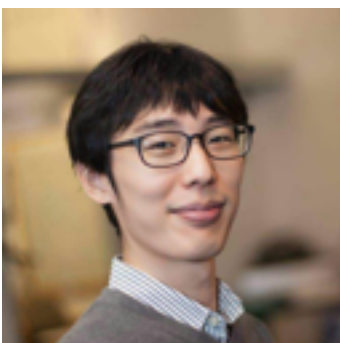




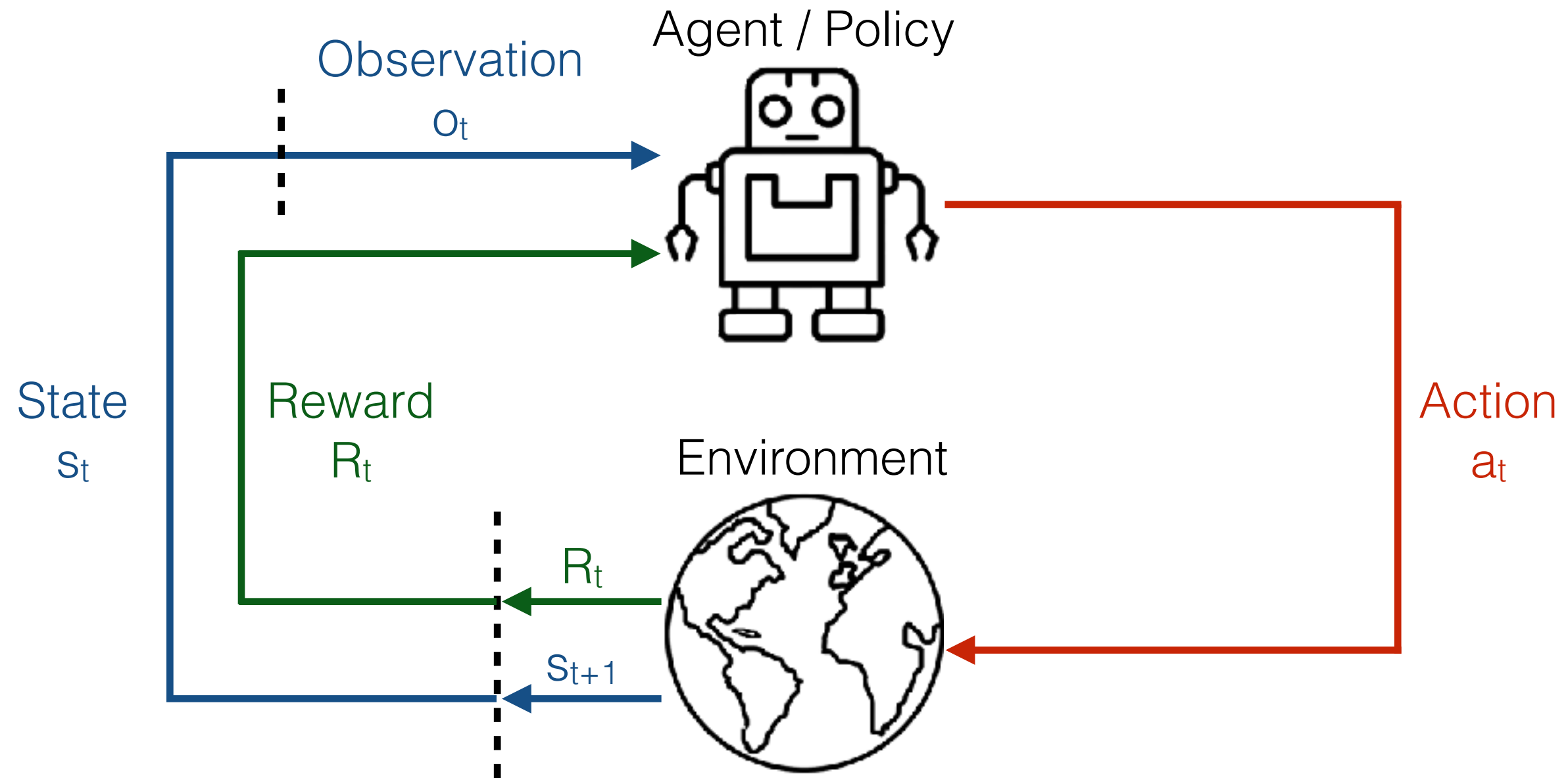
Learning to Synthesize Programs as *Interpretable* and *Generalizable Policies*

Shao-Hua Sun

Ph.D. candidate in Computer Science
at the University of Southern California (USC)

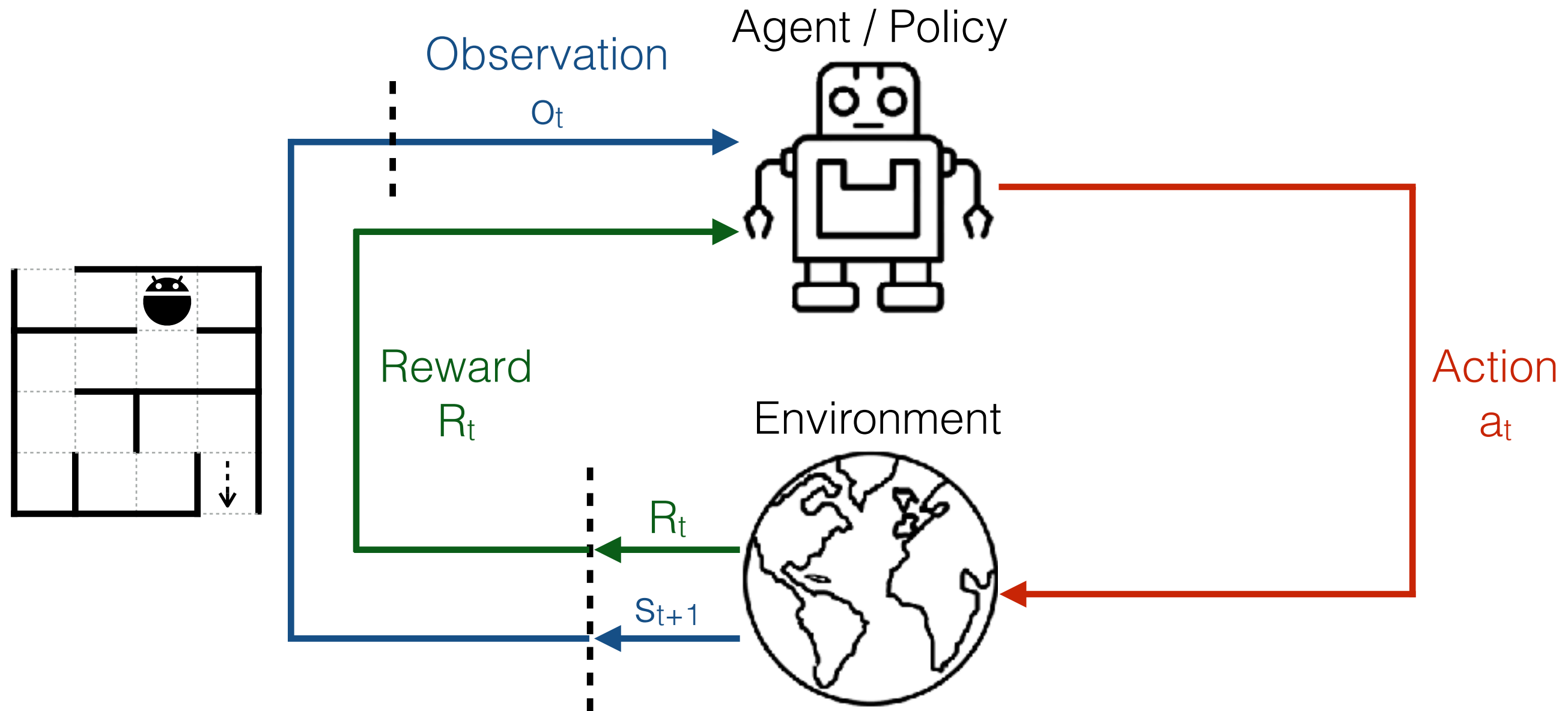


Reinforcement Learning



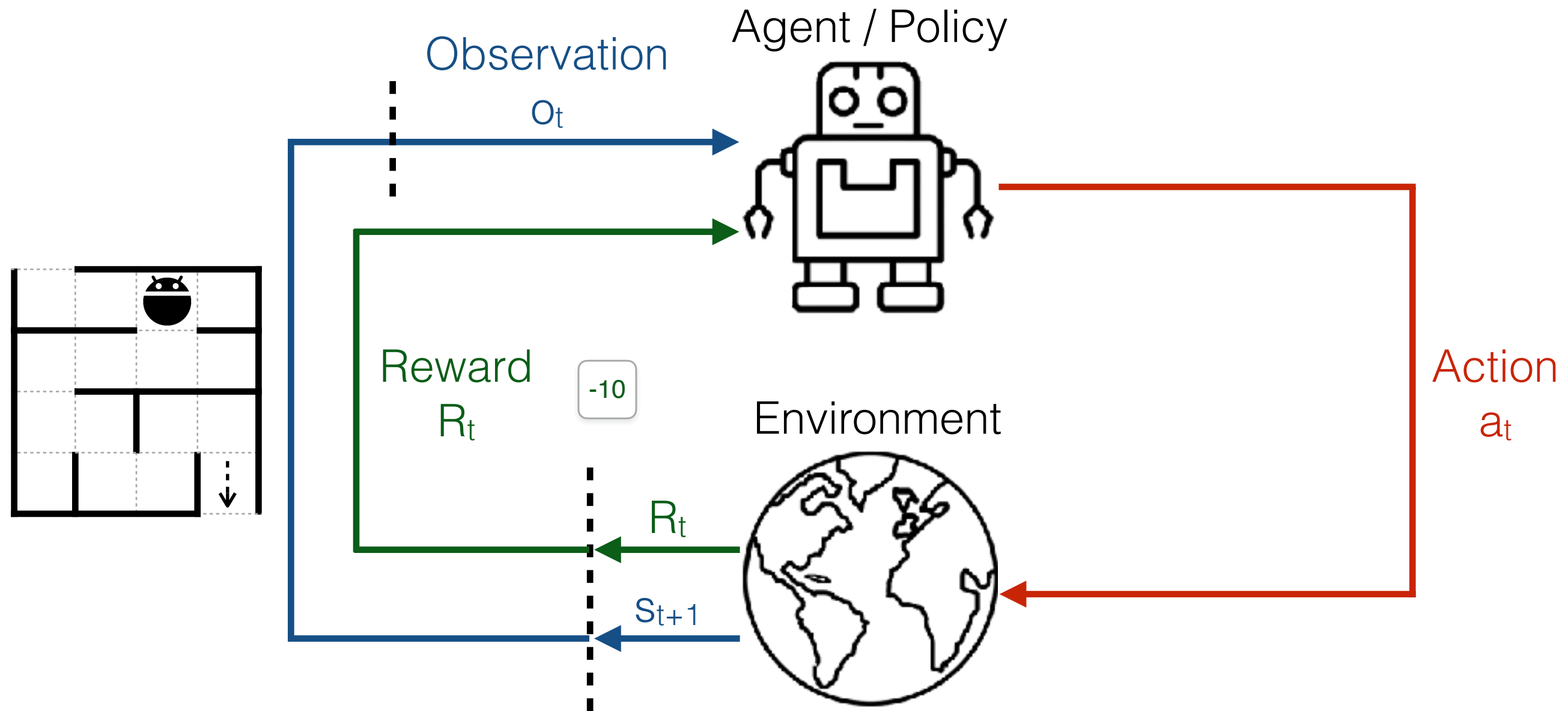
Goal: maximize $\sum_{t=0}^{t=H} \gamma^t R_t(s_t, a_t)$

Reinforcement Learning



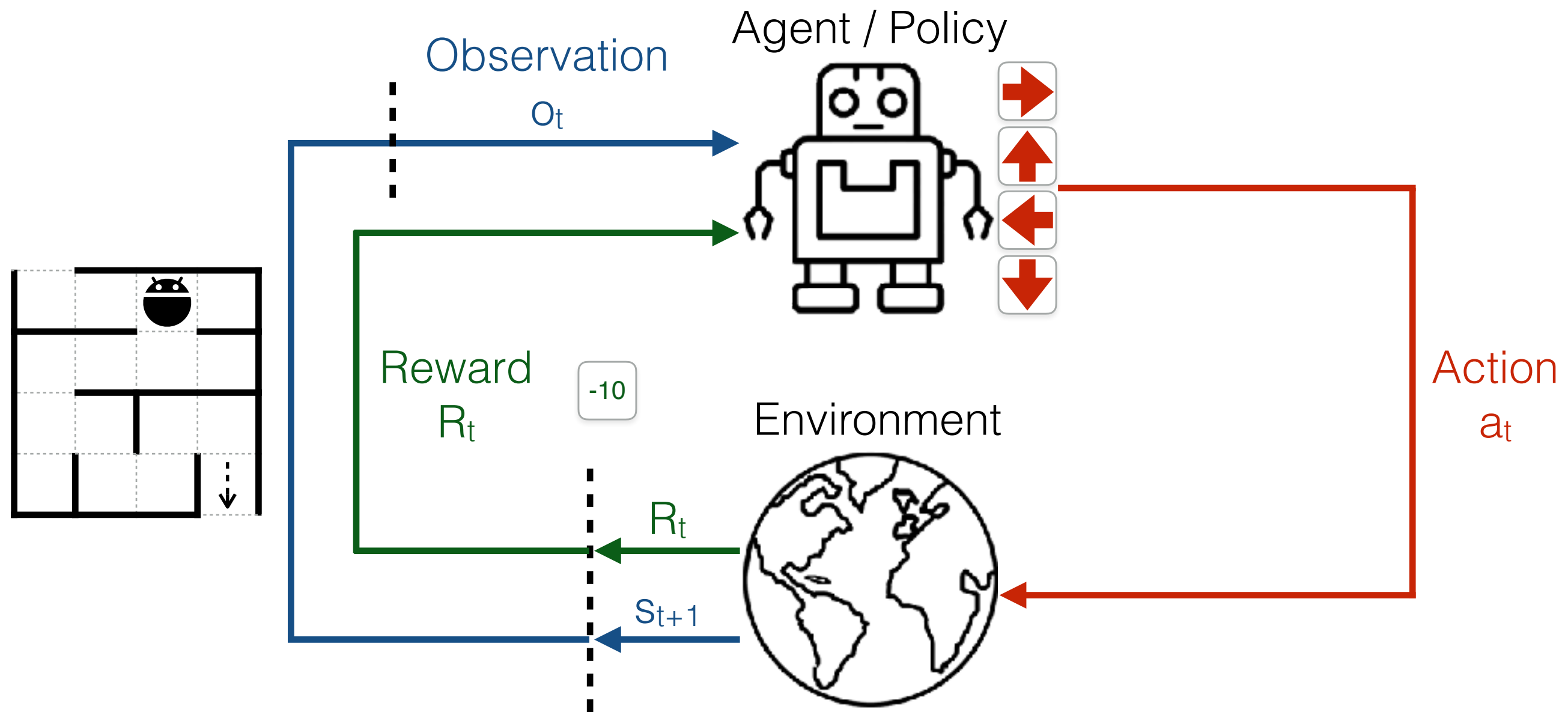
Goal: maximize $\sum_{t=0}^{t=H} \gamma^t R_t(s_t, a_t)$

Reinforcement Learning



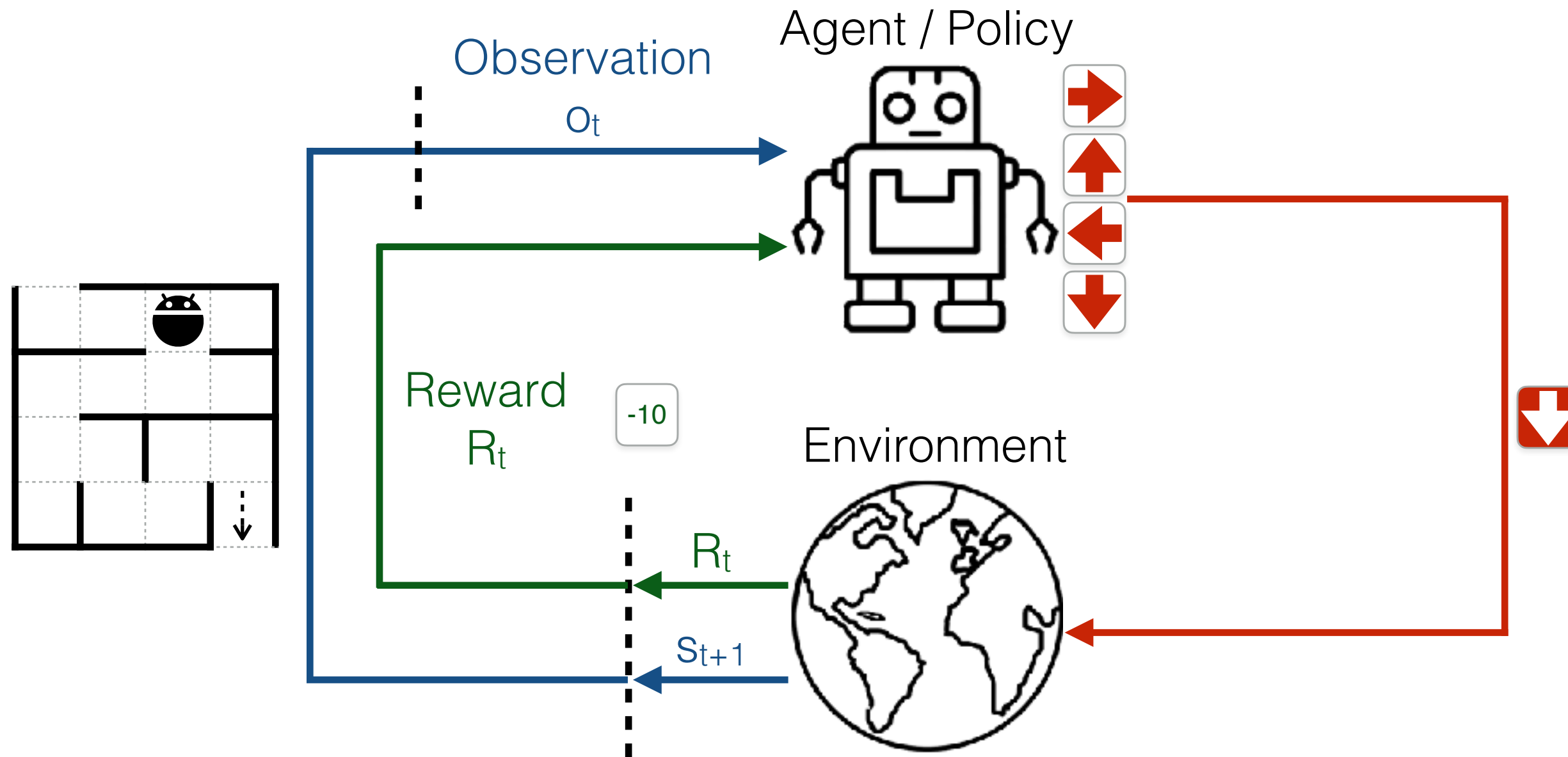
Goal: maximize $\sum_{t=0}^{t=H} \gamma^t R_t(s_t, a_t)$

Reinforcement Learning



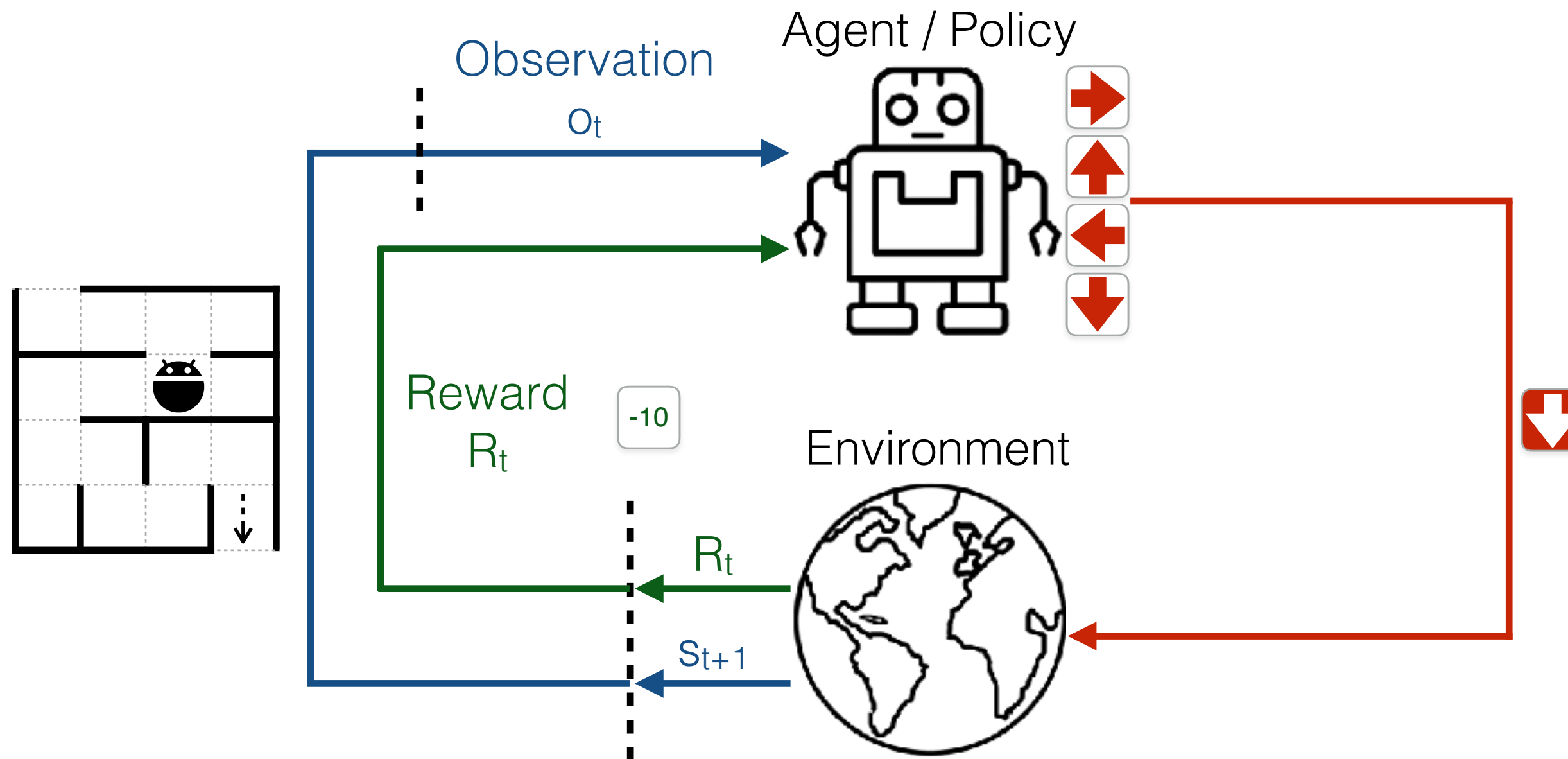
Goal: maximize $\sum_{t=0}^{t=H} \gamma^t R_t(s_t, a_t)$

Reinforcement Learning



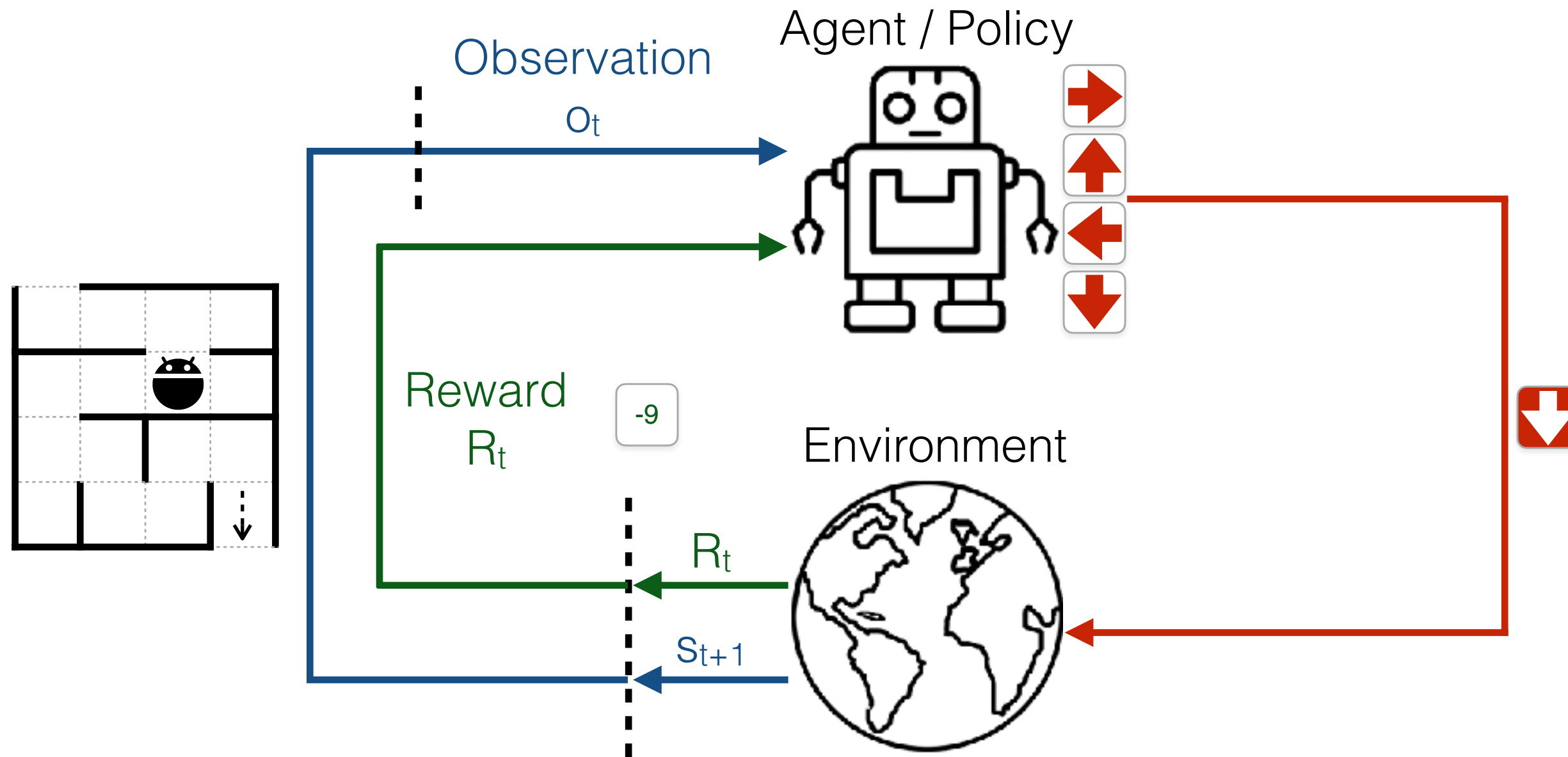
Goal: maximize $\sum_{t=0}^{t=H} \gamma^t R_t(s_t, a_t)$

Reinforcement Learning



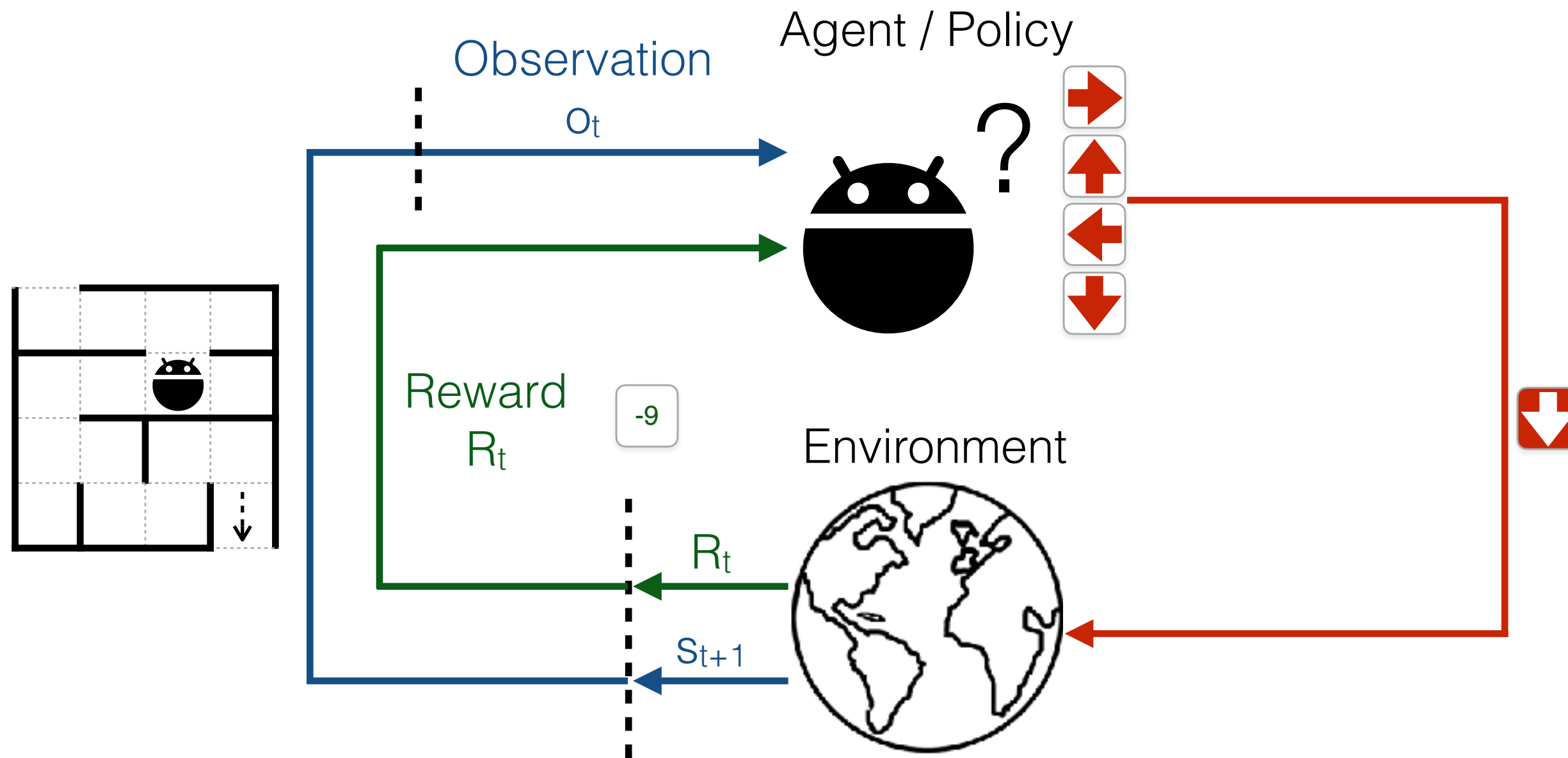
Goal: maximize $\sum_{t=0}^{t=H} \gamma^t R_t(s_t, a_t)$

Reinforcement Learning



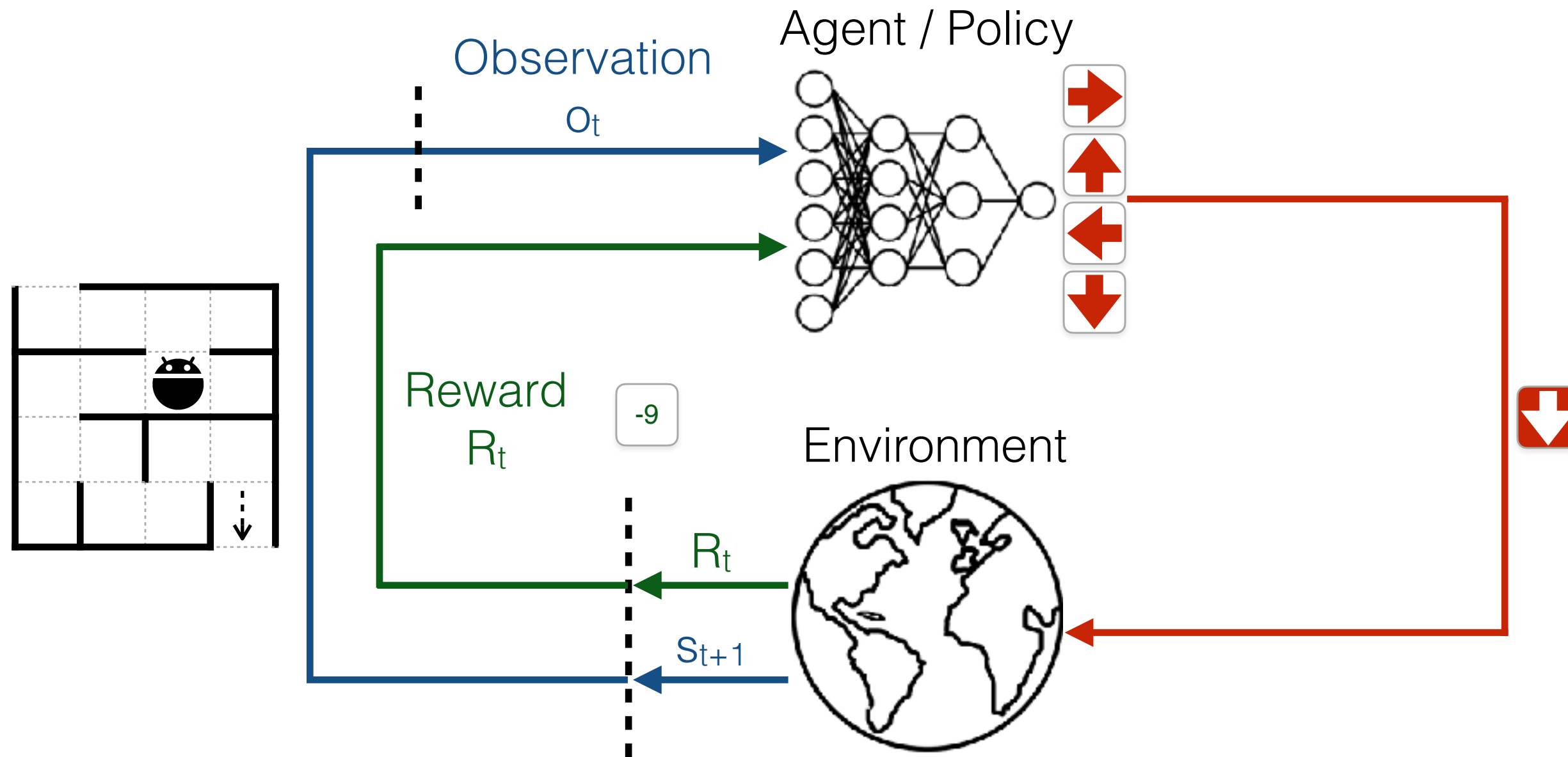
Goal: maximize $\sum_{t=0}^{t=H} \gamma^t R_t(s_t, a_t)$

Reinforcement Learning



Goal: maximize $\sum_{t=0}^{t=H} \gamma^t R_t(s_t, a_t)$

Reinforcement Learning



Goal: maximize $\sum_{t=0}^{t=H} \gamma^t R_t(s_t, a_t)$

Advances in Deep Reinforcement Learning



Robot Manipulation



Robot Locomotion

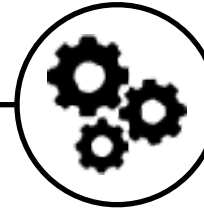


Autonomous Driving



Game AI

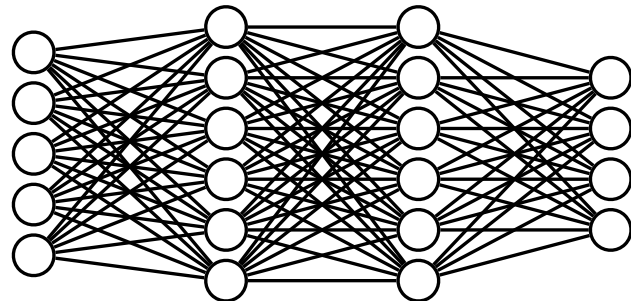
Execute



Environment



Deep Neural Network

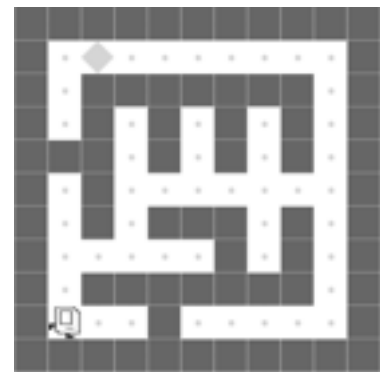


Reward

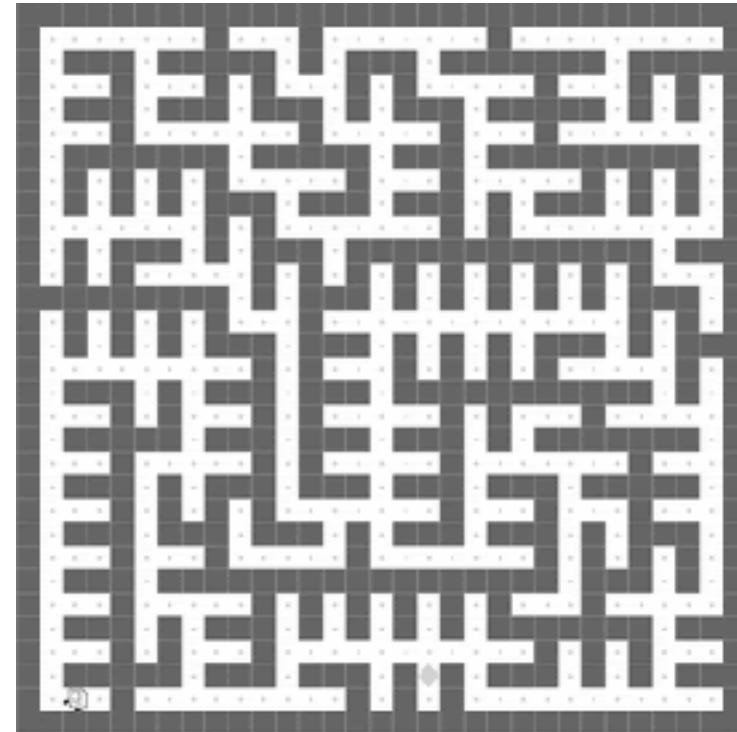


Issues with Deep Reinforcement Learning (DRL)

Generalization



Simple task



Complex task

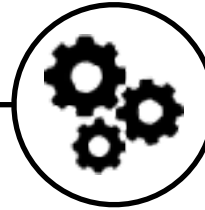
Interpretability

Trust

Safety

Contestability

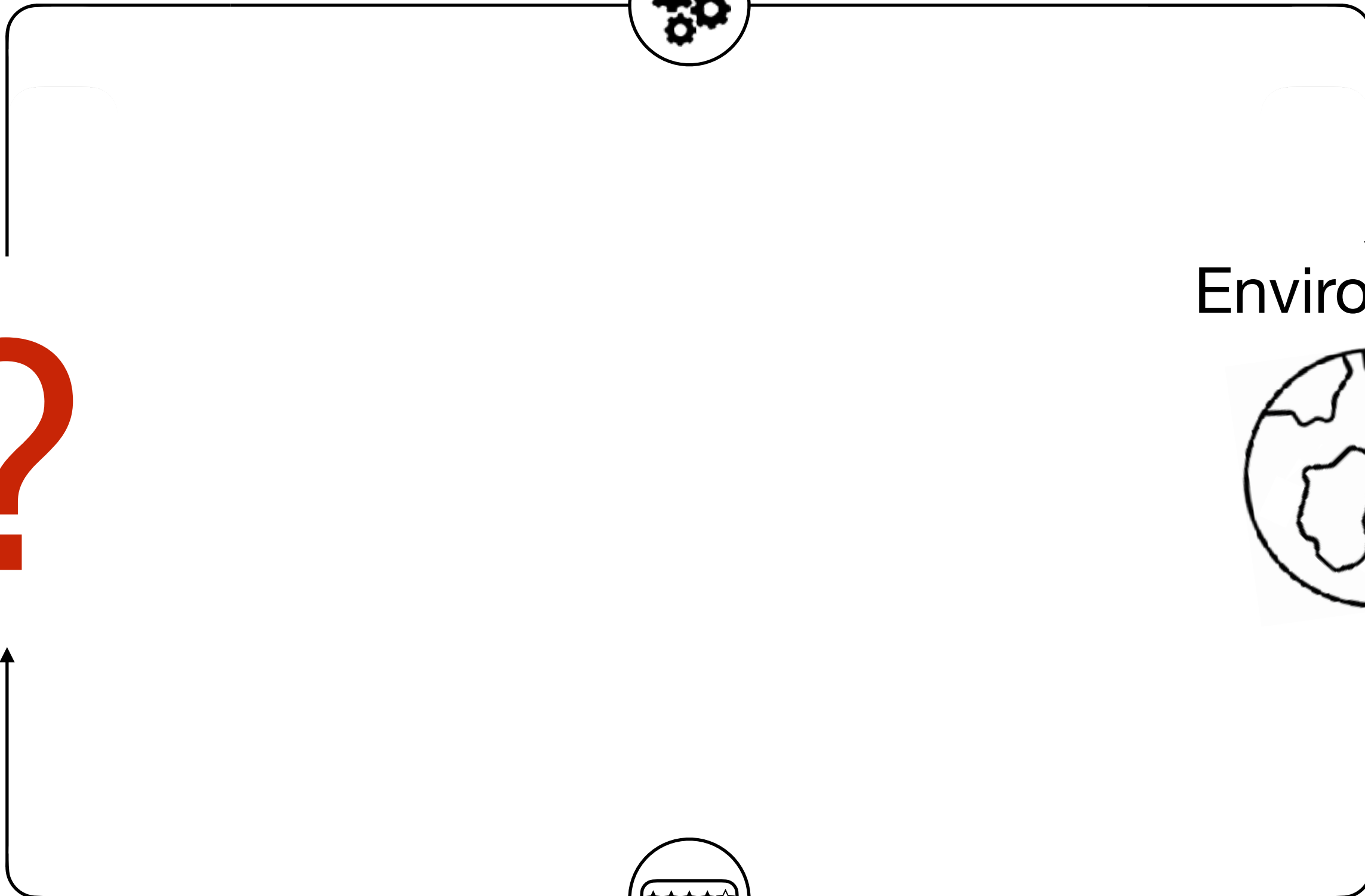
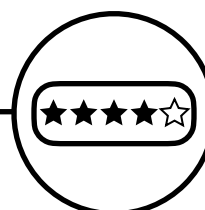
Execute



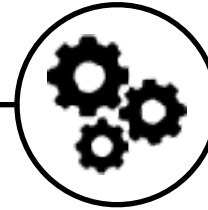
Environment



Reward



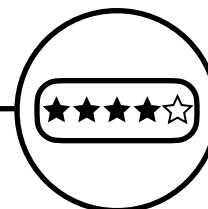
Execute



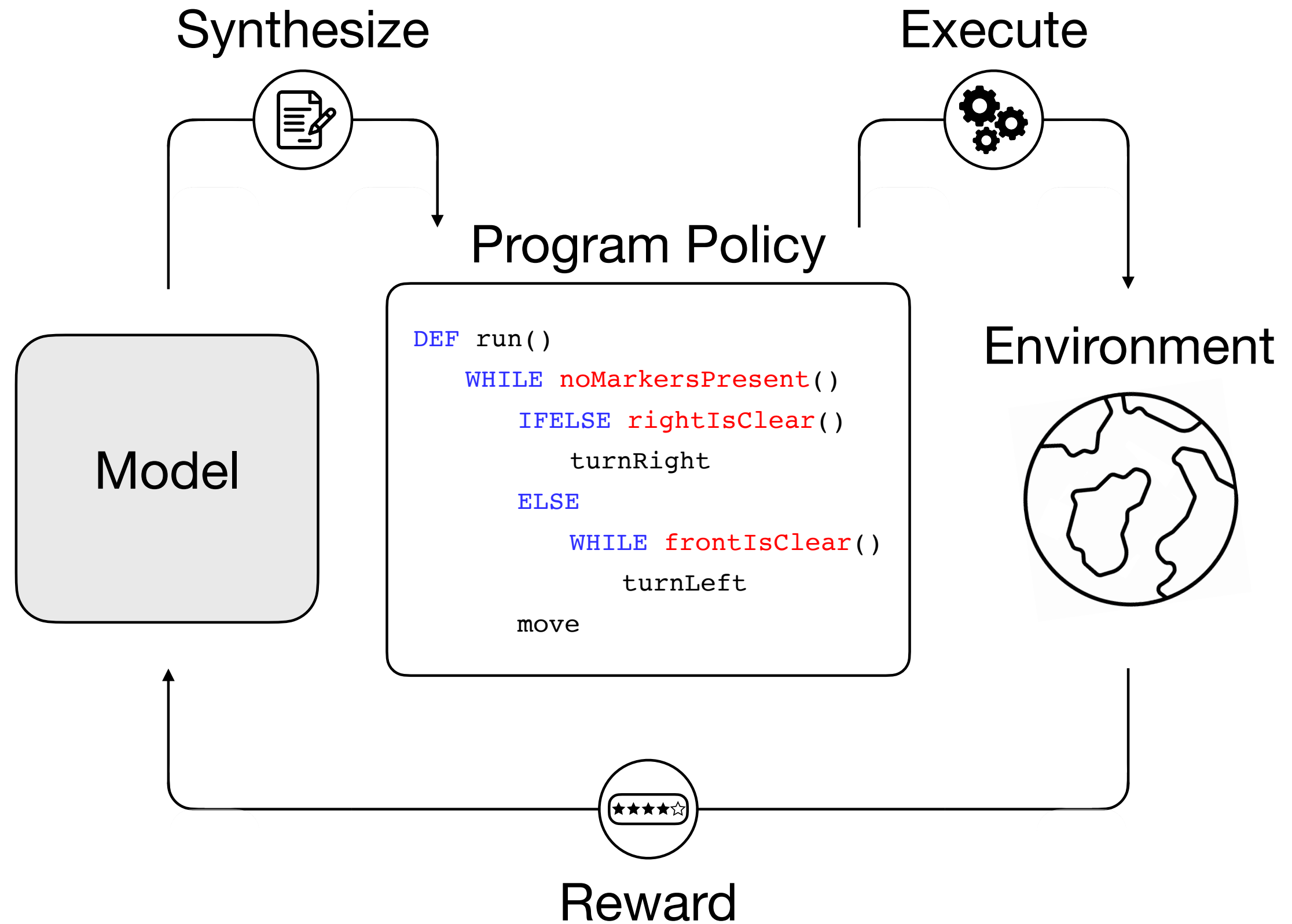
Environment

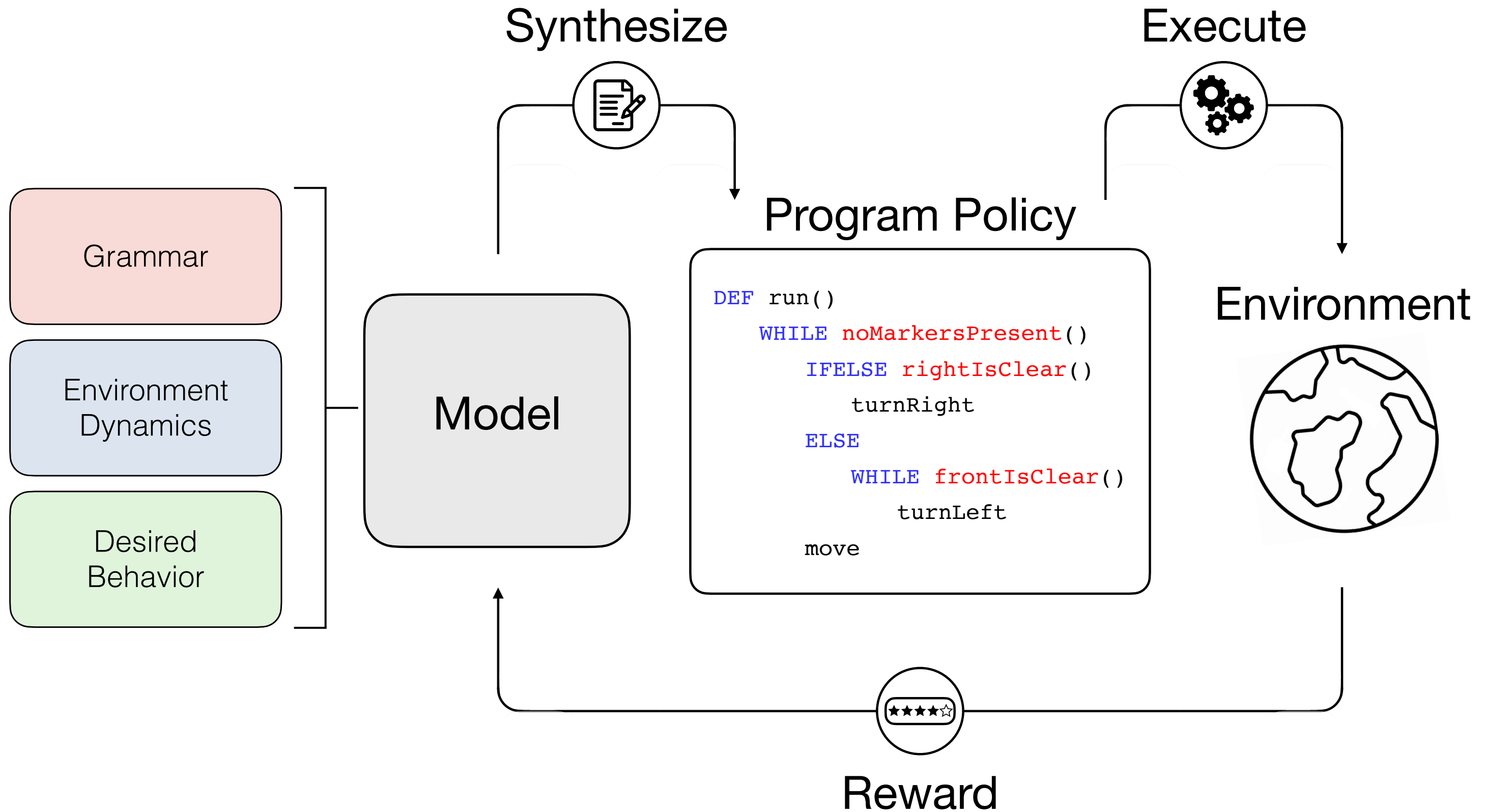


Reward



```
DEF run()  
  IF frontIsClear()  
    move  
  ELSE  
    IF frontIsClear()  
      turnLeft  
    ELSE  
      turnRight
```





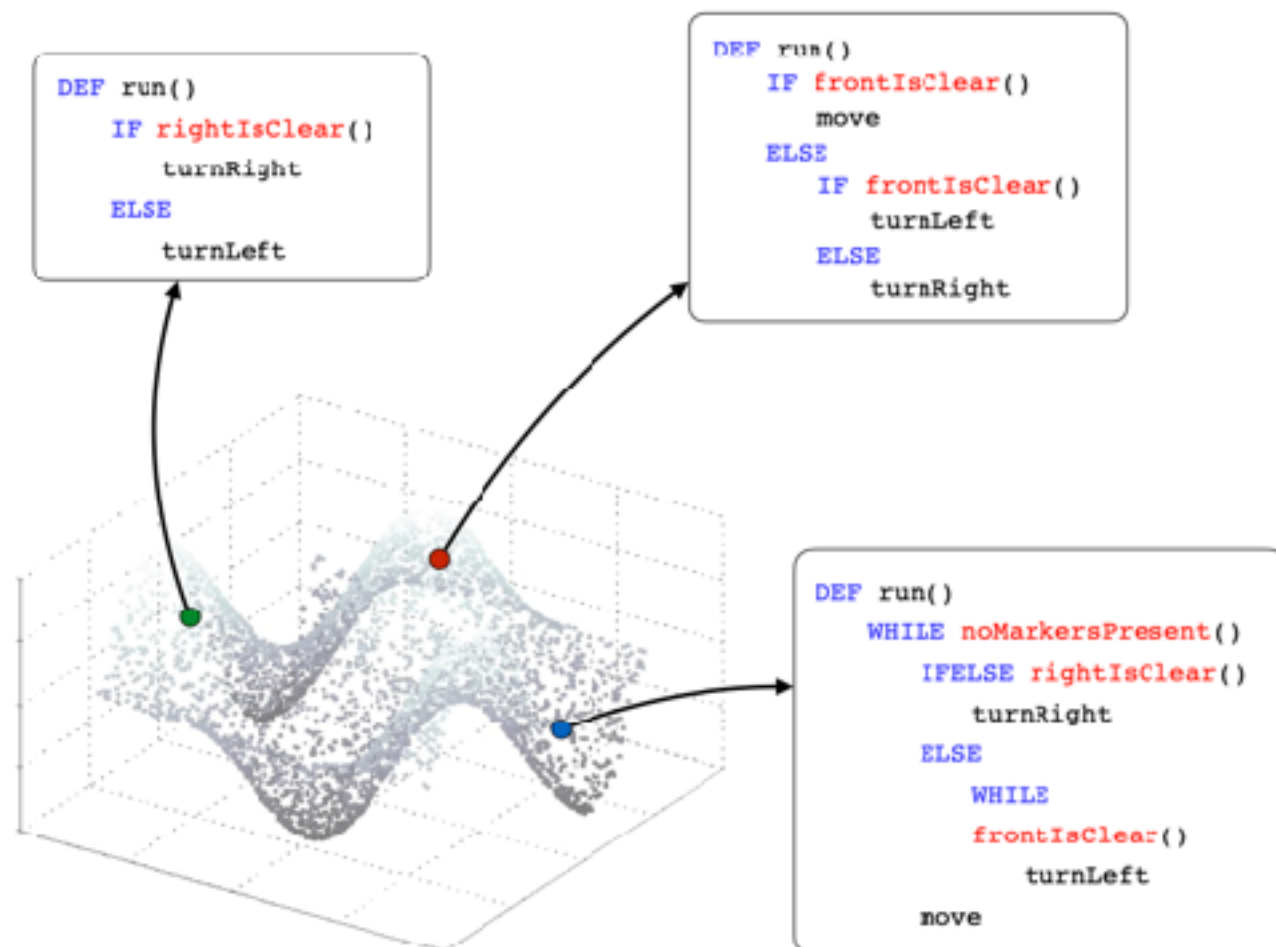
LEAPS: **L**earning **E**mbeddings for **I**ntent **P**rogram **S**ynthesis

Stage 1

Learning a program embedding space from randomly generated programs

Grammar

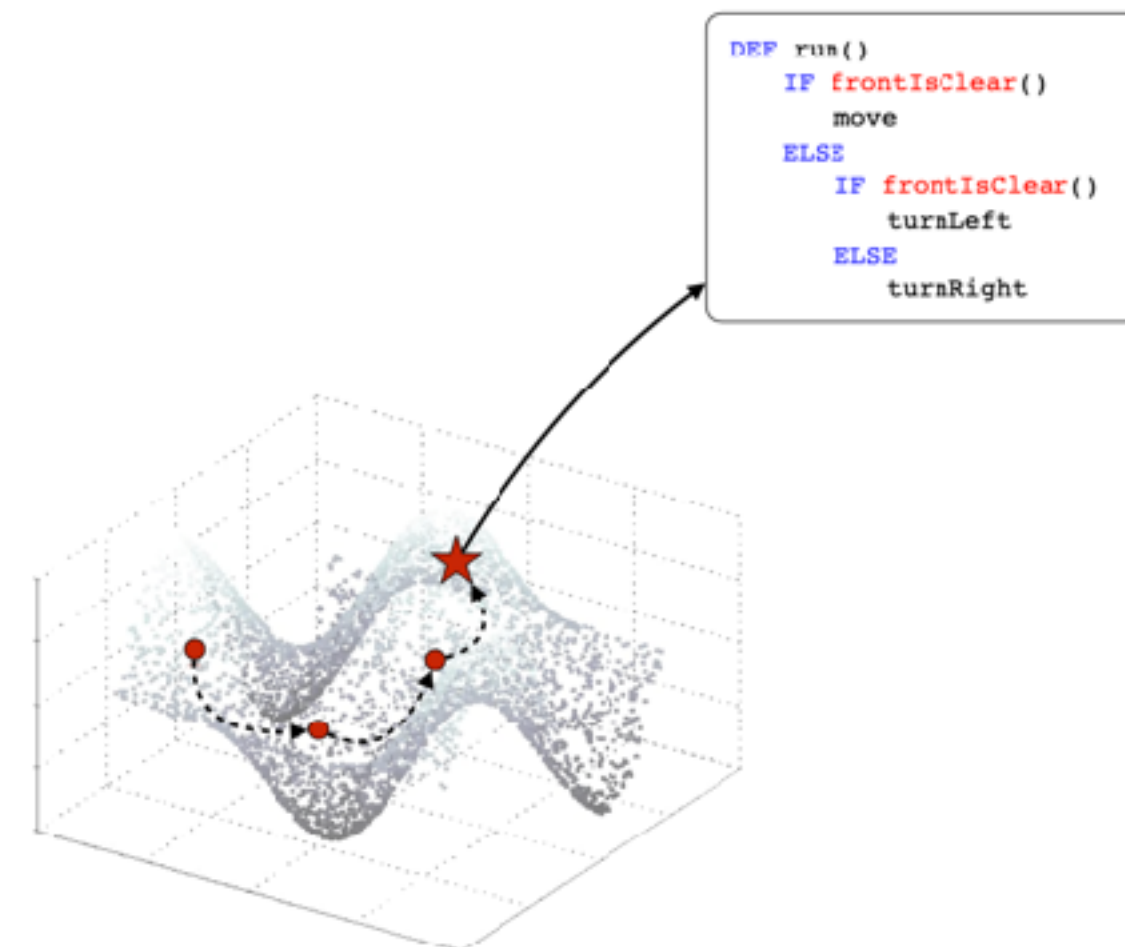
Environment
Dynamics



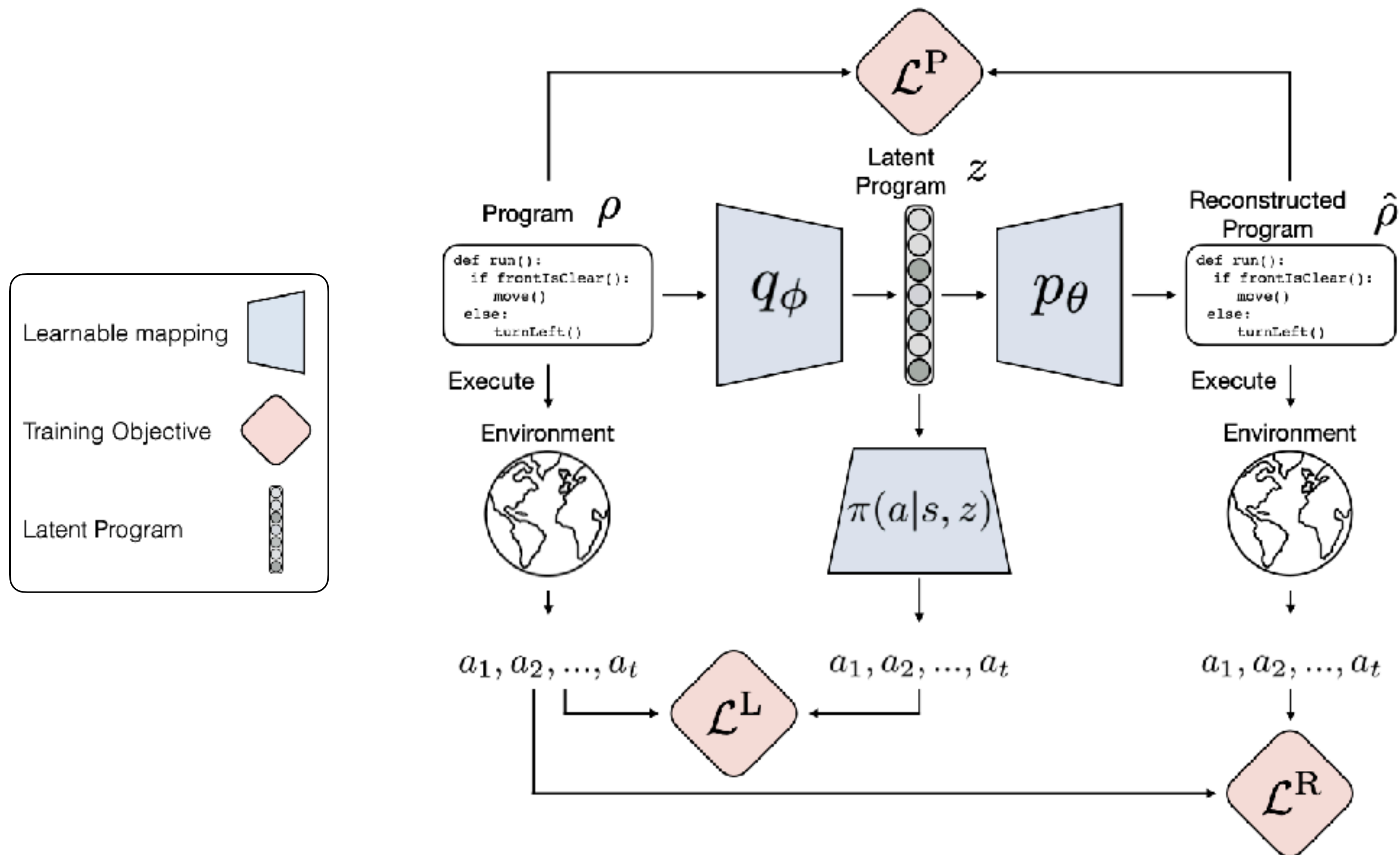
Stage 2

Searching for a task-solving program

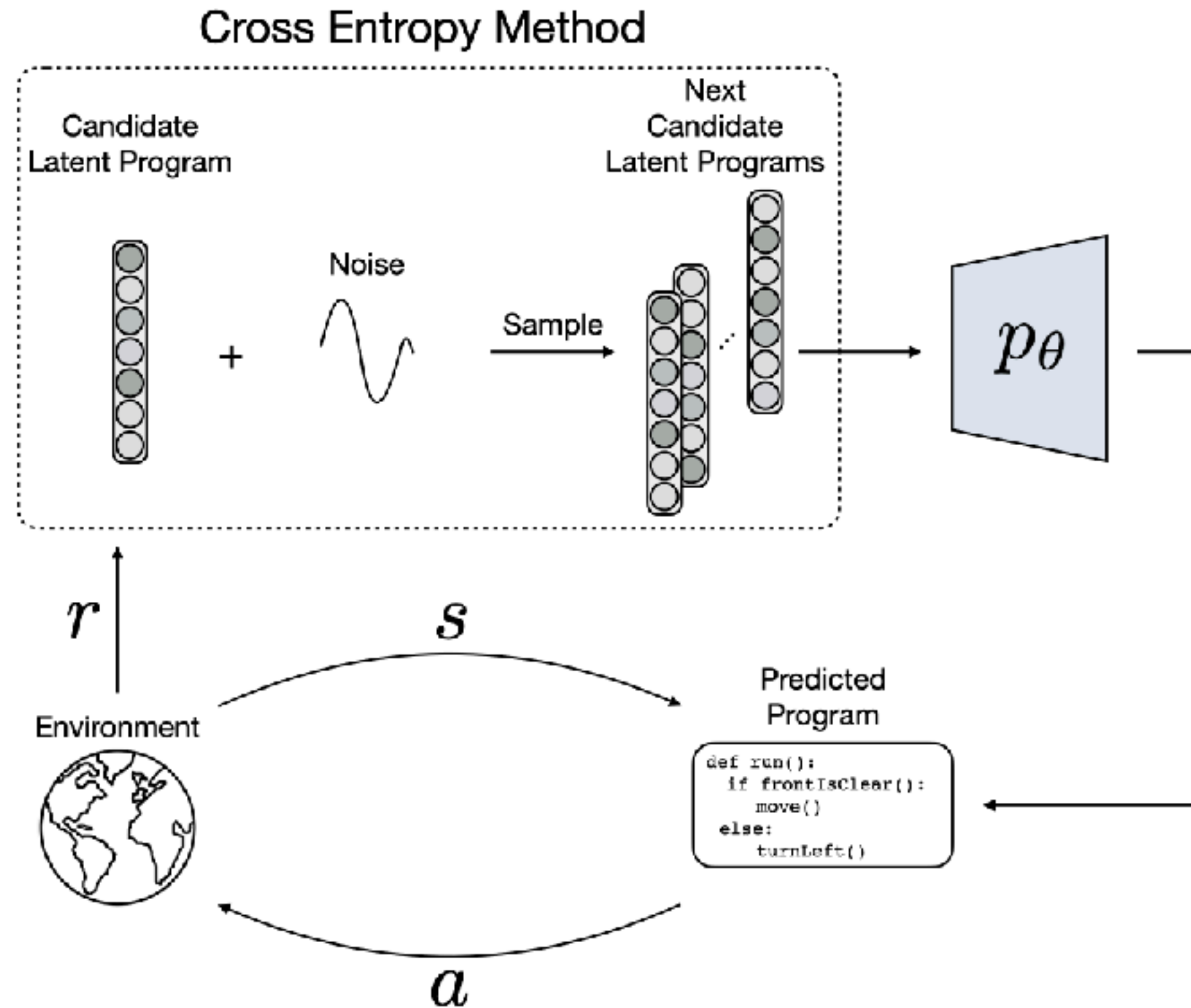
Desired
Behavior



Learning a Program Embedding Space

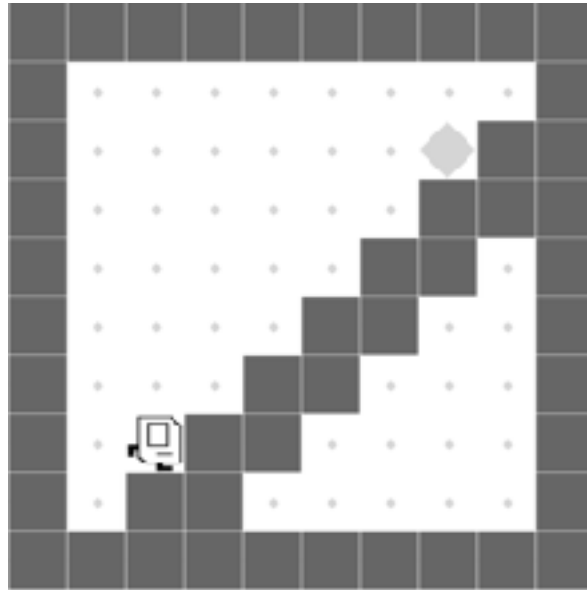


Latent Program Search with Cross-Entropy Method

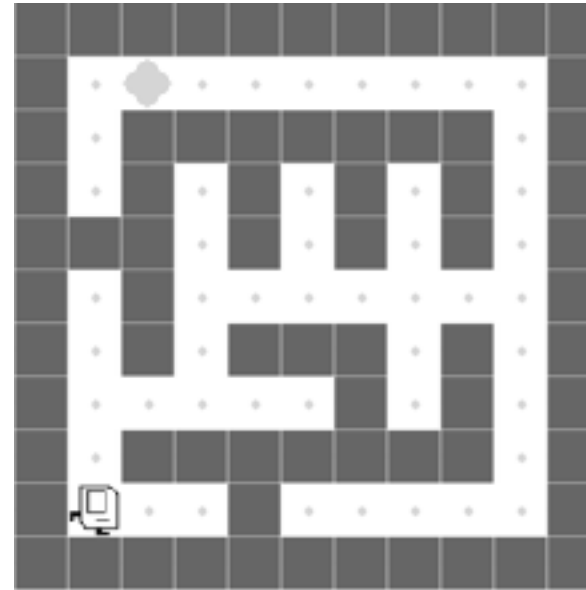


Karel Tasks

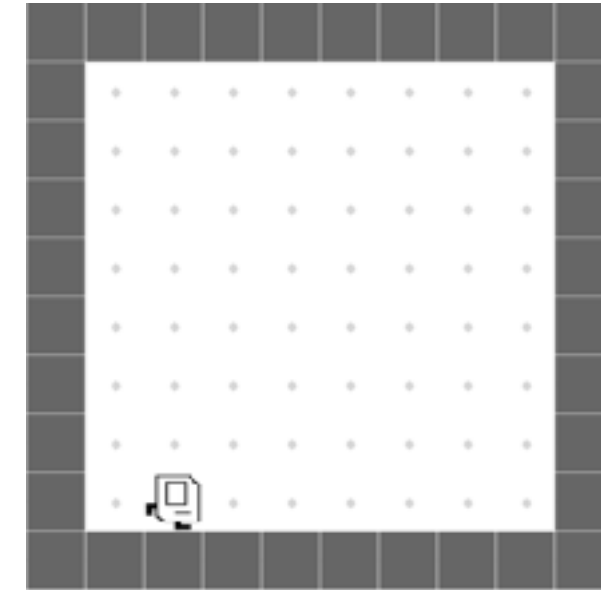
StairClimber



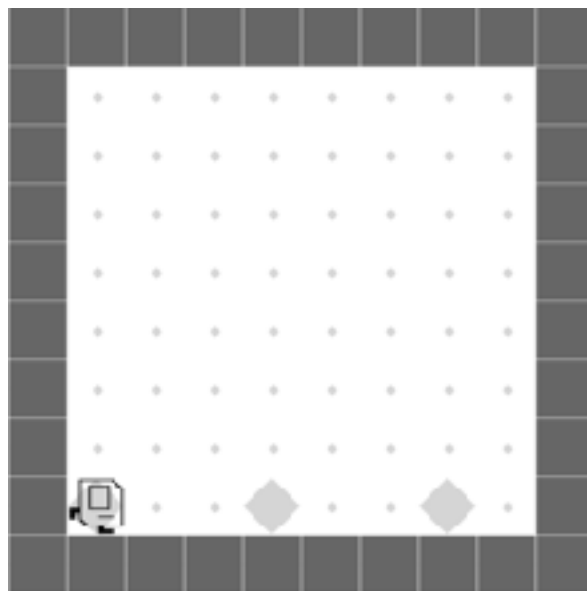
Maze



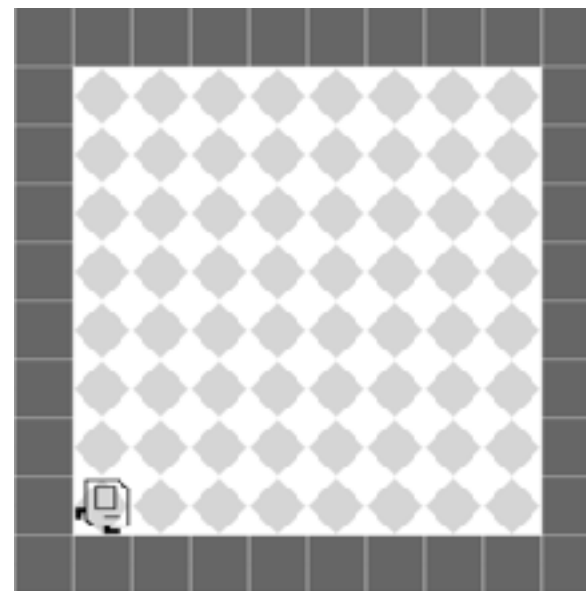
FourCorners



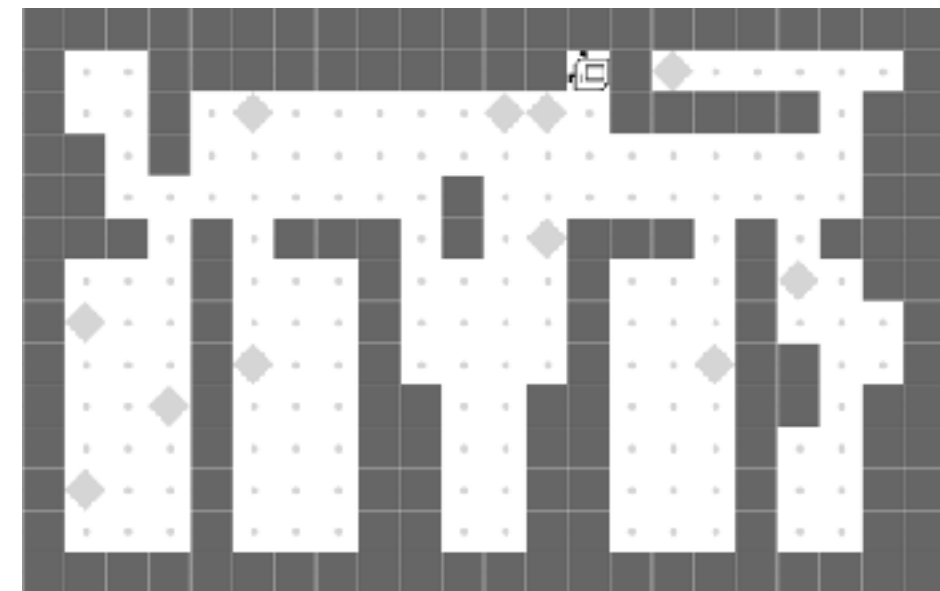
TopOff



Harvester



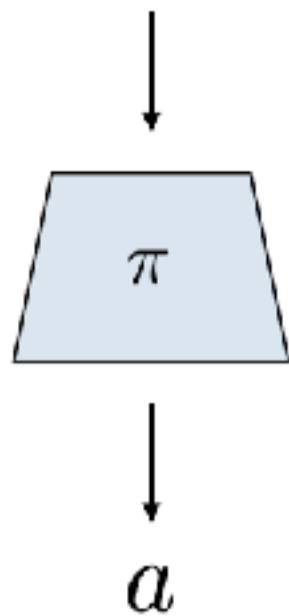
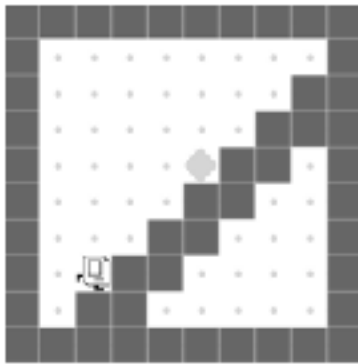
CleanHouse



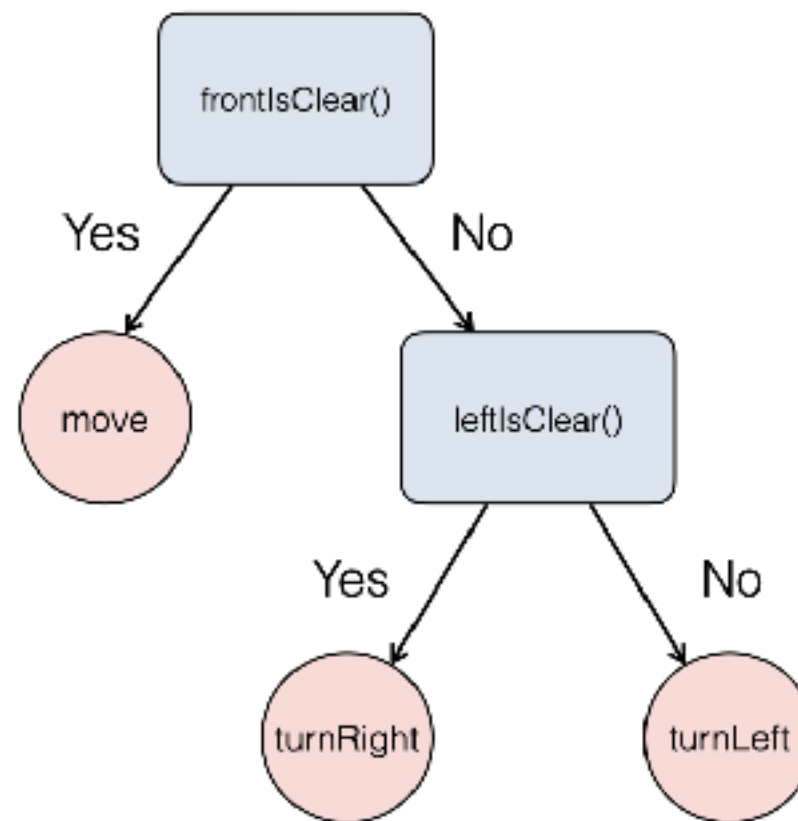
Baselines

DRL

Raw State

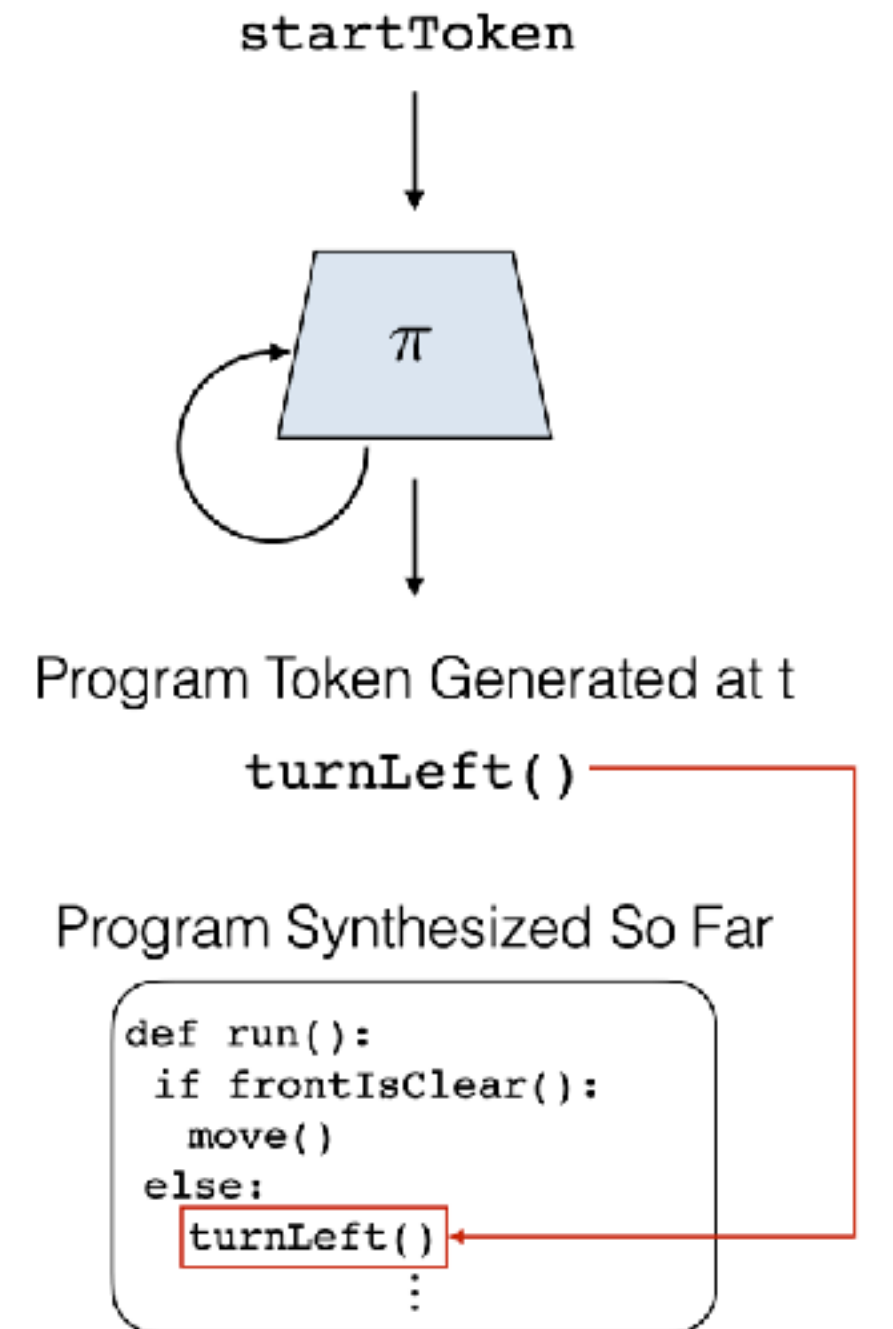


VIPER (Decision Tree)



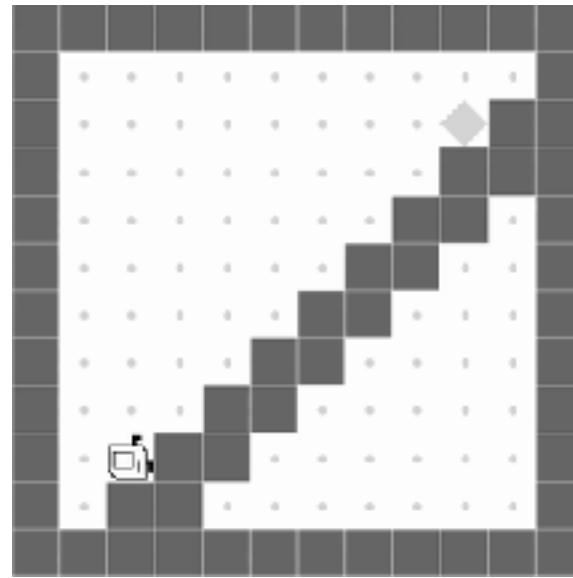
Distillation

Naive Program Synthesis

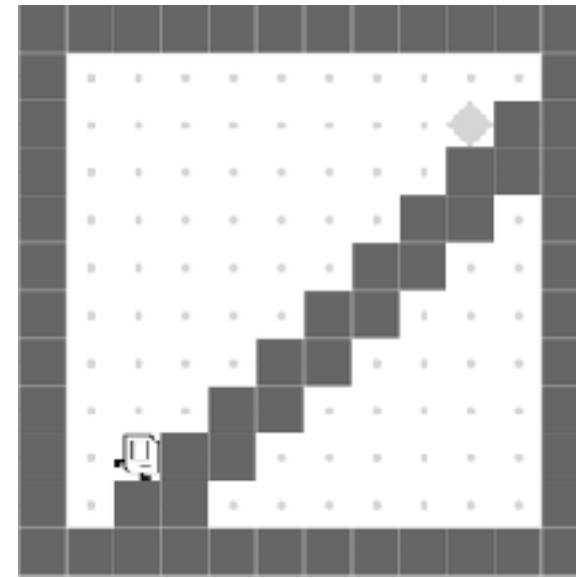


Qualitative Results

StairClimber

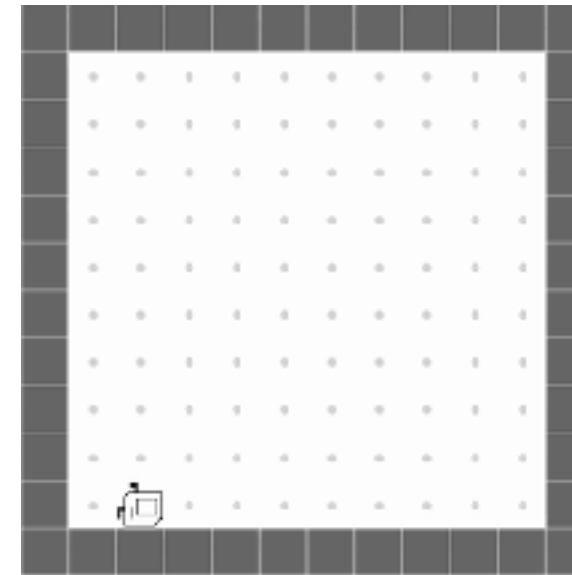


DRL

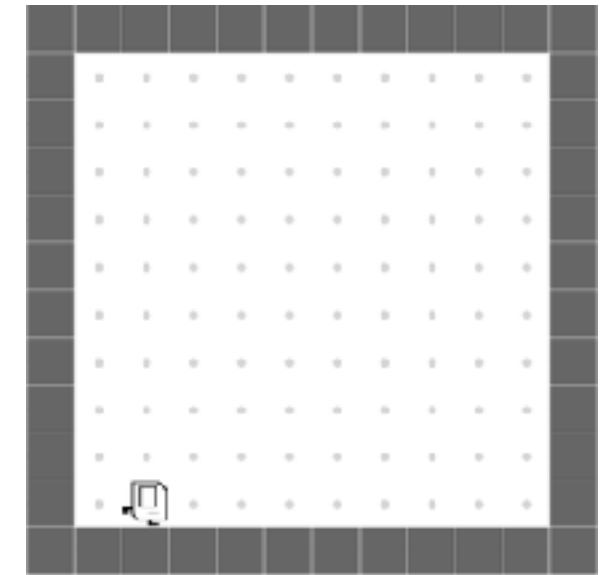


LEAPS

FourCorners

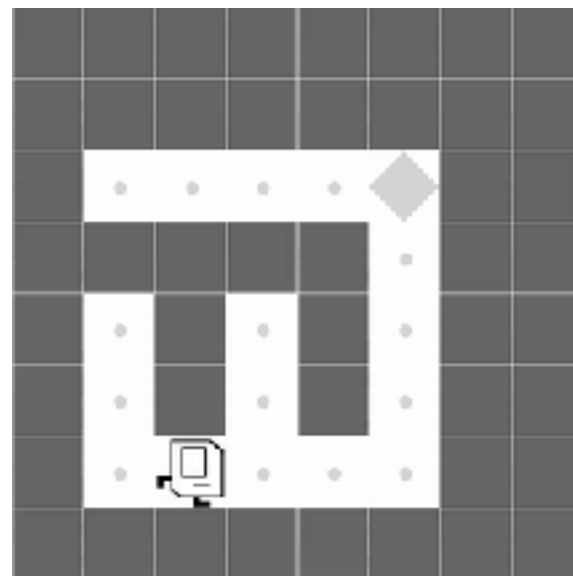


DRL

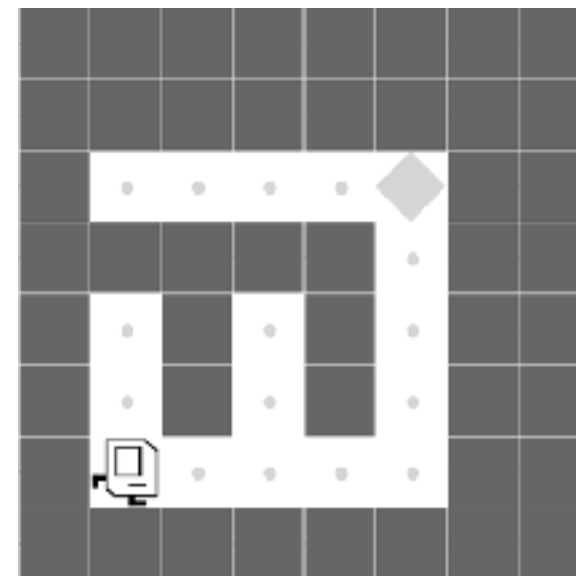


LEAPS

Maze

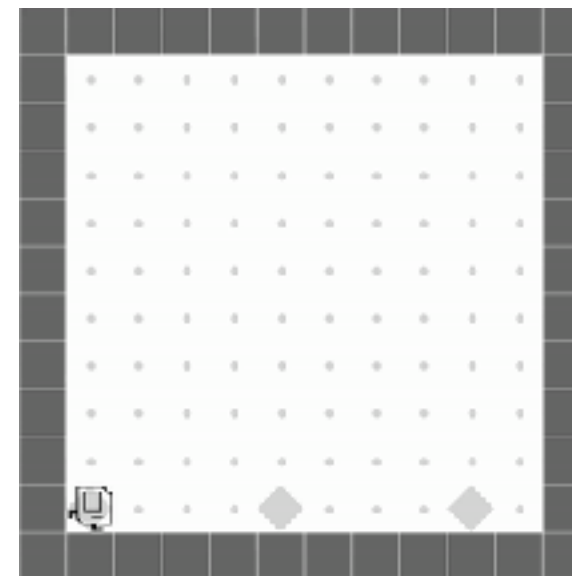


DRL

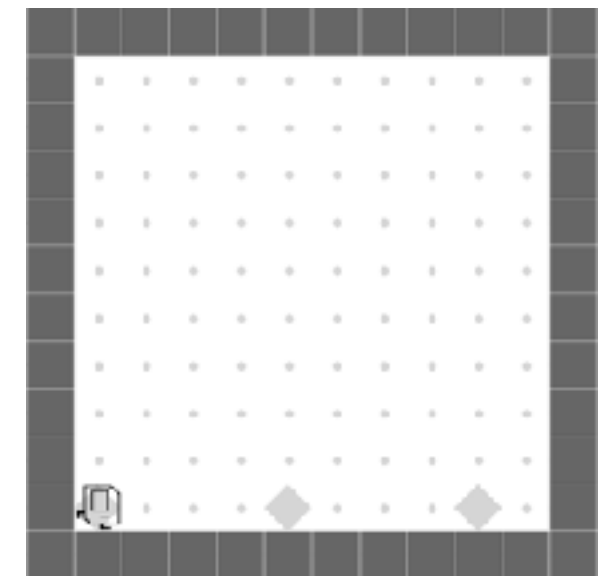


LEAPS

TopOff

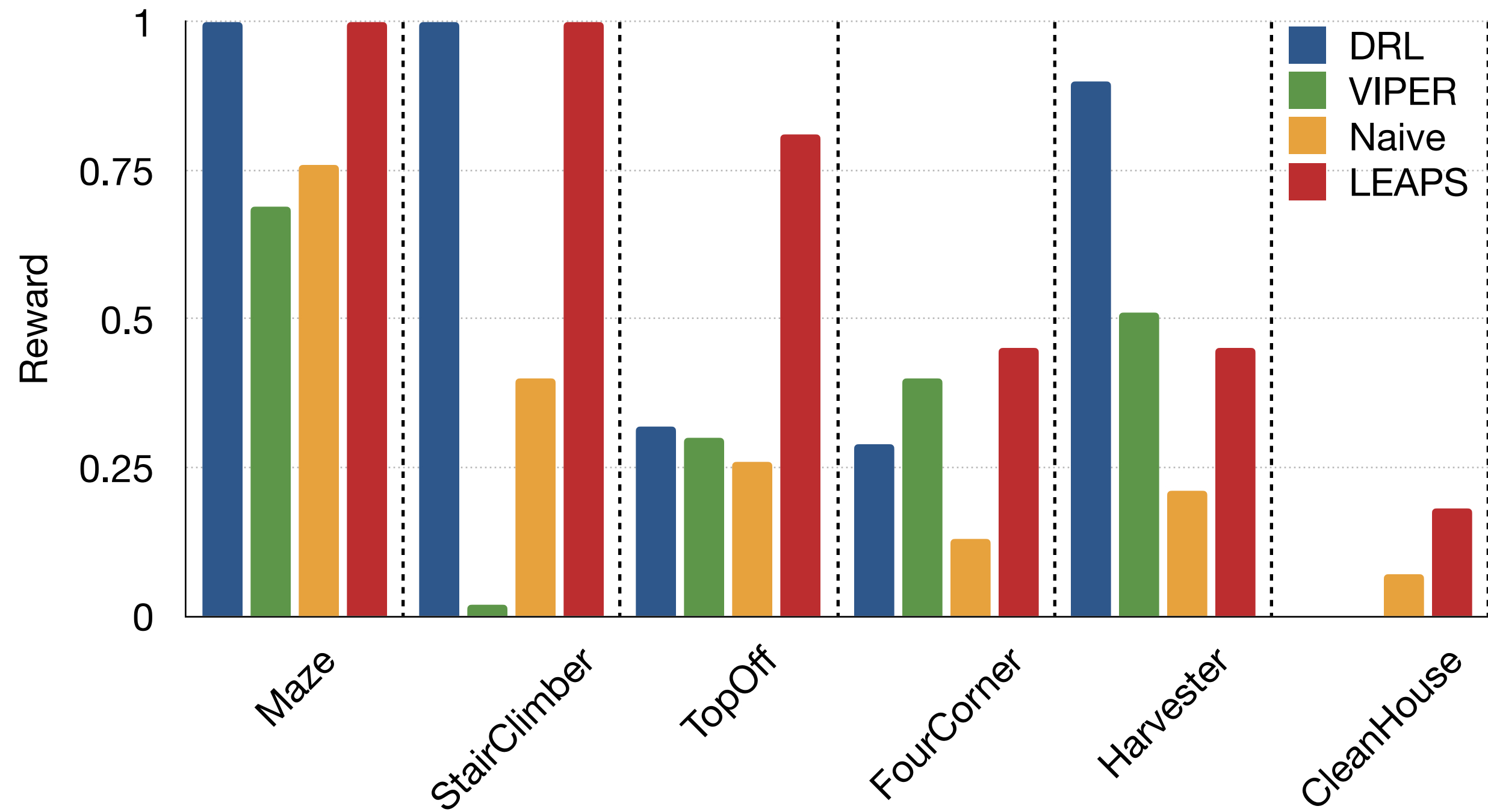


DRL

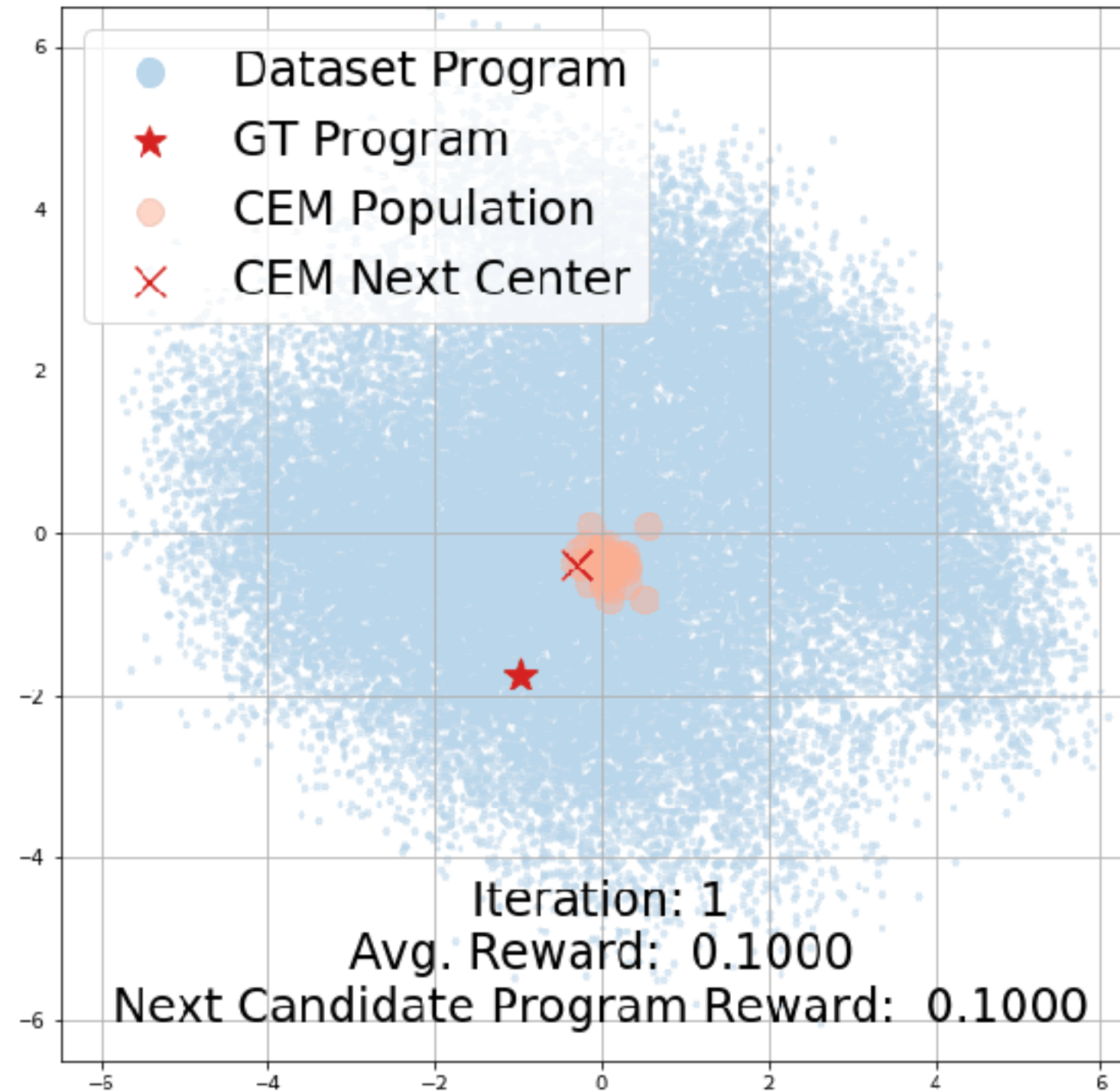


LEAPS

Quantitative Results



CEM trajectory Visualization

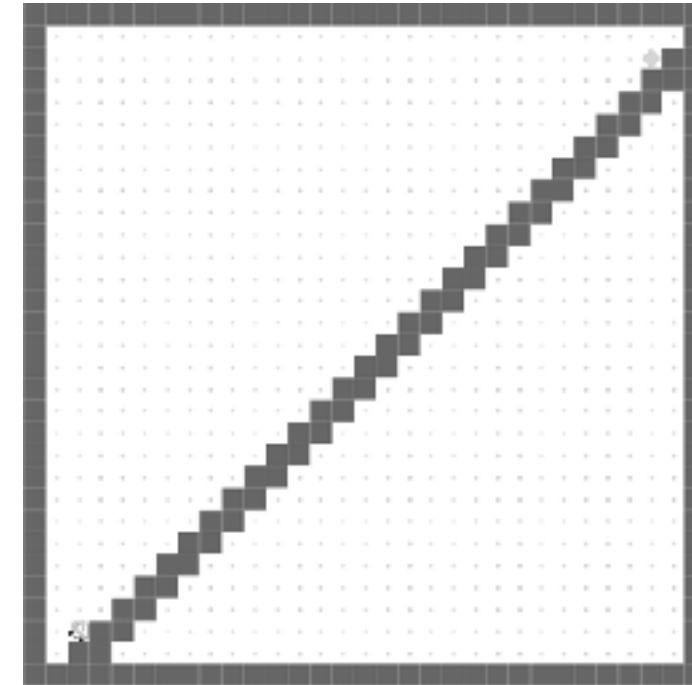
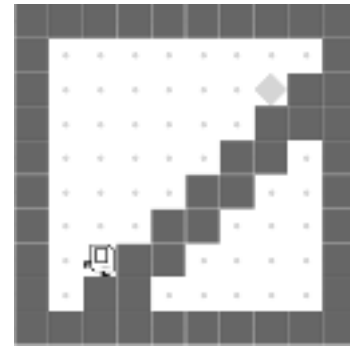


Zero-shot Generalization

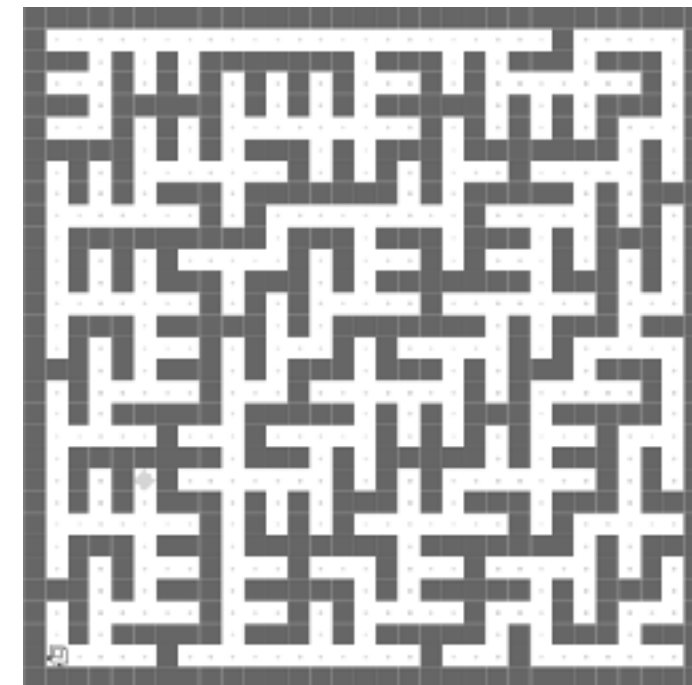
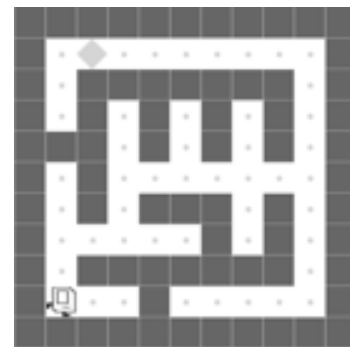
Learning on 8 x 8

Evaluation on 100 x 100

StairClimber



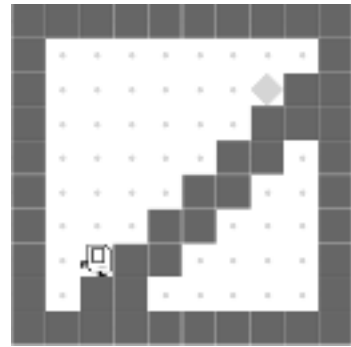
Maze



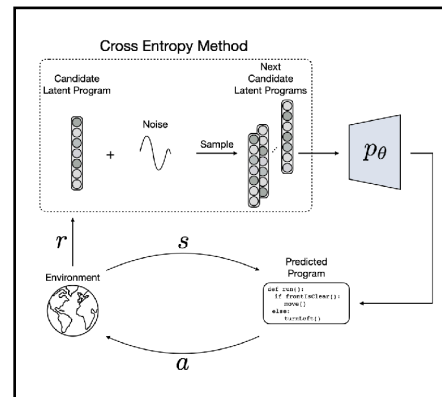
LEAPS Zero-shot Generalization

Learning on 8 x 8

StairClimber



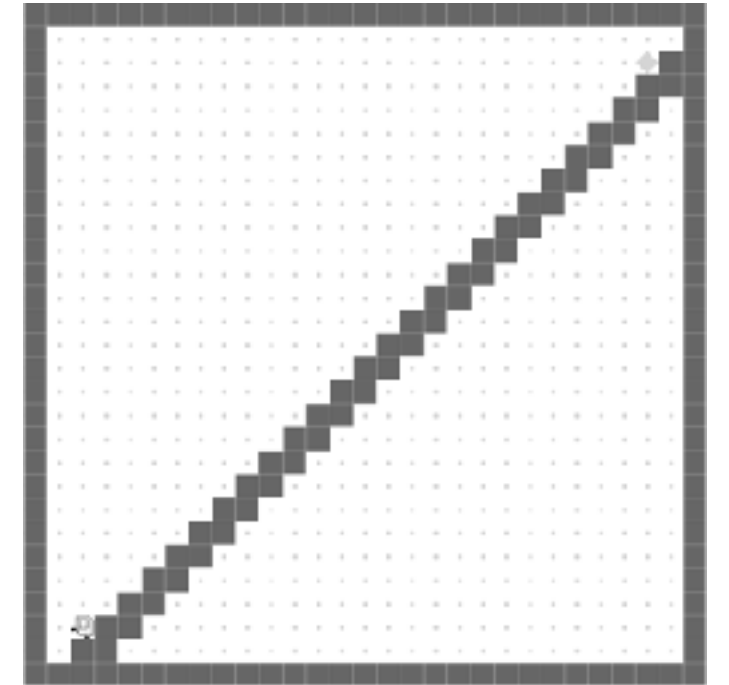
CEM search



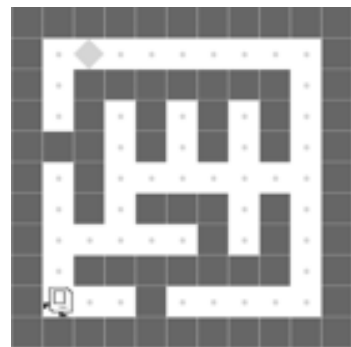
LEAPS Program Policy

```
DEF run()  
  WHILE noMarkersPresent()  
    turnRight  
    move  
  WHILE rightIsClear()  
    turnLeft
```

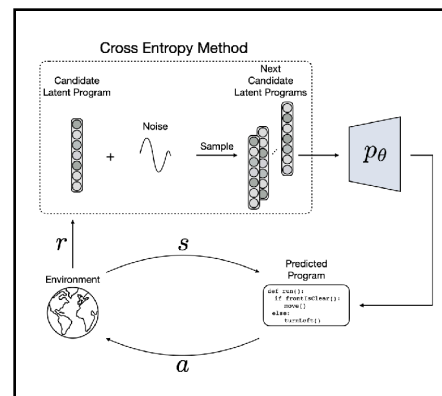
Evaluation on 100 x 100



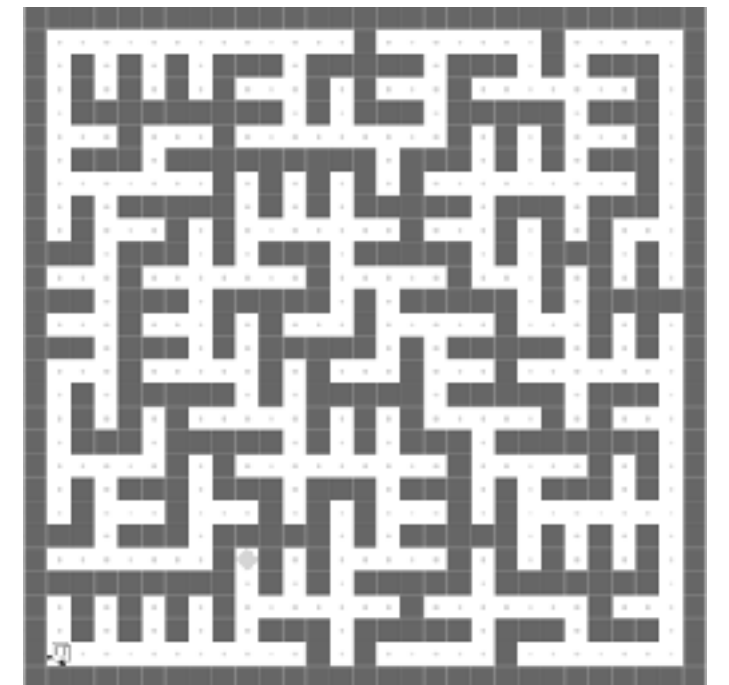
Maze



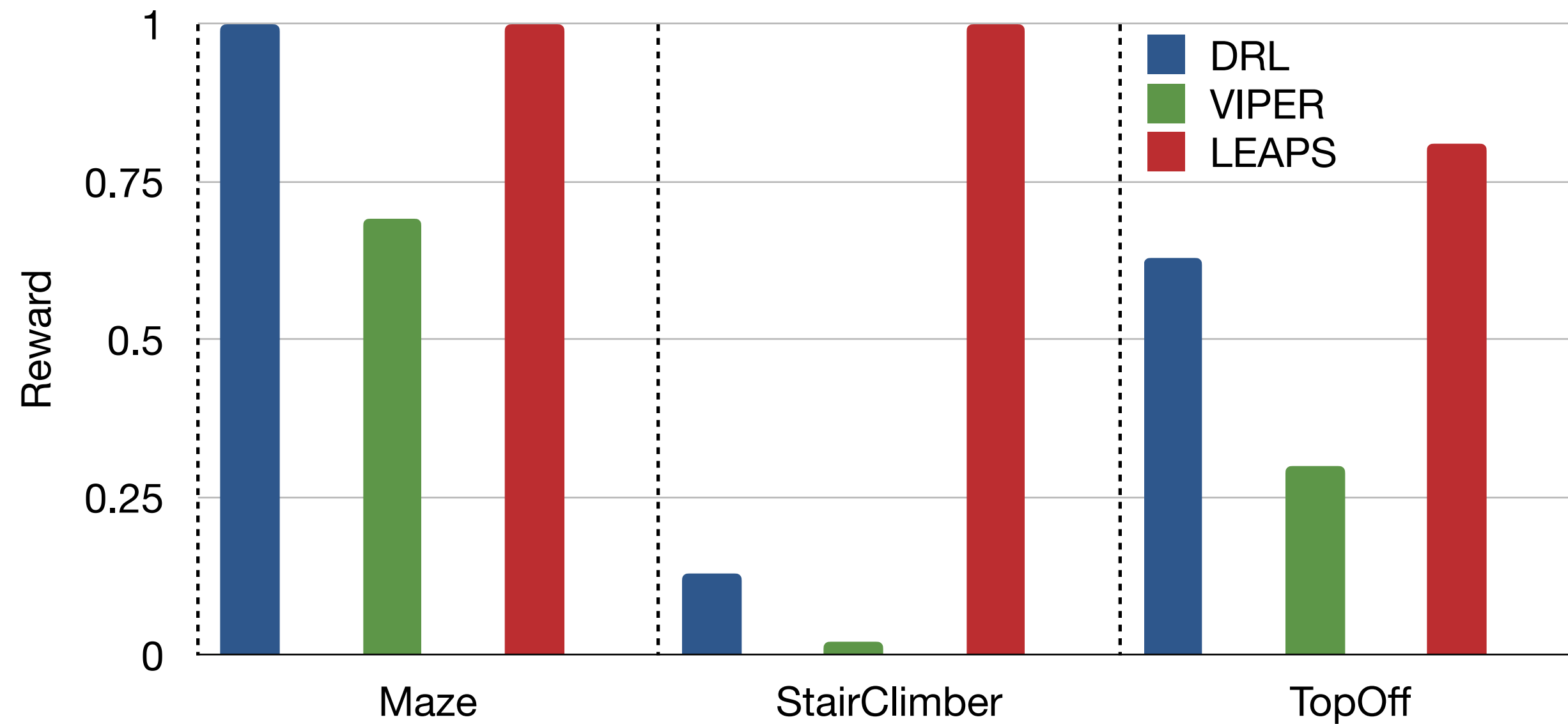
CEM search



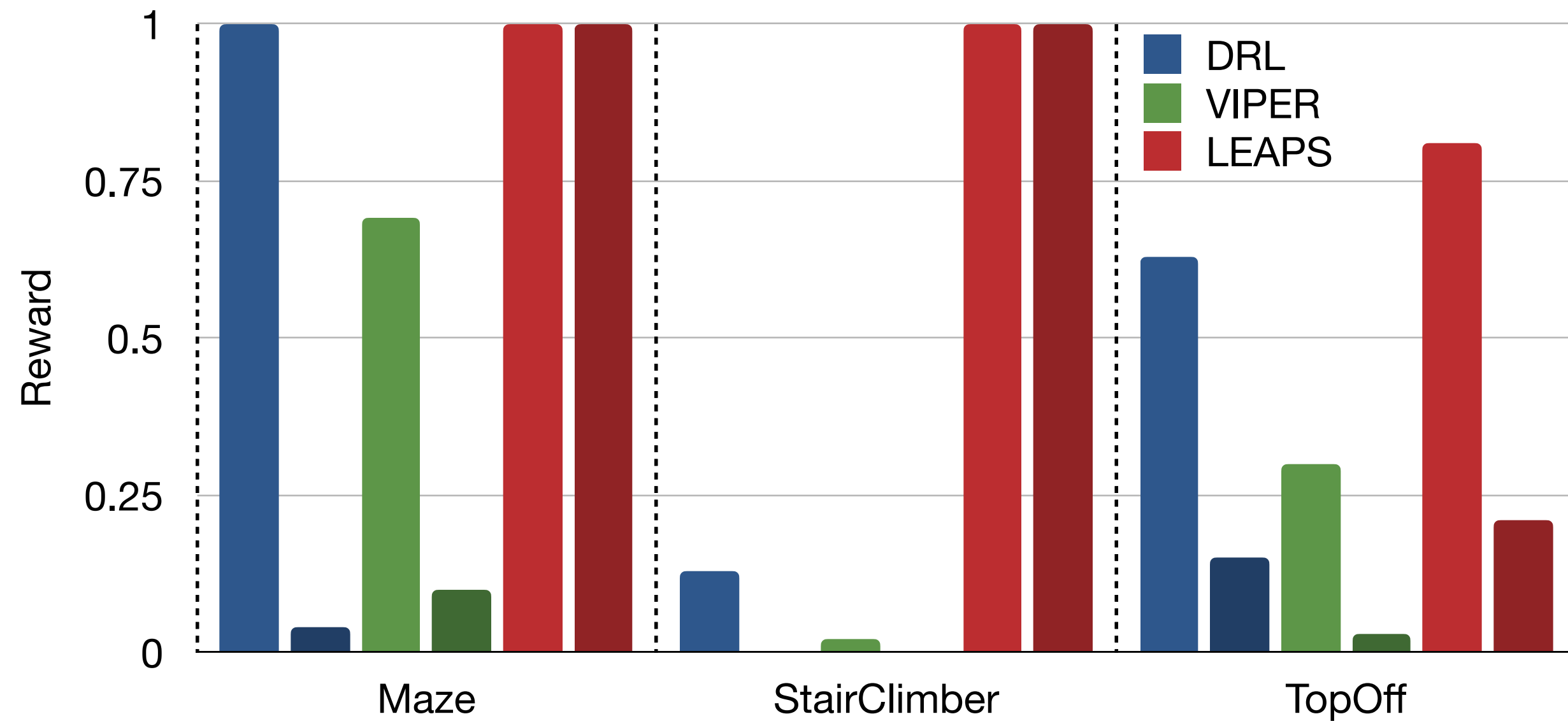
```
DEF run()  
  IF frontIsClear()  
    turnLeft  
  WHILE noMarkersPresent()  
    turnRight  
    move
```



Results - Zero-shot Generalization




Results - Zero-shot Generalization



Interpretability

Human Debugging Interface



Input Program:

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

Reset Code (Made a mistake?)

Issue with Code?

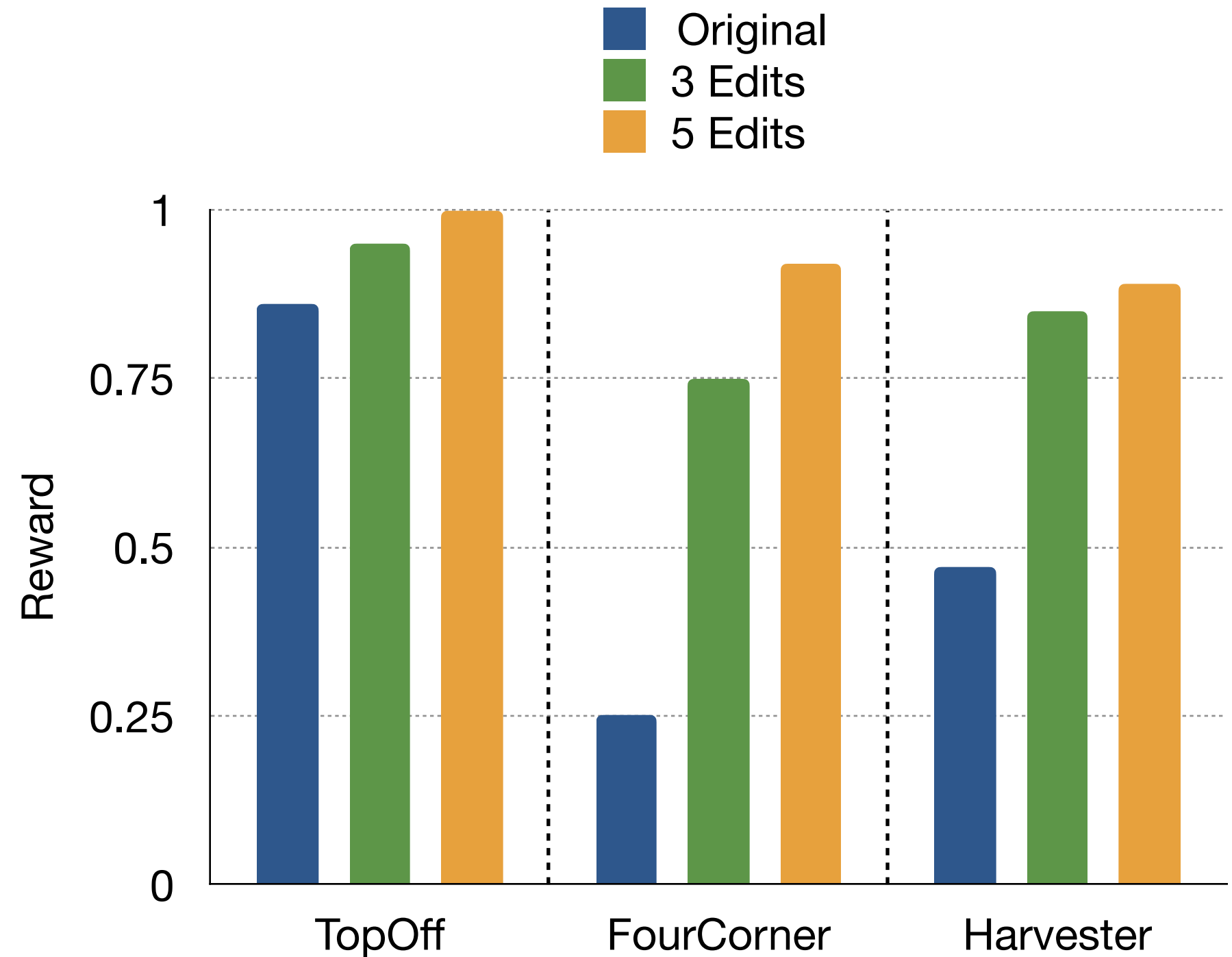
Submit Code (Runs your program)

New Reward:

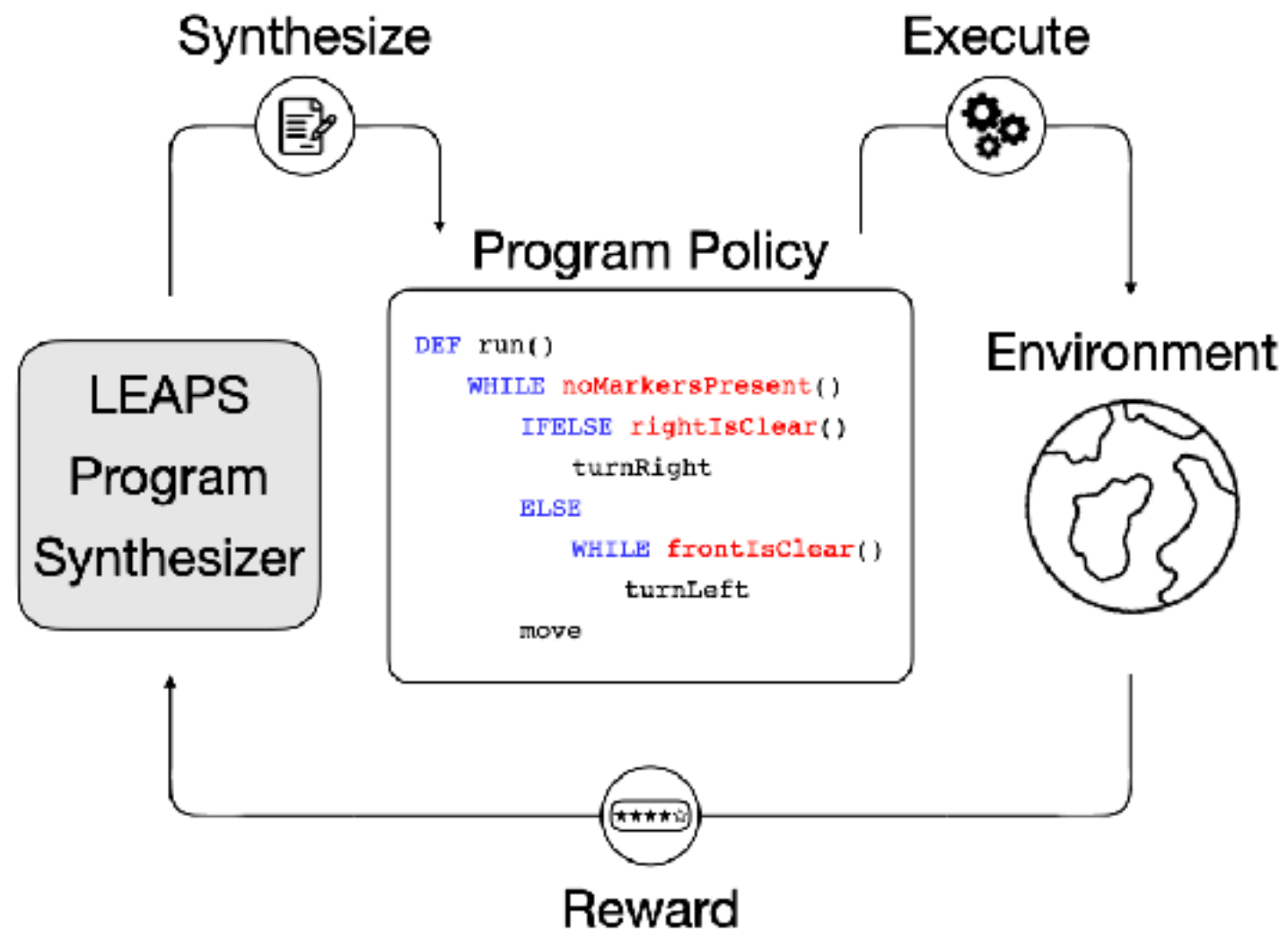
Orig Reward:

Best Reward:

Performance Improvement



Takeaways



- We learn to synthesize a program as a policy
- LEAPS
 - Learn a program embedding space
 - Search for a task-solving program
- Our synthesized programs achieve better
 - Task performance
 - Zero-shot *generalization*
 - *Interpretability*

[illegible]



Thank You

Questions?



Paper and code
clvrai.com/leaps