Learning to Synthesize Programs as Interpretable and Generalizable Policies

Dweep Trivedi*, Jesse Zhang*, Shao-Hua Sun*, Joseph J Lim

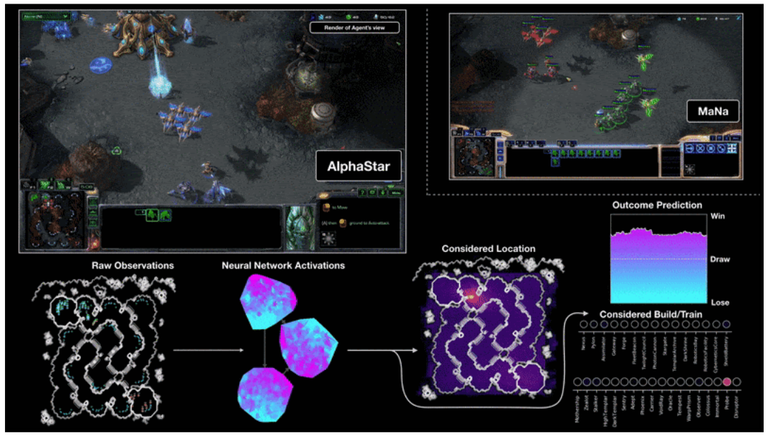
University of Southern California




Advances in Deep Reinforcement Learning

Autonomous Driving

Robotics

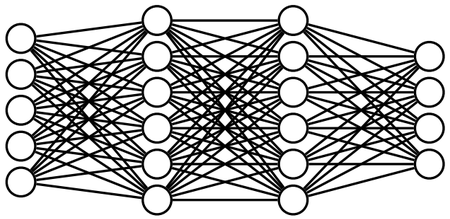


Game AI

Execute

Environment

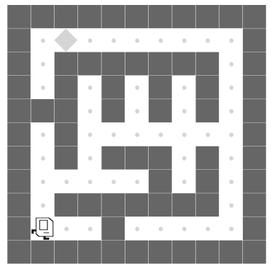
Deep Neural Network



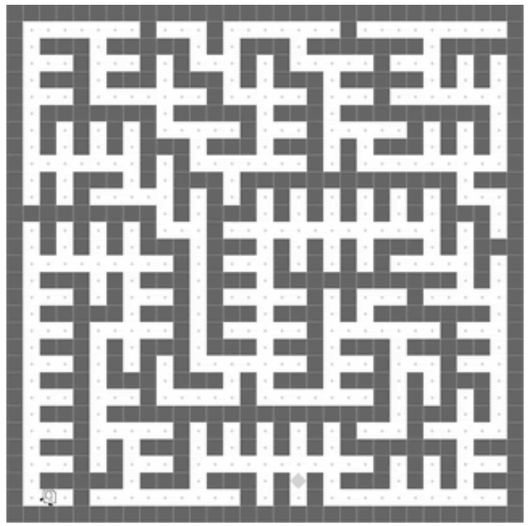
Reward

Issues with Deep Reinforcement Learning (DRL)

Generalization



Simple task

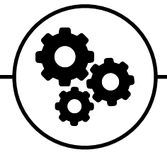


Complex task

Interpretability

- Trust
- Safety
- Contestability

Execute



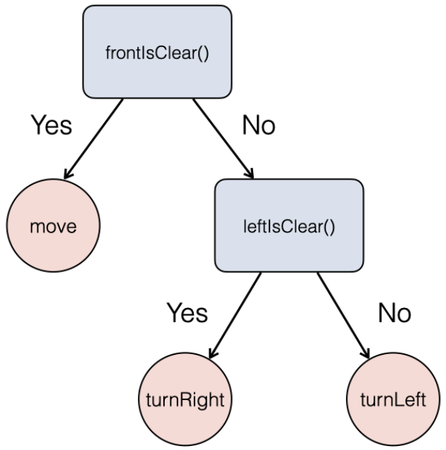
Environment

?

Reward

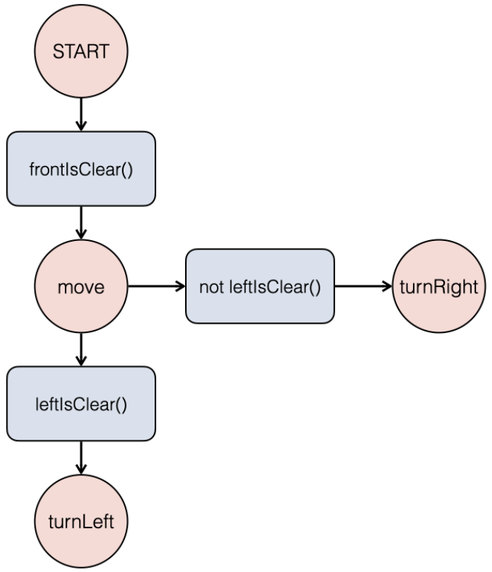
Programmatic Reinforcement Learning

Decision Trees



- Hard to represent repetitive behaviors

State Machines



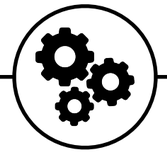
- Difficult to scale

Programs

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

- Flexible
- Human readable
- Hard to synthesize

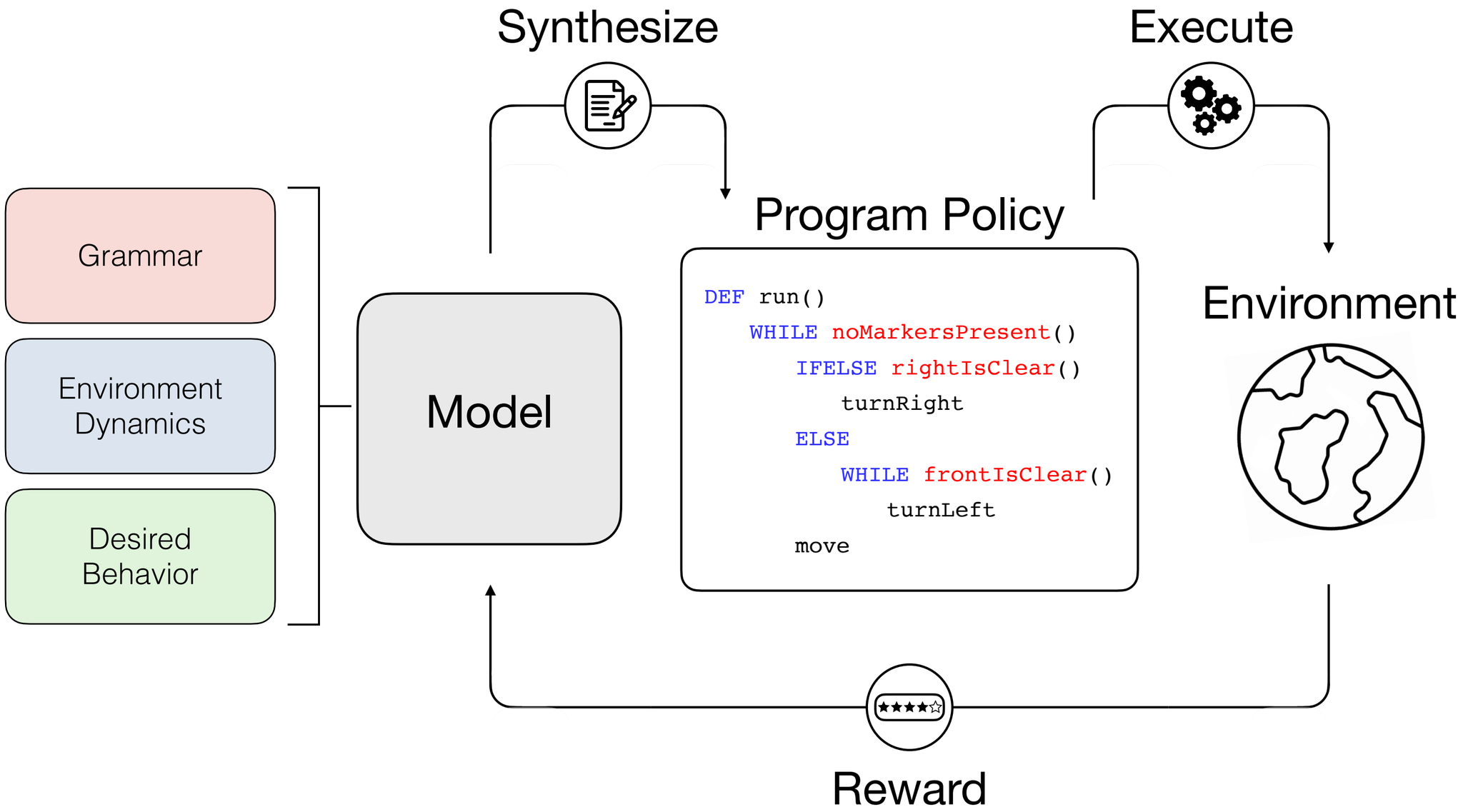
Execute

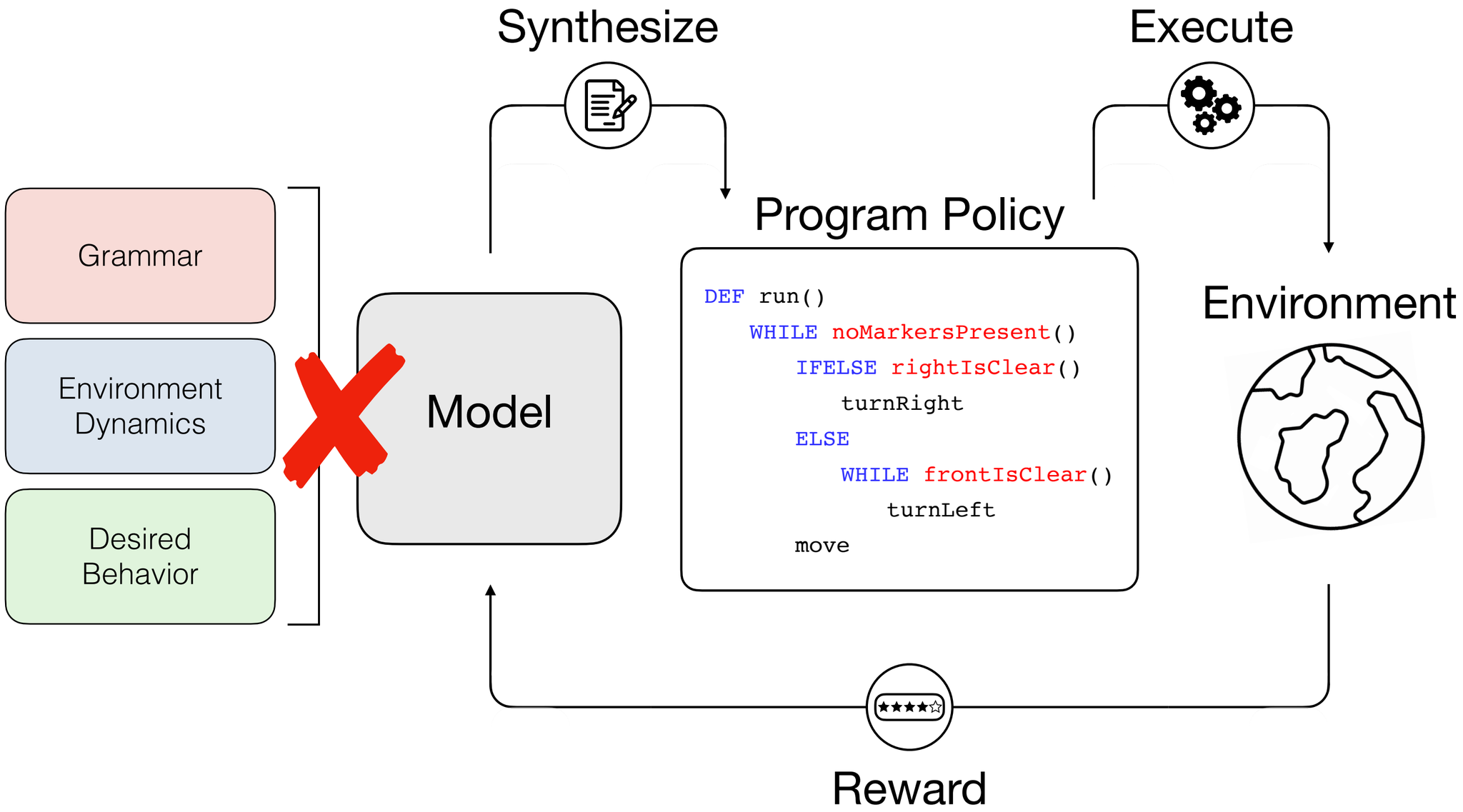


```
DEF run()
   IF frontIsClear()
      move
   ELSE
      IF frontIsClear()
         turnLeft
      ELSE
         turnRight
```

Environment

Reward


Synthesize
Execute
Grammar
Environment Dynamics
Desired Behavior
Model
Program Policy
DEF run()
WHILE noMarkersPresent()
IFELSE rightIsClear()
turnRight
ELSE
WHILE frontIsClear()
turnLeft
move
Environment
Reward


Synthesize
Execute
Grammar
Environment Dynamics
Desired Behavior
Model
Program Policy
DEF run()
WHILE noMarkersPresent()
IFELSE rightIsClear()
turnRight
ELSE
WHILE frontIsClear()
turnLeft
move
Environment
Reward

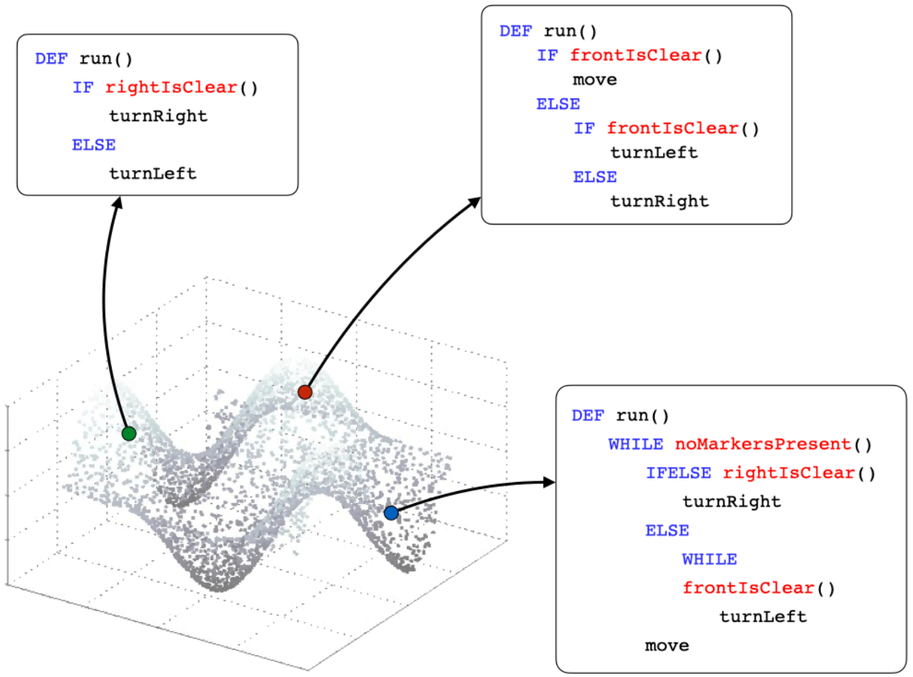
LEAPS: **L**earning **E**mbeddings for l**A**tent **P**rogram **S**ynthesis

Stage 1

Learning a program embedding space
from a set of randomly generated programs

Grammar

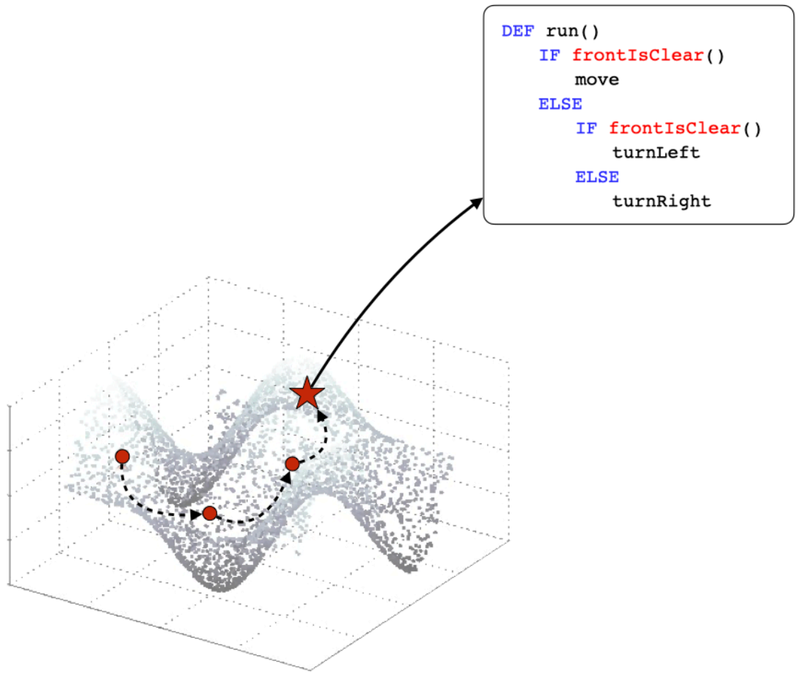
Environment
Dynamics



Stage 2

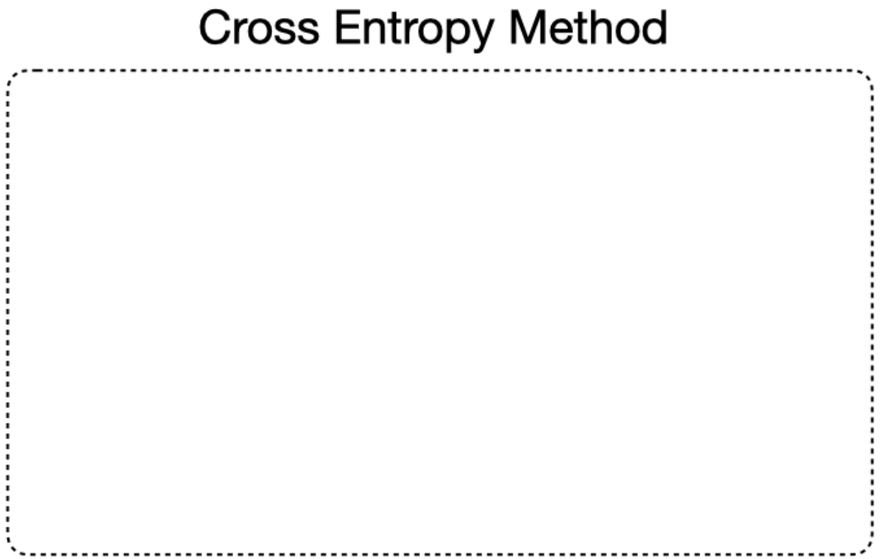
Searching for a task-solving program

Desired
Behavior

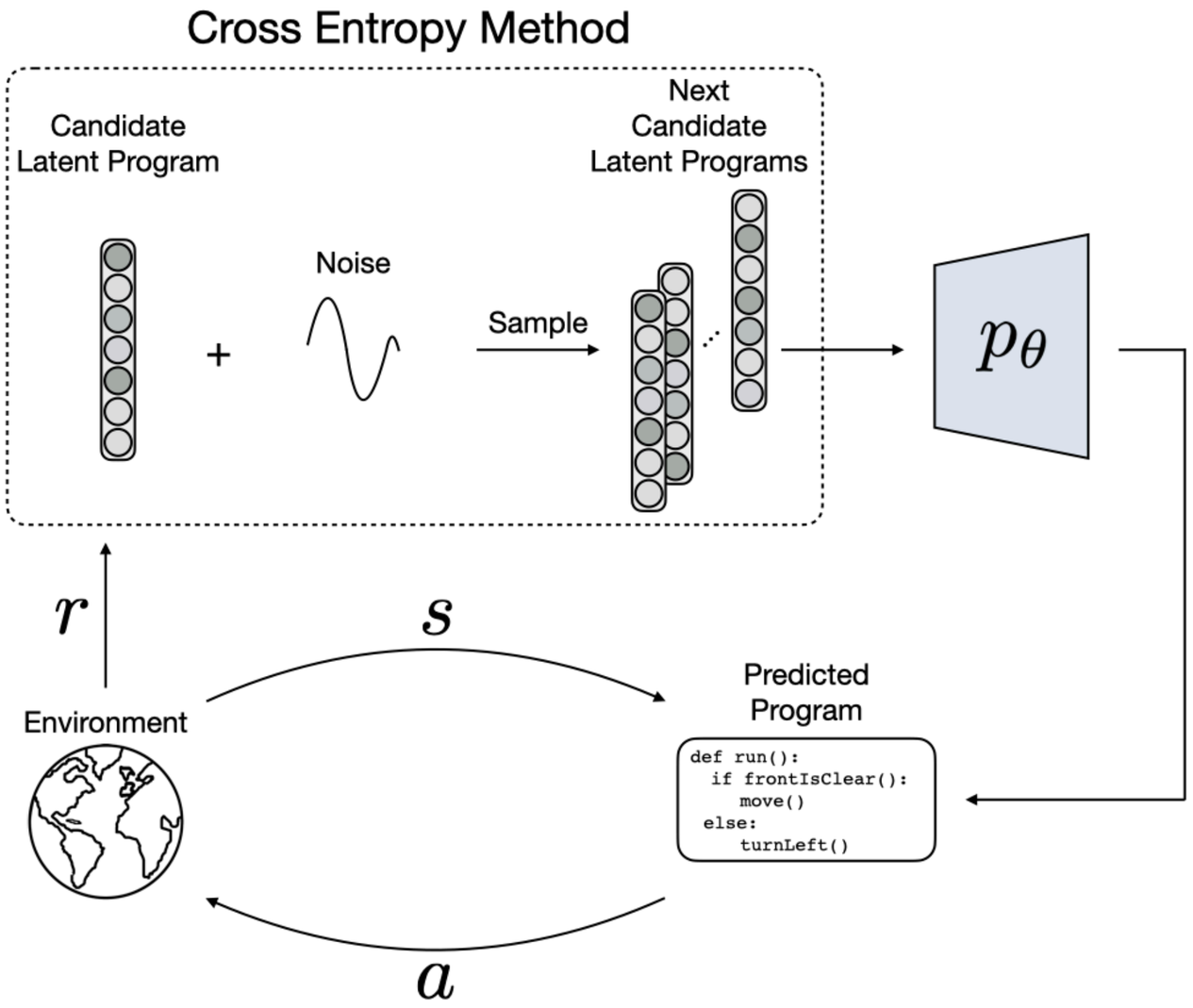


Learning a Program Embedding Space

Latent Program Search with Cross-Entropy Method

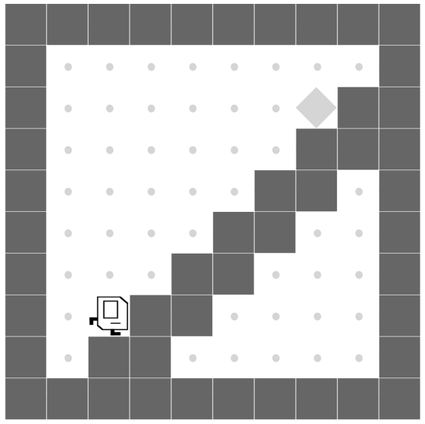


Latent Program Search with Cross-Entropy Method

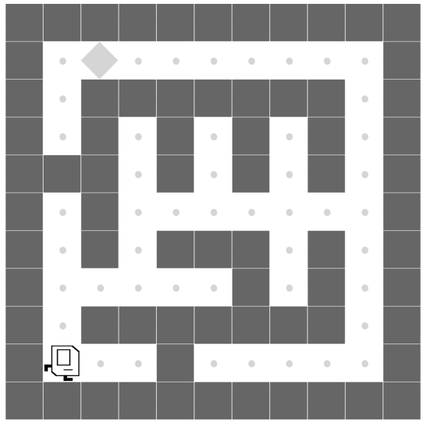


Karel Tasks

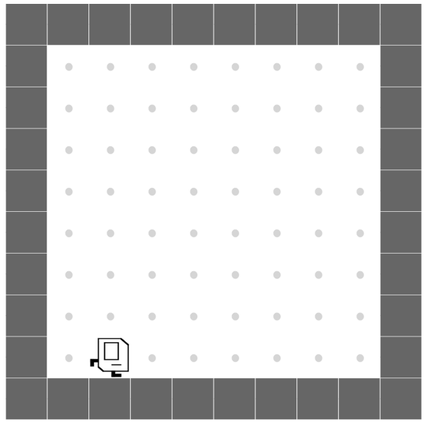
StairClimber



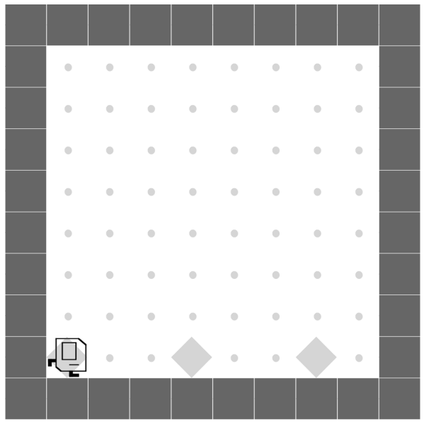
Maze



FourCorners

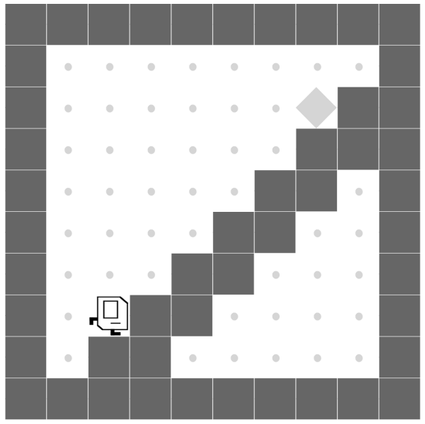


TopOff

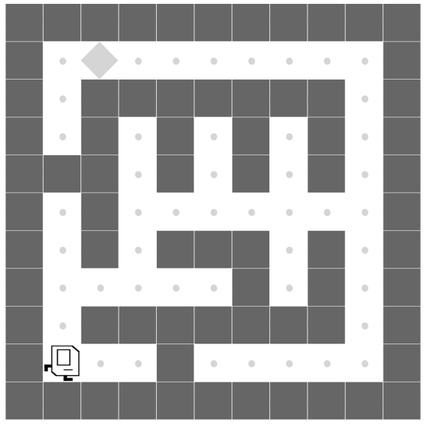


Karel Tasks

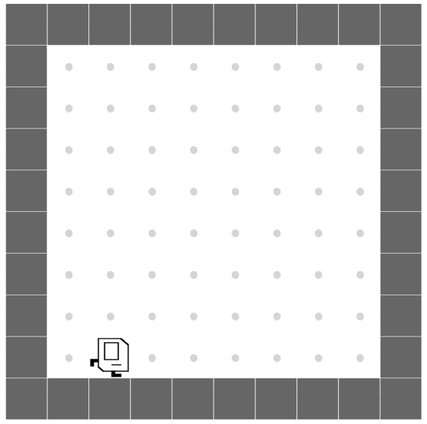
StairClimber



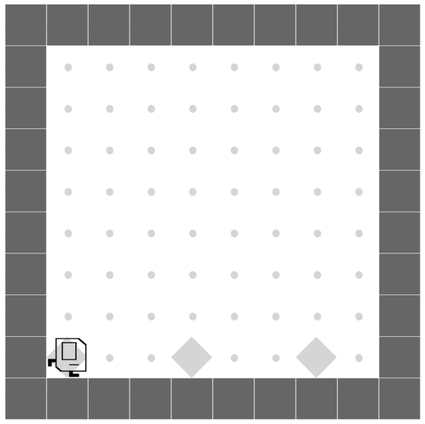
Maze



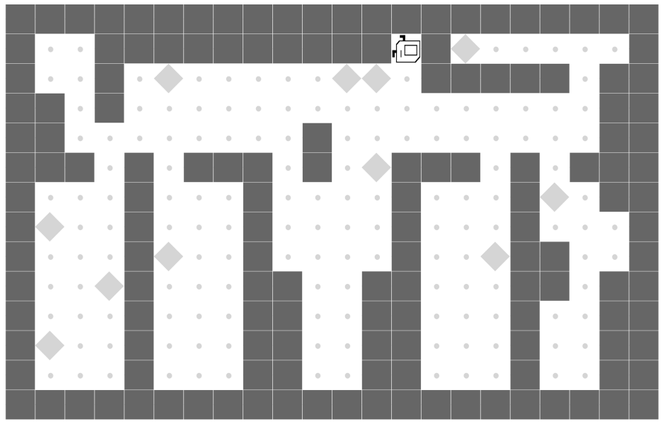
FourCorners



TopOff

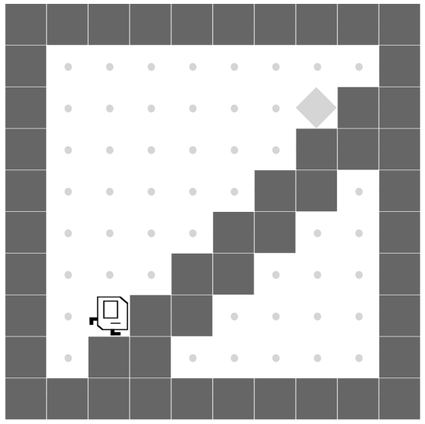


CleanHouse

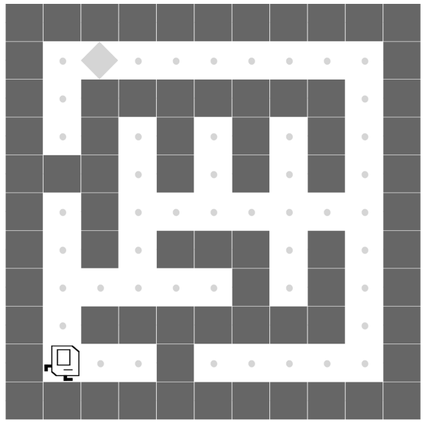


Karel Tasks

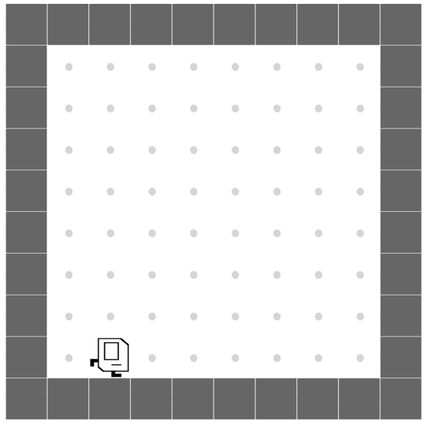
StairClimber



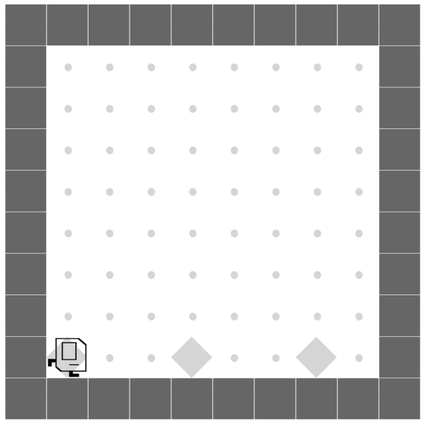
Maze



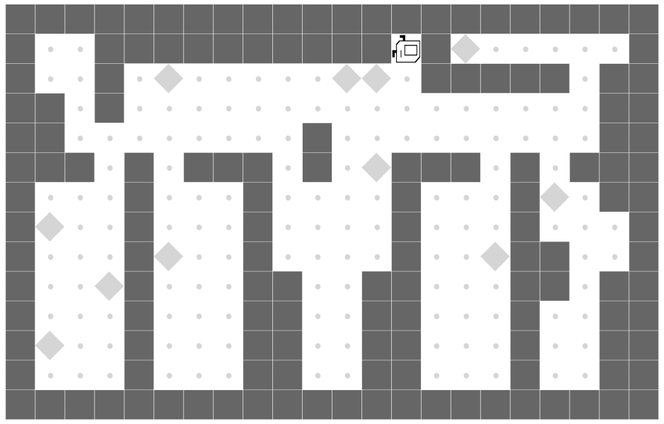
FourCorners



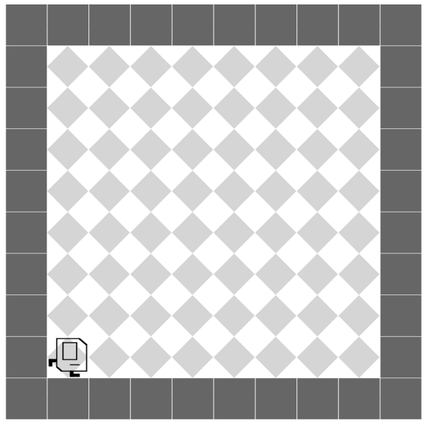
TopOff



CleanHouse



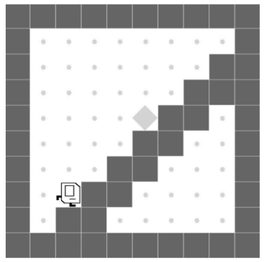
Harvester



Baselines

DRL

Raw State

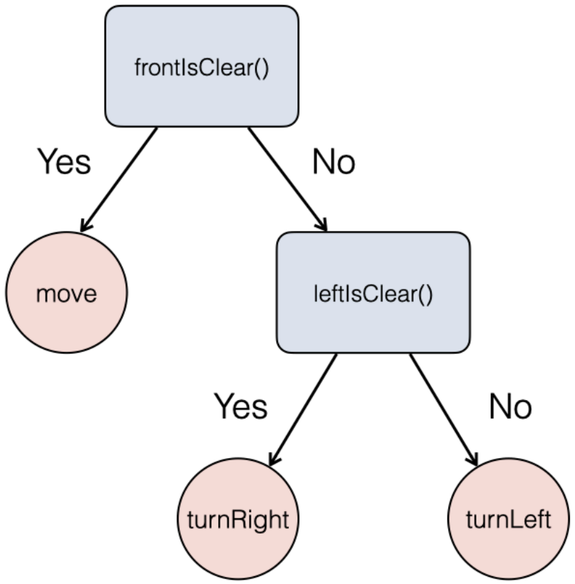


π

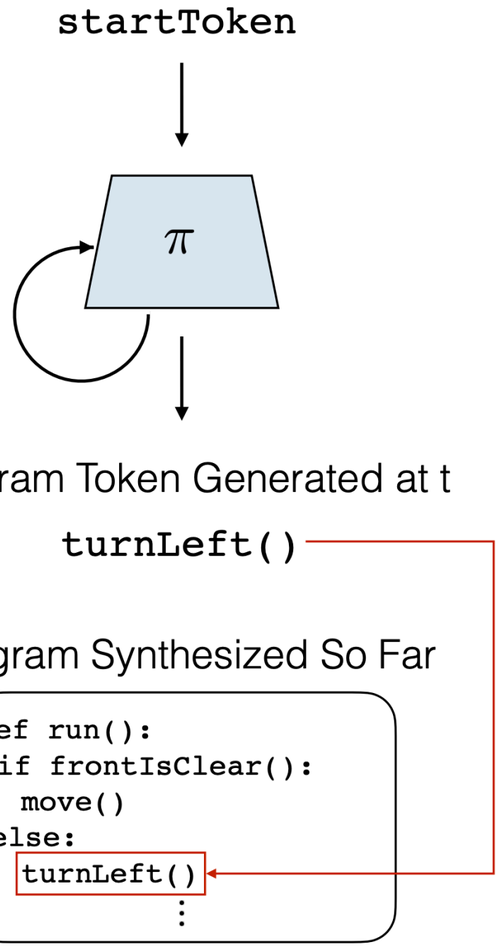
a

Distillation

VIPER (Decision Tree)

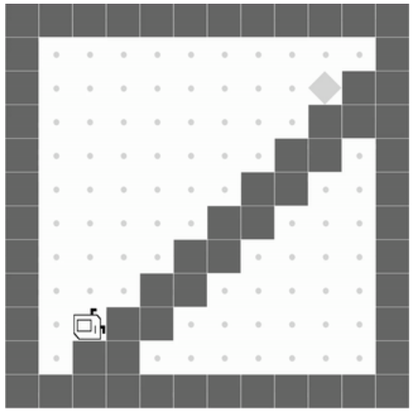


Naive Program Synthesis

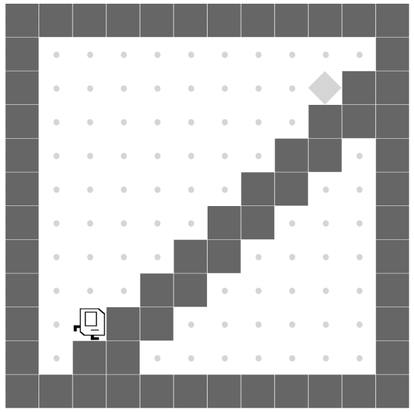


Results

StairClimber



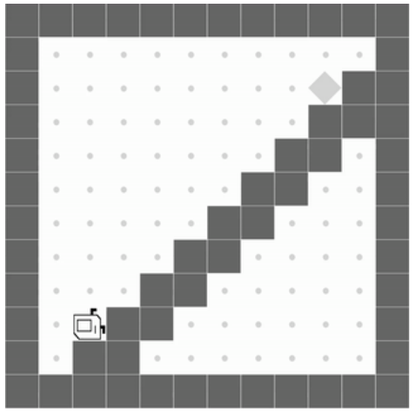
DRL



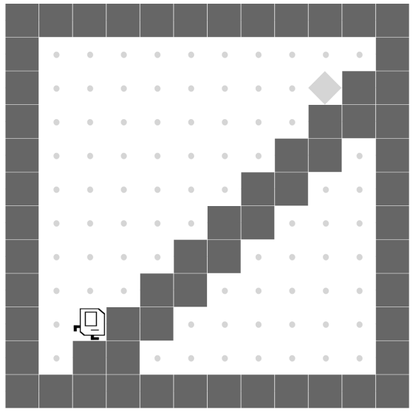
LEAPS

Results

StairClimber

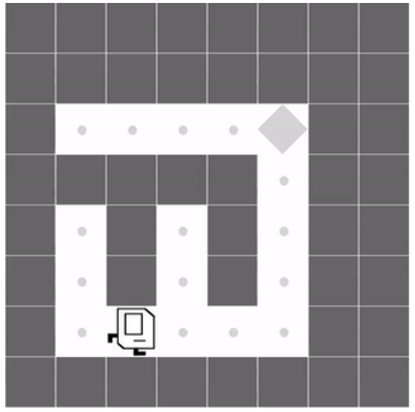


DRL

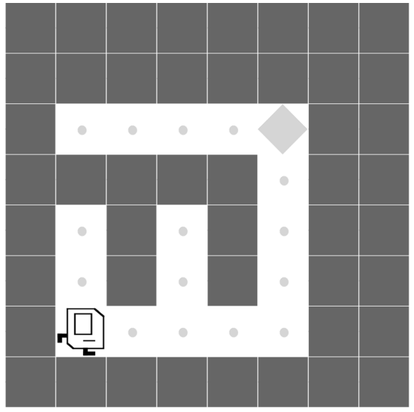


LEAPS

Maze



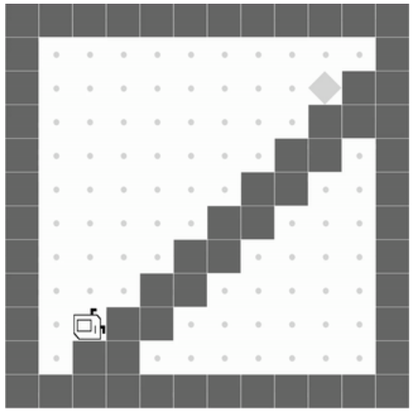
DRL



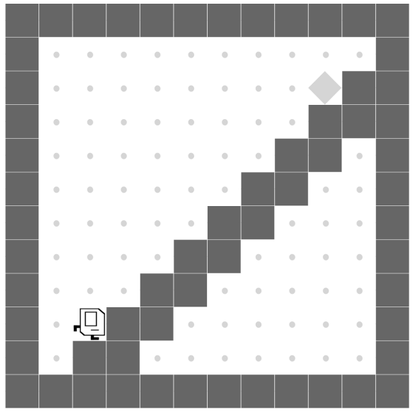
LEAPS

Results

StairClimber

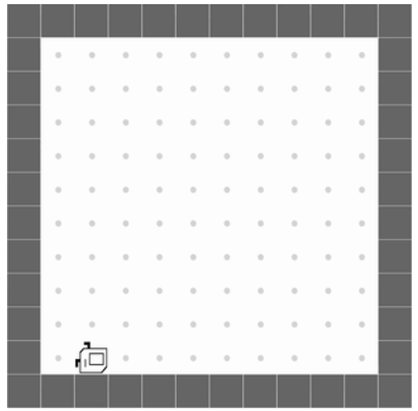


DRL

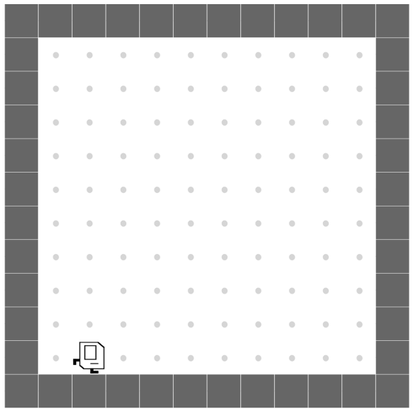


LEAPS

FourCorners

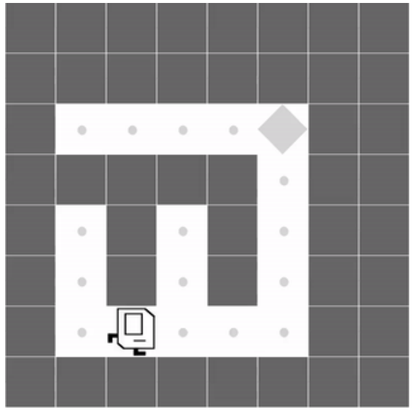


DRL

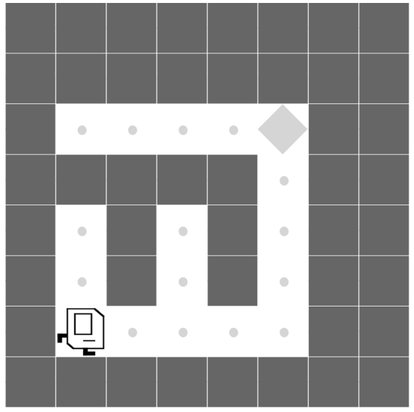


LEAPS

Maze



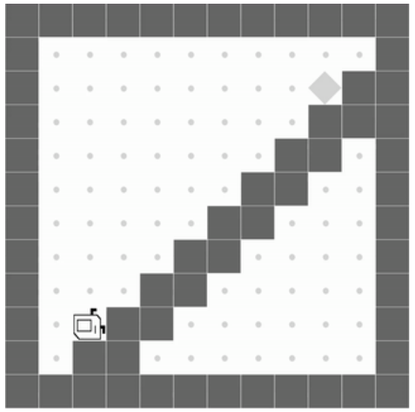
DRL



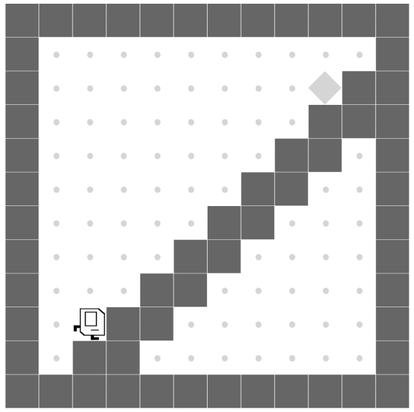
LEAPS

Results

StairClimber

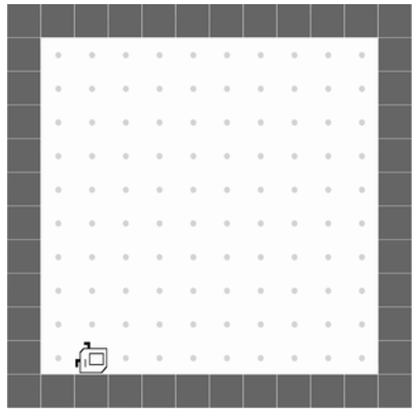


DRL

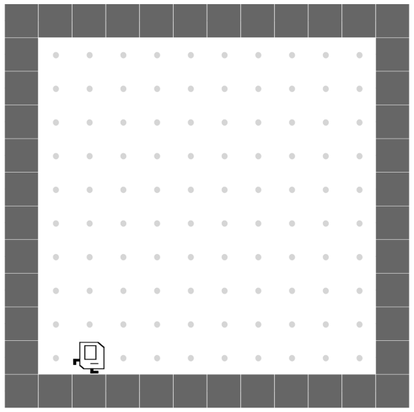


LEAPS

FourCorners

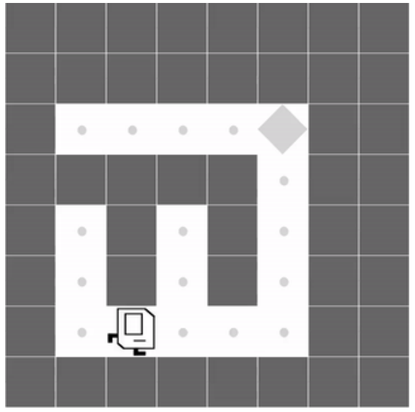


DRL

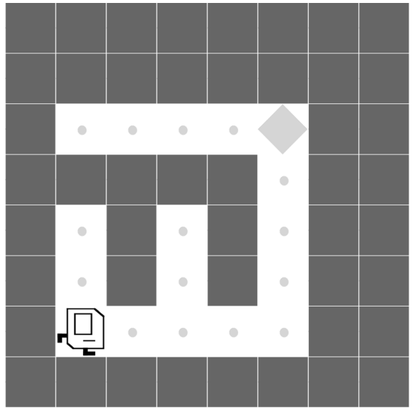


LEAPS

Maze

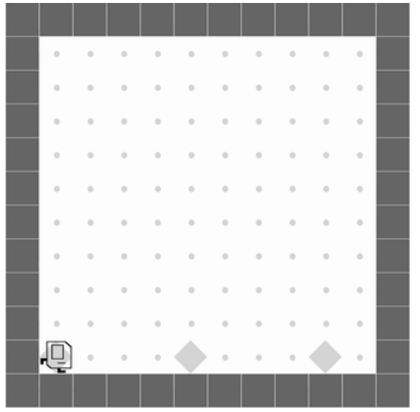


DRL

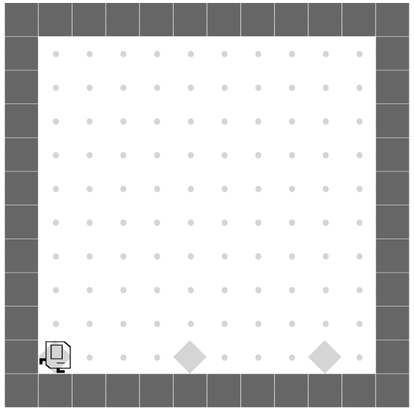


LEAPS

TopOff

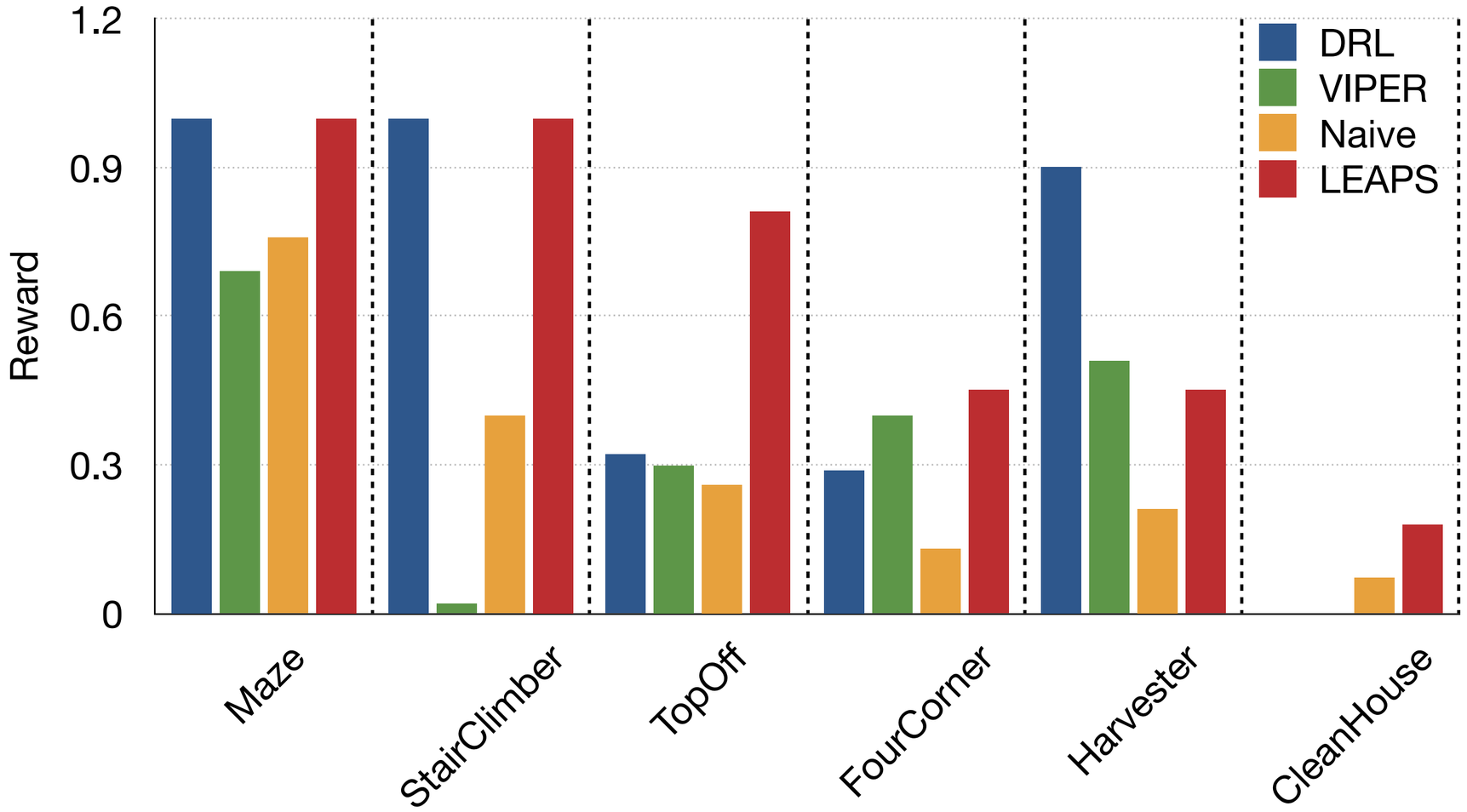


DRL

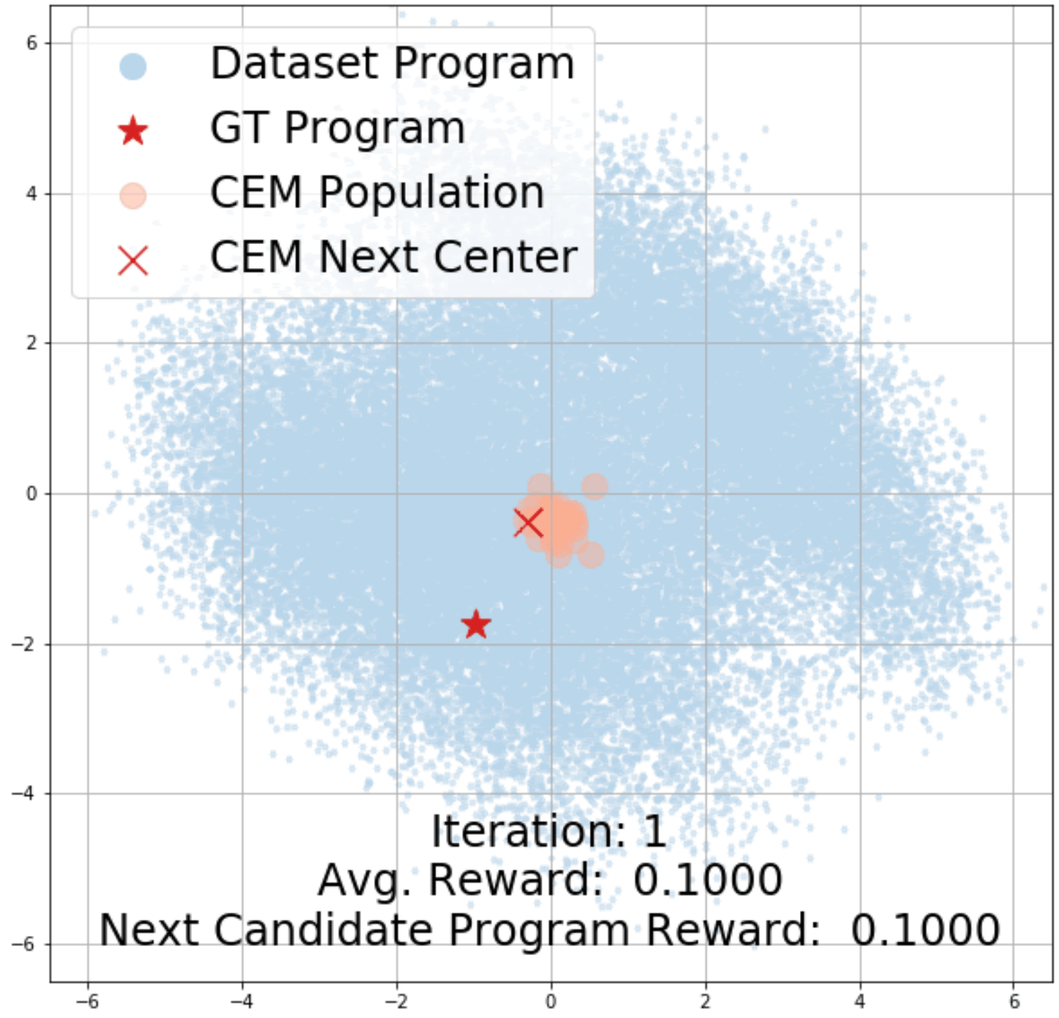


LEAPS

Results


1.2
0.9
0.6
0.3
0
Reward
DRL
VIPER
Naive
LEAPS
Maze
StairClimber
TopOff
FourCorner
Harvester
CleanHouse

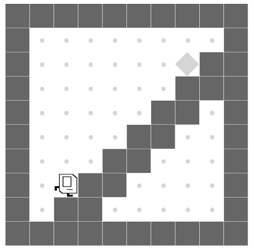
Results - CEM trajectory Visualization



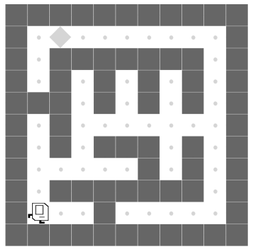
Zero-shot Generalization

Learning

StairClimber



Maze

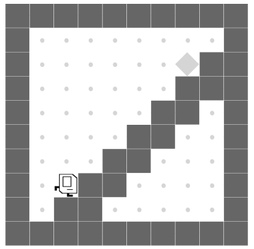


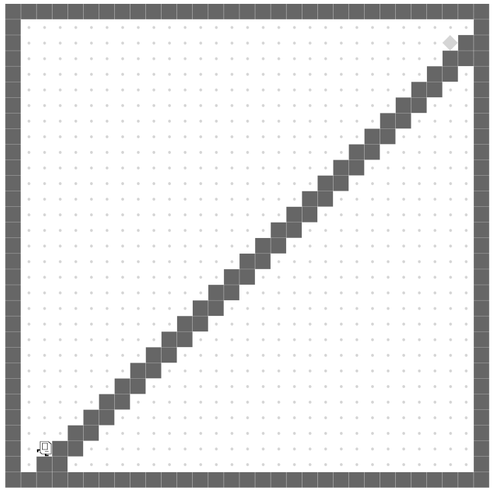
Zero-shot Generalization

Learning

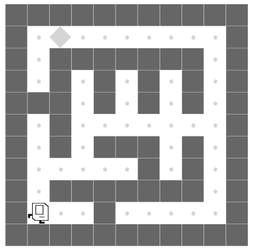
Evaluation on 100 x 100

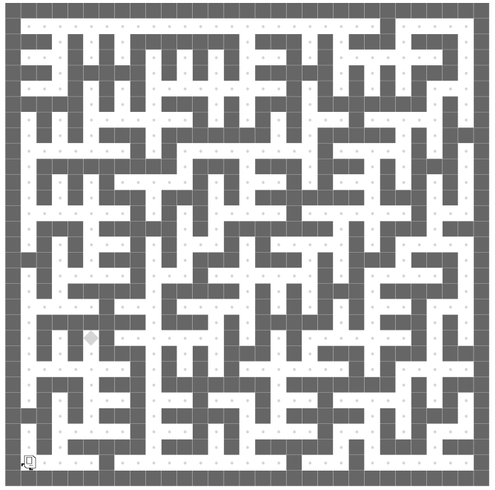
StairClimber





Maze

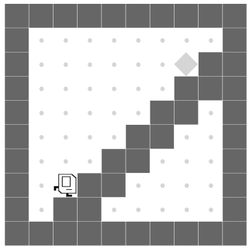




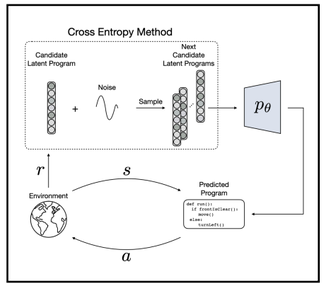
LEAPS Zero-shot Generalization

Learning

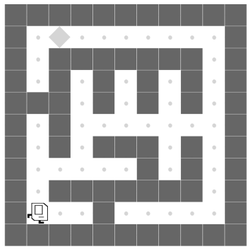
StairClimber



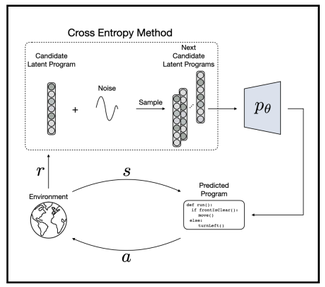
CEM search



Maze



CEM search

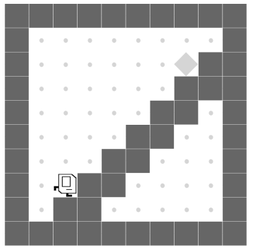


LEAPS Zero-shot Generalization

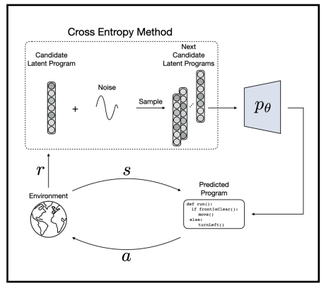
Learning

LEAPS policy

StairClimber

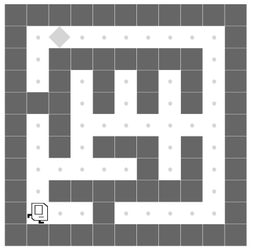


CEM search

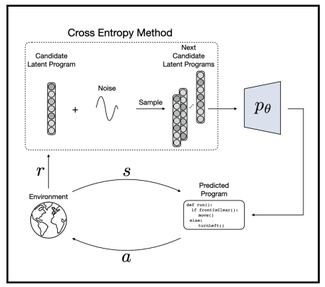



```
DEF run()
   WHILE noMarkersPresent()
      turnRight
      move
   WHILE rightIsClear()
      turnLeft
```

Maze



CEM search




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

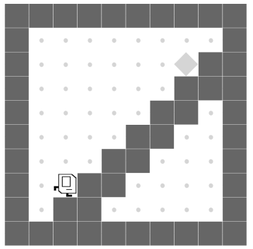
LEAPS Zero-shot Generalization

Learning

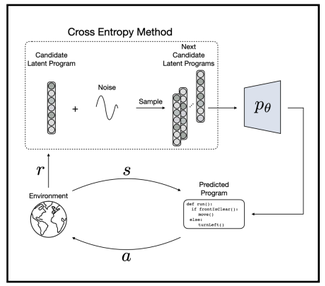
LEAPS policy

Evaluation on 100 x 100

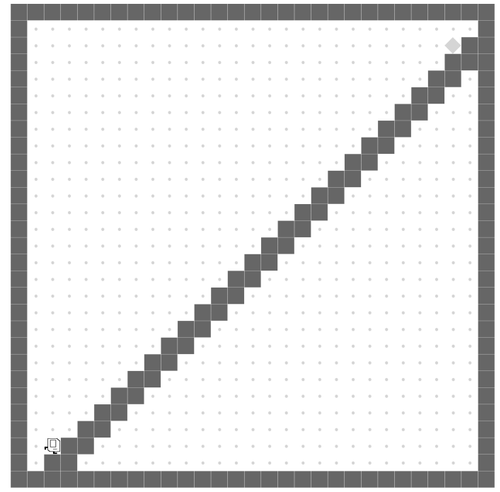
StairClimber



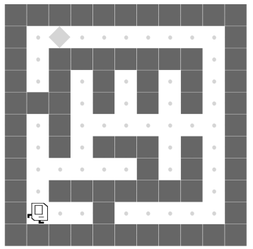
CEM search



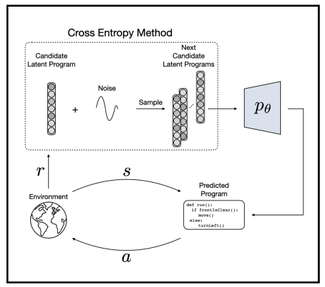

```
DEF run()
   WHILE noMarkersPresent()
      turnRight
      move
   WHILE rightIsClear()
      turnLeft
```



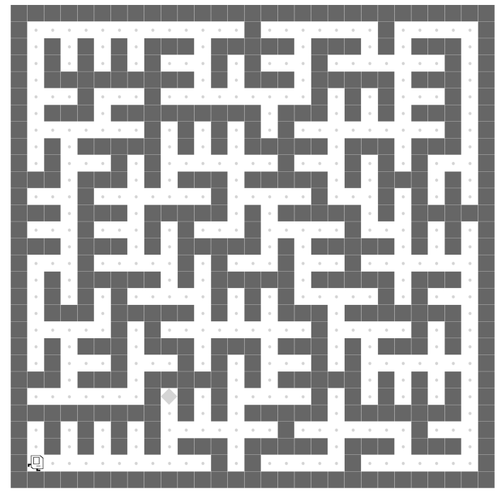
Maze



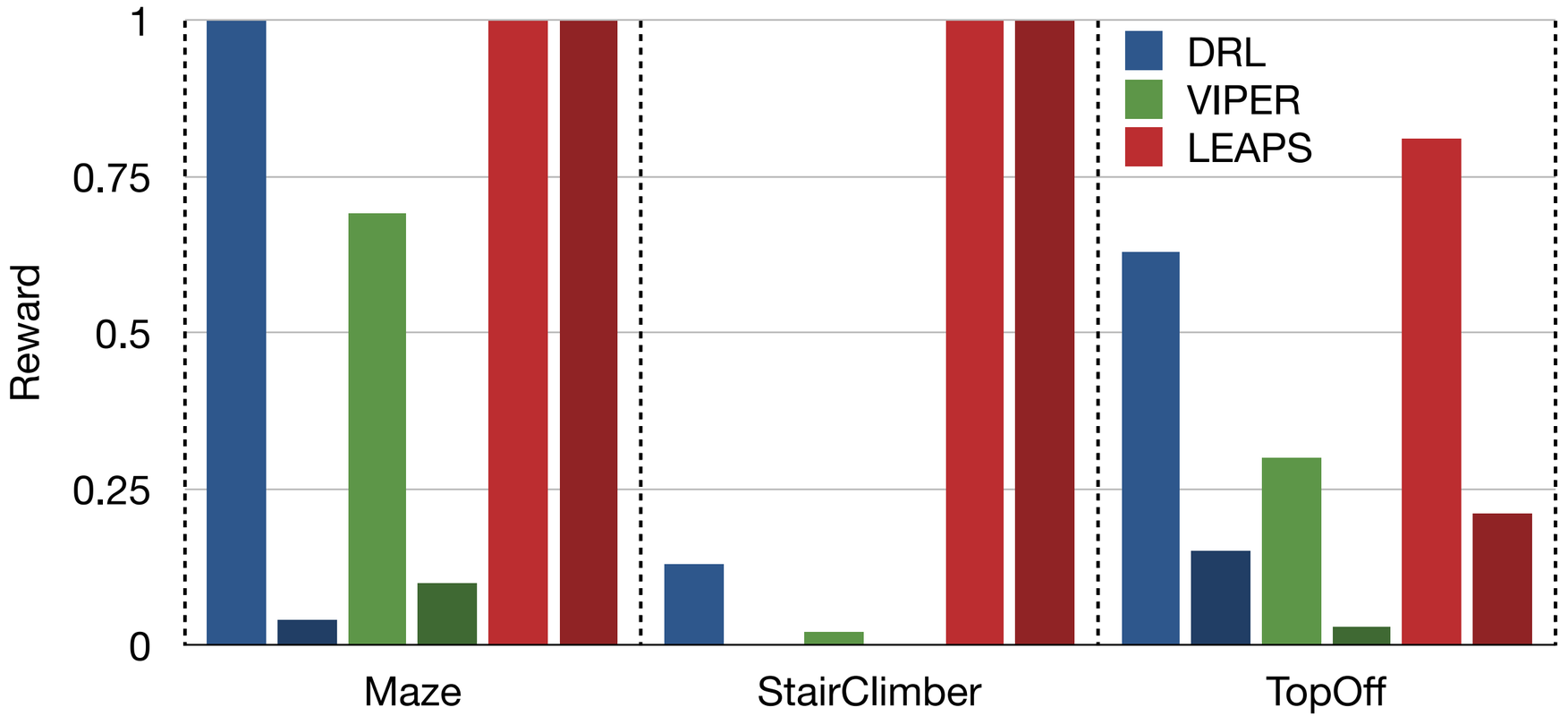
CEM search




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

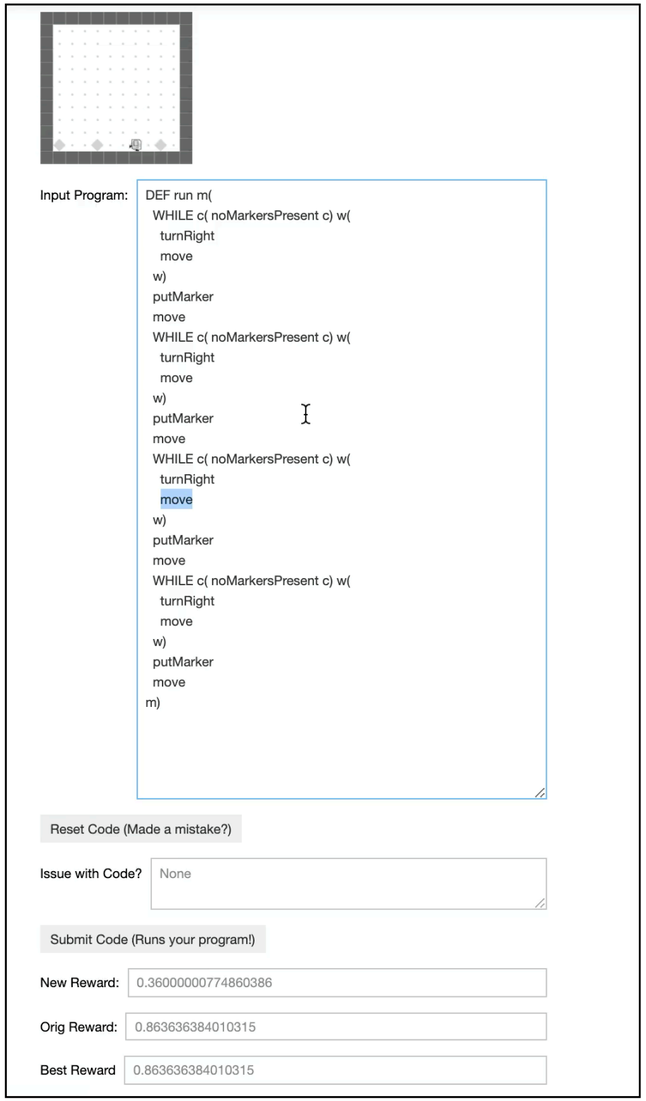


Results - Zero-shot Generalization

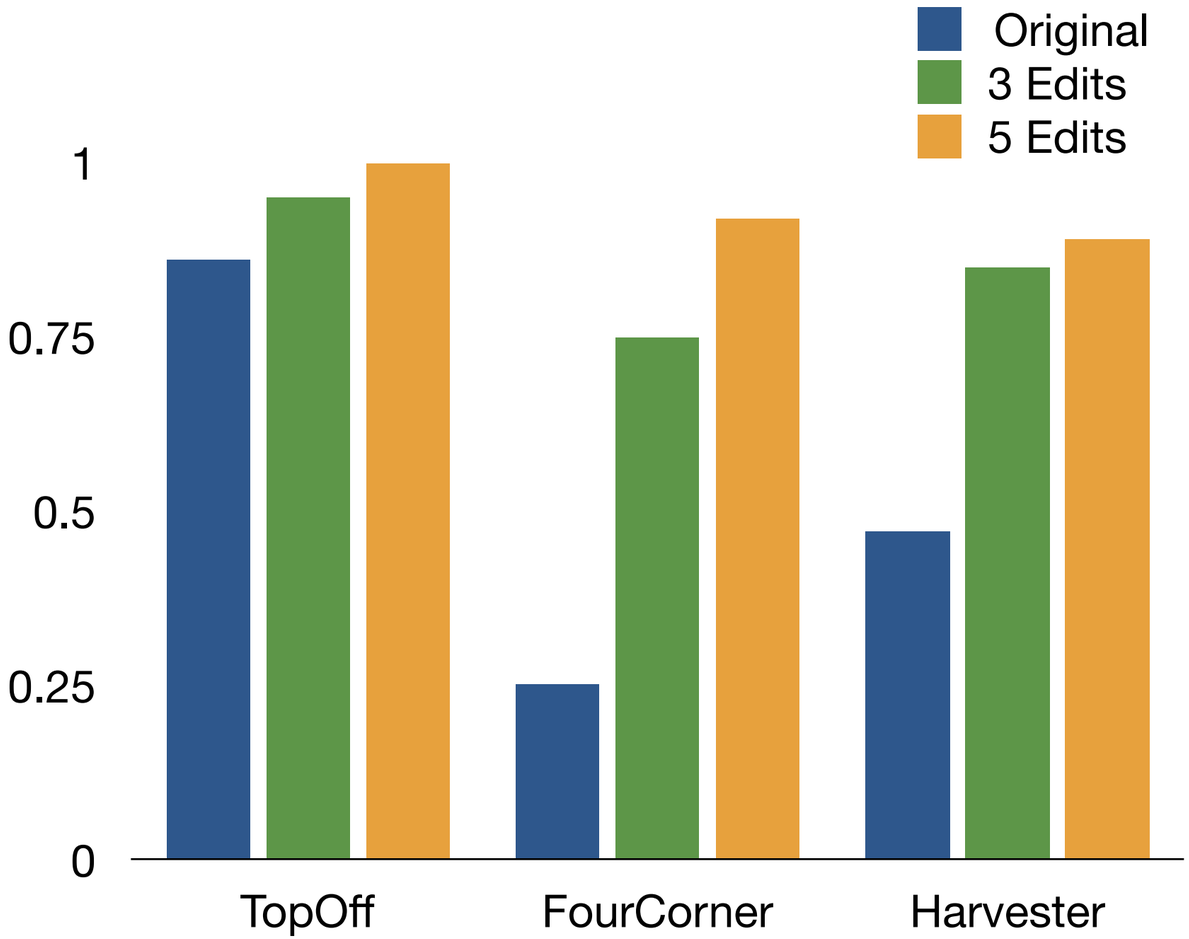


Interpretability

Human Debugging Interface

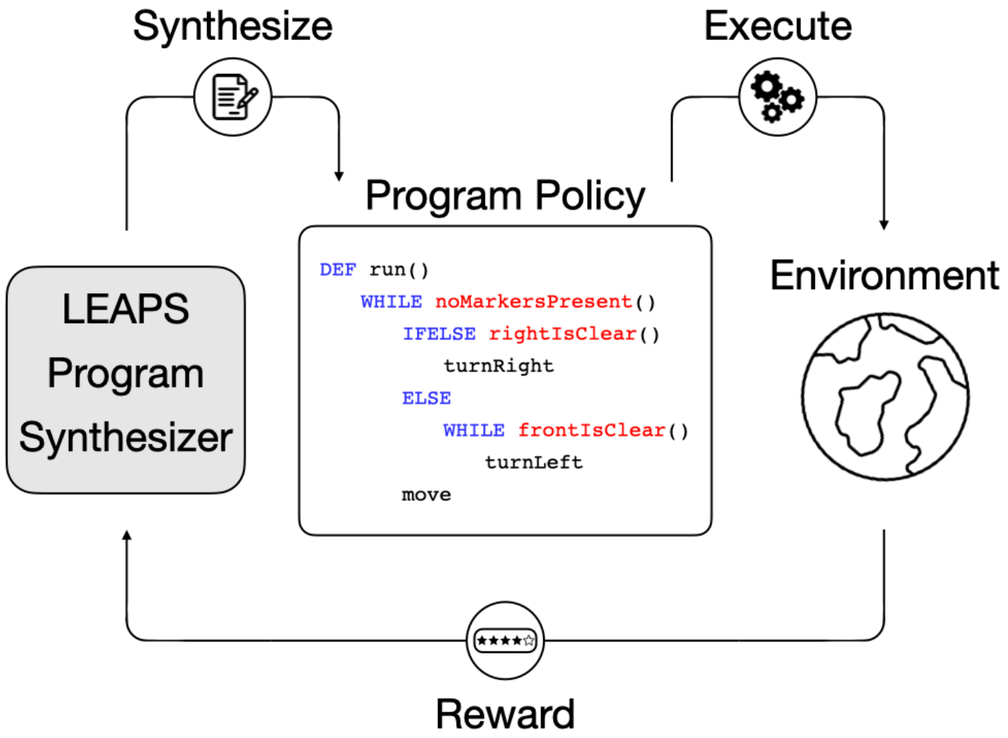


Performance Improvement



Takeaways

- We learn to synthesize a program as a policy purely from reward
- We first learn a program embedding space and then search for a task-solving program
- Our synthesized programs achieve good performance, and are more generalizable and interpretable



Learning to Synthesize Programs as Interpretable and Generalizable Policies

Dweep Trivedi*, Jesse Zhang*, Shao-Hua Sun*, Joseph J Lim

For more details...

Paper and code: clvrai.com/leaps