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

\[ t = H \]

\[ X_t = 0 \]

\[ t R_t(s_t, a_t) \]

Goal: maximize

\[ \sum_{t=0}^{t=H} \gamma^t R_t(s_t, a_t) \]
Reinforcement Learning

\[ t = H \]

\[ X_t = 0 \]

\[ R_t(s_t, a_t) \]

**Goal**: maximize

\[ \sum_{t=0}^{t=H} \gamma^t R_t(s_t, a_t) \]
Reinforcement Learning

Agent / Policy

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

\[ t = H \]

\[ X_t = 0 \]

\[ t \]

\[ R_t(s_t, a_t) \]

**Goal:** maximize

\[ \sum_{t=0}^{t=H} \gamma^t R_t(s_t, a_t) \]
Reinforcement Learning

\[ r_t = H \]

\[ s_t = 0 \]

\[ r_t(s_t, a_t) \]

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
Issues with Deep Reinforcement Learning (DRL)

Generalization

Simple task
Complex task

Interpretability

Trust
Safety
Contestability
DEF run()
    IF frontIsClear()
        move
    ELSE
        IF frontIsClear()
            turnLeft
        ELSE
            turnRight
    END IF
DEF run()
WHILE noMarkersPresent()
IFELSE rightIsClear()
turnRight
ELSE
WHILE frontIsClear()
turnLeft
move
DEF run()
WHILE noMarkersPresent()
IFELSE rightIsClear()
   turnRight
ELSE
   WHILE frontIsClear()
      turnLeft
      move
ELSE
LEAPS: Learning Embeddings for Latent Program Synthesis

Stage 1
Learning a program embedding space from randomly generated programs

Grammar

Environment Dynamics

Dynamics

Stage 2
Searching for a task-solving program

Desired Behavior

```
DEF run()
    IF rightIsClear()
        turnRight
    ELSE
        turnLeft

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

DEF run()
    WHILE noMobsPresent()
        IF rightIsClear()
            turnRight
        ELSE
            WHILE frontIsClear()
                move
            turnLeft
        move
```
Learning a Program Embedding Space
Karel Tasks

- StairClimber
- Maze
- FourCorners
- TopOff
- Harvester
- CleanHouse
Baselines

DRL

VIPER
(Decision Tree)

Naive Program Synthesis

Distillation
Qualitative Results

StairClimber

DRL  LEAPS

FourCorners

DRL  LEAPS

Maze

DRL  LEAPS

TopOff

DRL  LEAPS
CEM trajectory Visualization

Dataset Program
GT Program
CEM Population
CEM Next Center

Iteration: 1
Avg. Reward: 0.1000
Next Candidate Program Reward: 0.1000
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

Maze

CEM search

LEAPS Program Policy

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

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

Evaluation on 100 x 100
Results - Zero-shot Generalization

- Maze
- StairClimber
- TopOff

DRL
VIPER
LEAPS

Reward
Interpretability

Human Debugging Interface

Performance Improvement

- Original
- 3 Edits
- 5 Edits

![Performance Improvement Graph](image)
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*
Thank You

Questions?

Paper and code
clvrai.com/leaps