Execute

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

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)
```

Environment

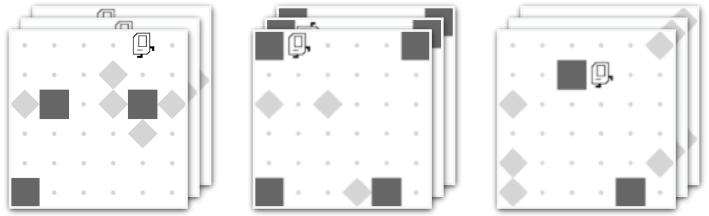
Shao-Hua Sun (孫紹華)

Assistant Professor

Dept. of Electrical Engineering (EE)

National Taiwan University

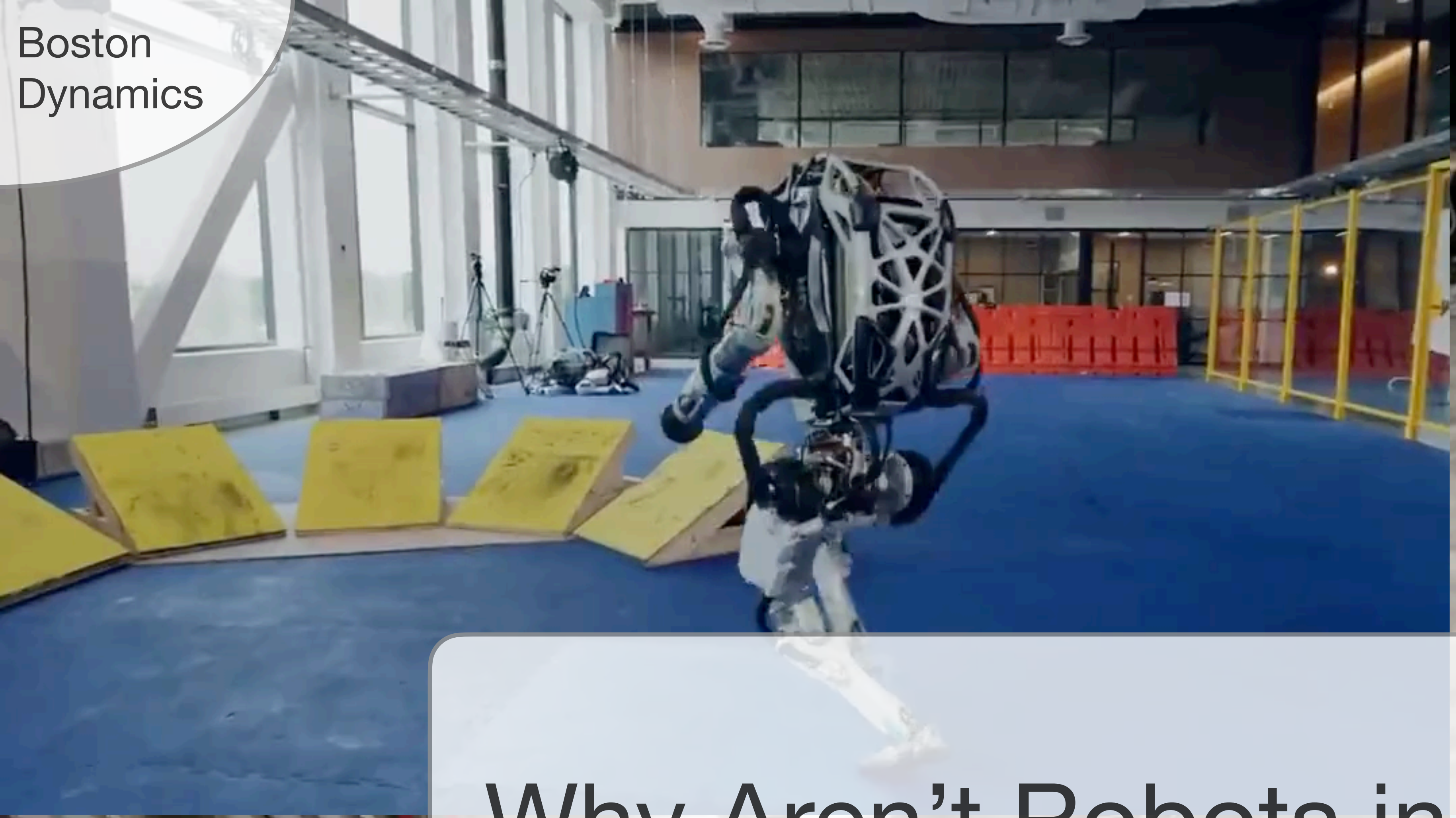
Demonstrations



Machine Learning Summer Schools 2024 @ OIST

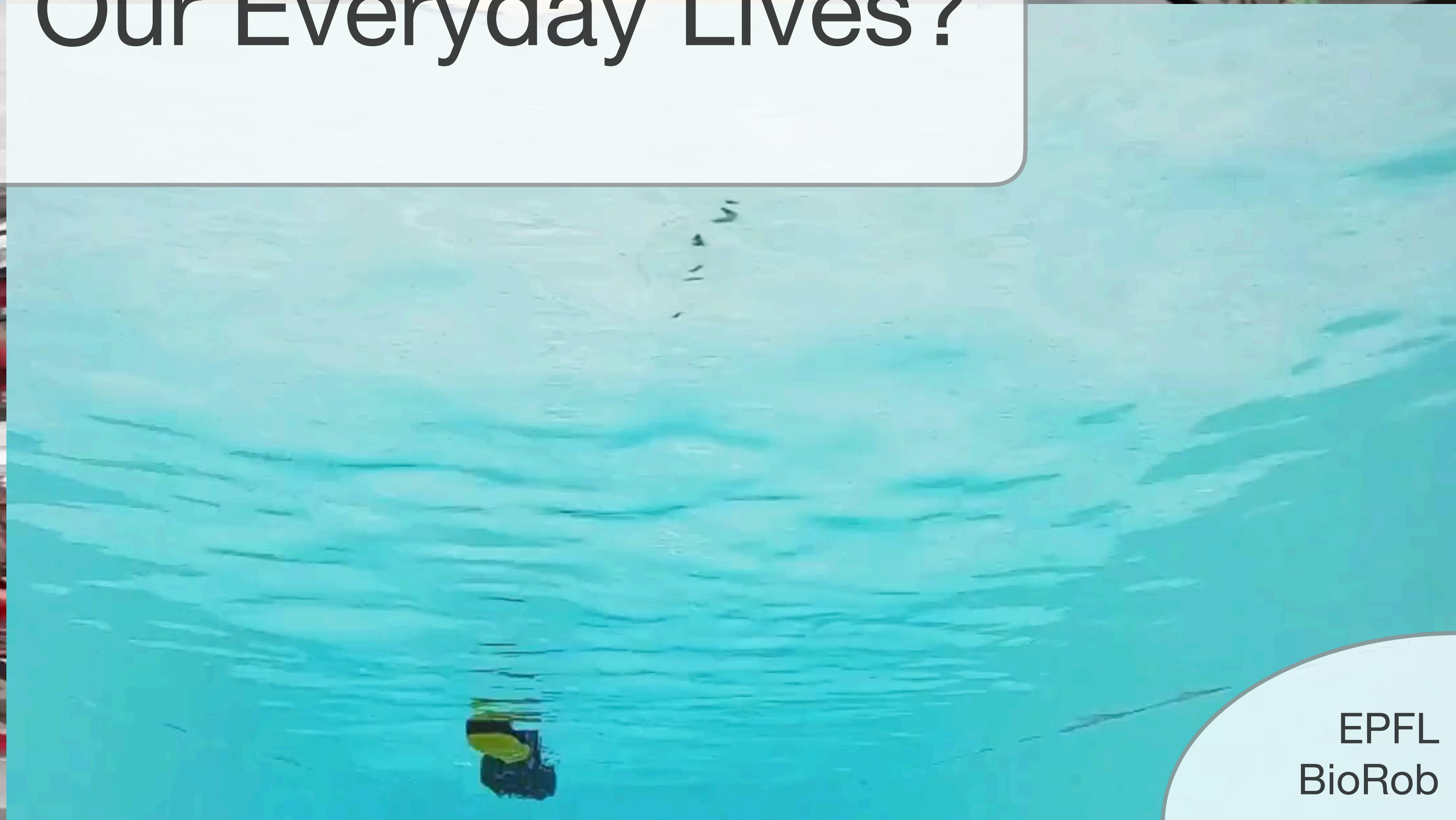
Reward

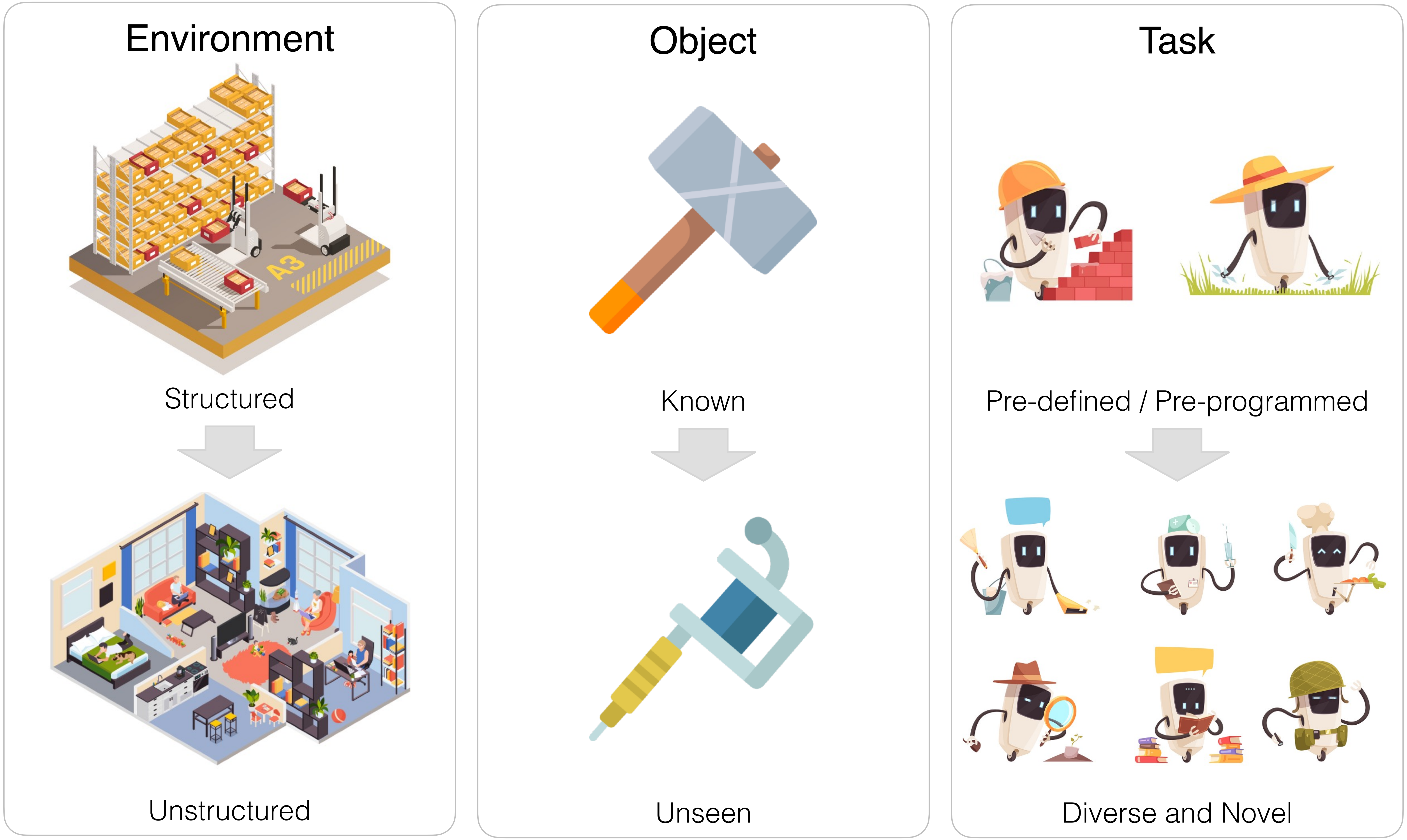
Why Aren’t Robots in Our Everyday Lives?

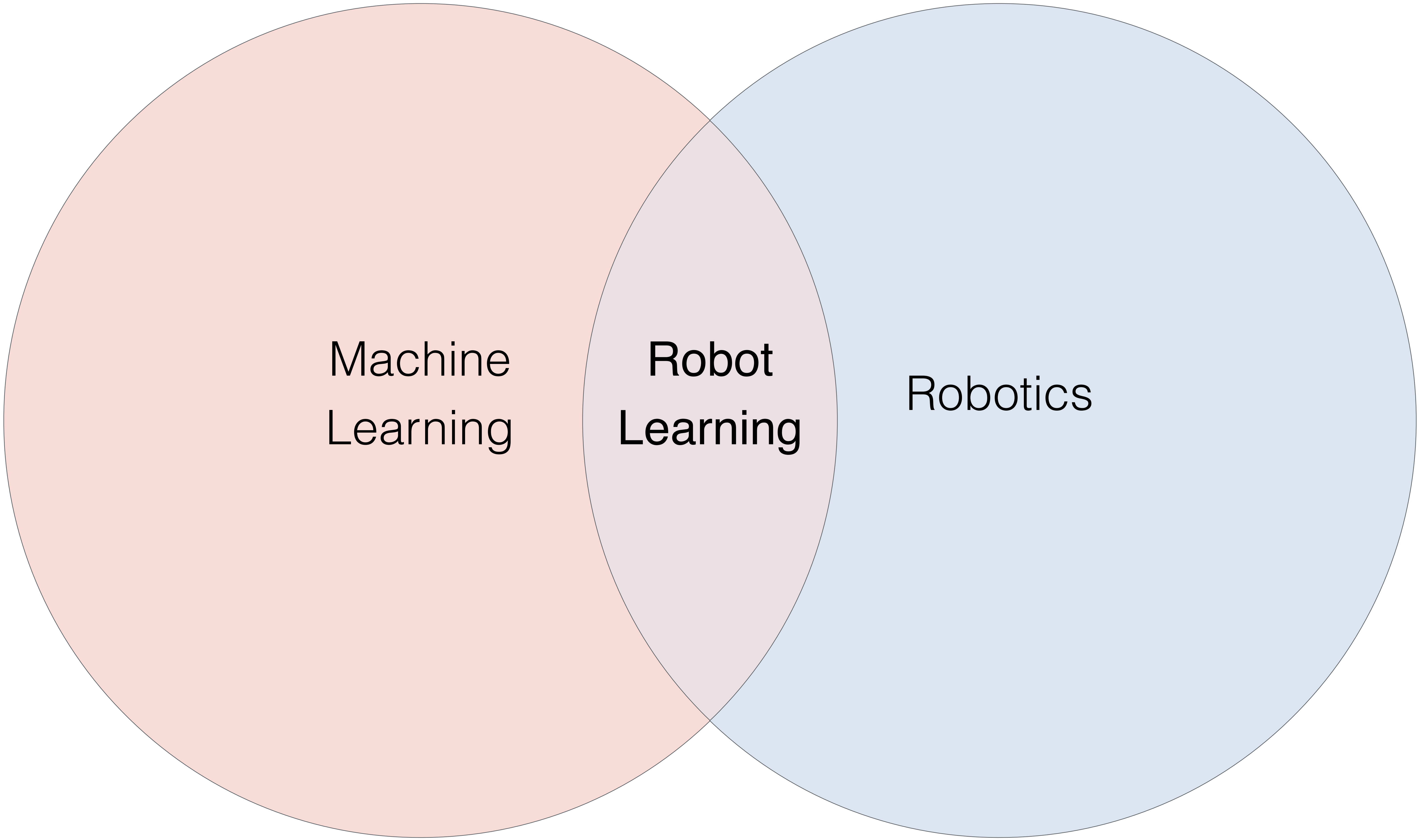

Boston Dynamics

Google Robotics


Tesla


EPFL BioRob


Environment
Structured
Unstructured
Object
Known
Unseen
Task
Pre-defined / Pre-programmed
Diverse and Novel
A3



Machine
Learning
Robot
Learning
Robotics

Supervised Learning

Image Classification

Model	image size	# parameters	Mult-Adds	Top 1 Acc. (%)	Top 5 Acc. (%)
Inception V2 [29]	224×224	11.2 M	1.94 B	74.8	92.2
NASNet-A (5 @ 1538)	**299×299**	**10.9 M**	**2.35 B**	**78.6**	**94.2**
Inception V3 [60]	299×299	23.8 M	5.72 B	78.8	94.4
Xception [9]	299×299	22.8 M	8.38 B	79.0	94.5
Inception ResNet V2 [58]	299×299	55.8 M	13.2 B	80.1	95.1
NASNet-A (7 @ 1920)	**299×299**	**22.6 M**	**4.93 B**	**80.8**	**95.3**
ResNeXt-101 (64 x 4d) [68]	320×320	83.6 M	31.5 B	80.9	95.6
PolyNet [69]	331×331	92 M	34.7 B	81.3	95.8
DPN-131 [8]	320×320	79.5 M	32.0 B	81.5	95.8
SENet [25]	**320×320**	**145.8 M**	**42.3 B**	**82.7**	**96.2**
NASNet-A (6 @ 4032)	**331×331**	**88.9 M**	**23.8 B**	**82.7**	**96.2**

Table 2. Performance of architecture search and other published state-of-the-art models on ImageNet classification. Mult-Adds indicate the number of composite multiply-accumulate operations for a single image. Note that the composite multiple-accumulate operations are calculated for the image size reported in the table. Model size for [25] calculated from open-source implementation.

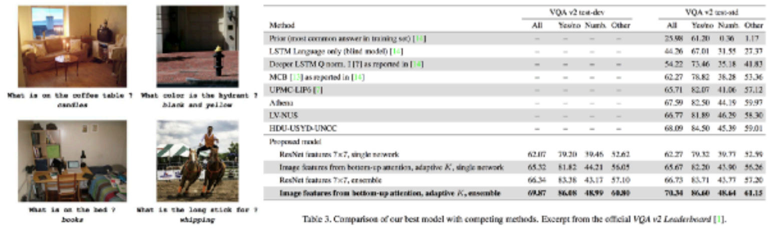
Zoph *et al.* Learning Transferable Architectures for Scalable Image Recognition

Instance Segmentation

Figure 5. More results of **Mask R-CNN** on COCO test images, using ResNet-101-FPN and running at 5 fps, with 35.7 mask AP (Table 1).

He *et al.* Mask R-CNN

Visual Question Answering



Teney *et al.* Tips and Tricks for Visual Question Answering: Learnings from the 2017 Challenge

Machine Translation

Source	"The reason Boeing are doing this is to cram more seats in to make their plane more competitive with our products," said Kevin Keniston, head of passenger comfort at Europe's Airbus.	
PBMT	"La raison pour laquelle Boeing sont en train de faire, c'est de concentrer davantage de sièges pour prendre leur avion plus compétitive avec nos produits", a déclaré Kevin M. Keniston, chef du confort des passagers de l'Airbus de l'Europe.	3.0
GNMT	"La raison pour laquelle Boeing fait cela est de créer plus de sièges pour rendre son avion plus compétitif avec nos produits", a déclaré Kevin Keniston, chef du confort des passagers chez Airbus.	6.0
Human	"Boeing fait ça pour pouvoir caser plus de sièges et rendre ses avions plus compétitifs par rapports à nos produits", a déclaré Kevin Keniston, directeur de Confort Passager chez l'avionneur européen Airbus.	6.0
Source	When asked about this, an official of the American administration replied: "The United States is not conducting electronic surveillance aimed at offices of the World Bank and IMF in Washington."	
PBMT	Interrogé à ce sujet, un responsable de l'administration américaine a répondu : "Les Etats-Unis n'est pas effectuer une surveillance électronique destiné aux bureaux de la Banque mondiale et du FMI à Washington".	3.0
GNMT	Interrogé à ce sujet, un fonctionnaire de l'administration américaine a répondu: "Les États-Unis n'effectuent pas de surveillance électronique à l'intention des bureaux de la Banque mondiale et du FMI à Washington".	6.0
Human	Interrogé sur le sujet, un responsable de l'administration américaine a répondu: "les Etats-Unis ne mènent pas de surveillance électronique visant les sièges de la Banque mondiale et du FMI à Washington".	6.0

Wu *et al.* Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Word Embeddings

On word embeddings - Part 1 by Ruder

Named Entity Recognition

Named Entity Recognition and Classification with Scikit-Learn by Susan Li

Esteves *et al.* Named Entity Recognition in Twitter using Images and Text

Question Answering

- Input Question:

  `Where do water droplets collide with ice crystals to form precipitation?`

- Input Paragraph:

  `... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. ...`

- Output Answer:

  `within a cloud`

System	Dev EM	Dev F1	Test EM	Test F1
Leaderboard (Oct 8th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
#1 Single - nlnet	-	-	83.5	90.1
#2 Single - QANet	-	-	82.5	89.3
Published				
BiDAF+ELMo (Single)	-	85.8	-	-
R.M. Reader (Single)	78.9	86.3	79.5	86.6
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

Table 2: SQuAD results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

Devlin *et al.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

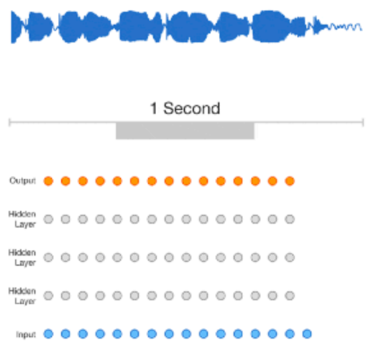
Speech Recognition

Word Error Rate

Senone set	Model/combination step	WER devset (ngram-LM)	WER test (ngram-LM)	WER devset (LSTM-LMs)	WER test (LSTM-LMs)
9k	BLSTM	11.5	8.3	9.2	6.3
27k	BLSTM	11.4	8.0	9.3	6.3
27k-puhpum	BLSTM	11.3	8.0	9.2	6.3
9k	BLSTM+ResNet+LACE+CNN-BLSTM	9.6	7.2	7.7	5.4
9k-puhpum	BLSTM+ResNet+LACE	9.7	7.4	7.8	5.4
9k-puhpum	BLSTM+ResNet+LACE+CNN-BLSTM	9.7	7.3	7.8	5.5
27k	BLSTM+ResNet+LACE	10.0	7.5	8.0	5.8
-	Confusion network combination			7.4	5.2
-	+ LSTM rescoring			7.3	5.2
-	+ ngram rescoring			7.2	5.2
-	+ backchannel penalty			7.2	5.1

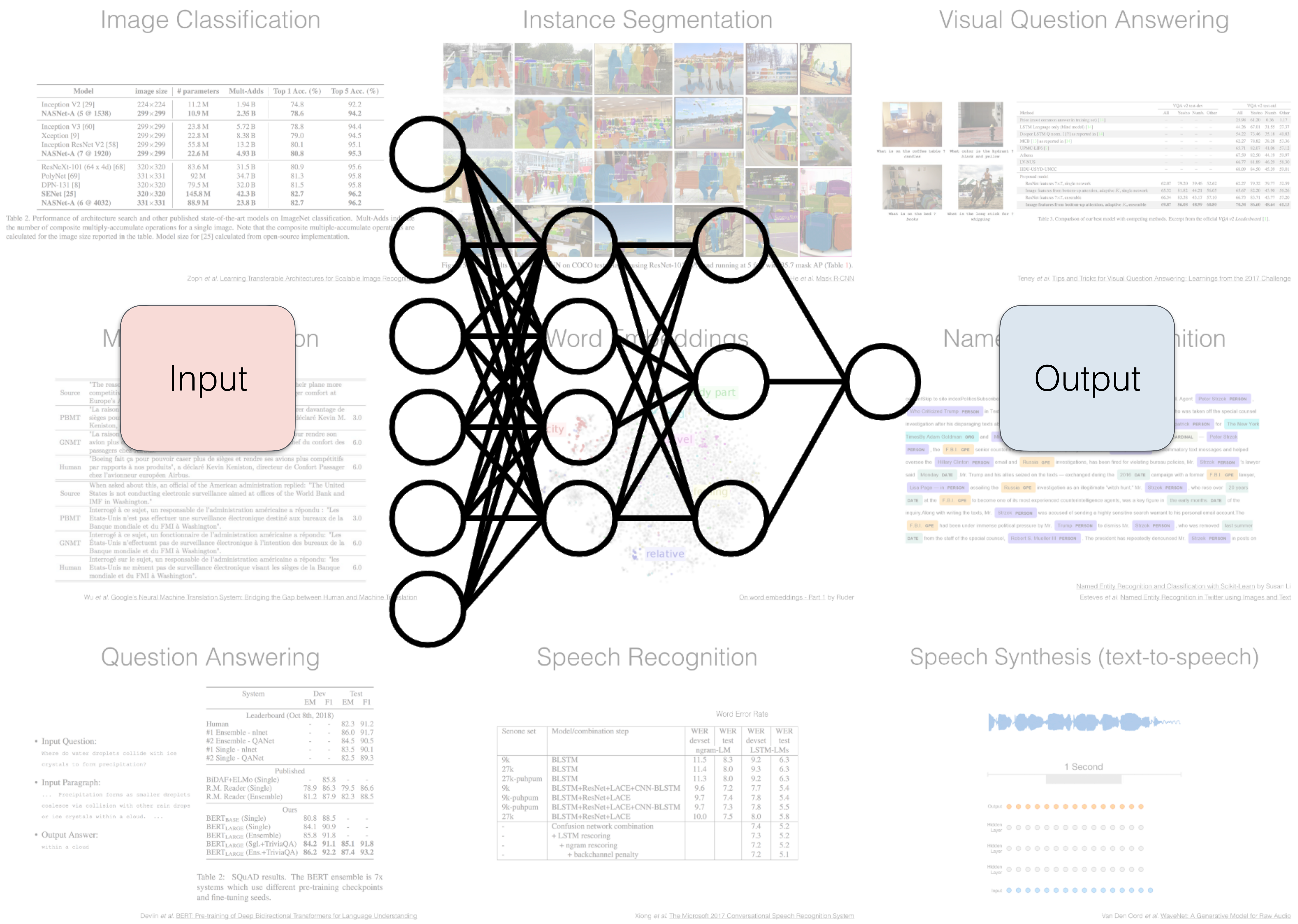
Xiong *et al.* The Microsoft 2017 Conversational Speech Recognition System

Speech Synthesis (text-to-speech)

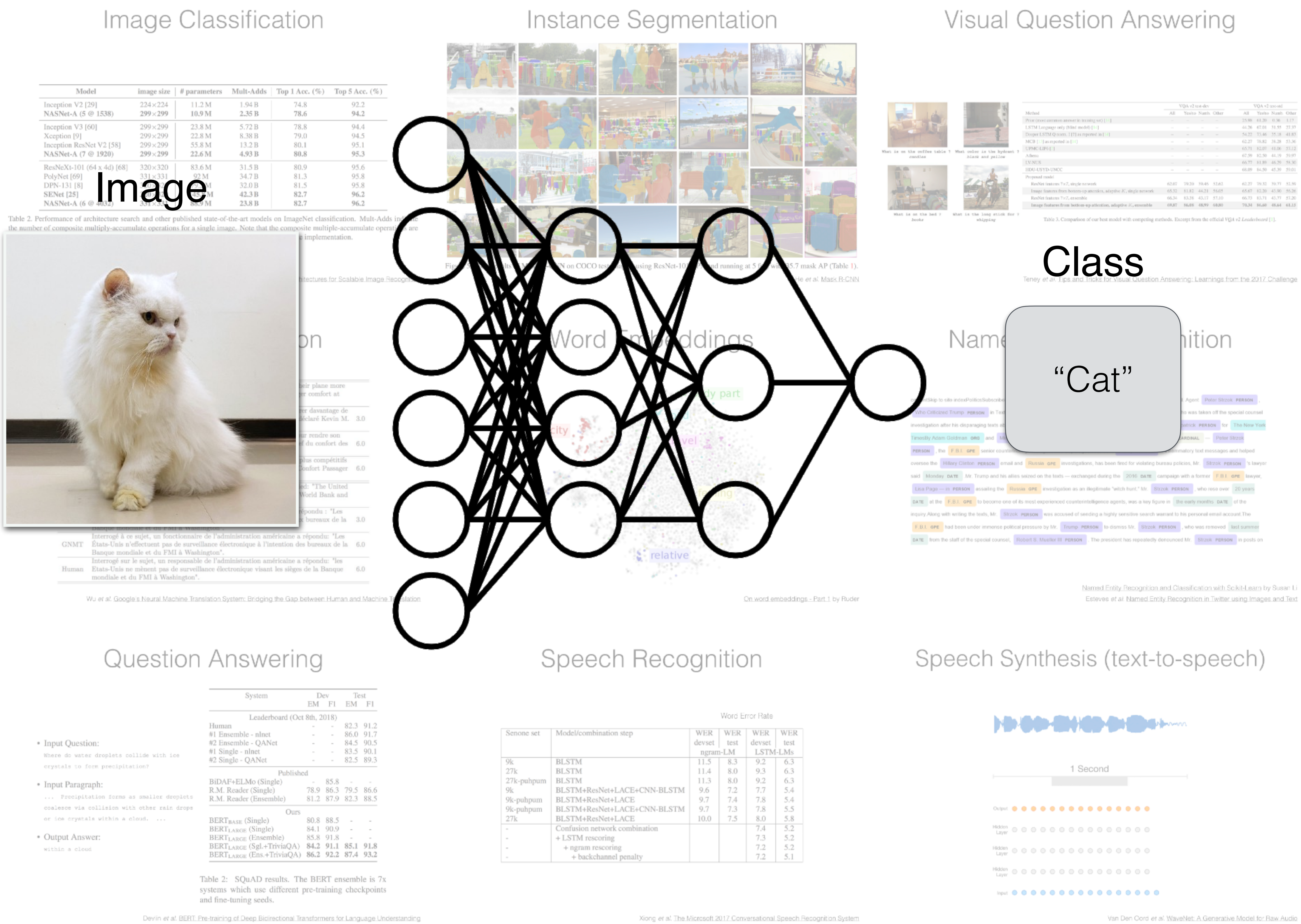


Van Den Oord *et al.* WaveNet: A Generative Model for Raw Audio

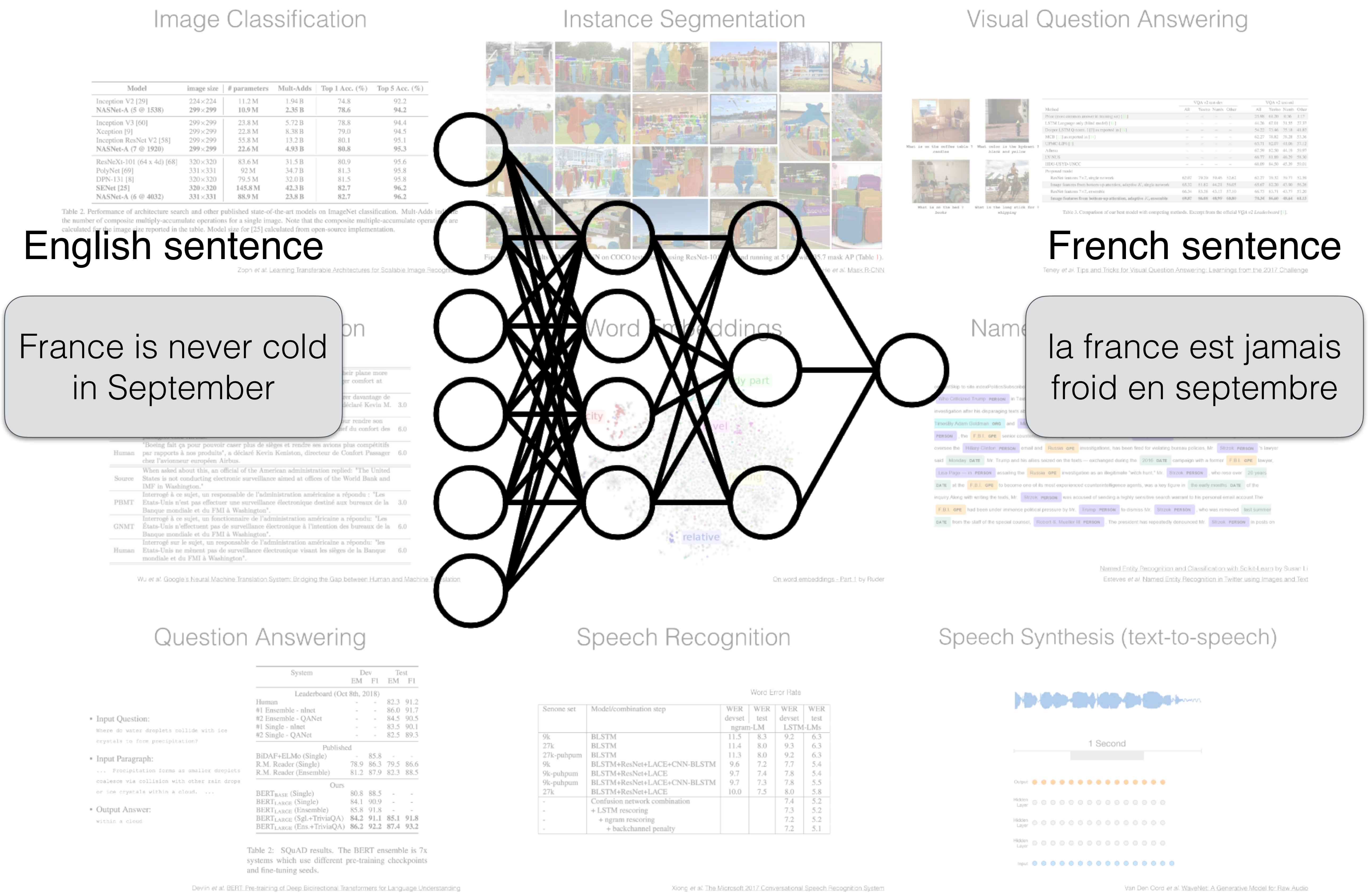
Supervised Learning



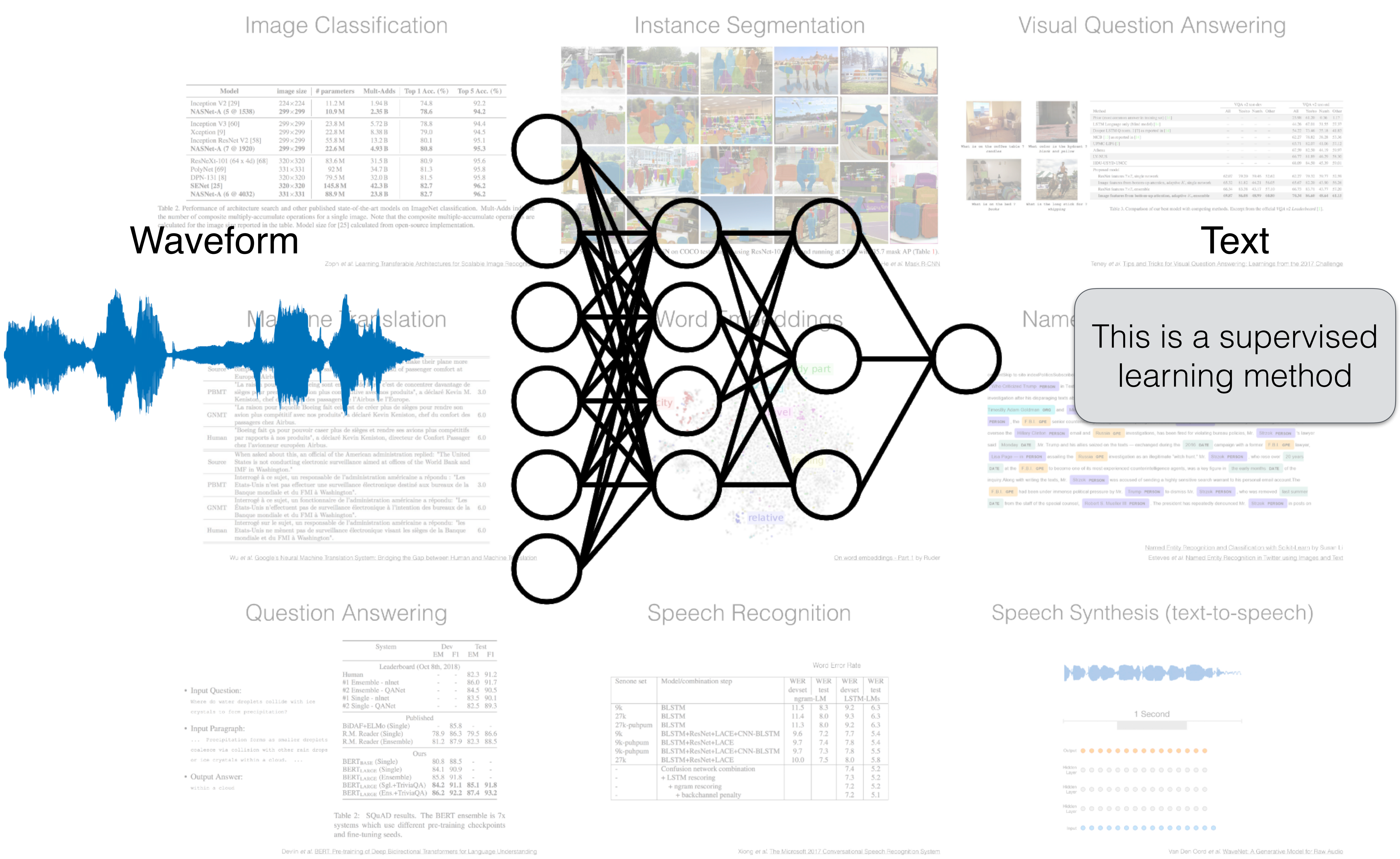
Supervised Learning



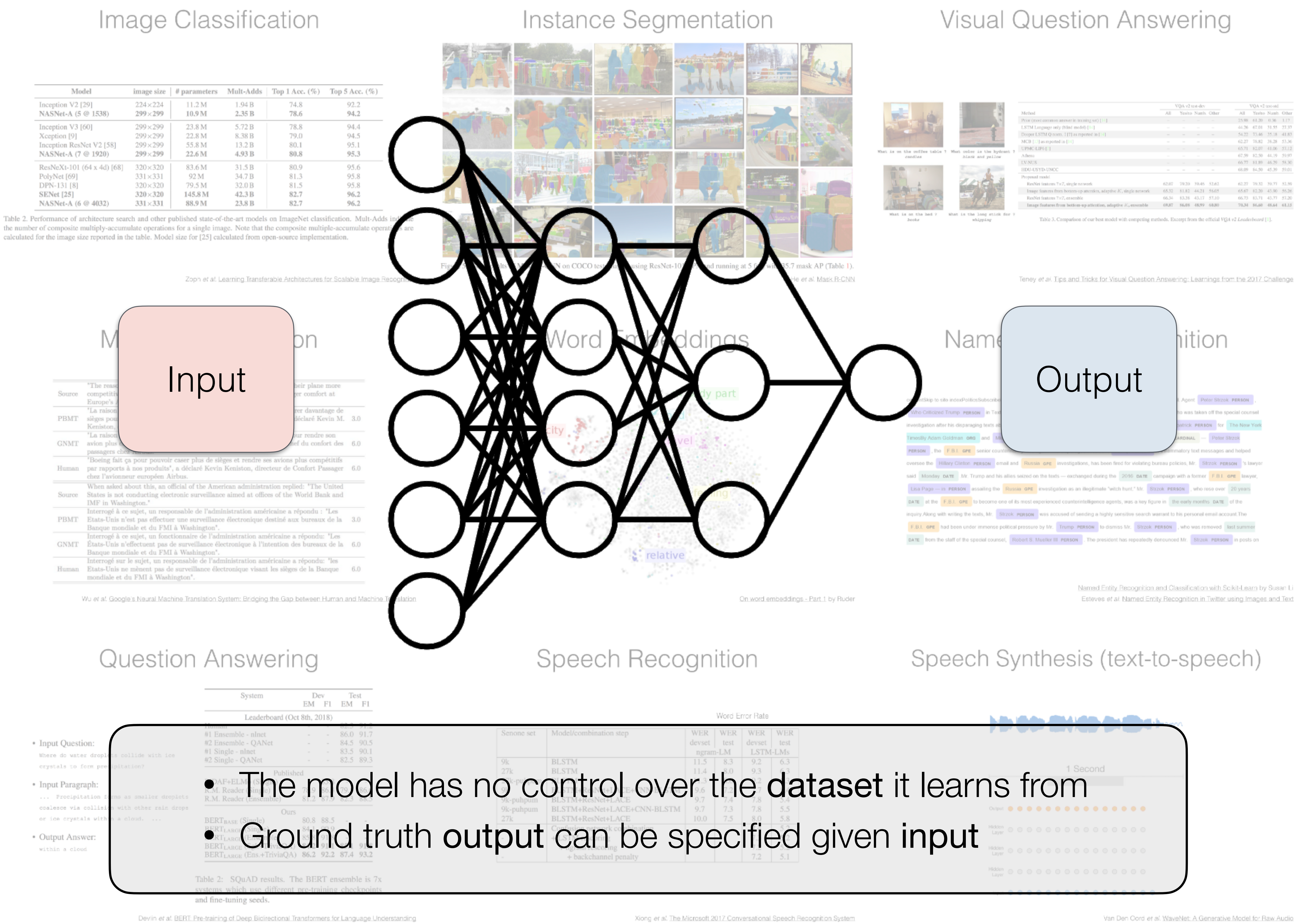
Supervised Learning



Supervised Learning

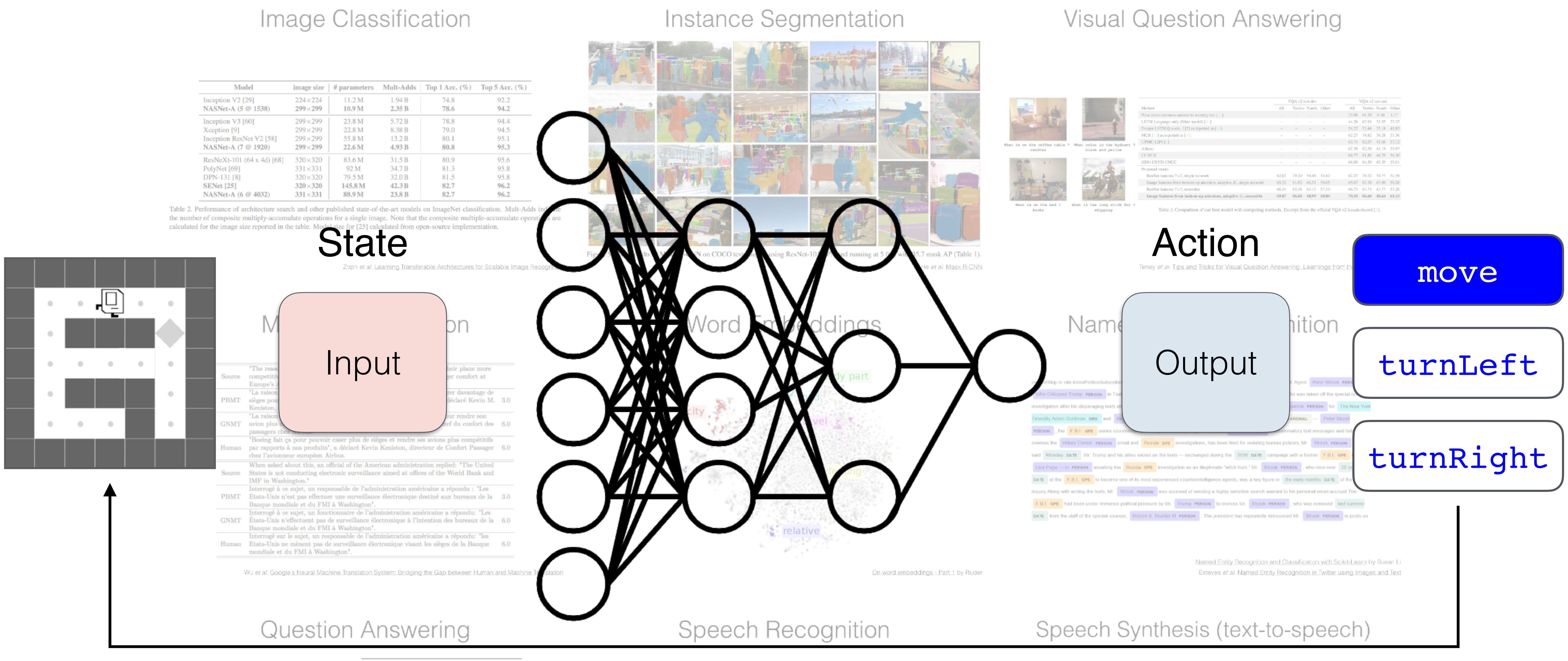


Supervised Learning



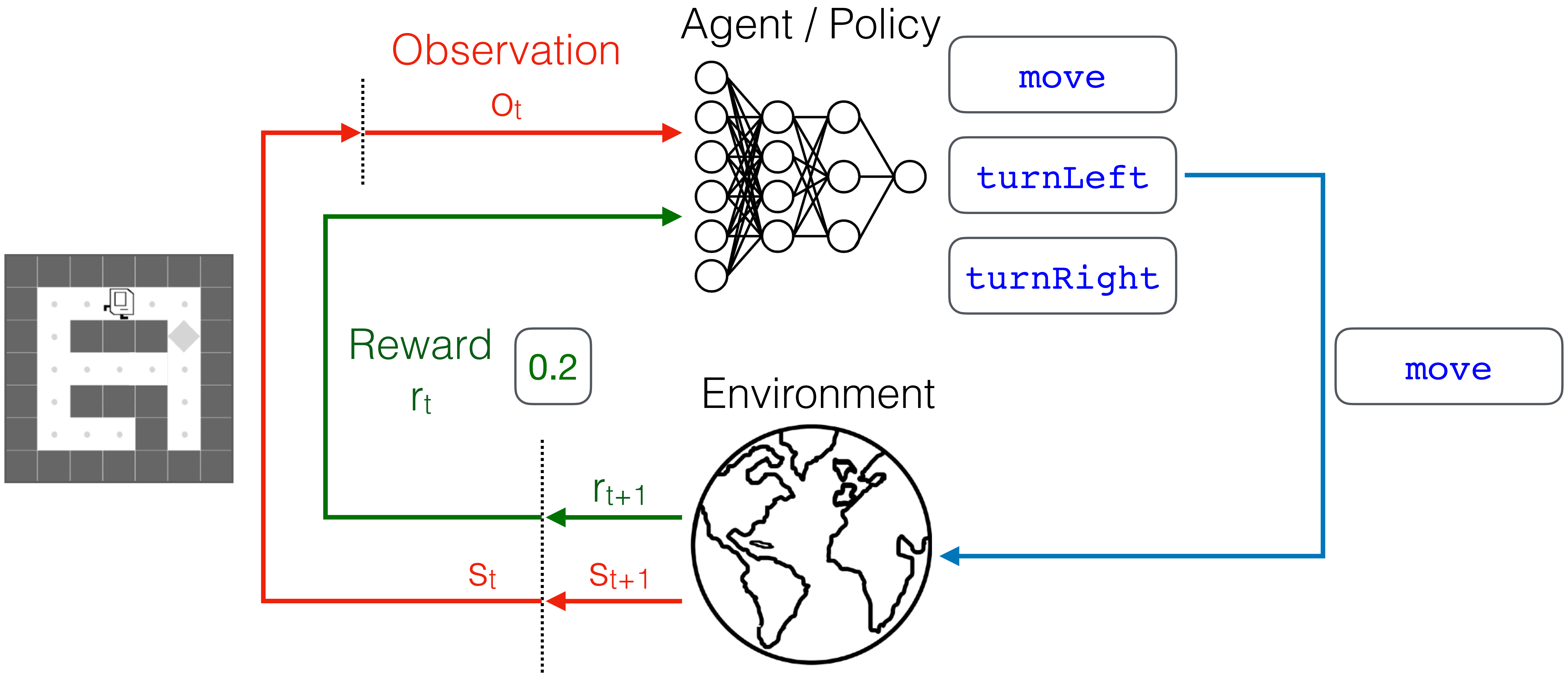
- The model has no control over the **dataset** it learns from
- Ground truth **output** can be specified given **input**

Robot Learning via Supervised Learning



- The model affects the **data** it learns from
- Difficult to specify desired **output (action)** for every **input (state)**

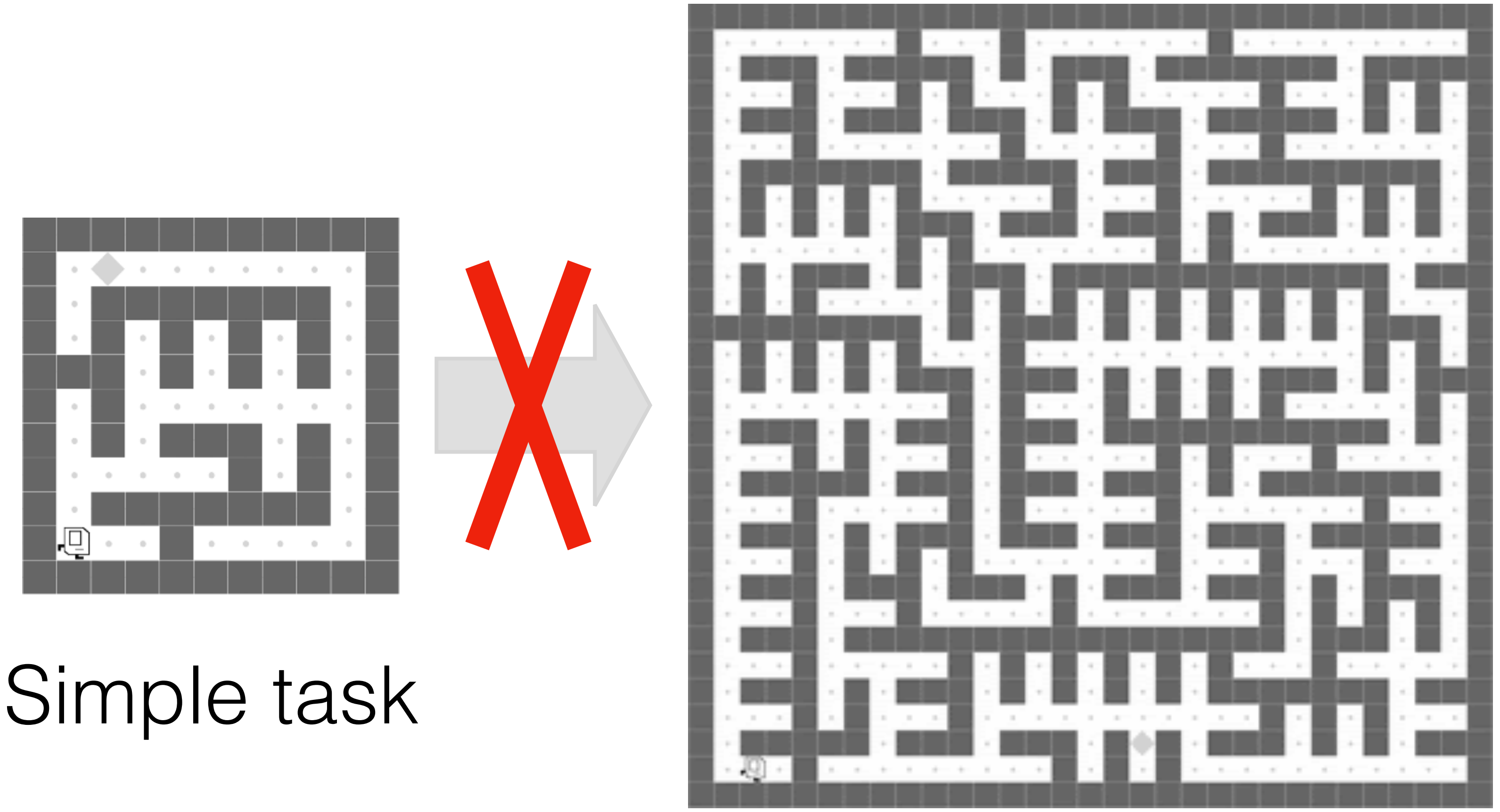
Robot Learning via Deep Reinforcement Learning



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

Robot Learning via Deep Reinforcement Learning - Issues

Generalization

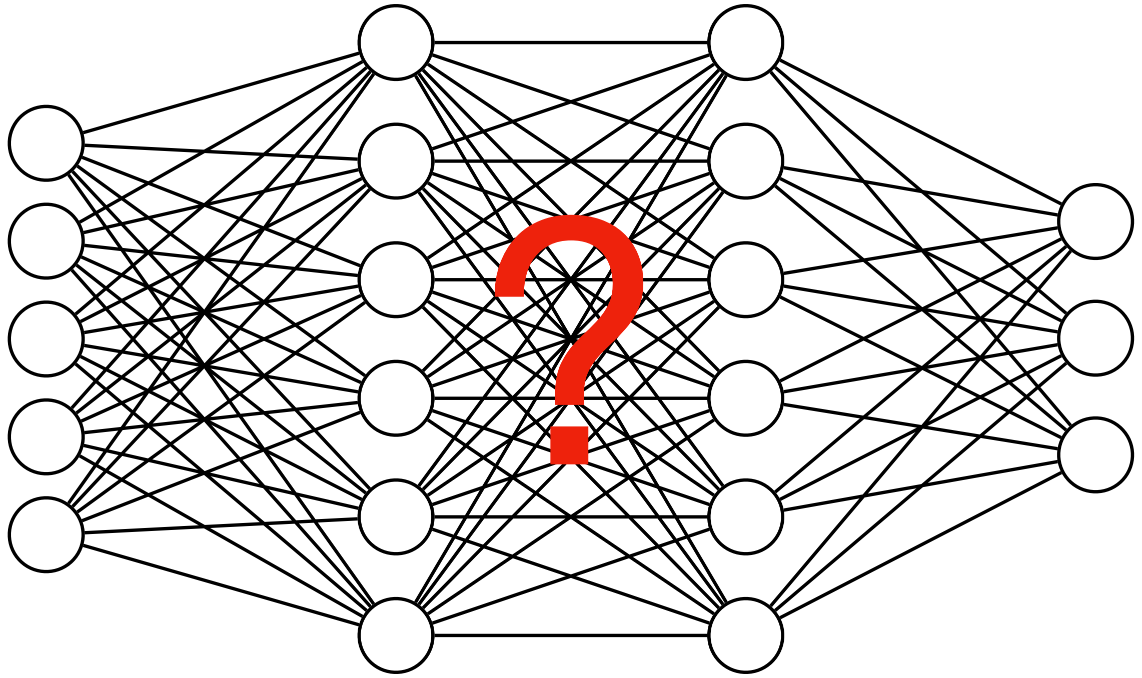


Simple task

Complex task

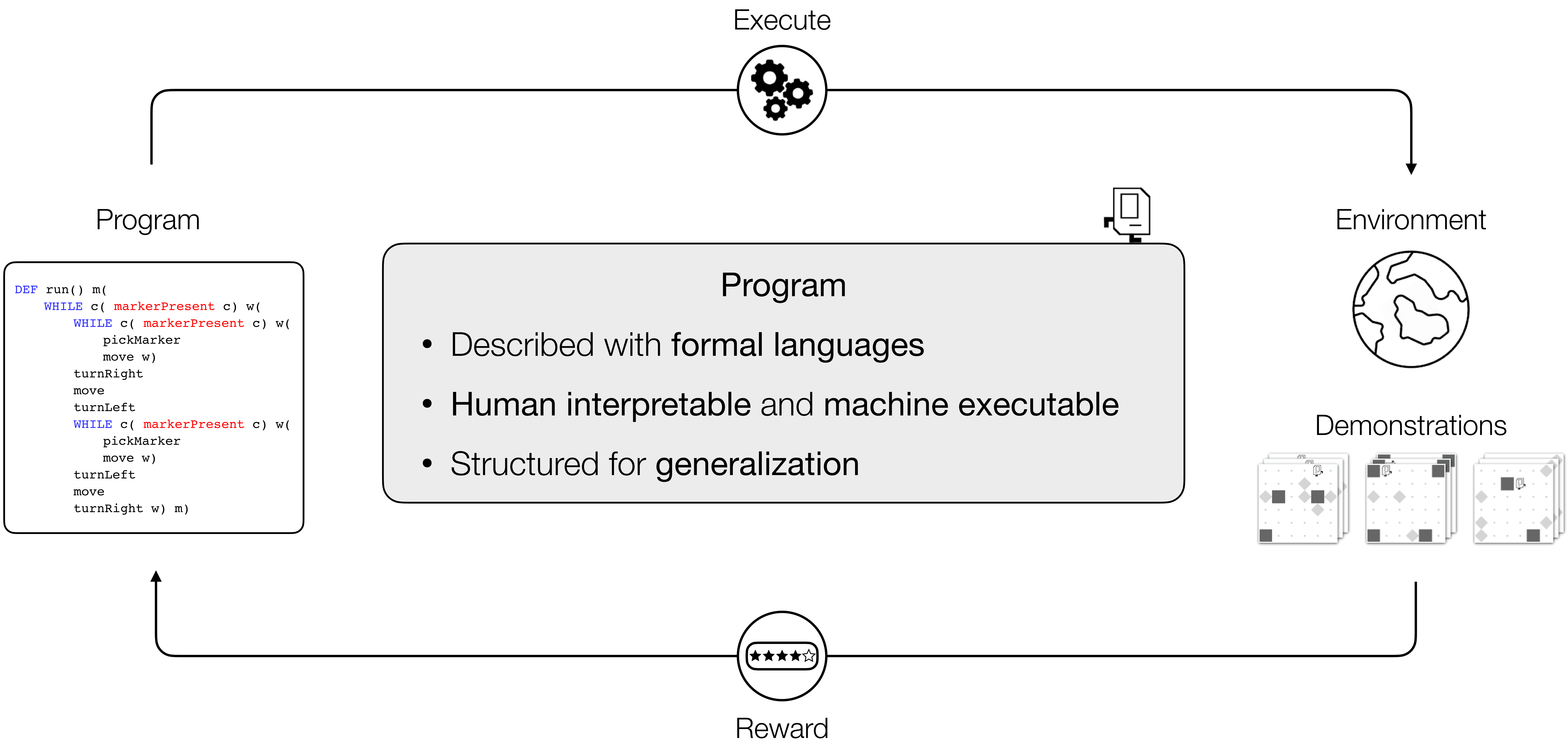
Interpretability

Trust, Safety, and Contestability



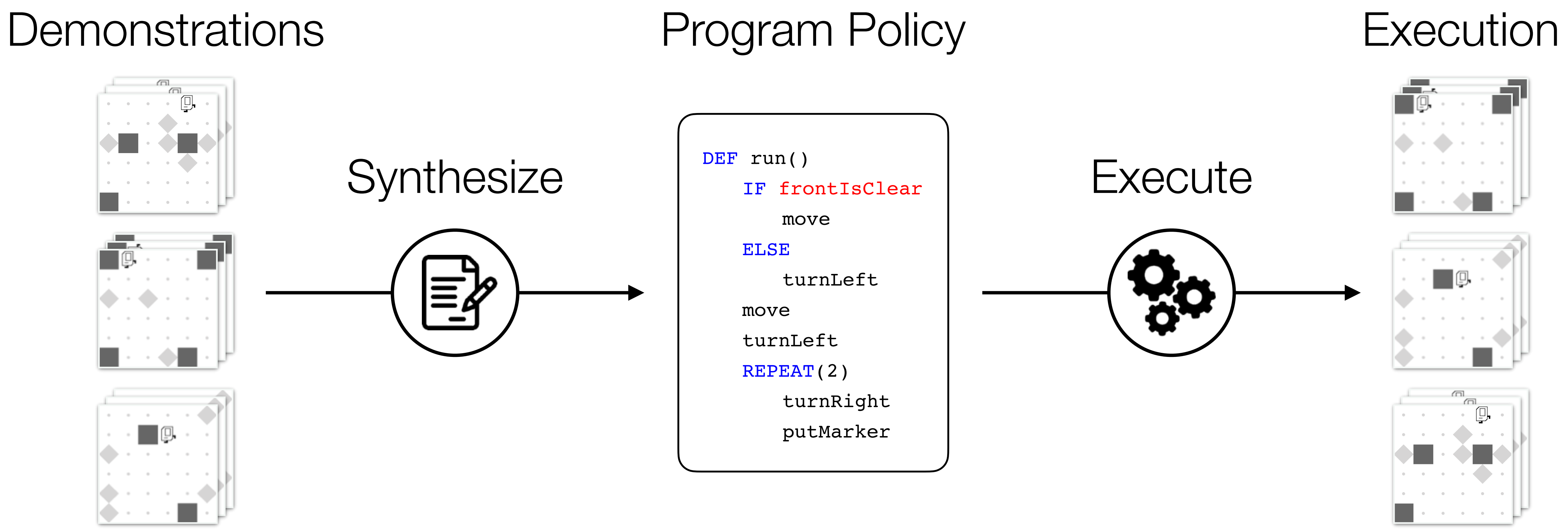
Deep neural network policy

Program as Reinforcement Learning Policies



Neural Program Synthesis from Diverse Demonstration Videos

ICML 2018



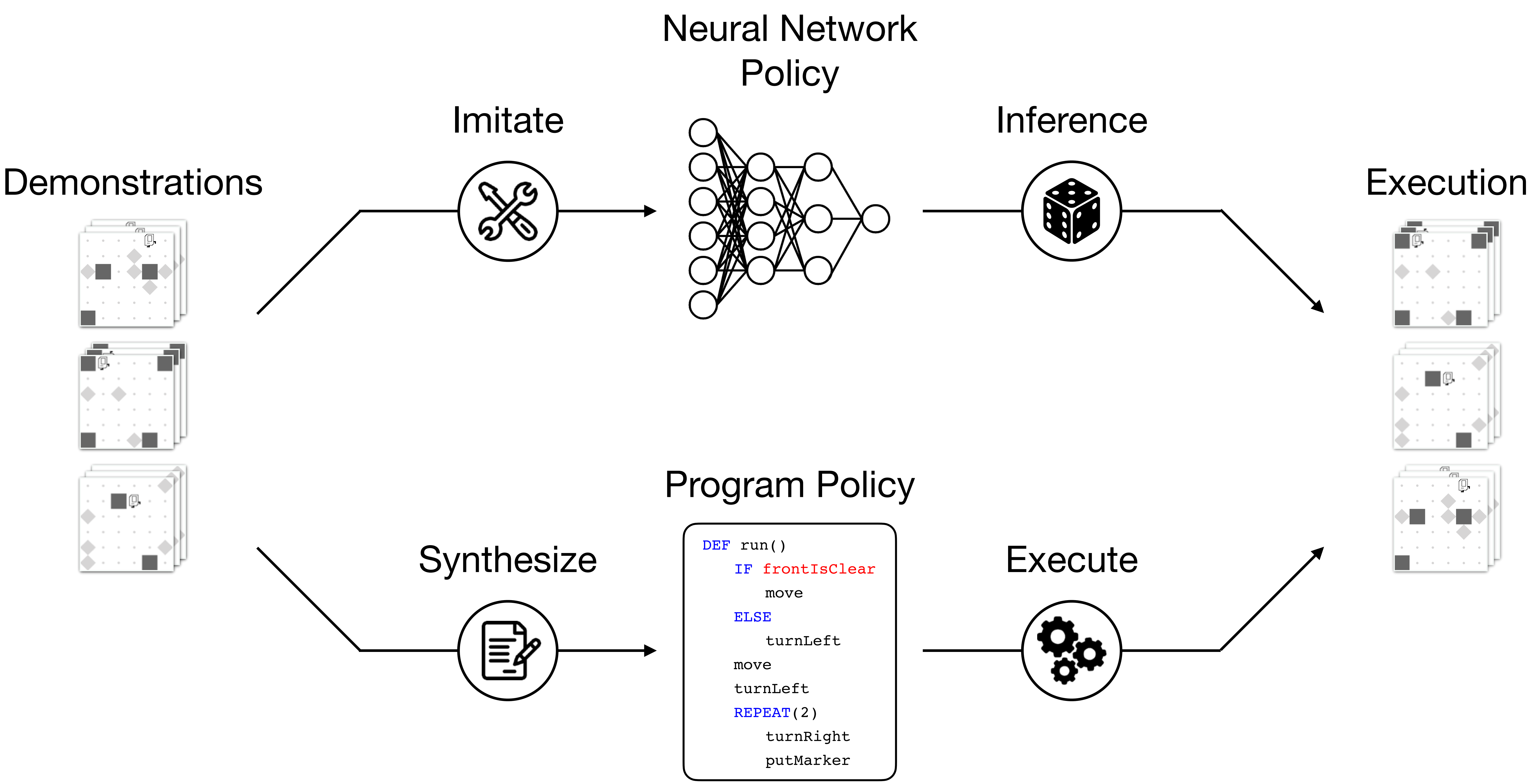

Shao-Hua Sun*


Hyeonwoo Noh*


Sriram Somasundaram


Joseph J. Lim

Imitation Learning via Synthesizing Programs



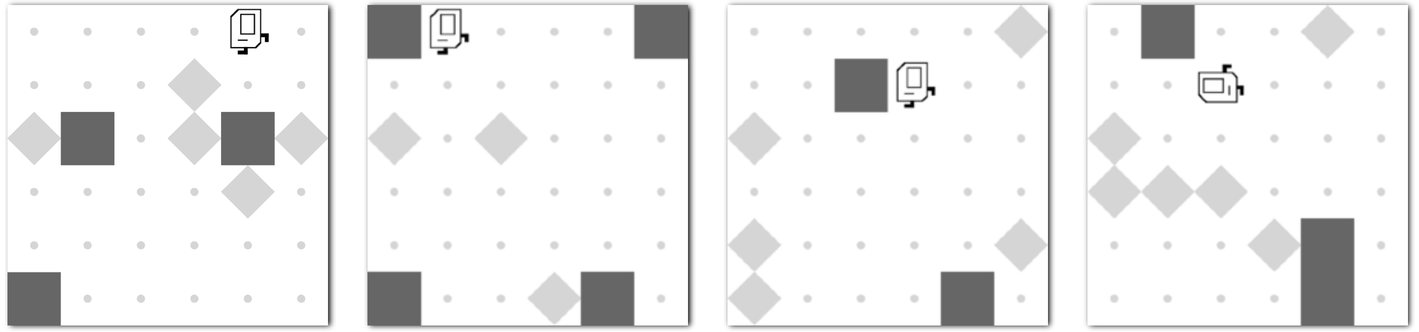
Environments

Karel

Program

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

Demonstrations



Richard E Pattis. “Karel the robot: a gentle introduction to the art of programming.” John Wiley & Sons, Inc., 1981

ViZDoom

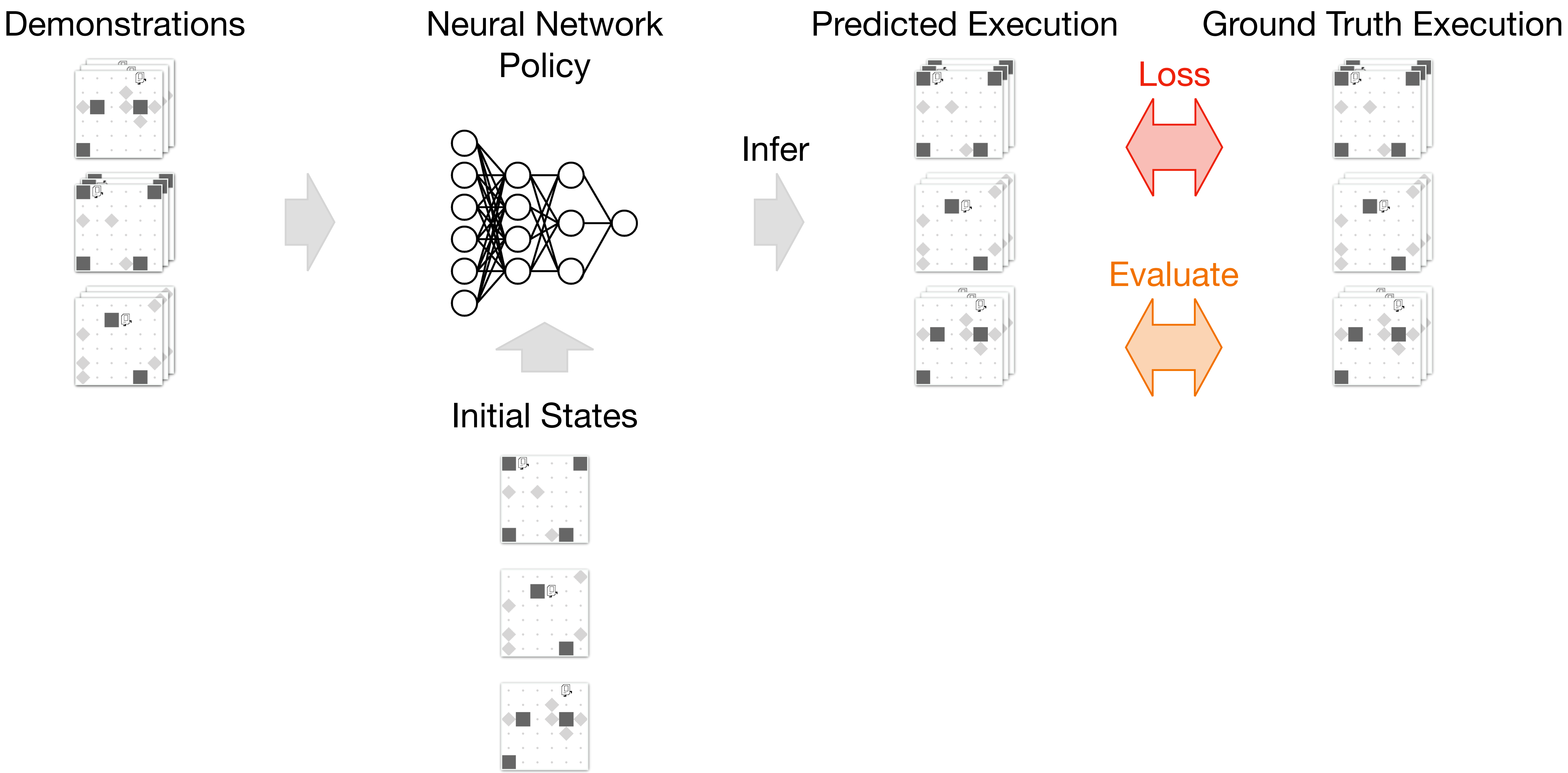
Program

```
DEF run()
   WHILE frontIsClear(HellKnight)
      attack
      moveForward
   IF thereIs(Demon)
      moveRight
   ELSE
      moveLeft
      moveBackward
```

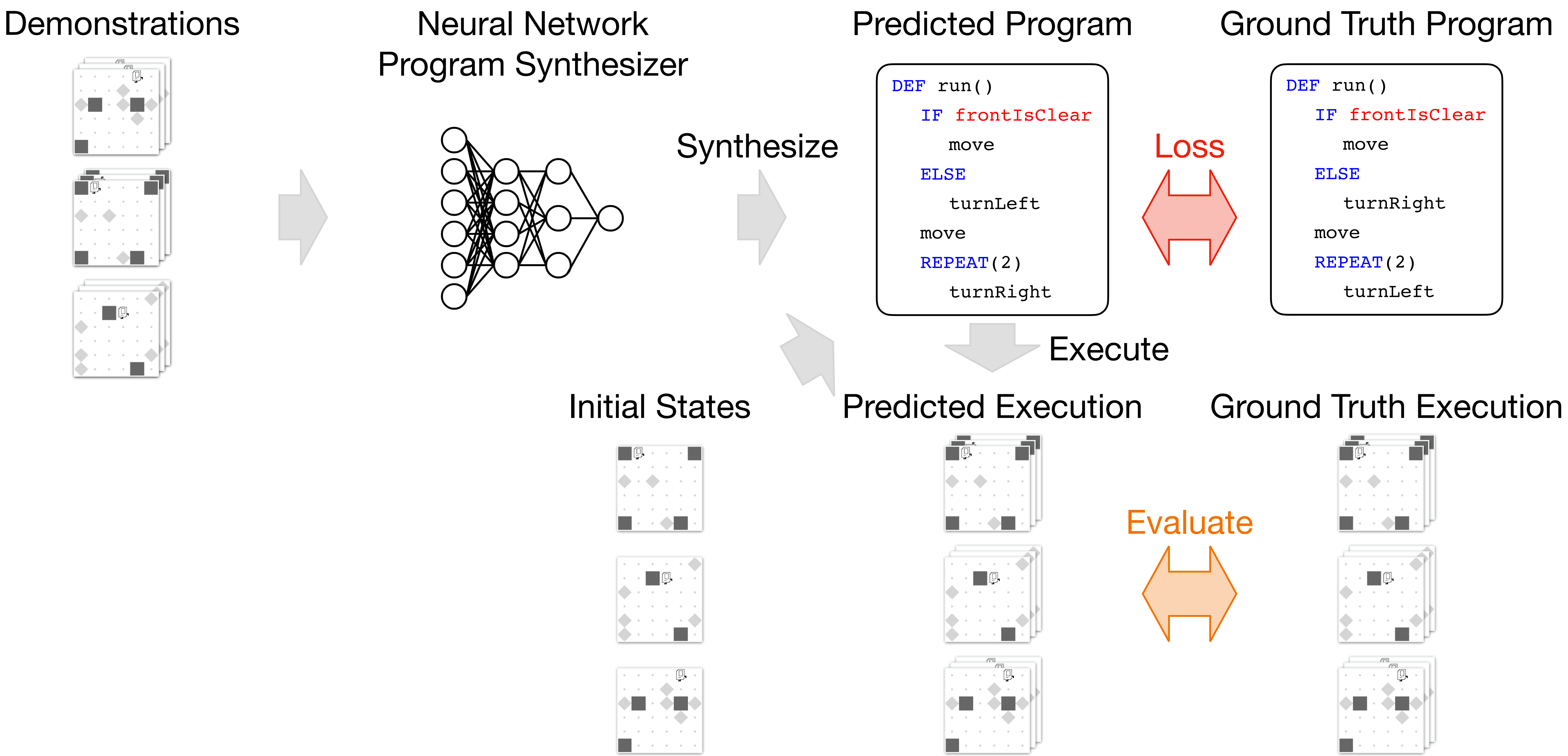
Demonstrations

Kempka et al., “Vizdoom: A doom-based ai research platform for visual reinforcement learning.” in CIG, 2016

Imitation Learning with Neural Network Policy

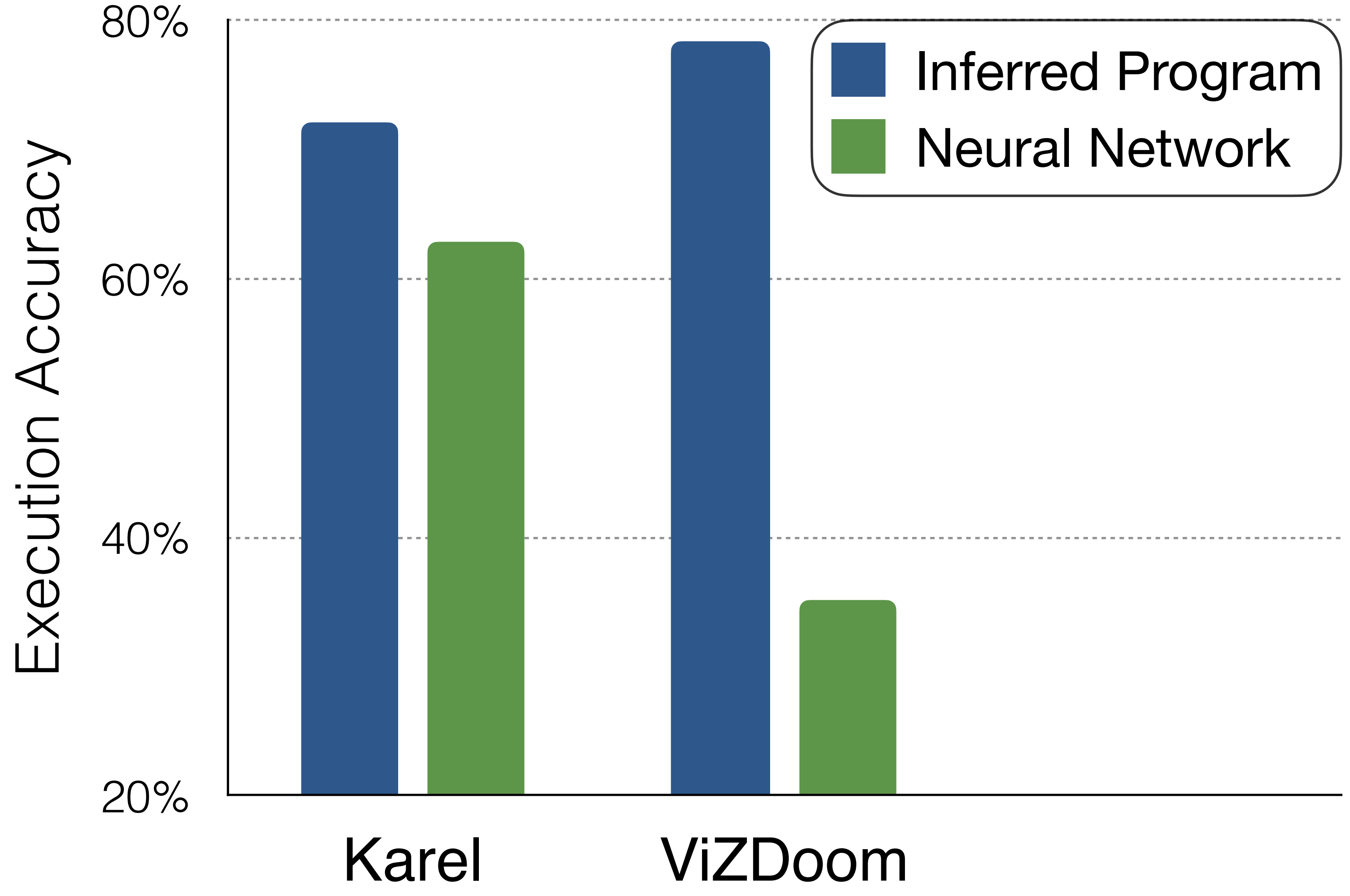


Imitation Learning with Program Policy



Experimental Results

Quantitative Results



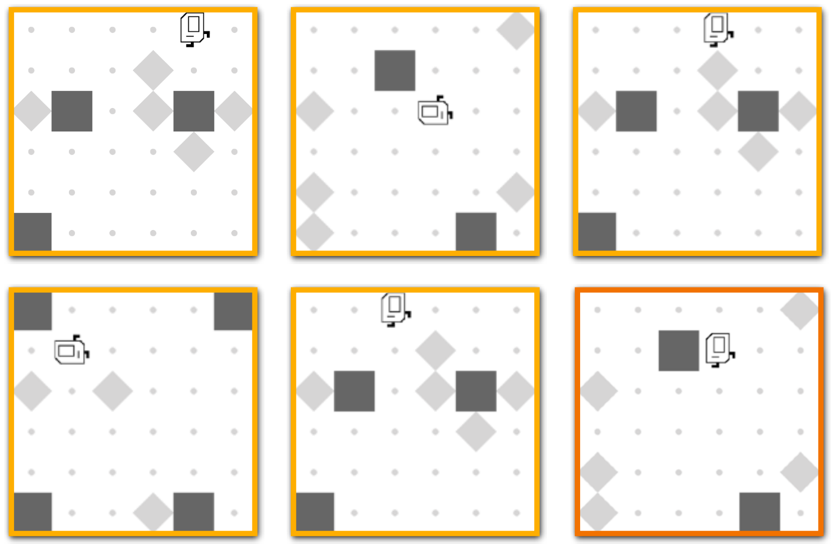
Evaluation: Execute the inferred program and the learned neural network policy on a set of unseen initial states and compare them to the ground truth demonstrations

Observation

Program

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

Demonstrations



Neural Network Policy

```
move
```

- Learn to mimic **the general tendency** of the expert **behaviors**

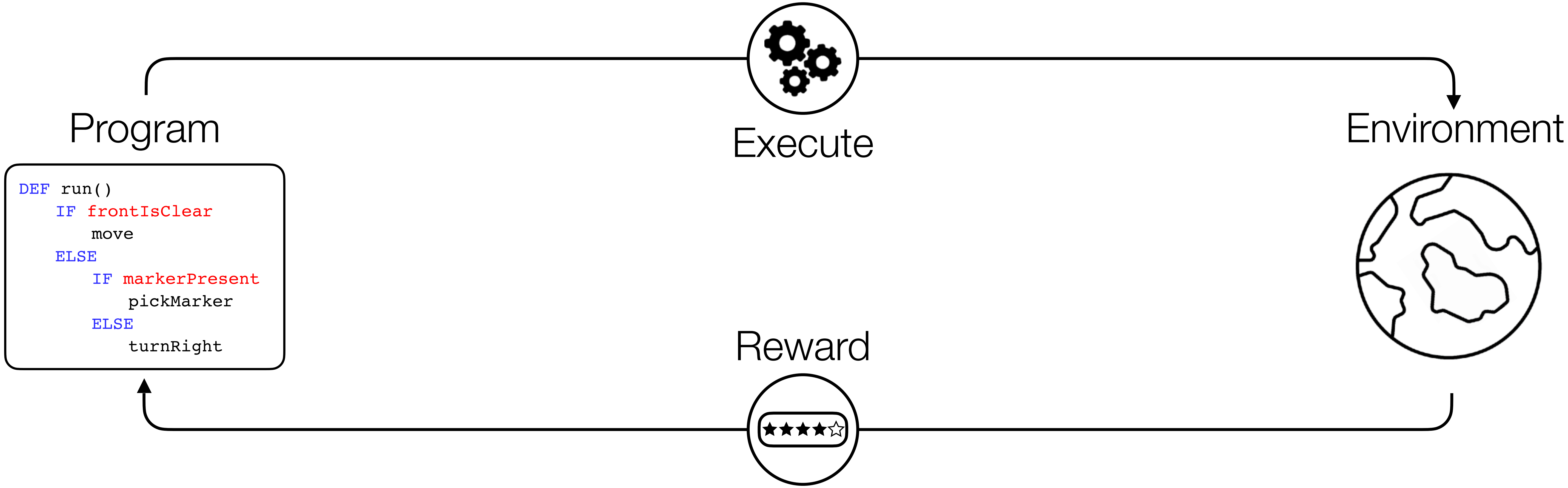
Inferred Program

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

- Learn to capture **the decision-making logics** of the expert

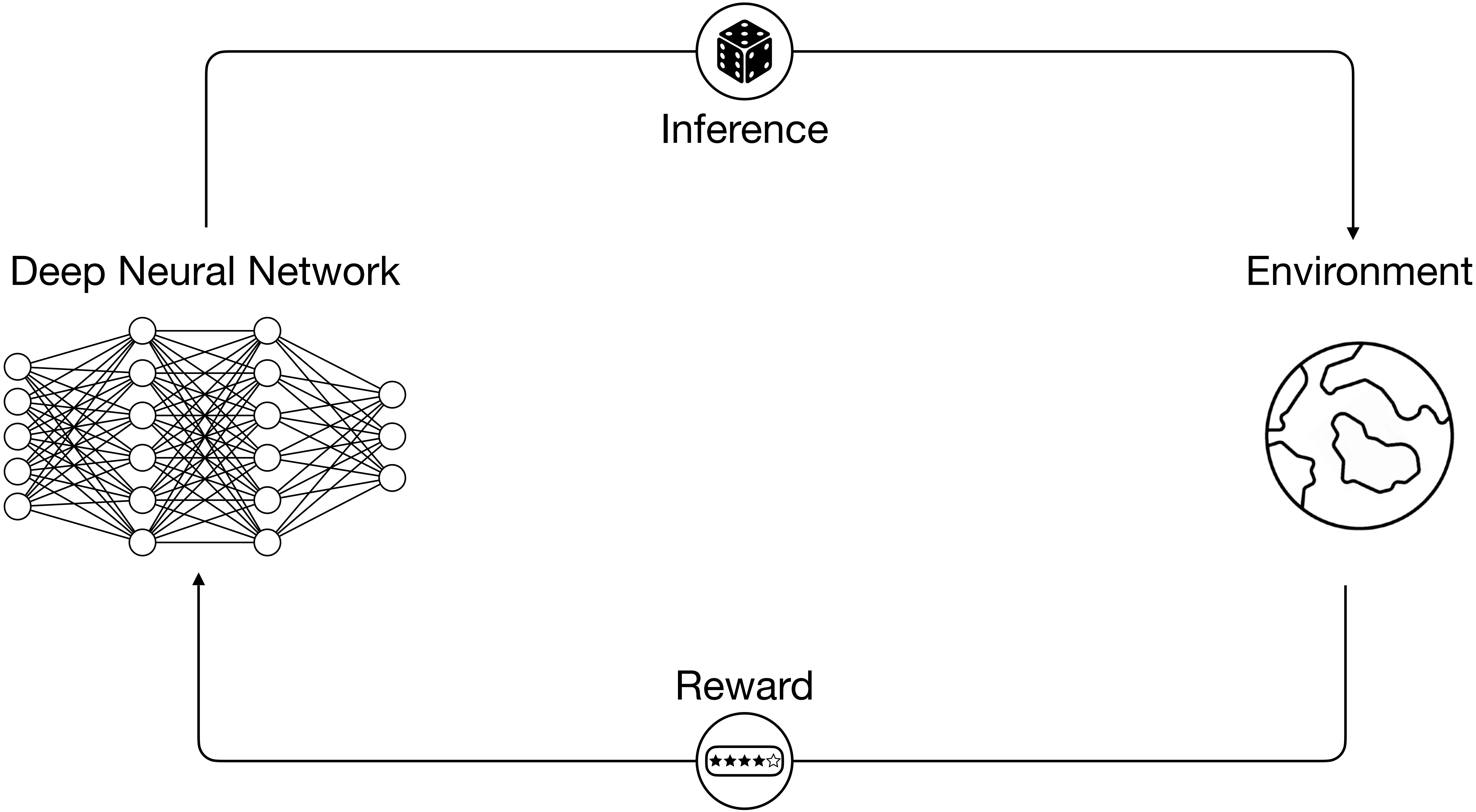
Learning to Synthesize Programs as Interpretable and Generalizable Policies

NeurIPS 2021

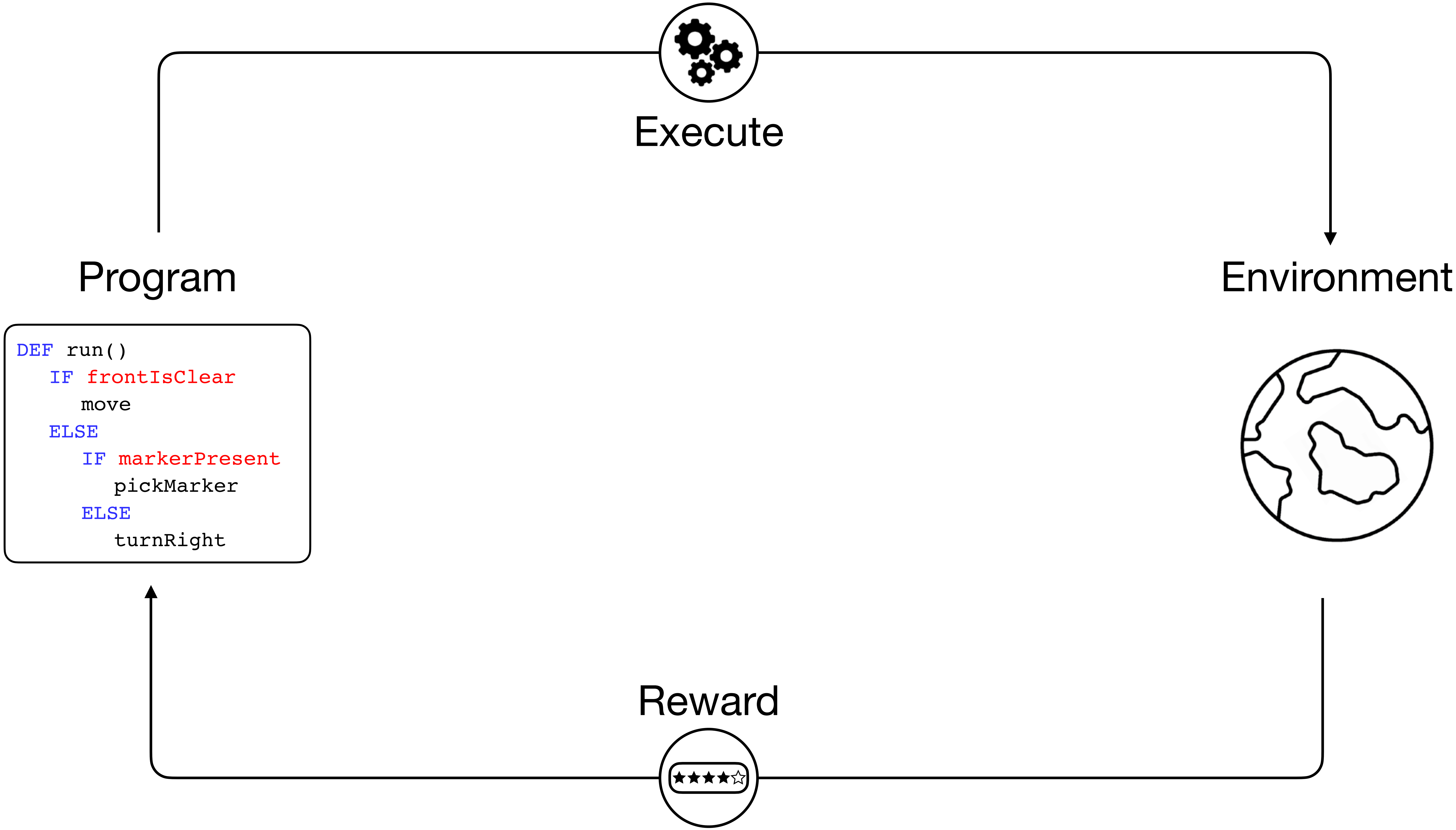


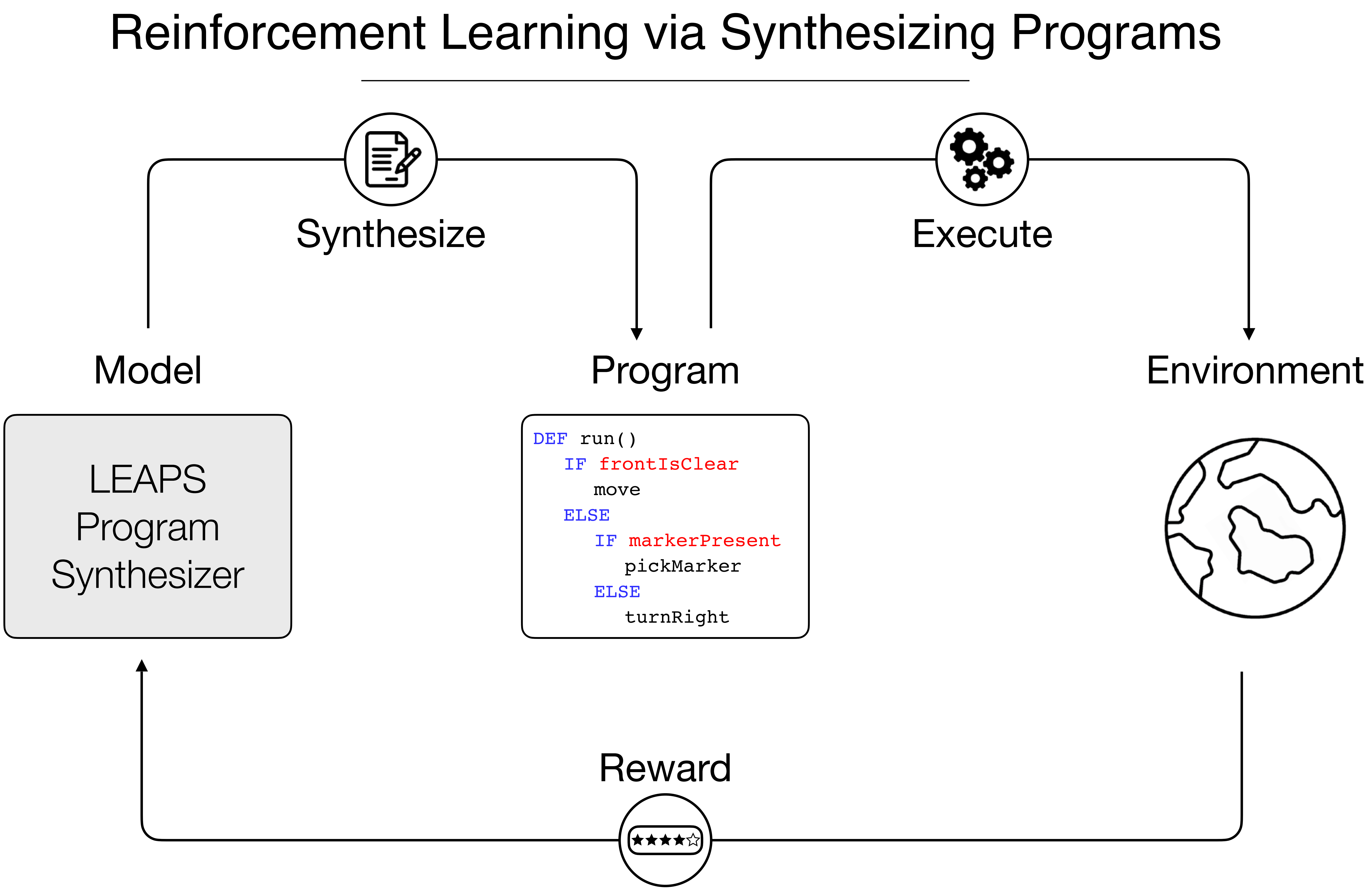
Dweep Trivedi* Jesse Zhang* Shao-Hua Sun Joseph J. Lim

Deep Reinforcement Learning



Reinforcement Learning via Synthesizing Programs

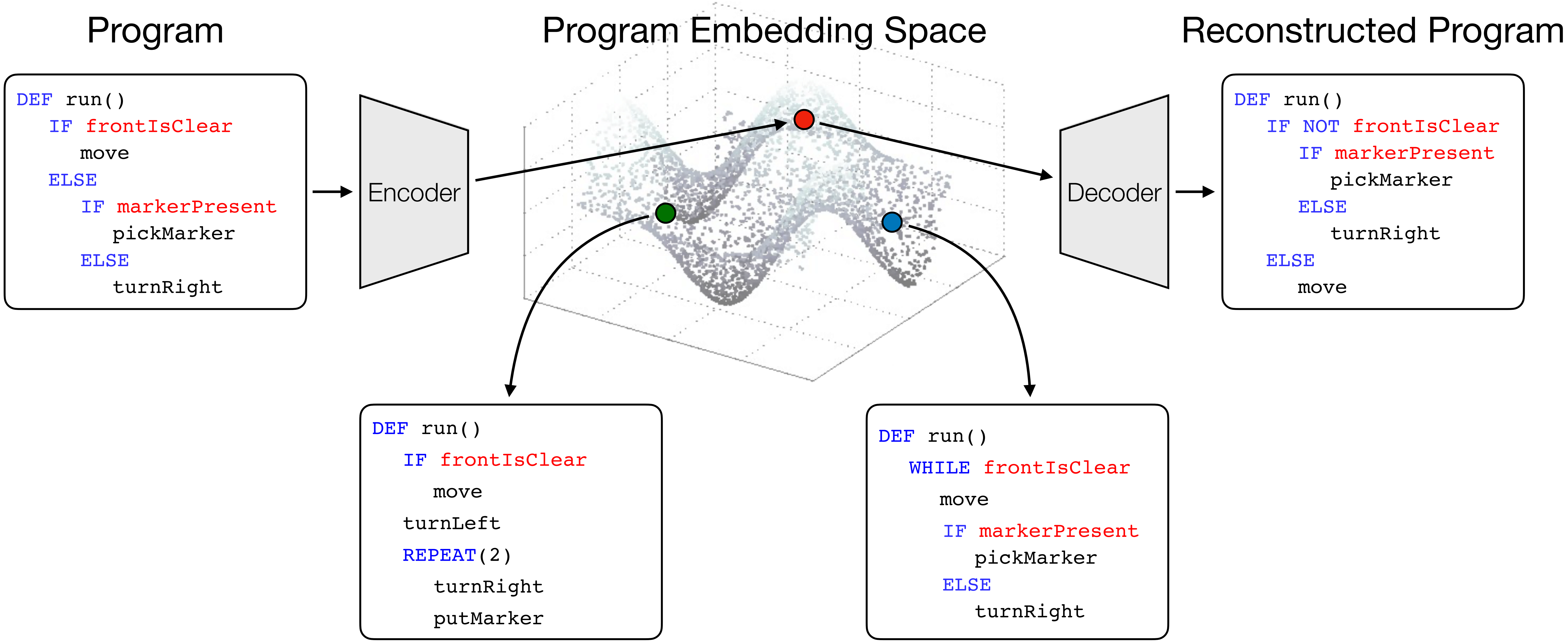



Reinforcement Learning via Synthesizing Programs
Synthesize
Execute
Model
Program
Environment
LEAPS
Program
Synthesizer
DEF run()
IF frontIsClear
move
ELSE
IF markerPresent
pickMarker
ELSE
turnRight
Reward

LEAPS: Learning Embeddings for lAtent Program Synthesis

Stage 1 Learn a **program embedding space** from randomly generated programs

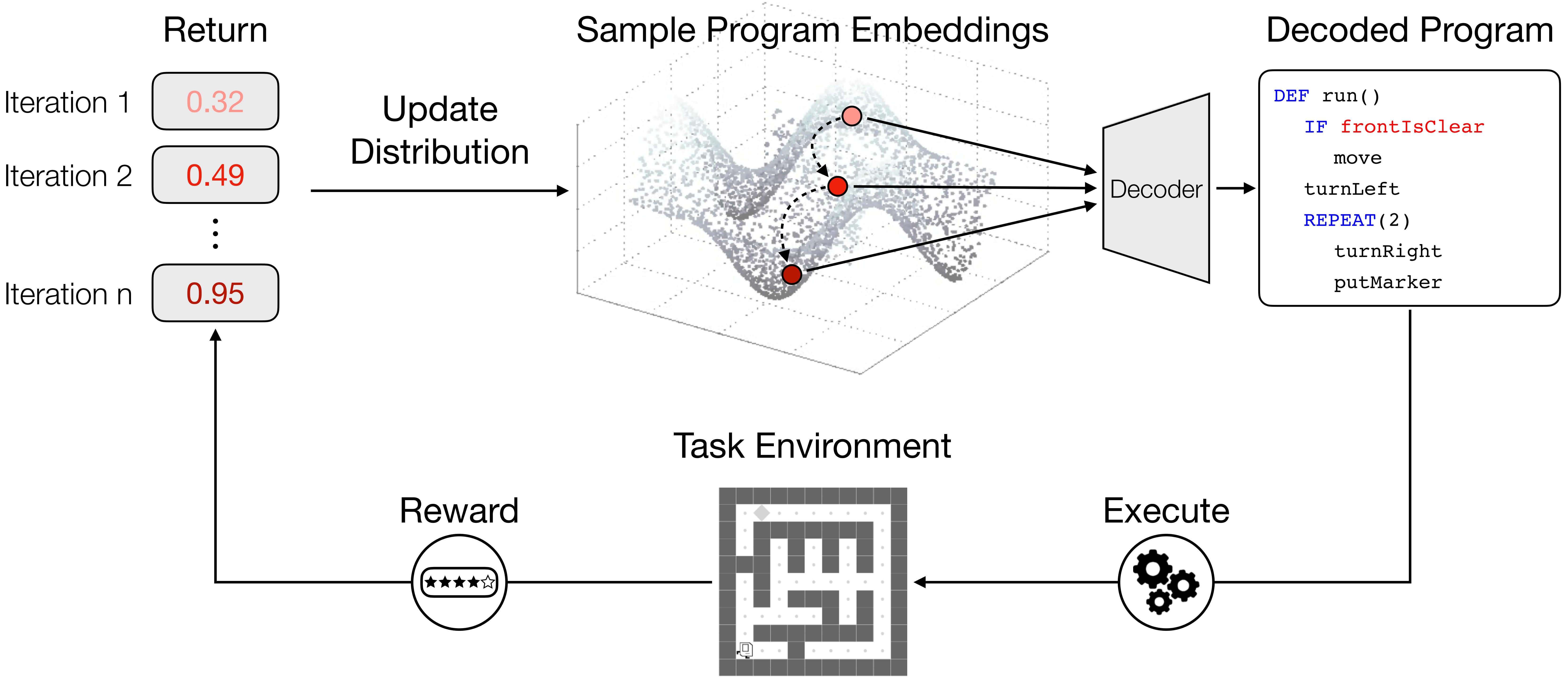
Goal Learn the grammar and the environment dynamics



LEAPS: Learning Embeddings for lAtent Program Synthesis

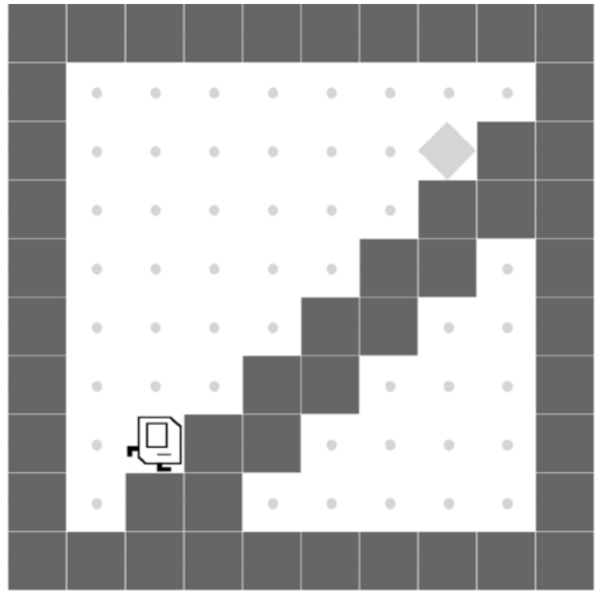
Stage 2 Search for a **task-solving program** using the **cross-entropy method (CEM)**

Goal Optimize the task performance

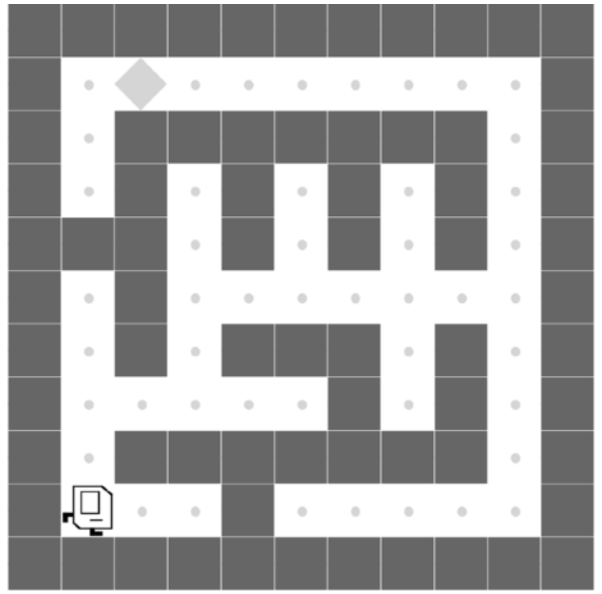


Karel Tasks

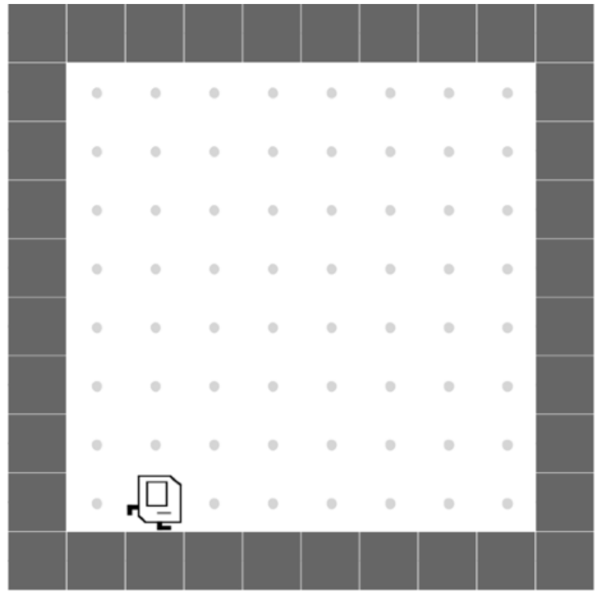
StairClimber



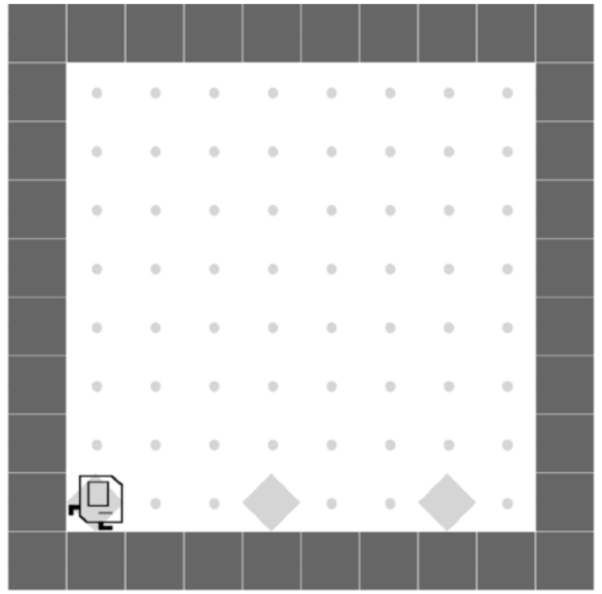
Maze



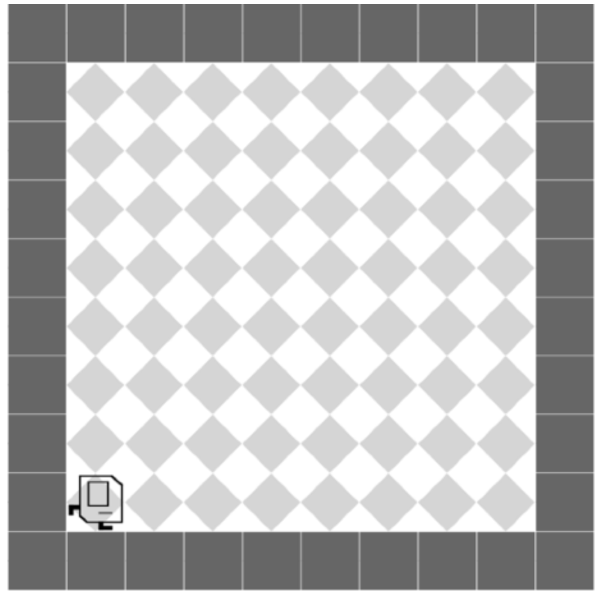
FourCorners



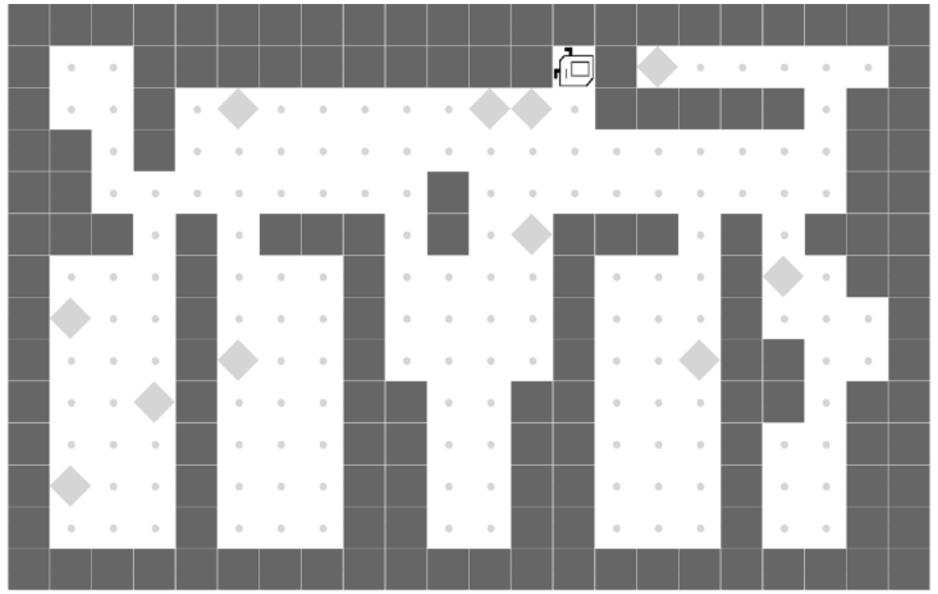
TopOff



Harvester

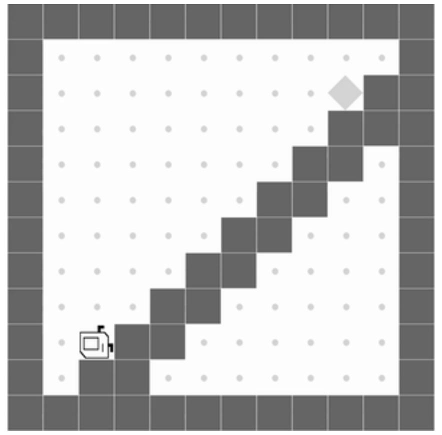


CleanHouse

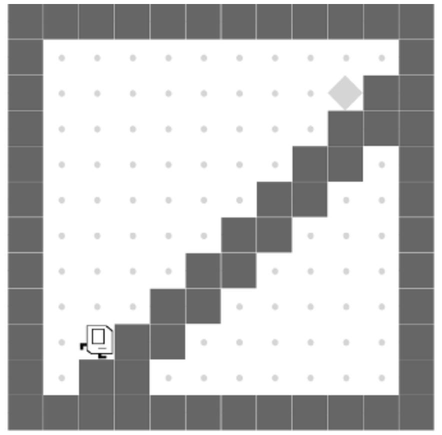


Qualitative Results

StairClimber

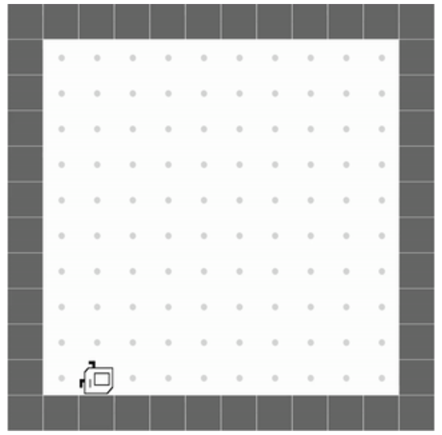


Deep RL

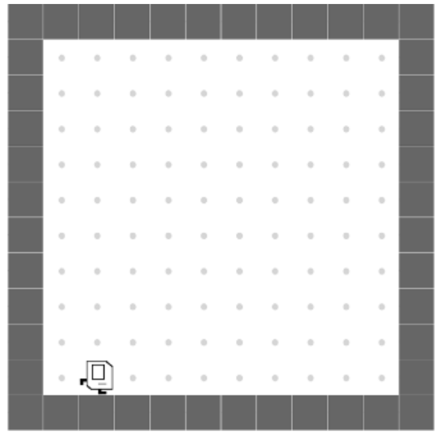


LEAPS

FourCorners

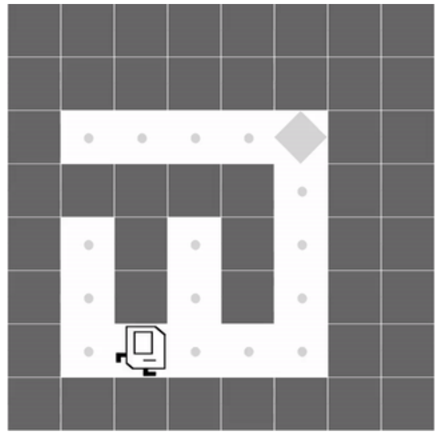


Deep RL

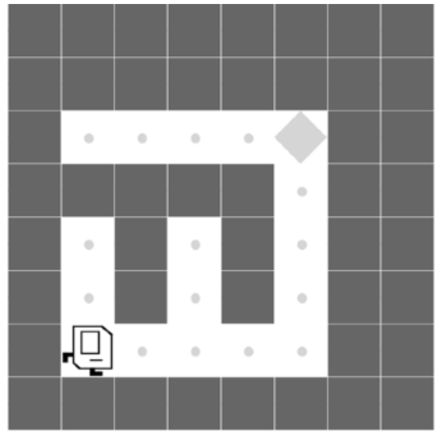


LEAPS

Maze

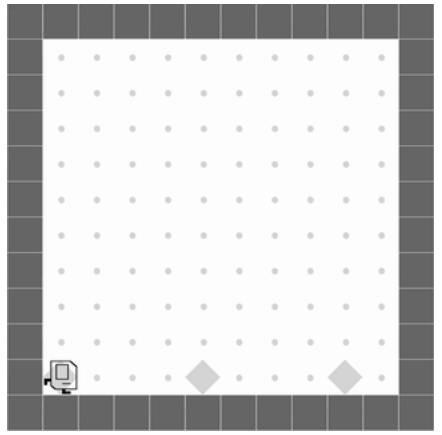


Deep RL

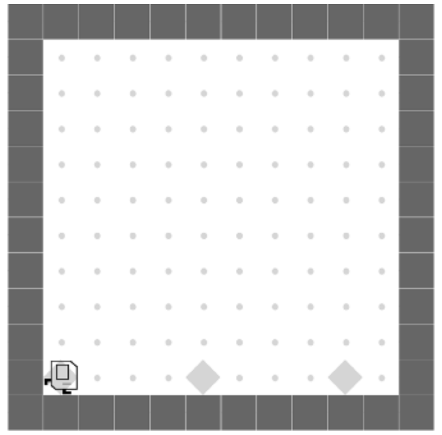


LEAPS

TopOff

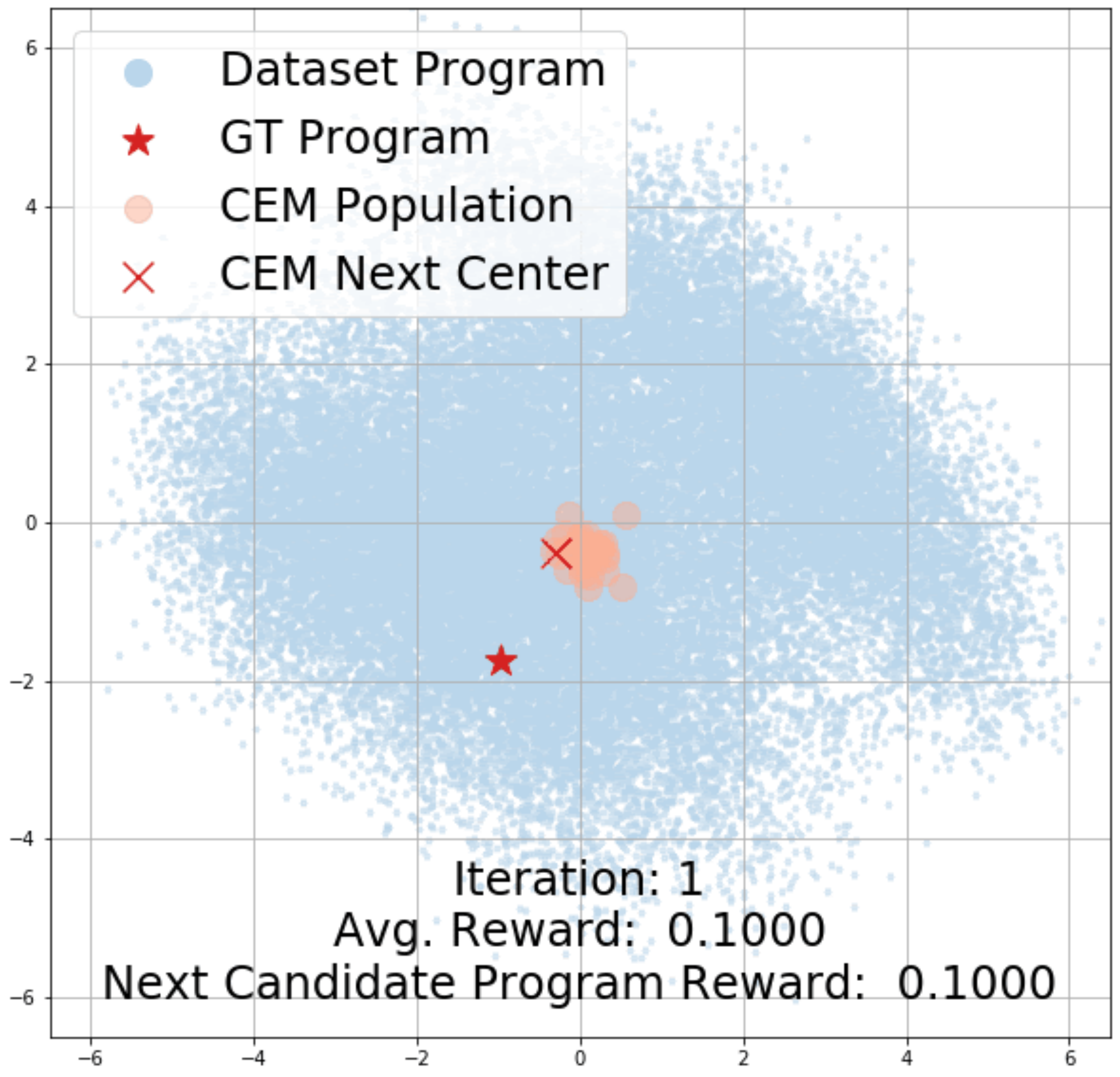


Deep RL



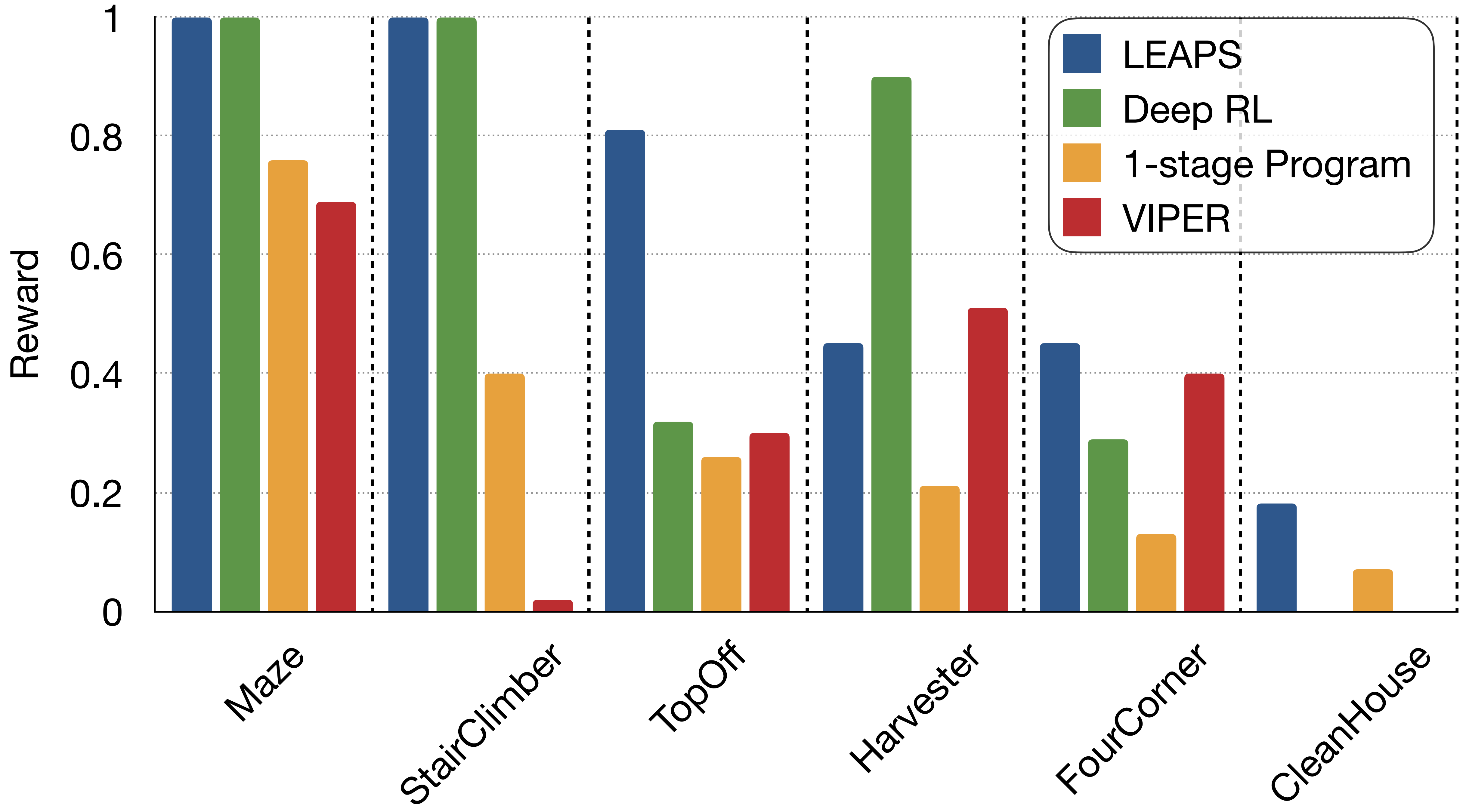
LEAPS

CEM trajectory Visualization



Goal: Search for a StairClimber **program** in the **learned program embedding space**

Quantitative Results



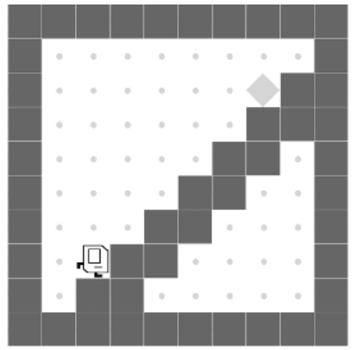
LEAPS Zero-shot Generalization

Learning on 8 x 8

LEAPS Program Policy

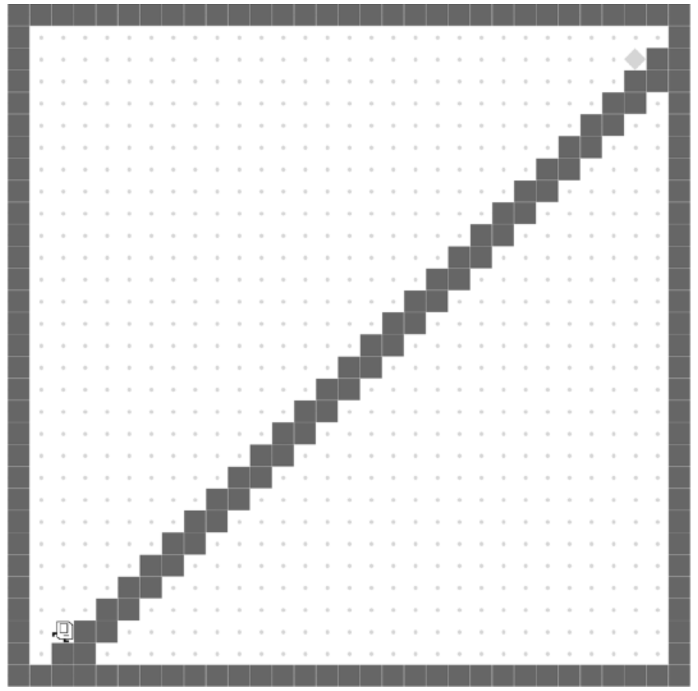
Evaluation on 100 x 100

StairClimber

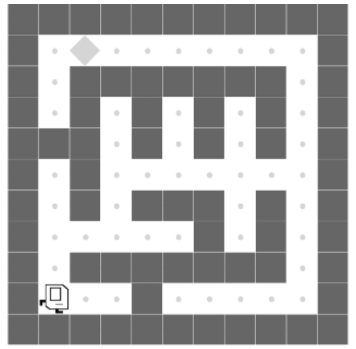


LEAPS Program Synthesizer

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

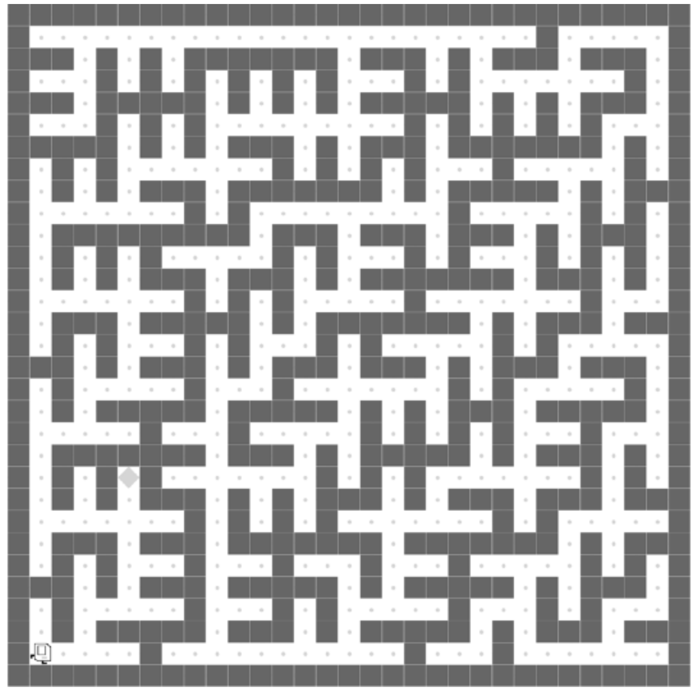


Maze

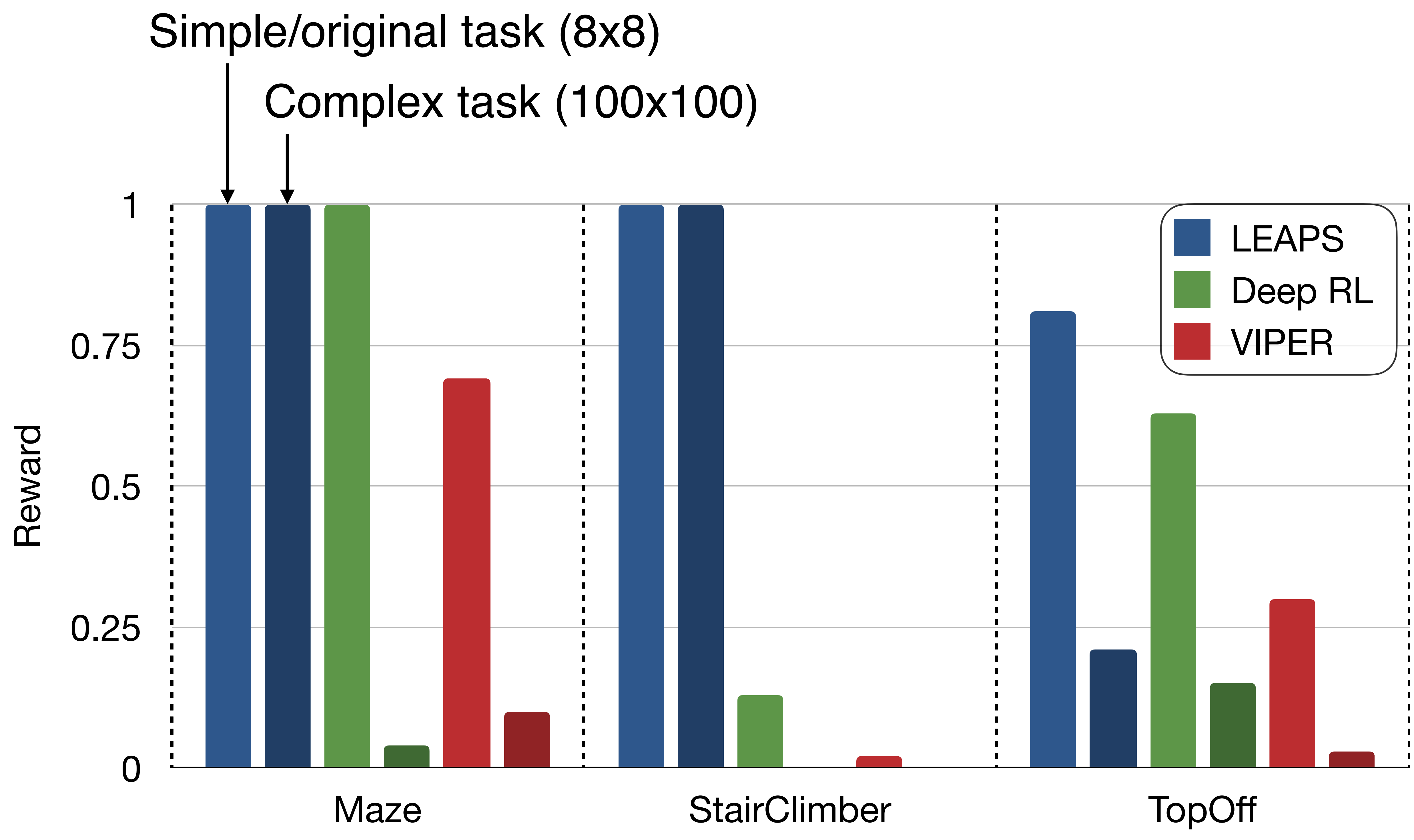


LEAPS Program Synthesizer

```
DEF run()
   if frontIsClear()
     turnLeft
   while noMarkersPresent()
      turnRight
      move
```

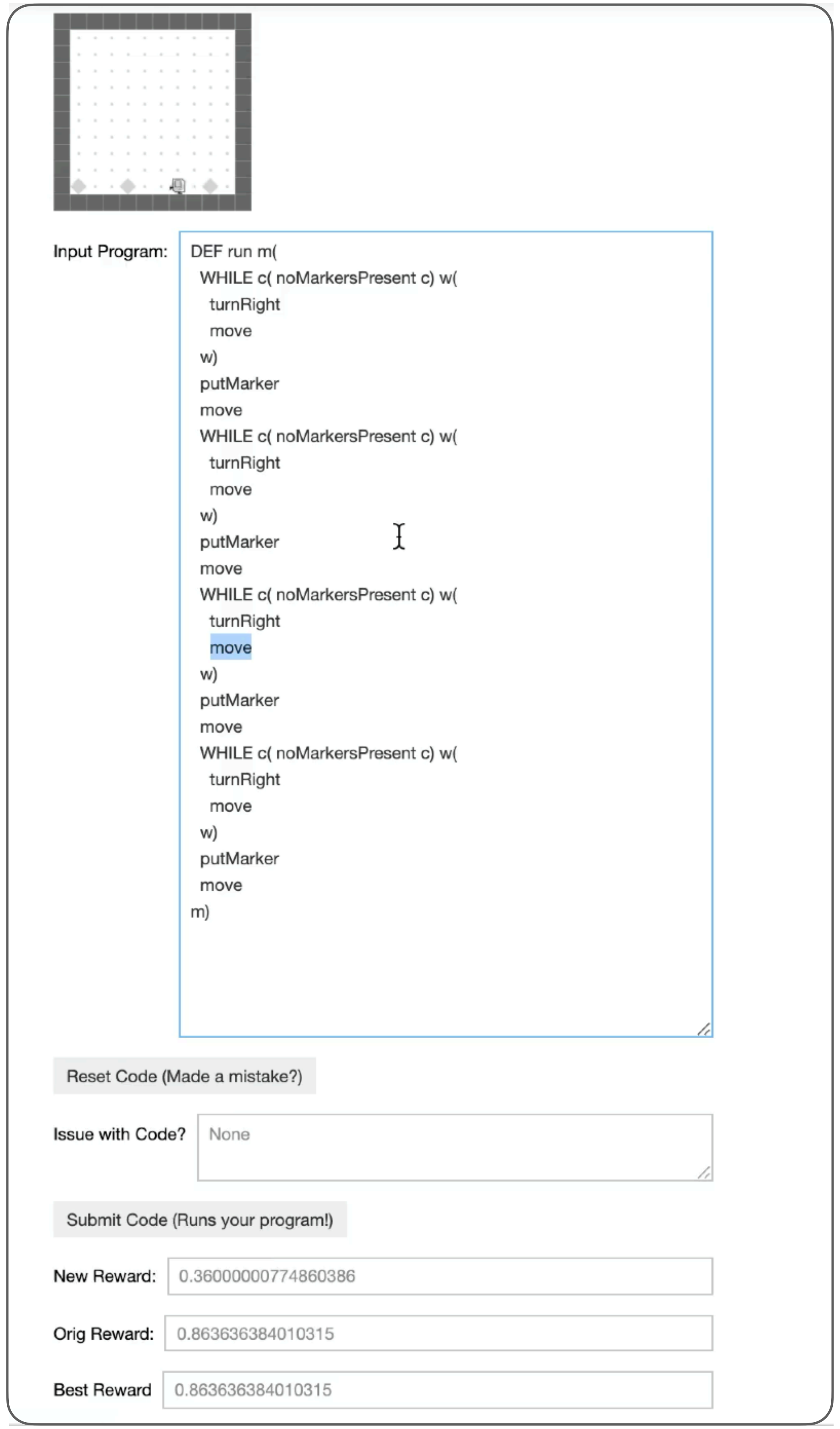


Experimental Results - Zero-shot Generalization

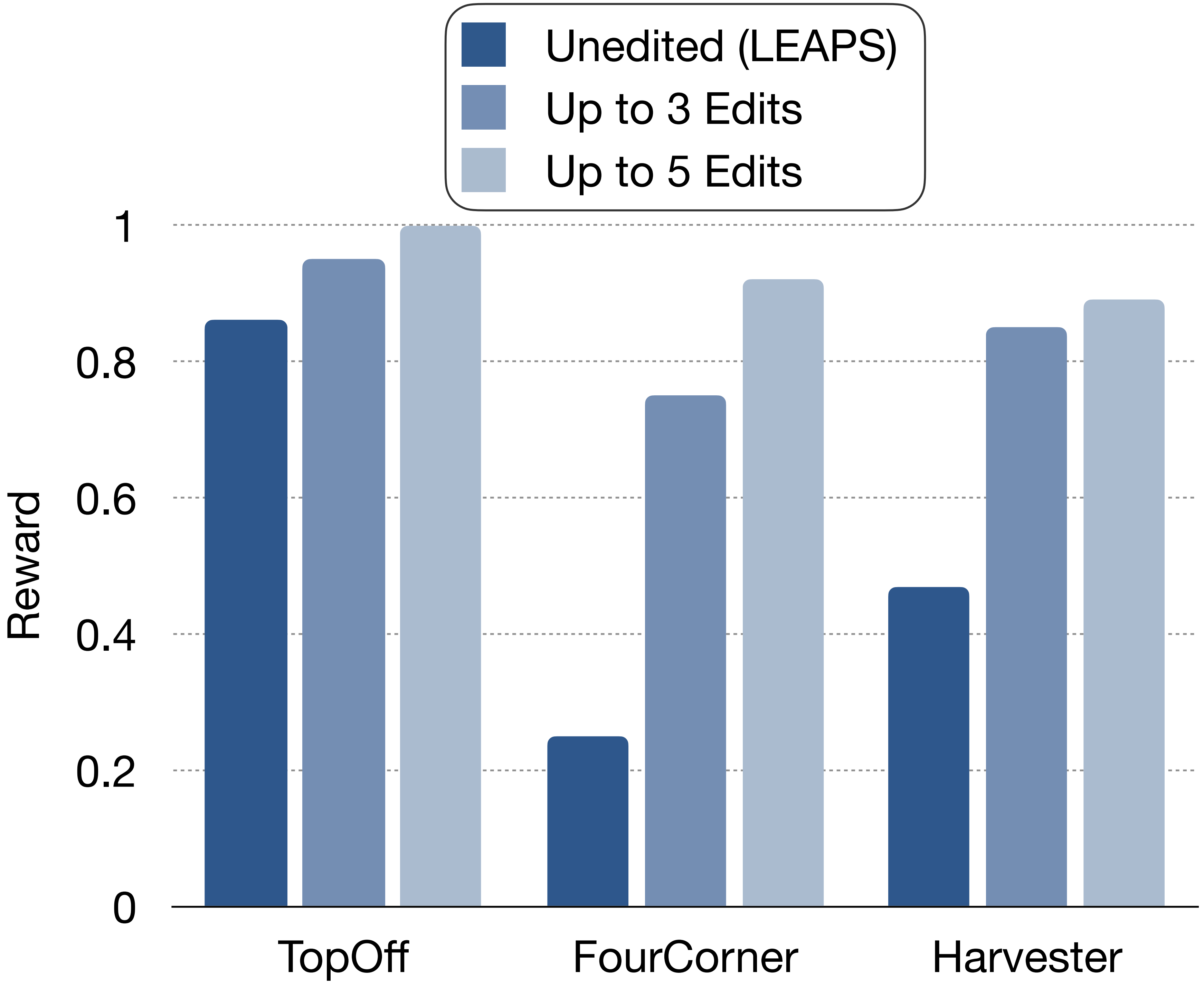


Interpretability & Interactability

Interactive Debugging Interface

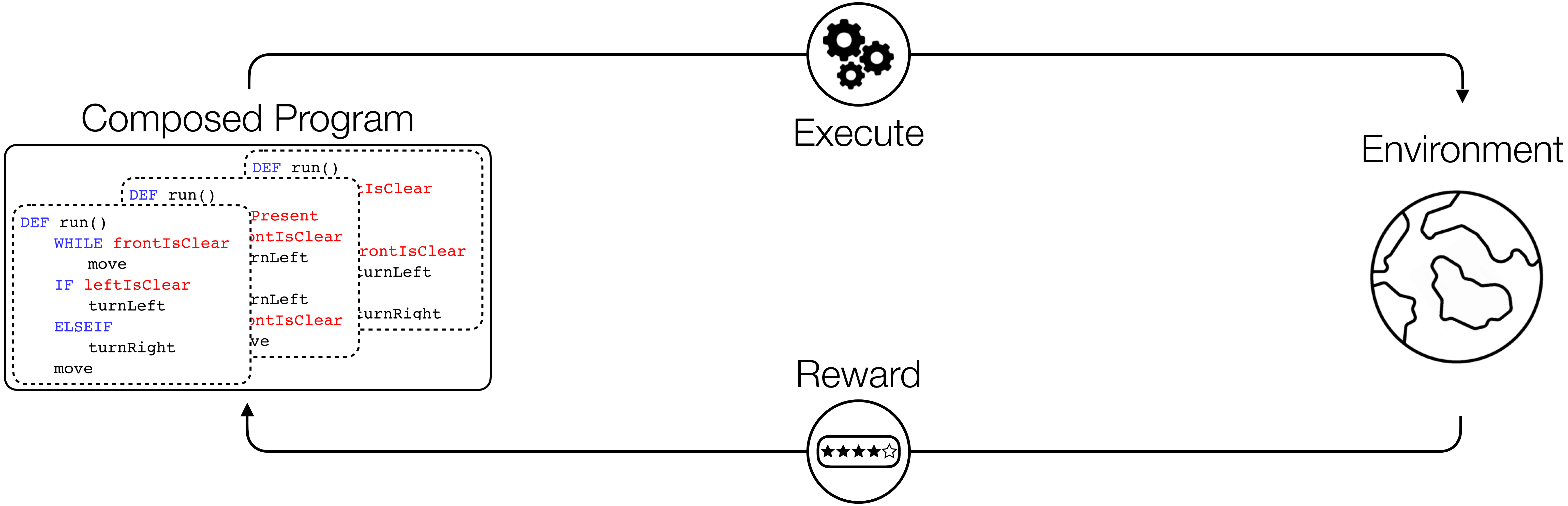


Performance Improvement



Hierarchical Programmatic Reinforcement Learning via Learning to Compose Programs

ICML 2023

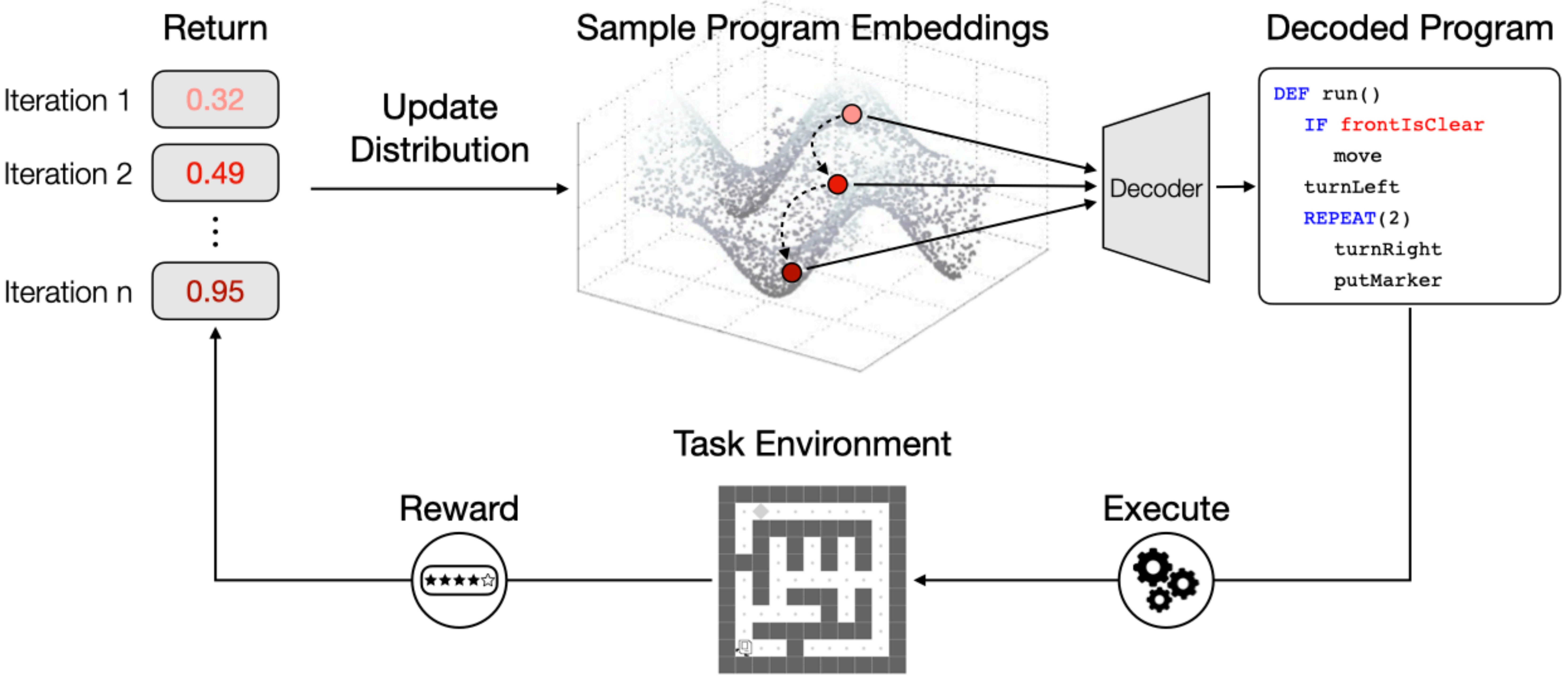


Guan-Ting Liu* En-Pei Hu* Pu-Jen Cheng Hung-Yi Lee Shao-Hua Sun

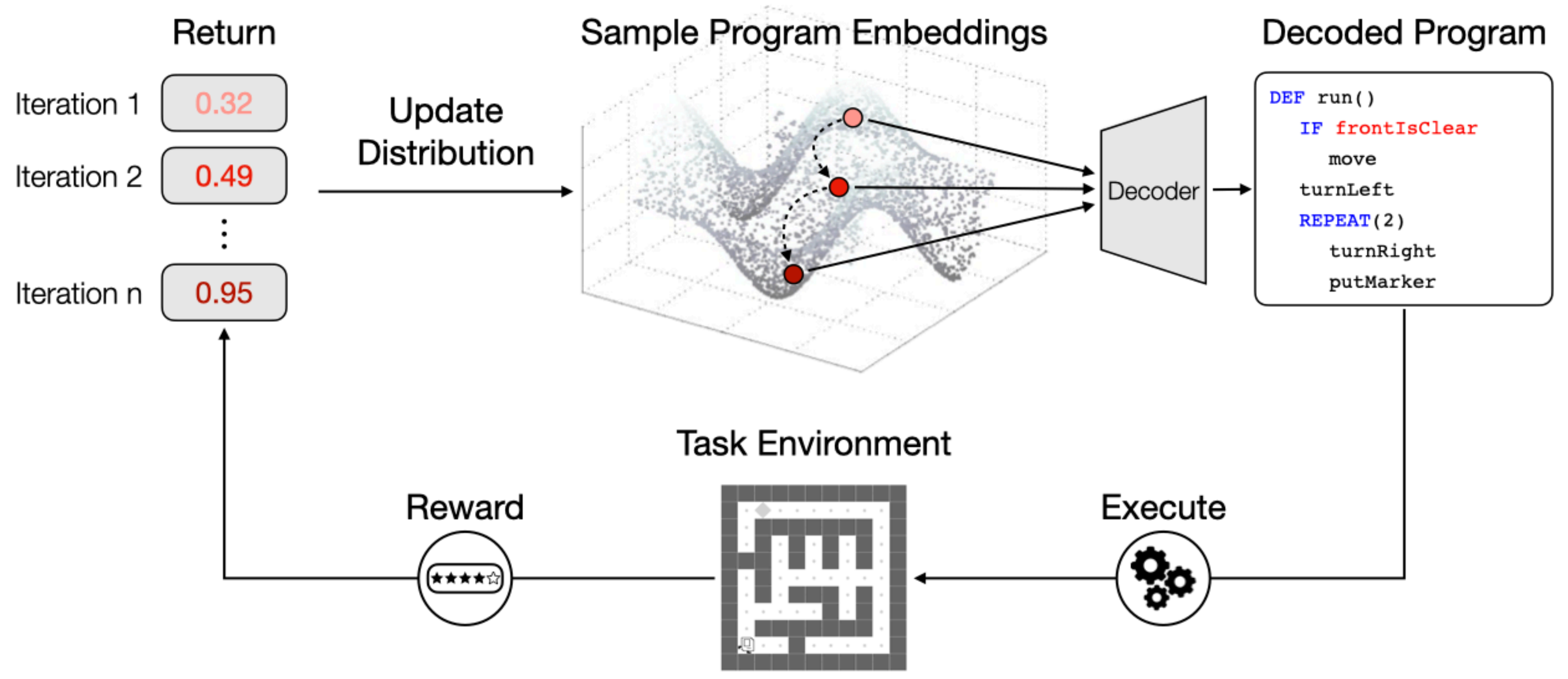
LEAPS: Learning Embeddings for lAtent Program Synthesis

Stage 2 Searching for a **task-solving program** using the **cross-entropy method**



LEAPS: Learning Embeddings for lAtent Program Synthesis

Stage 2 Searching for a **task-solving program** using the **cross-entropy method**



Limited program distribution

Search in the program embedding space spanned by **the dataset programs**

Cannot **synthesize longer or more complex** programs

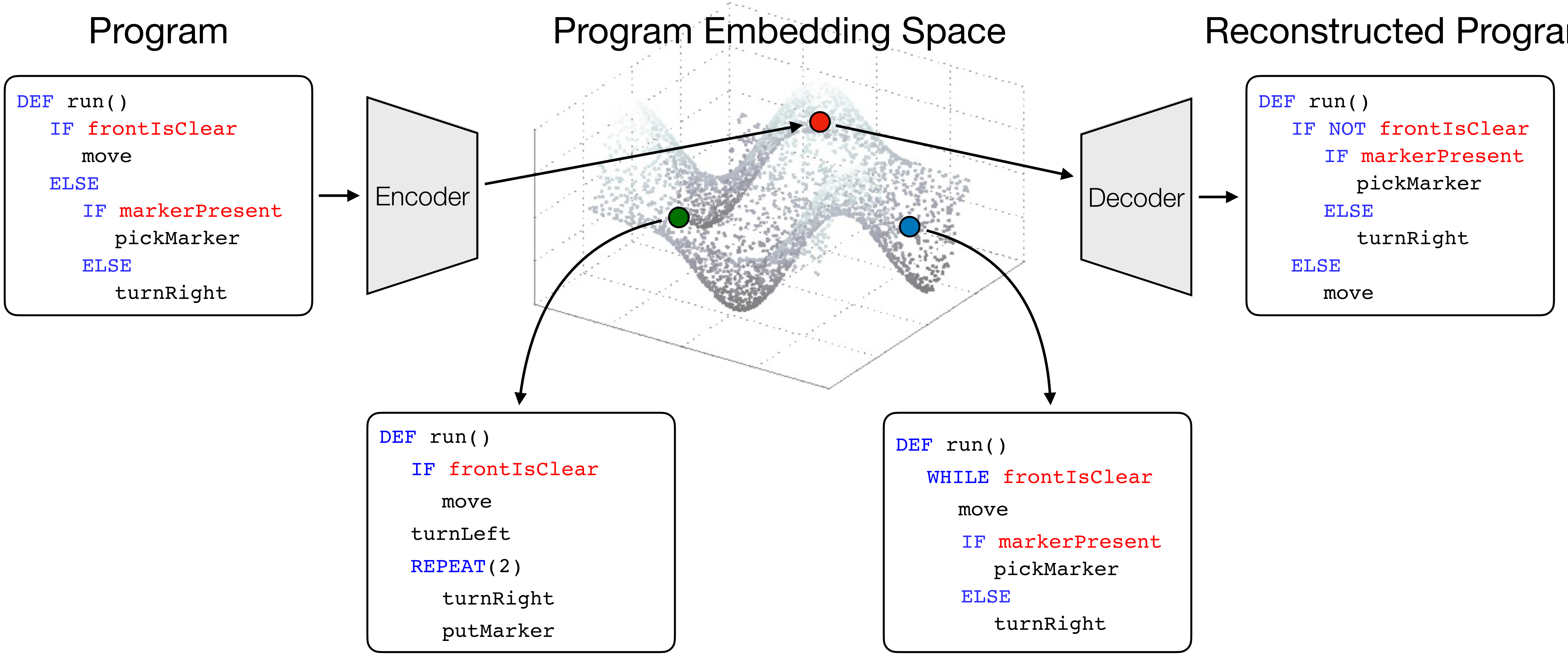
Poor credit assignment

Evaluate each candidate program solely based on the **cumulative return** of its execution trace

Cannot accurately **attribute rewards** to corresponding program parts

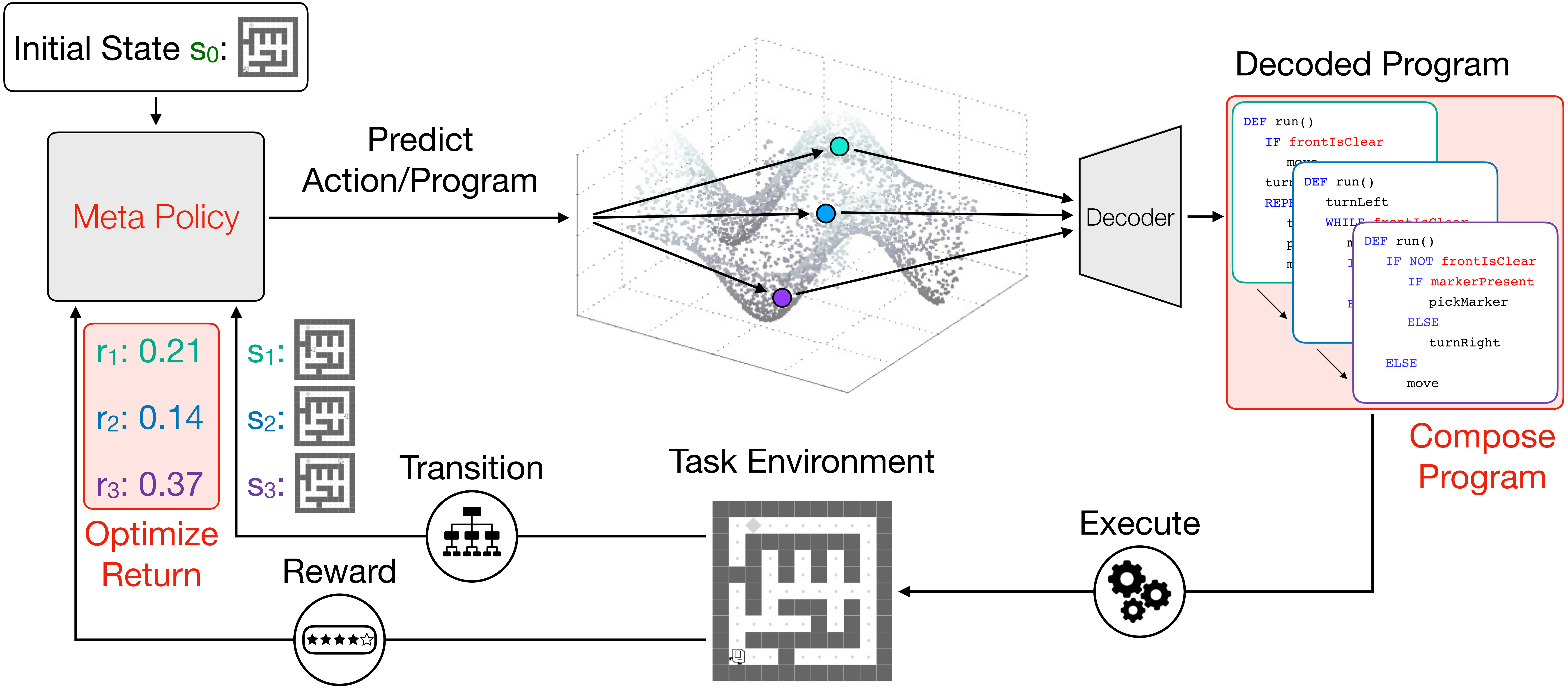
HPRL: Hierarchical Programmatic Reinforcement Learning

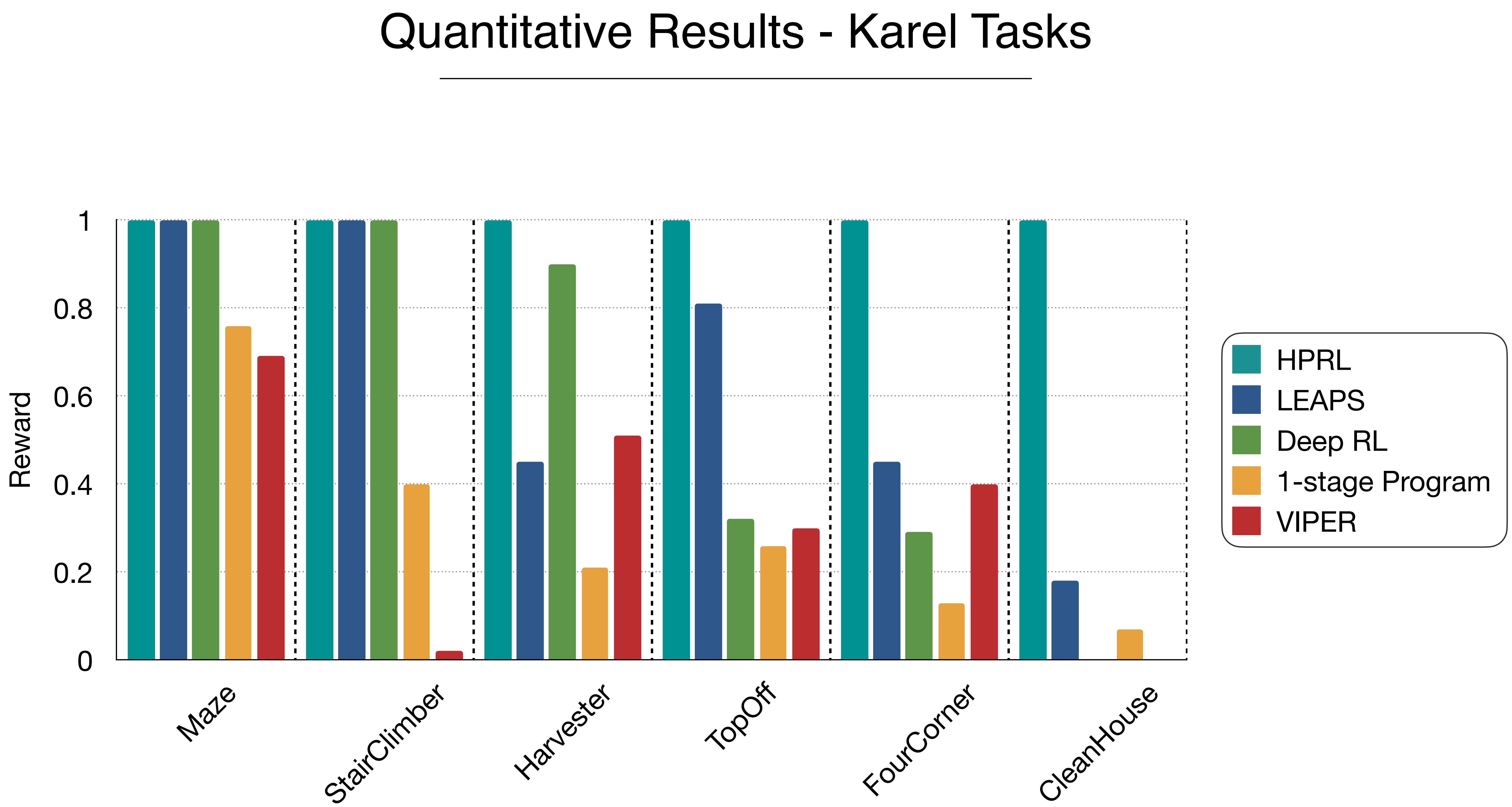
Stage 1 Learning a compressed program embedding space from randomly generated programs



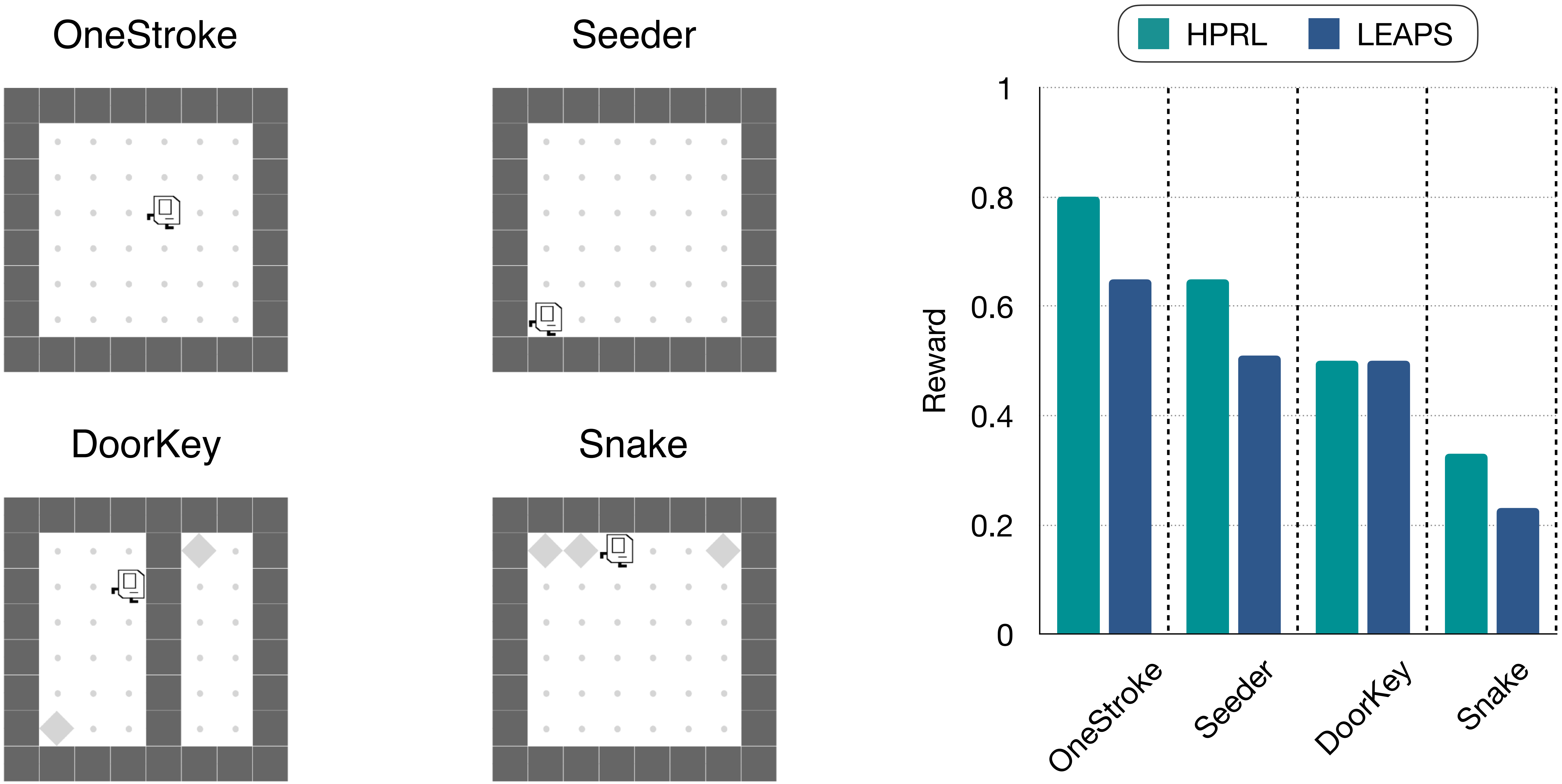
HPRL: Hierarchical Programmatic Reinforcement Learning

Stage 2 Learning a **meta policy** to produce **a series of programs** (*i.e.*, predict a series of **actions**) to yield a **composed task-solving program**




Quantitative Results - Karel Tasks
Reward
1
0.8
0.6
0.4
0.2
0
Maze
StairClimber
Harvester
TopOff
FourCorner
CleanHouse
HPRL
LEAPS
Deep RL
1-stage Program
VIPER

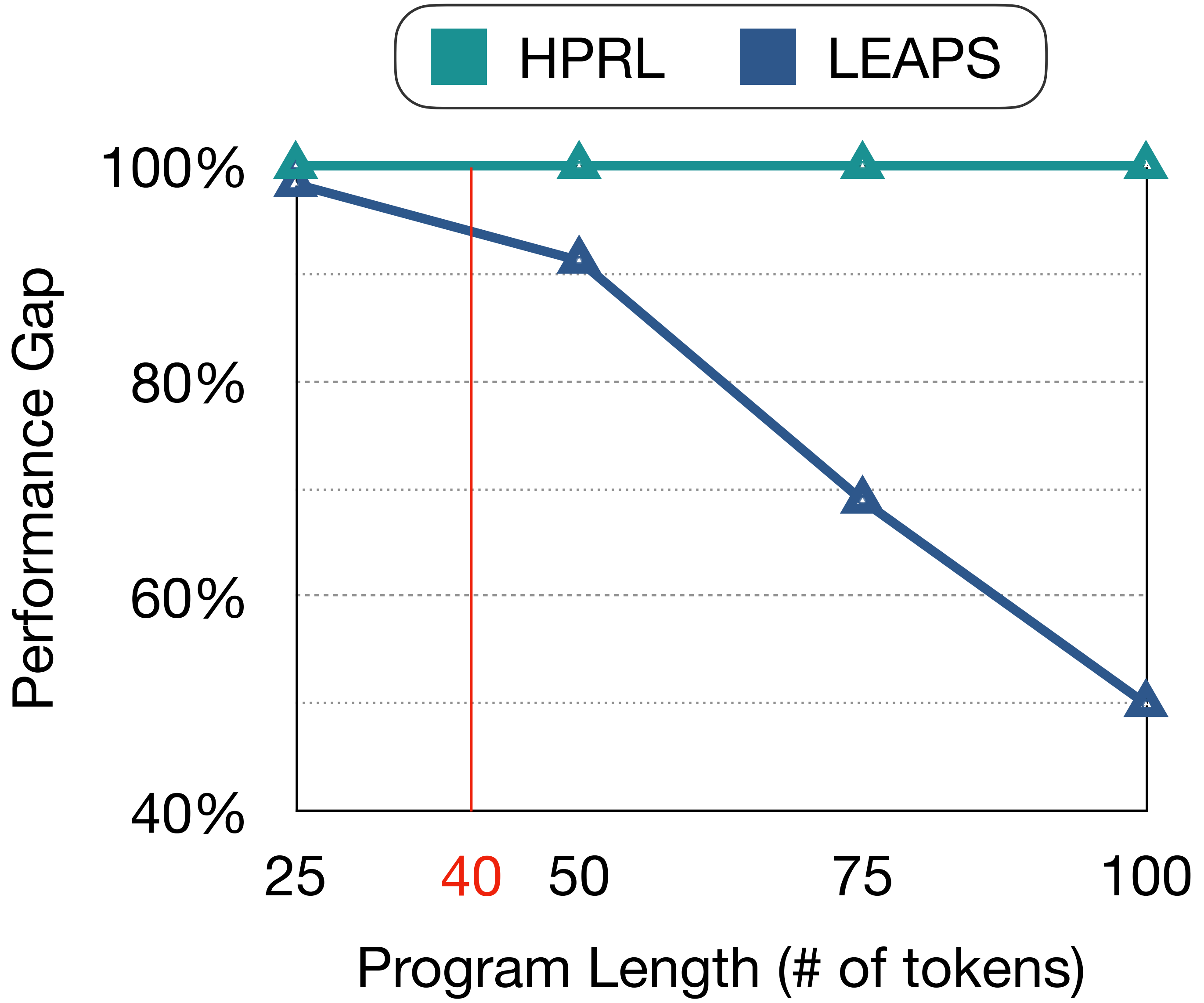
Karel-Hard Tasks



Additional Experiments

Limited program distribution

Synthesize out-of-distributionally long programs

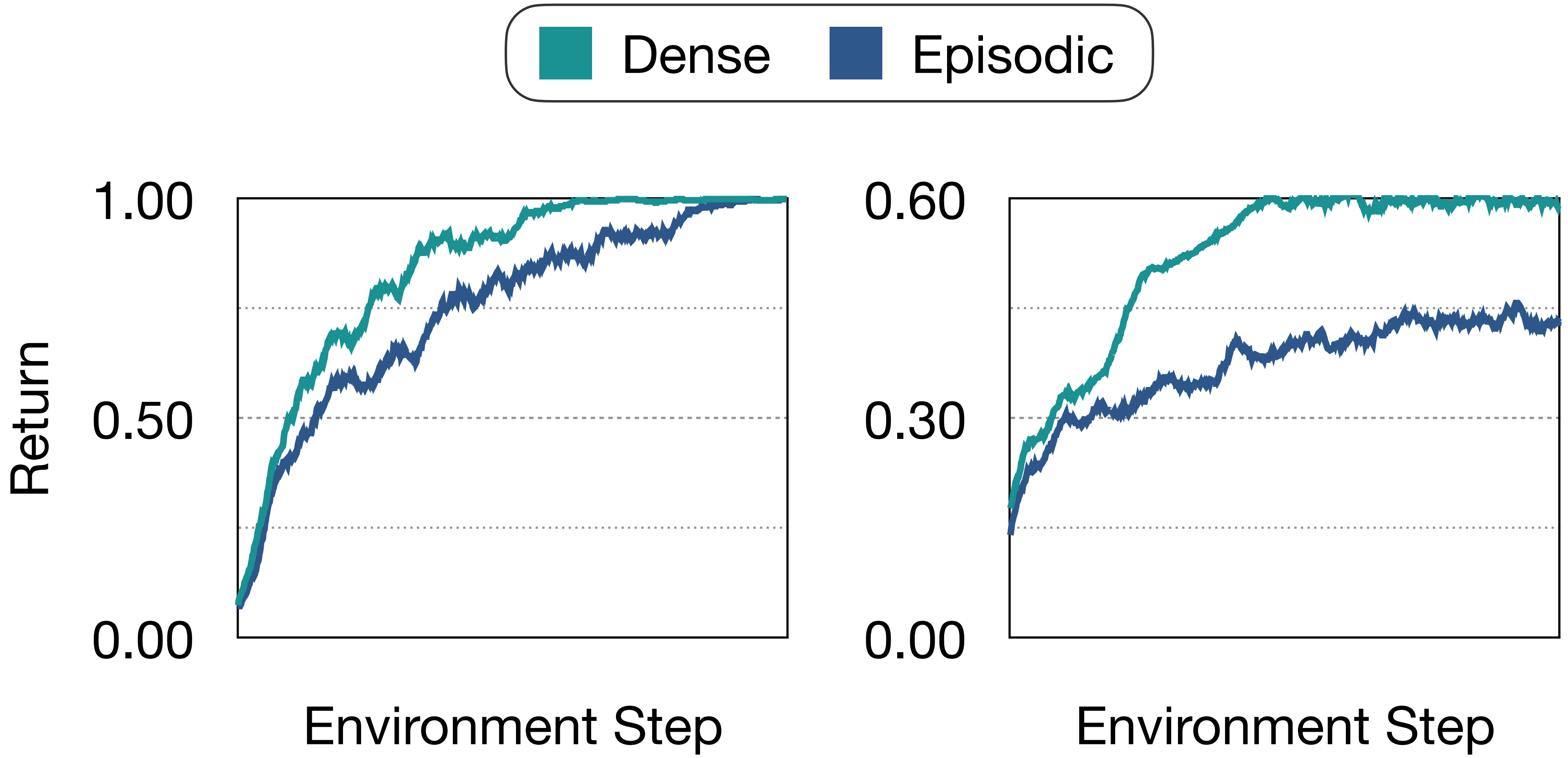


- HPRL can synthesize programs **longer** than the dataset programs (< 40 tokens) better than LEAPS

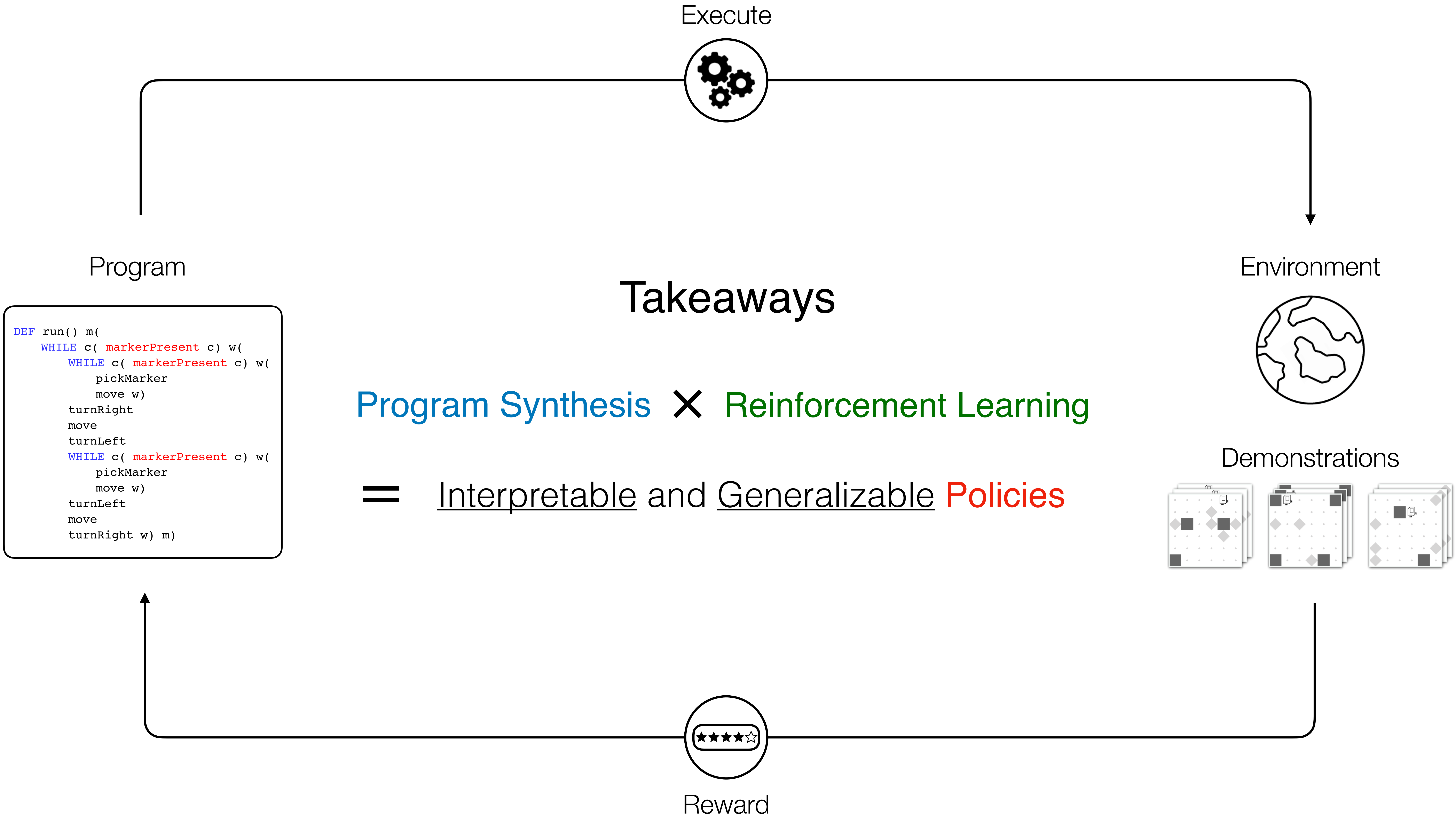
Poor credit assignment

Learning from episodic reward

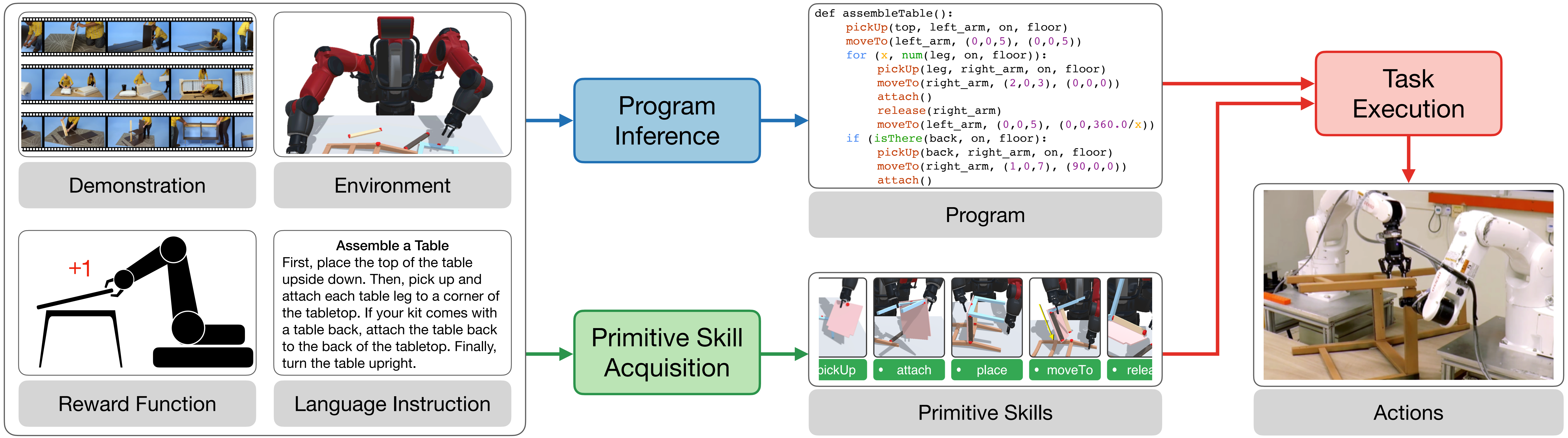
- **Dense:** Reward each subprogram based on its execution trace
- **Episodic:** Reward the entire composed program at the end



- **The hierarchical design** of HPRL allows for better **credit assignment** with dense rewards, facilitating the learning progress


Execute
Program
Takeaways
Environment
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)
Program Synthesis × Reinforcement Learning
= Interpretable and Generalizable Policies
Demonstrations
Reward

Program-Guided Robot Learning

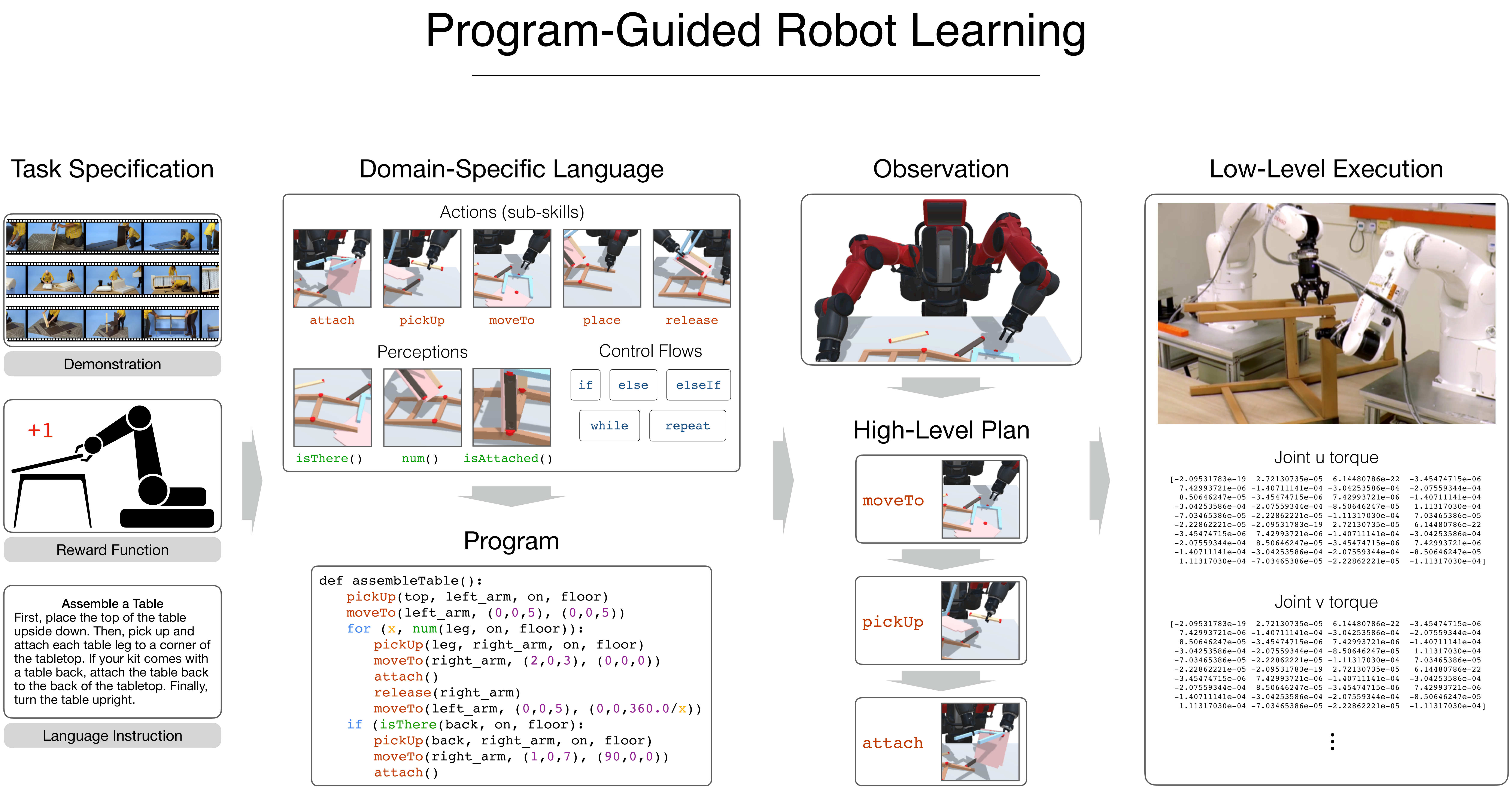


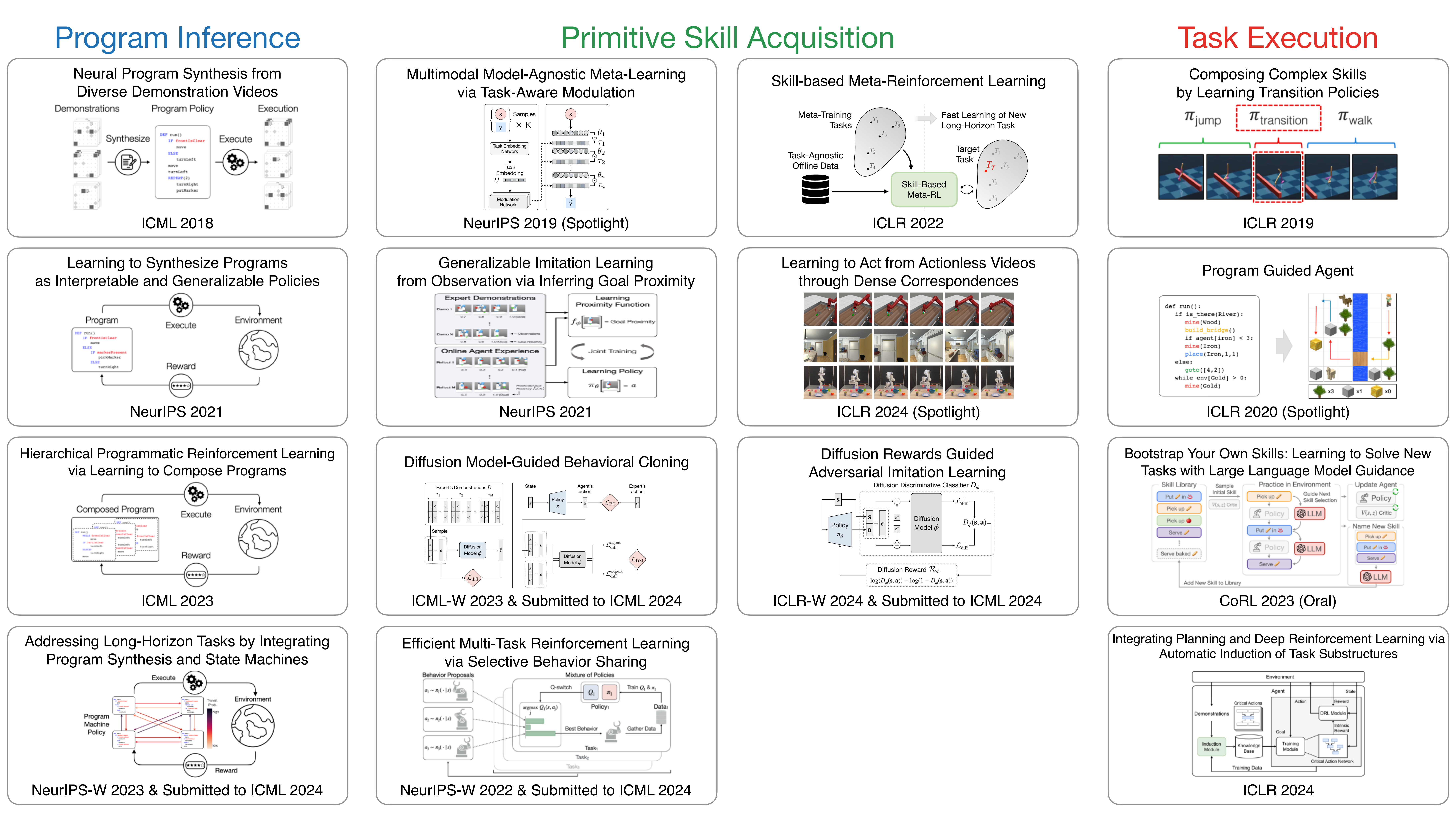
Key idea

- Represent **robot behaviors** using **programs** based on pre-defined and acquired primitive skills
- Decouple **learning a skill** as performing program inference and task execution

Program

- Described with formal languages
- **Human interpretable** and **machine executable**
- Structured for **generalization**


Program-Guided Robot Learning
Task Specification
Demonstration
+1
Reward Function
Assemble a Table
First, place the top of the table upside down. Then, pick up and attach each table leg to a corner of the tabletop. If your kit comes with a table back, attach the table back to the back of the tabletop. Finally, turn the table upright.
Language Instruction
Domain-Specific Language
Actions (sub-skills)
attach
pickUp
moveTo
place
release
Perceptions
isThere()
num()
isAttached()
Control Flows
if
else
elseIf
while
repeat
Program
def assembleTable():
pickUp(top, left_arm, on, floor)
moveTo(left_arm, (0,0,5), (0,0,5))
for (x, num(leg, on, floor)):
pickUp(leg, right_arm, on, floor)
moveTo(right_arm, (2,0,3), (0,0,0))
attach()
release(right_arm)
moveTo(left_arm, (0,0,5), (0,0,360.0/x))
if (isThere(back, on, floor):
pickUp(back, right_arm, on, floor)
moveTo(right_arm, (1,0,7), (90,0,0))
attach()
Observation
High-Level Plan
moveTo
pickUp
attach
Low-Level Execution
Joint u torque
[-2.09531783e-19 2.72130735e-05 6.14480786e-22 -3.45474715e-06
7.42993721e-06 -1.40711141e-04 -3.04253586e-04 -2.07559344e-04
8.50646247e-05 -3.45474715e-06 7.42993721e-06 -1.40711141e-04
-3.04253586e-04 -2.07559344e-04 -8.50646247e-05 1.11317030e-04
-7.03465386e-05 -2.22862221e-05 -1.11317030e-04 7.03465386e-05
-2.22862221e-05 -2.09531783e-19 2.72130735e-05 6.14480786e-22
-3.45474715e-06 7.42993721e-06 -1.40711141e-04 -3.04253586e-04
-2.07559344e-04 8.50646247e-05 -3.45474715e-06 7.42993721e-06
-1.40711141e-04 -3.04253586e-04 -2.07559344e-04 -8.50646247e-05
1.11317030e-04 -7.03465386e-05 -2.22862221e-05 -1.11317030e-04]
Joint v torque
[-2.09531783e-19 2.72130735e-05 6.14480786e-22 -3.45474715e-06
7.42993721e-06 -1.40711141e-04 -3.04253586e-04 -2.07559344e-04
8.50646247e-05 -3.45474715e-06 7.42993721e-06 -1.40711141e-04
-3.04253586e-04 -2.07559344e-04 -8.50646247e-05 1.11317030e-04
-7.03465386e-05 -2.22862221e-05 -1.11317030e-04 7.03465386e-05
-2.22862221e-05 -2.09531783e-19 2.72130735e-05 6.14480786e-22
-3.45474715e-06 7.42993721e-06 -1.40711141e-04 -3.04253586e-04
-2.07559344e-04 8.50646247e-05 -3.45474715e-06 7.42993721e-06
-1.40711141e-04 -3.04253586e-04 -2.07559344e-04 -8.50646247e-05
1.11317030e-04 -7.03465386e-05 -2.22862221e-05 -1.11317030e-04]
⋮


Program Inference
Primitive Skill Acquisition
Task Execution
Neural Program Synthesis from Diverse Demonstration Videos
Demonstrations
Program Policy
Execution
Synthesize
Execute
ICML 2018
Multimodal Model-Agnostic Meta-Learning via Task-Aware Modulation
NeurIPS 2019 (Spotlight)
Skill-based Meta-Reinforcement Learning
Meta-Training Tasks
Fast Learning of New Long-Horizon Task
Task-Agnostic Offline Data
Target Task
Skill-Based Meta-RL
ICLR 2022
Composing Complex Skills by Learning Transition Policies
ICLR 2019
Learning to Synthesize Programs as Interpretable and Generalizable Policies
Program
Execute
Environment
Reward
NeurIPS 2021
Generalizable Imitation Learning from Observation via Inferring Goal Proximity
Expert Demonstrations
Learning Proximity Function
Joint Training
Online Agent Experience
Learning Policy
NeurIPS 2021
Learning to Act from Actionless Videos through Dense Correspondences
ICLR 2024 (Spotlight)
Program Guided Agent
ICLR 2020 (Spotlight)
Hierarchical Programmatic Reinforcement Learning via Learning to Compose Programs
Composed Program
Execute
Environment
Reward
ICML 2023
Diffusion Model-Guided Behavioral Cloning
ICML-W 2023 & Submitted to ICML 2024
Diffusion Rewards Guided Adversarial Imitation Learning
Diffusion Discriminative Classifier
Policy
Diffusion Model
Diffusion Reward
ICLR-W 2024 & Submitted to ICML 2024
Bootstrap Your Own Skills: Learning to Solve New Tasks with Large Language Model Guidance
Skill Library
Practice in Environment
Update Agent
Name New Skill
CoRL 2023 (Oral)
Addressing Long-Horizon Tasks by Integrating Program Synthesis and State Machines
Execute
Environment
Program Machine Policy
Reward
NeurIPS-W 2023 & Submitted to ICML 2024
Efficient Multi-Task Reinforcement Learning via Selective Behavior Sharing
Behavior Proposals
Mixture of Policies
NeurIPS-W 2022 & Submitted to ICML 2024
Integrating Planning and Deep Reinforcement Learning via Automatic Induction of Task Substructures
Environment
Agent
ICLR 2024

Shao-Hua Sun (孫紹華)

Assistant Professor
at National Taiwan University
shaohuas@ntu.edu.tw

Bio

I am an **Assistant Professor** at **National Taiwan University (NTU)** with a joint appointment in **the Department of Electrical Engineering** and **the Graduate Institute of Communication Engineering**. Prior to joining NTU, I recently completed my Ph.D. in Computer Science at the University of Southern California, where I worked in the Cognitive Learning for Vision and Robotics Lab (CLVR). Before that, I received my B.S. degree in Electrical Engineering from NTU. My research interests span **Robot Learning**, **Reinforcement Learning**, **Program Synthesis**, and **Machine Learning**.

Prospective students: I am looking for students interested in machine learning, robot learning, reinforcement learning, and program synthesis. Specifically, I am hiring **M.S.** and **Ph.D. students** admitted to the Data Science and Smart Networking Group at the Graduate Institute of Communication Engineering (電信所丙組/資料科學與智慧網路組) or the Data Science Degree Program (資料科學學位學程) at NTU. Also, I am seeking **undergraduate students**, **research assistants**, and **visitors** with different experience levels. If you are interested in joining my group, please check out this slide and fill in the Google form.

Edit profile

Shao-Hua Sun
@shaohua0116

Assistant Professor @ National Taiwan University (NTU) | CS Ph.D. @USC | Robot Learning, Reinforcement Learning, Program Synthesis | 台大電機系助理教授

Taipei, Taiwan   shaohua0116.github.io   Joined October 2015

868 Following   **2,987** Followers

Execute
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)
Thank You
Questions?
Environment
Demonstrations
Reward