELSE
         turnRight
```



Reconstructed Program

```
DEF run()
   IF NOT frontIsClear
      IF markerPresent
         pickMarker
      ELSE
         turnRight
   ELSE
      move
```

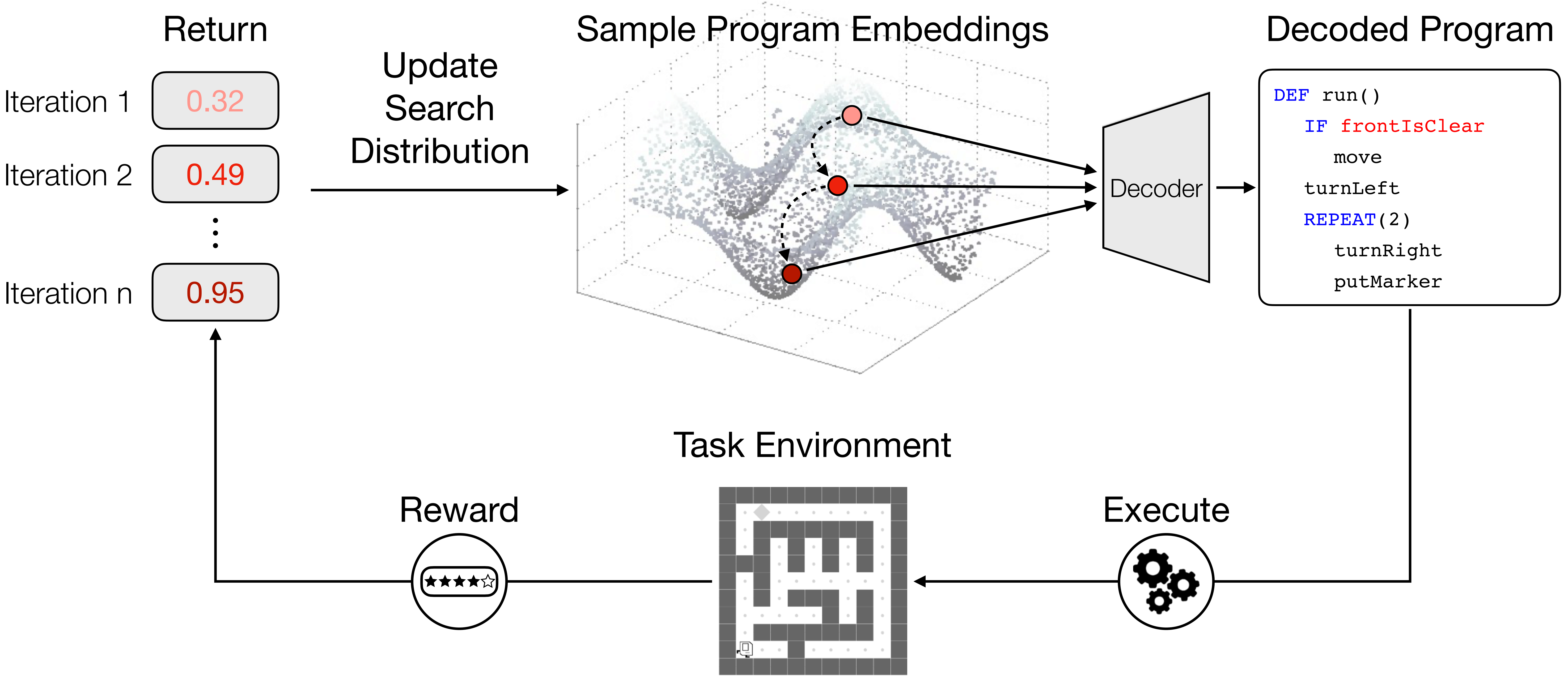
```
DEF run()
   IF frontIsClear
      move
   turnLeft
   REPEAT(2)
      turnRight
      putMarker
```

```
DEF run()
   WHILE frontIsClear
      move
      IF markerPresent
         pickMarker
      ELSE
         turnRight
```

Trivedi et al. "[Learning to Synthesize Programs as Interpretable and Generalizable Policies](#)." NeurIPS 2021.

LEAPS - Latent Program Search

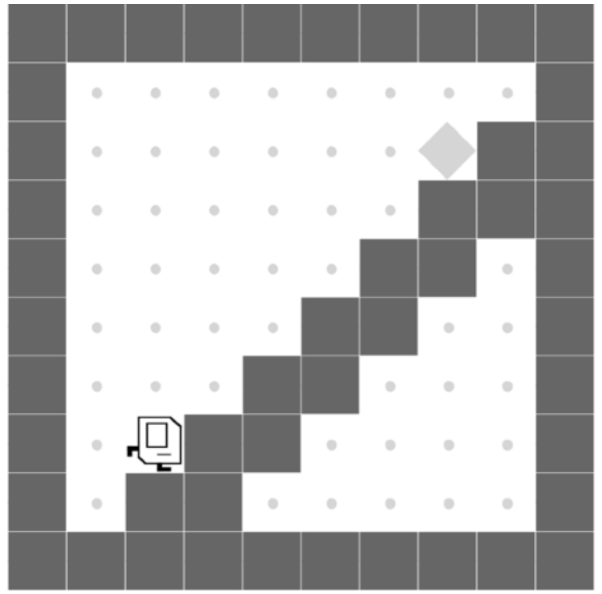
Search for a **task-solving program** using the **cross-entropy method (CEM)**



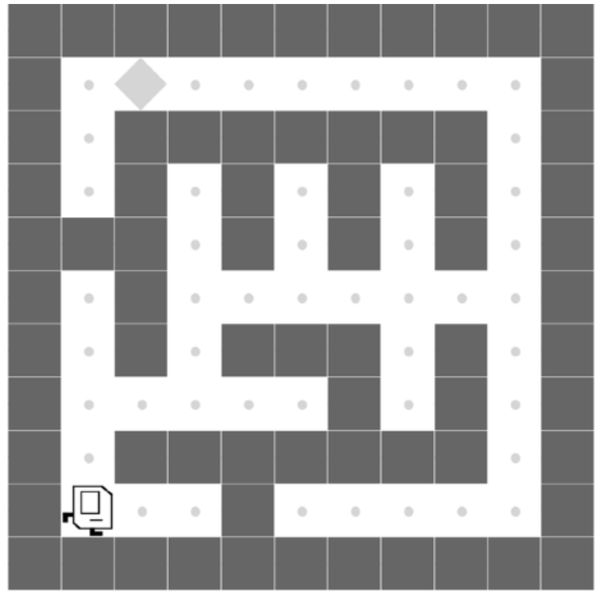
Trivedi et al. "Learning to Synthesize Programs as Interpretable and Generalizable Policies." NeurIPS 2021.

Karel Tasks

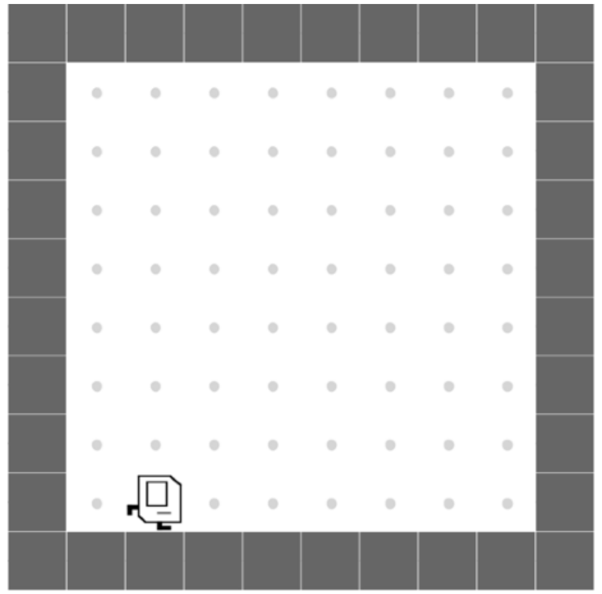
StairClimber



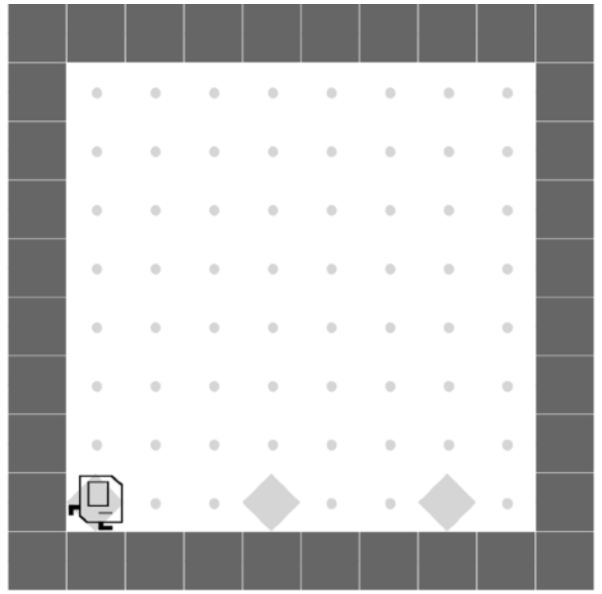
Maze



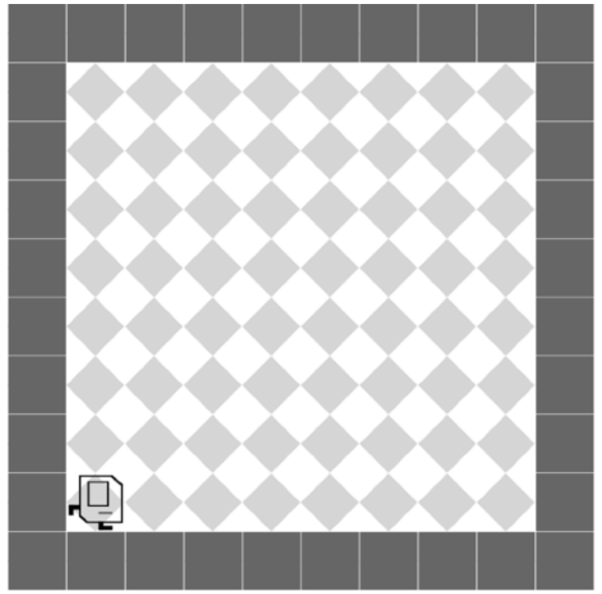
FourCorners



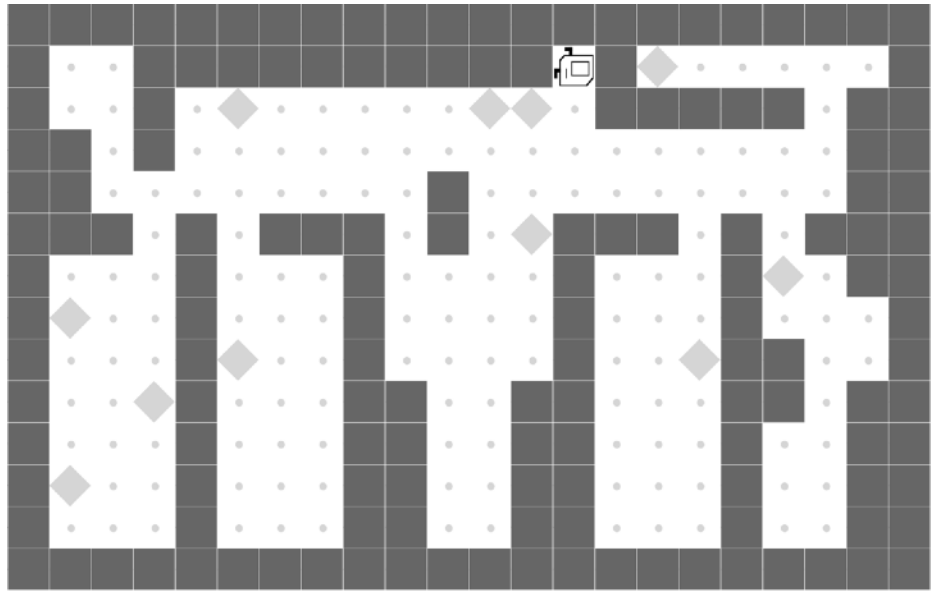
TopOff



Harvester

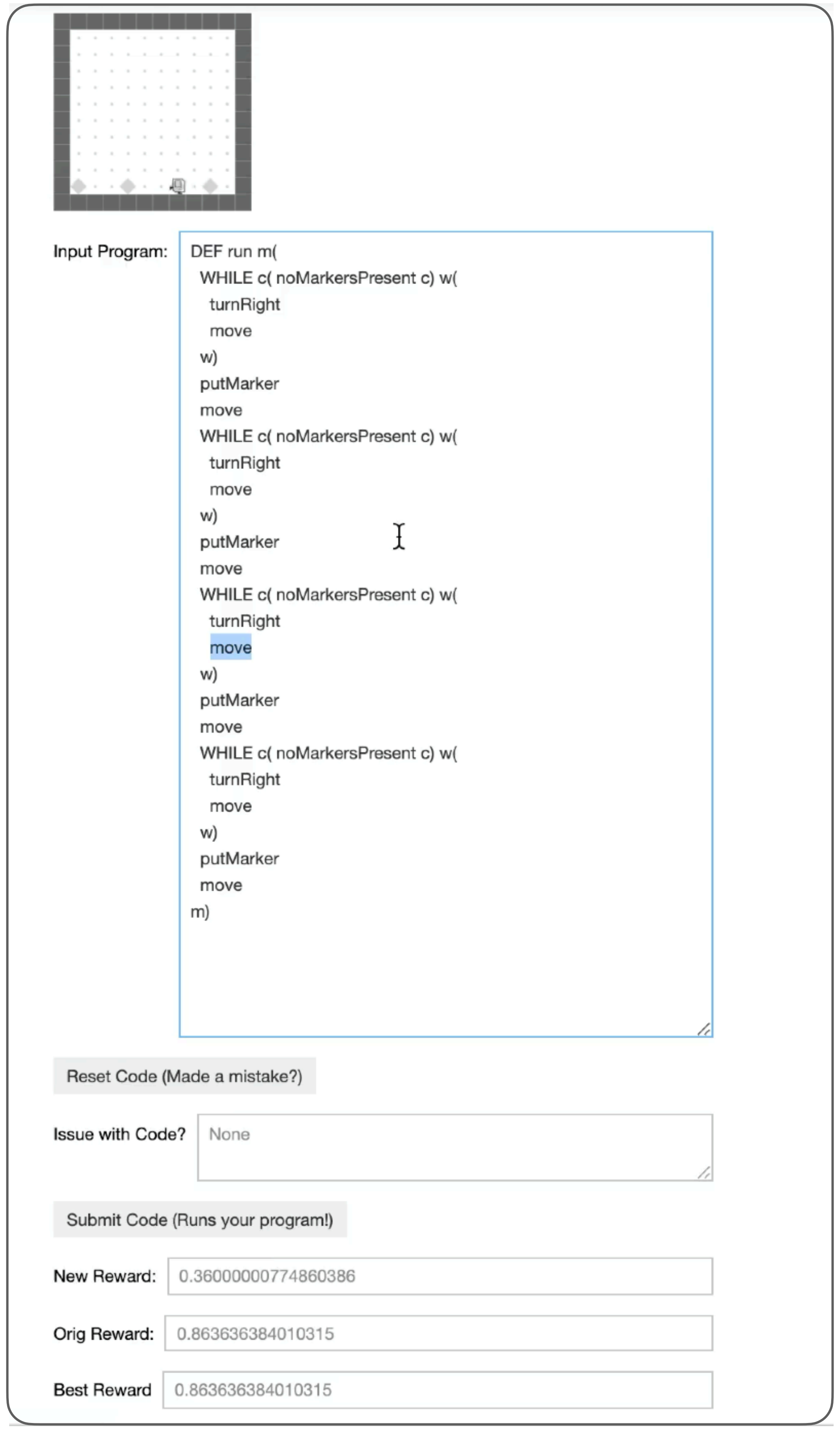


CleanHouse

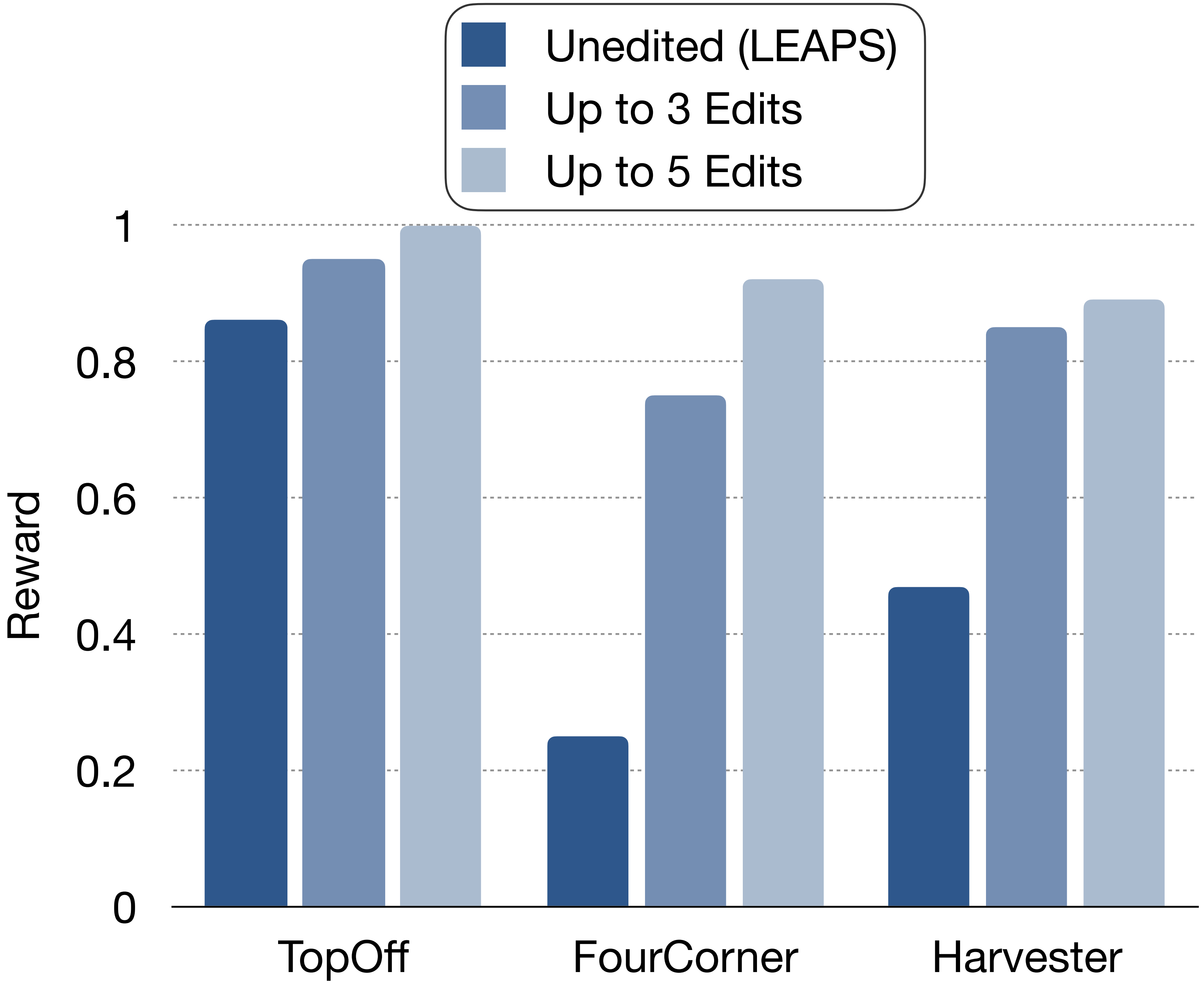


Interpretability & Interactability

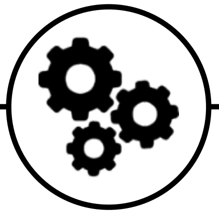
Interactive Debugging Interface



Performance Improvement



Programmatic Reinforcement Learning - State Machine

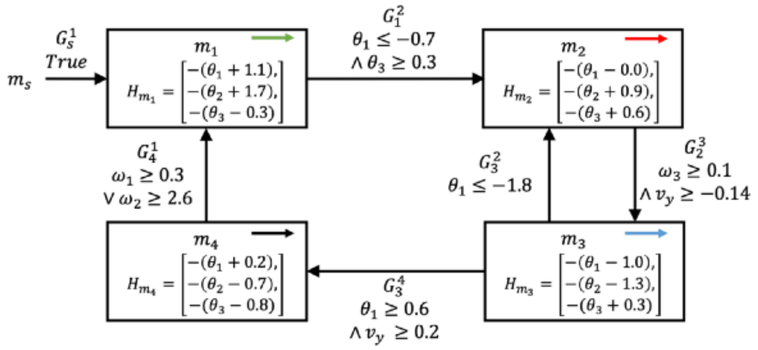


Execute

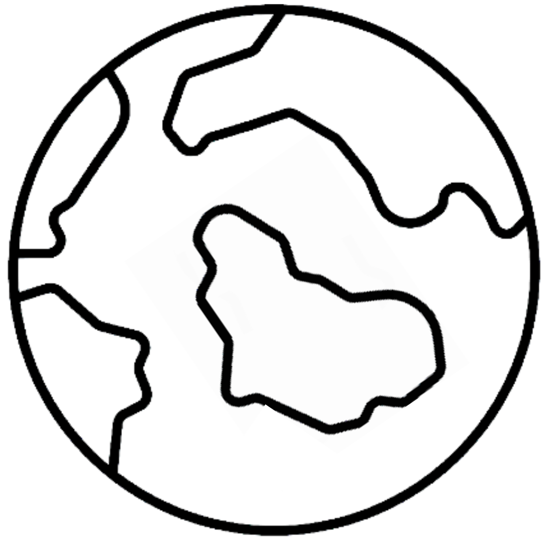
State Machine

- **Inductively generalize** to longer-horizon tasks

Environment



Symbolic Modes



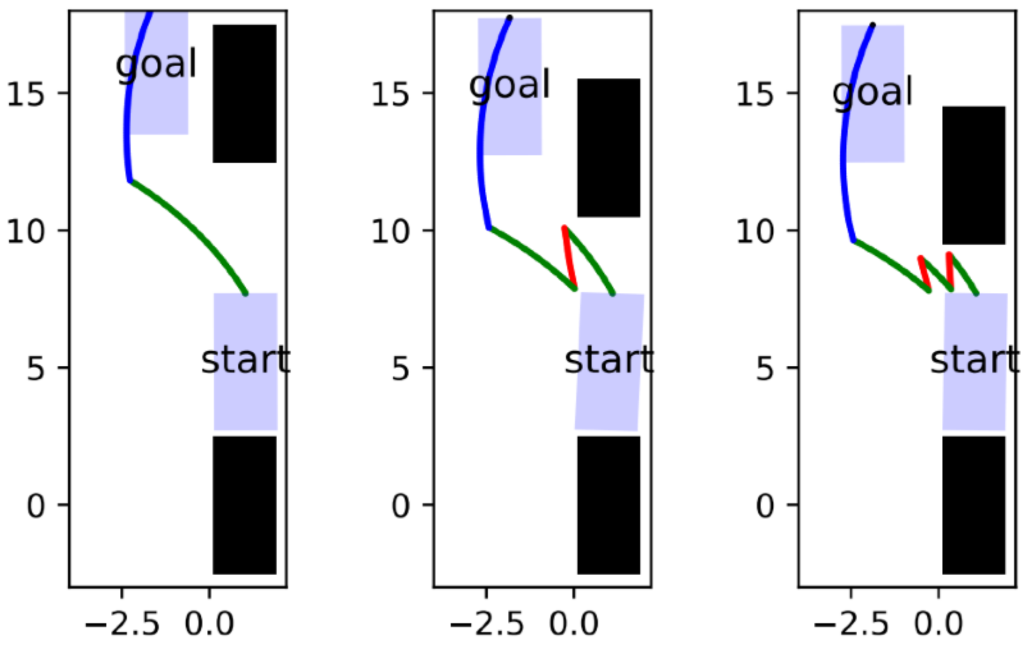
Reward

Koul et al. "Learning Finite State Representations of Recurrent Policy Networks." ICLR 2019.
Inala et al. "Synthesizing Programmatic Policies that Inductively Generalize." ICLR 2020.

Inductive Generalization

Task: Retrieve a car from tight parking spots

Training Tasks

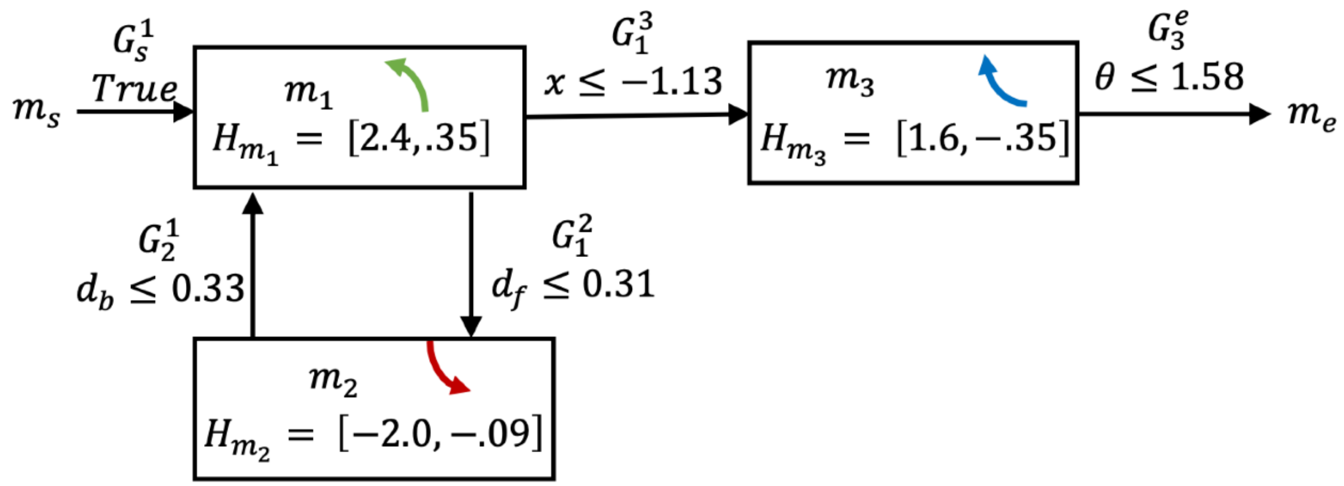


(a) Train 1 (b) Train 2 (c) Train 3

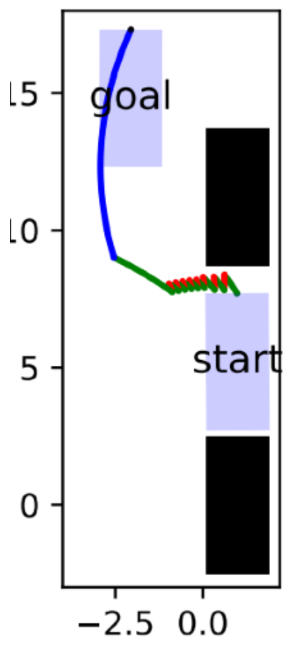
Learned State Machine Policy

Symbolic Modes (i.e., actions) +

Transition Function



Testing Task



(d) Test

This Work: Integrating Program Synthesis and State Machine

Program

```
DEF run() m(
    WHILE c( markerPresent c) w(
        WHILE c( markerPresent c) w(
            pickMarker
            move w)
        turnRight
        move
        turnLeft
        WHILE c( markerPresent c) w(
            pickMarker
            move w)
        turnLeft
        move
        turnRight w) m)
```

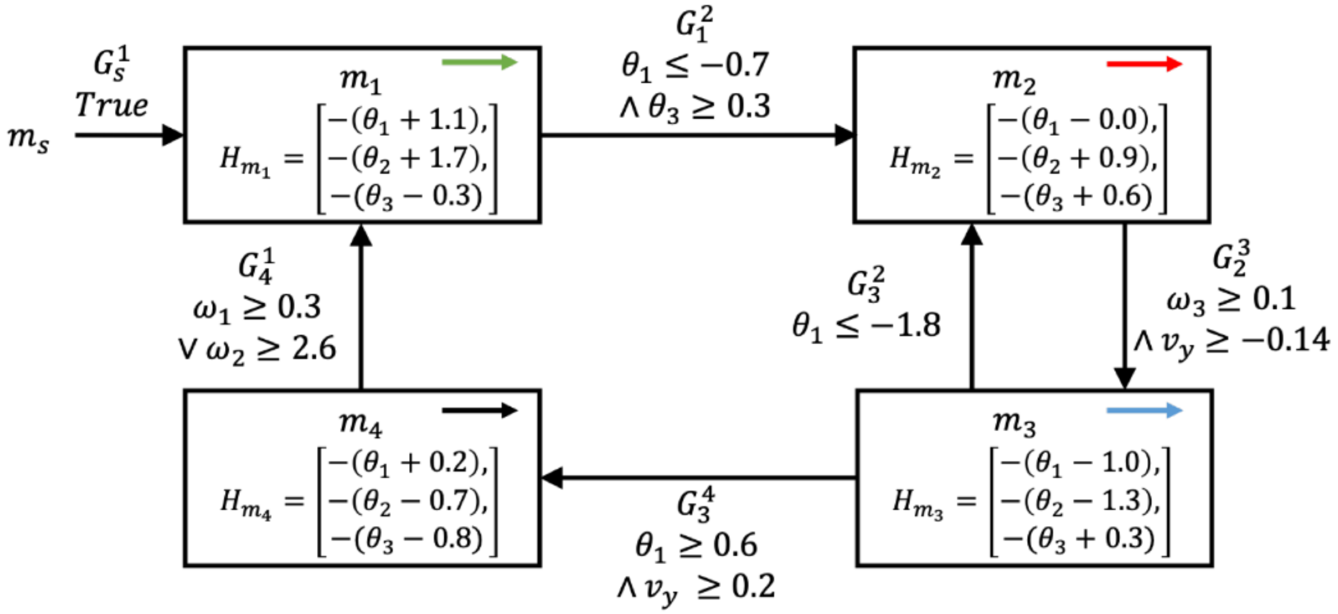
Pros

- Interpretable

Cons

- Often fail to generalize to longer-horizon tasks

State Machine



Pros

- Inductively generalizable

Cons

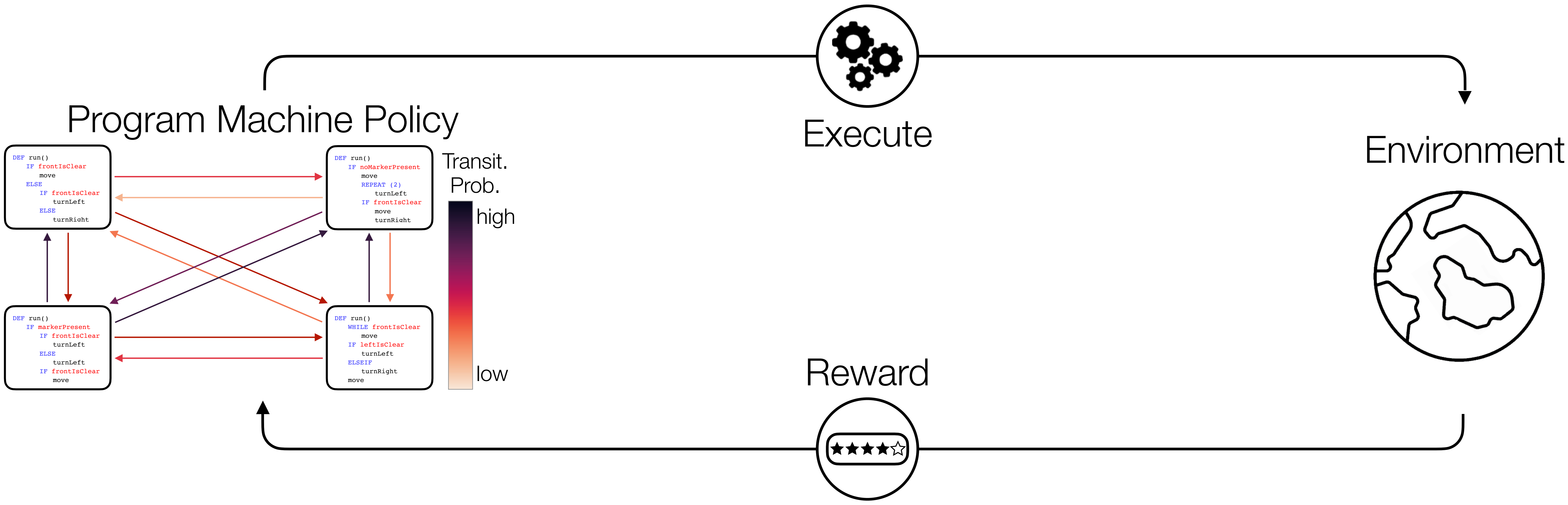
- Difficult to interpret

PrOgram Machine Policy (POMP)

Modes: A set of **programs** that can be executed

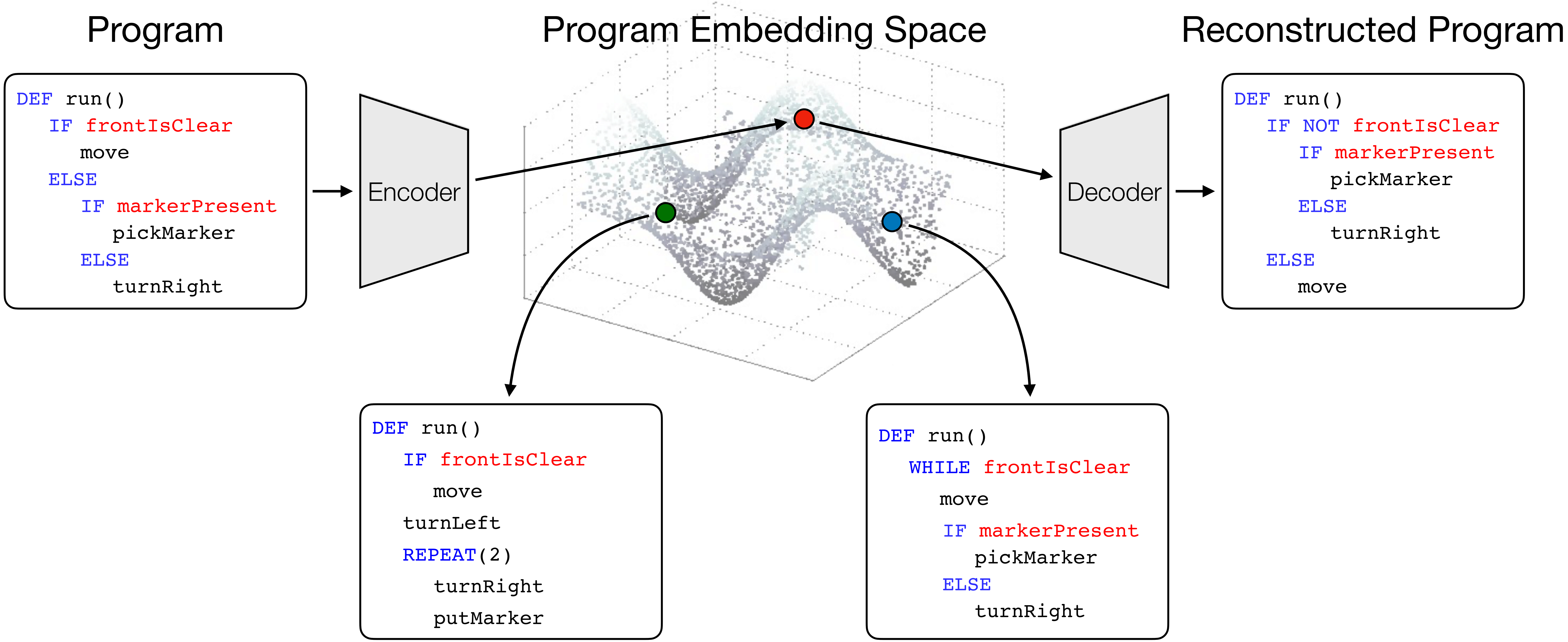
Transition function: Alter between a set of program modes

- Mapping: current environment state **x** current mode → next mode



Step 1 - Learning a Program Embedding Space

Goal: Learn the **grammar** and the **environment dynamics**



Step 2 - Retrieving Effective, Diverse, Compatible Programs

Goal: Retrieve a set of programs as state machine modes

Effectiveness: Retrieved programs should (partially) solve the task

→ Task reward: $\sum_{t=0}^{T} \gamma^t \mathbb{E}_{(s_t,a_t)\sim EXEC(\rho_z)}[r_t]$

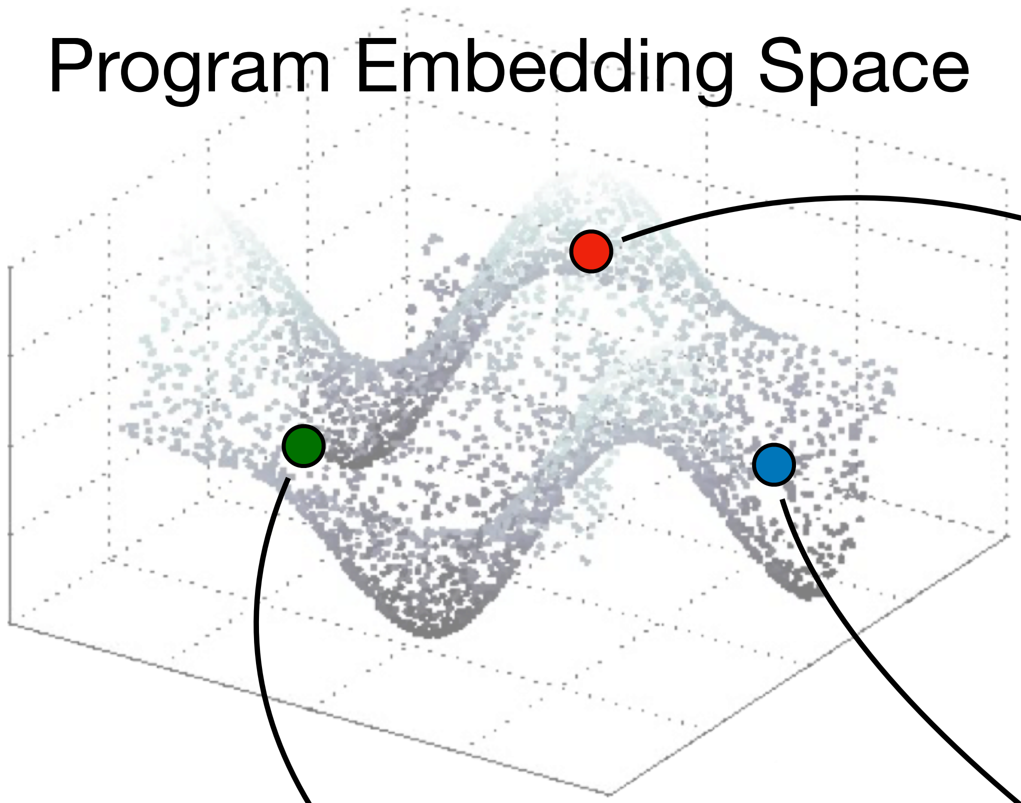
Diversity: Retrieved programs should induce non-overlapping, diverse behaviors

→ Diversity bonus: $-\max_{z_i \in Z} \frac{z \cdot z_i}{\|z\| \|z_i\|}$

Compatibility: Composing retrieved programs in some order should yield good performance

→ Randomly execute retrieved programs before/after executing the current program candidate

Program Embedding Space



Mode 1

```
DEF run()
   IF NOT frontIsClear
      IF markerPresent
         pickMarker
      ELSE
         turnRight
   ELSE
      move
```

```
DEF run()
   IF frontIsClear
      move
   turnLeft
   REPEAT(2)
      turnRight
      putMarker
```

Mode 2

```
DEF run()
   WHILE frontIsClear
      move
      IF markerPresent
         pickMarker
      ELSE
         turnRight
```

Mode 3

Step 3 - Learning Transition Function

Goal: Learn a **transition function**, i.e., current environment state **x** current mode → next mode

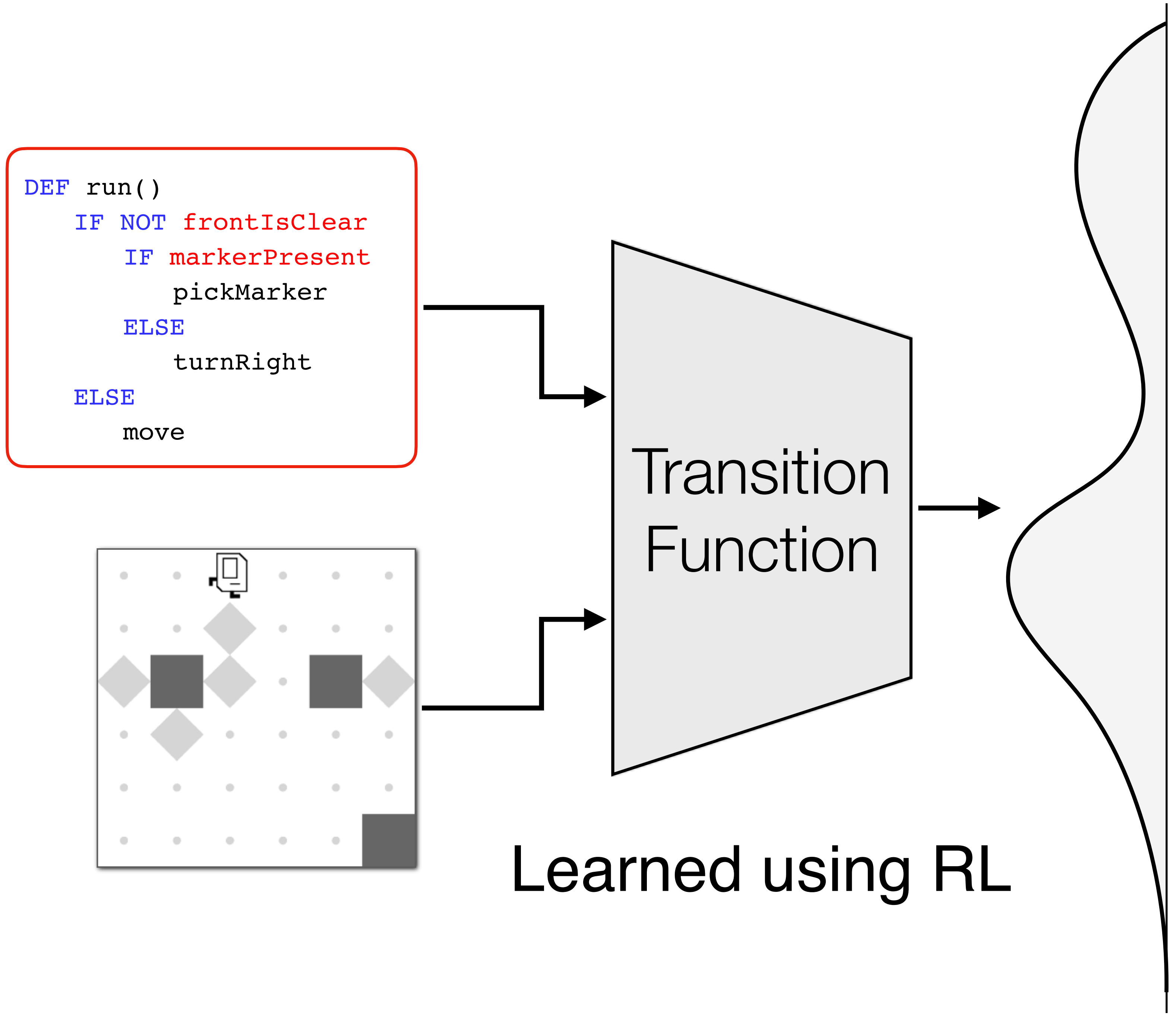
Modes (Retrieved Programs)

```
DEF run()
    IF NOT frontIsClear
        IF markerPresent
            pickMarker
        ELSE
            turnRight
    ELSE
        move
```

```
DEF run()
    WHILE frontIsClear
        move
        IF markerPresent
            pickMarker
        ELSE
            turnRight
```

```
DEF run()
    IF frontIsClear
        move
    turnLeft
    REPEAT(2)
        turnRight
        putMarker
```

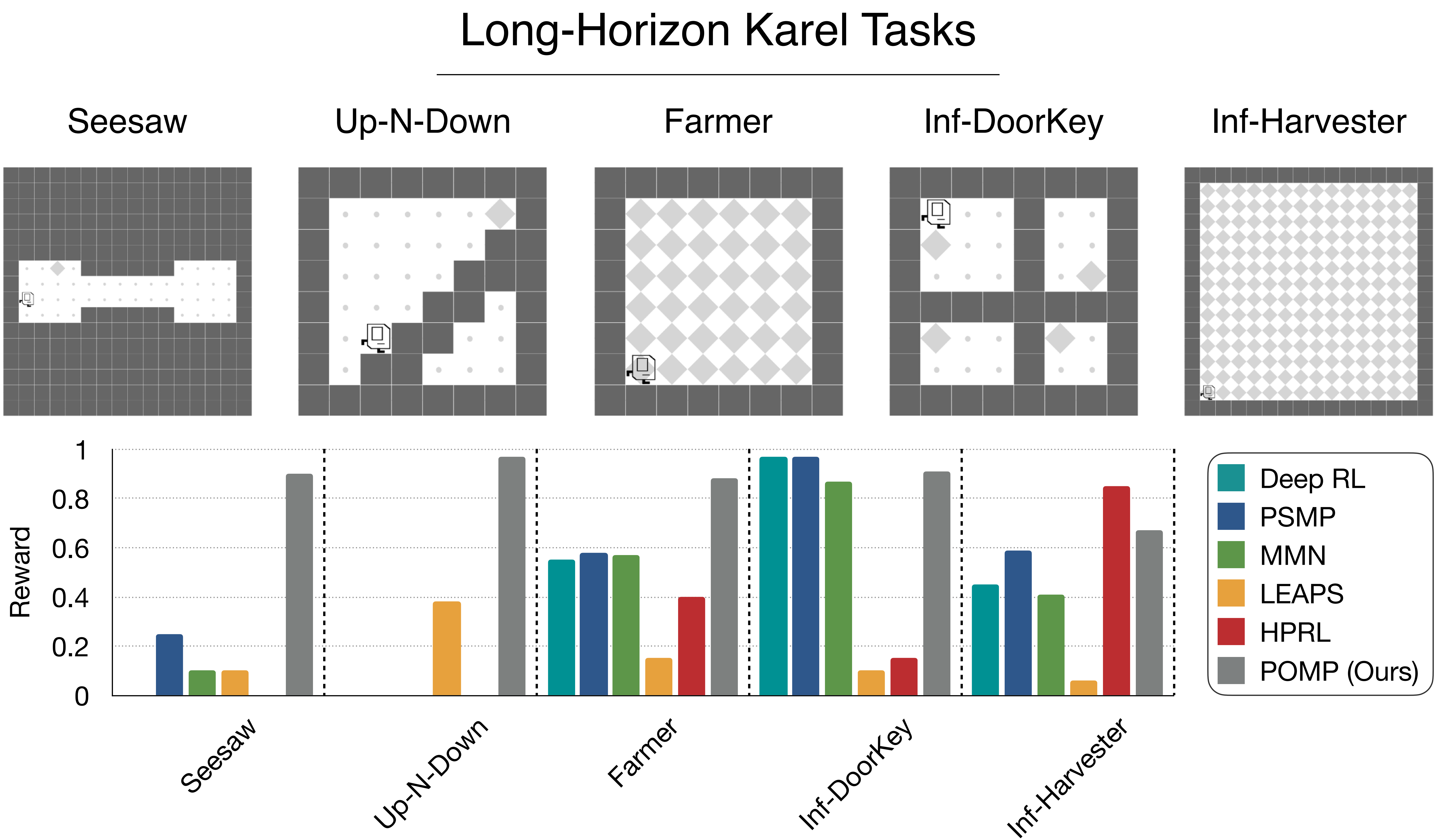
Transition Probability



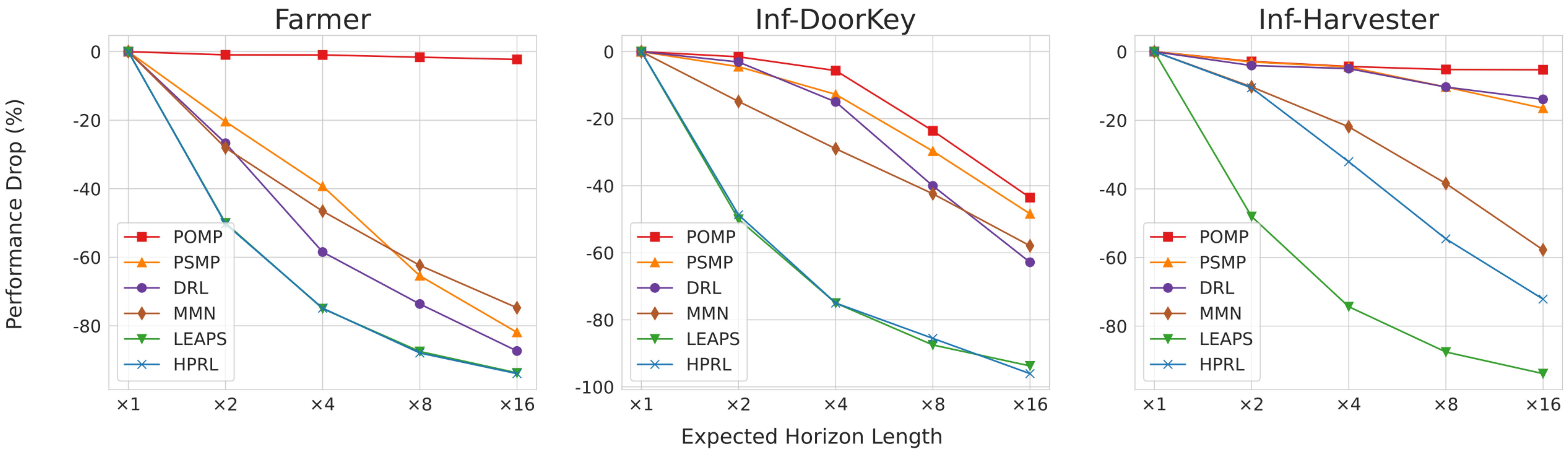

```
DEF run()
    IF NOT frontIsClear
        IF markerPresent
            pickMarker
        ELSE
            turnRight
    ELSE
        move
```

```
DEF run()
    WHILE frontIsClear
        move
        IF markerPresent
            pickMarker
        ELSE
            turnRight
```

```
DEF run()
    IF frontIsClear
        move
    turnLeft
    REPEAT(2)
        turnRight
        putMarker
```


Long-Horizon Karel Tasks
Seesaw
Up-N-Down
Farmer
Inf-DoorKey
Inf-Harvester
Reward
1
0.8
0.6
0.4
0.2
0
Deep RL
PSMP
MMN
LEAPS
HPRL
POMP (Ours)
Seesaw
Up-N-Down
Farmer
Inf-DoorKey
Inf-Harvester

Inductive Generalization



Learned Program Machine Policy

Task: Inf-DoorKey

Retrieved Programs

Mode 1

```
DEF run m(
    WHILE c(
        noMarkersPresent c)
        w(
        WHILE c(
            noMarkersPresent
            c) w(
            turnRight
            move
            w)
        pickMarker
        putMarker
        w)
    WHILE c( rightIsClear c)
        w(
        move
        w)
    WHILE c( rightIsClear c)
        w(
        move
        w)
    pickMarker
    putMarker
    move
    m)
```

Mode 2

```
DEF run m(
    WHILE c(
        noMarkersPresent c)
        w(
        move
        turnLeft
        w)
    WHILE c(
        noMarkersPresent c)
        w(
        move
        w)
    m)
```

Mode 3

```
DEF run m(
    move
    turnLeft
    move
    turnLeft
    move
    turnLeft
    m)
```

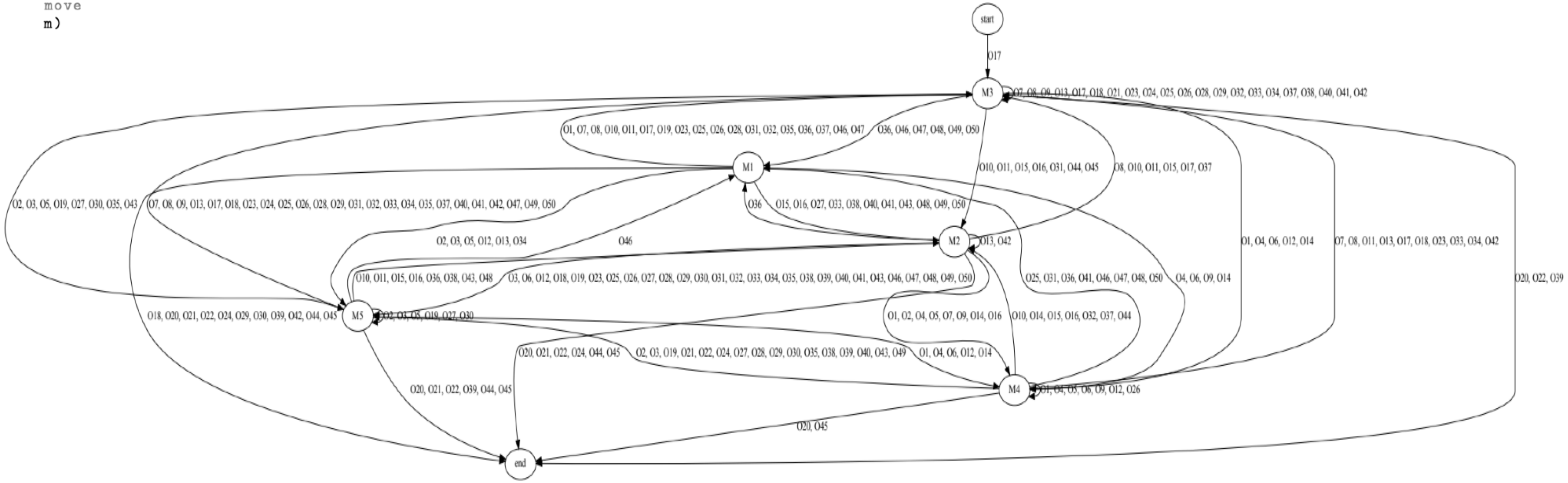
Mode 4

```
DEF run m(
    pickMarker
    IF c( not c( leftIsClear
        c) c) i(
        move
        i)
    IF c( not c( leftIsClear
        c) c) i(
        turnRight
        i)
    m)
```

Mode 5

```
DEF run m(
    putMarker
    REPEAT R=13 r(
        pickMarker
        turnRight
        move
        putMarker
        move
        pickMarker
        move
        turnRight
        move
        r)
    move
    pickMarker
    move
    move
    m)
```

Extracted State Machine

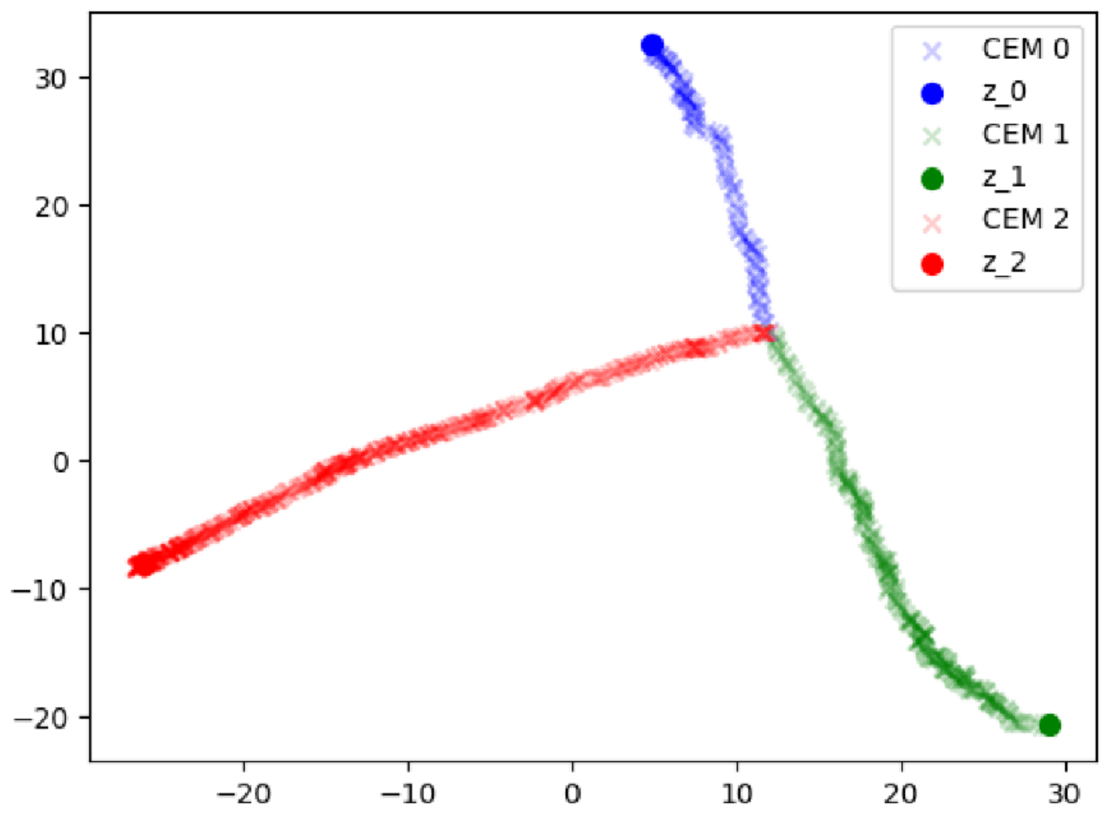


Ablation Study

Diversity & Compatibility

Method	SEESAW	UP-N-DOWN	FARMER	INF-DOORKEY	INF-HARVESTER
CEM $\times\|M\|$	0.31 ± 0.18	0.88 ± 0.12	0.22 ± 0.00	0.39 ± 0.46	0.56 ± 0.12
CEM+diversity top k, $k = \|M\|$	0.09 ± 0.11	0.72 ± 0.36	0.23 ± 0.00	0.92 ± 0.01	0.71 ± 0.02
CEM+diversity $\times\|M\|$	0.47 ± 0.39	0.76 ± 0.31	0.24 ± 0.03	0.89 ± 0.02	0.66 ± 0.07
POMP (Ours)	**0.90** ± 0.02	**0.97** ± 0.00	**0.88** ± 0.01	0.91 ± 0.01	0.67 ± 0.03

CEM Search Trajectories with Diversity Bonus

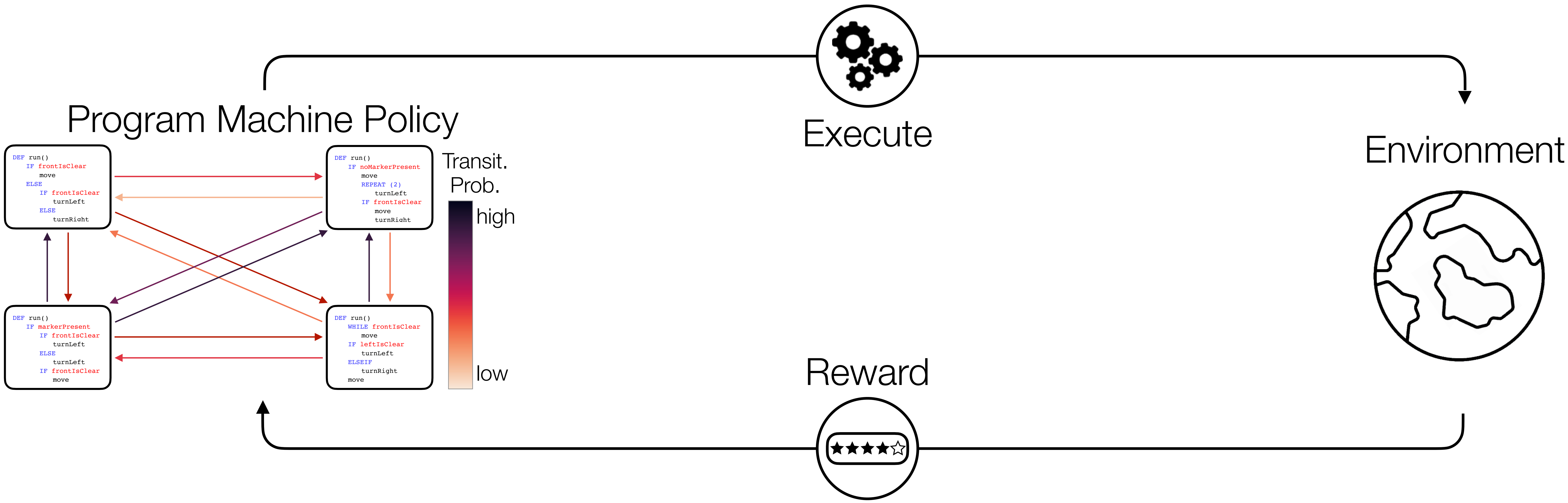



Addressing Long-Horizon Tasks by Integrating Program Synthesis and State Machines
SCAN ME

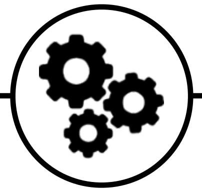
Takeaway

Program Synthesis × State Machine =

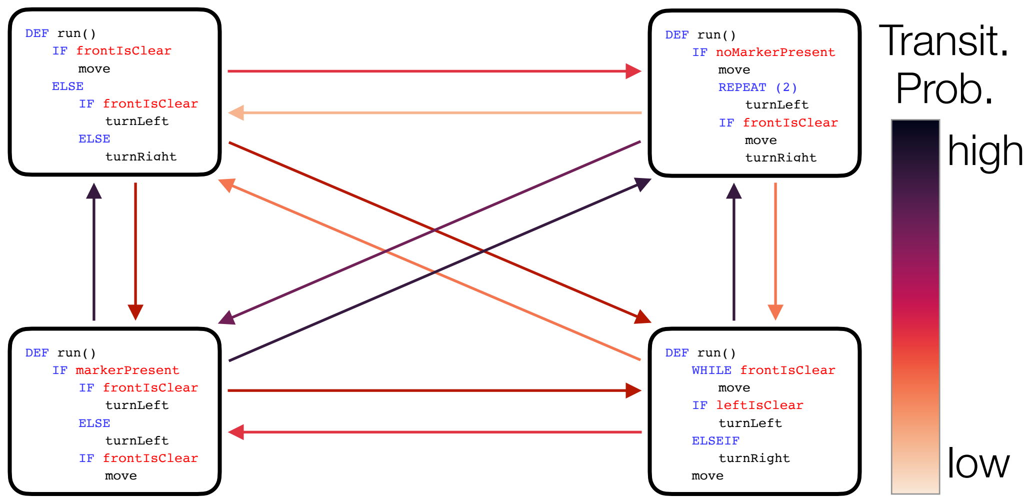
Interpretable and Inductively Generalizable Policies



Execute



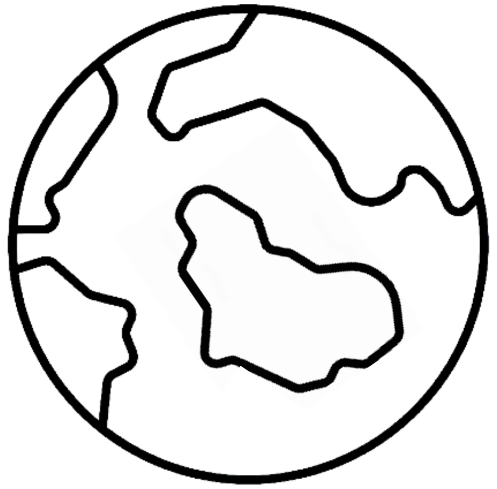
Program Machine Policy



Thank You

Questions?

Environment



Reward