Explaining Deep Adaptive Programs via Reward Decomposition

Martin Erwig, Alan Fern, Magesh Murali, Anurag Koul

Oregon State University

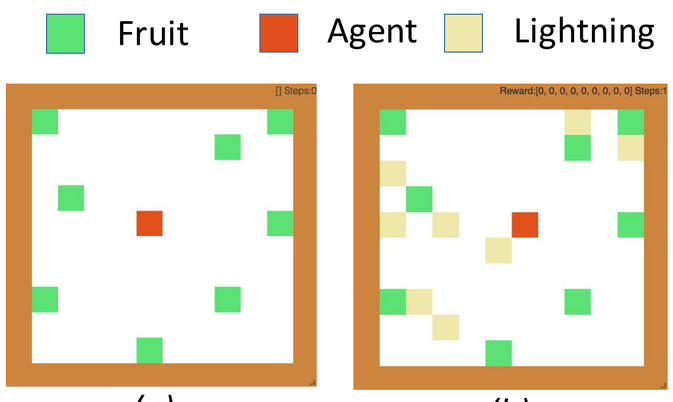
1. Adaptive Programs

- Employ choice points to replace uncertain logic in the code
- Train choice points via Reinforcement Learning to maximize assigned reward
- Decisions at choice points are learned via deep neural networks

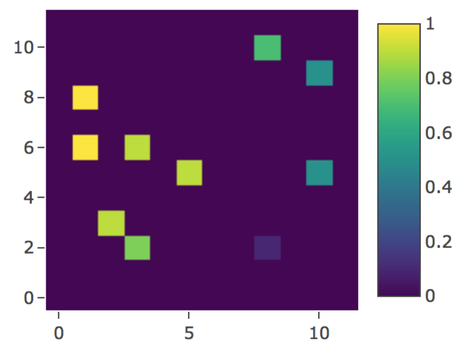
```
state = env.reset()
move = Adaptive(choices = [UP, DOWN,LEFT,RIGHT])
while not done:
   direction = move.choose(state)
   state, reward, done = env.step(direction)
   move.adapt(reward)
```

2. Example Environment

- Goal: Collect maximum number of fruits
- Fruits have fixed location
- Lightning kills the agent
- Horizon: 100 steps



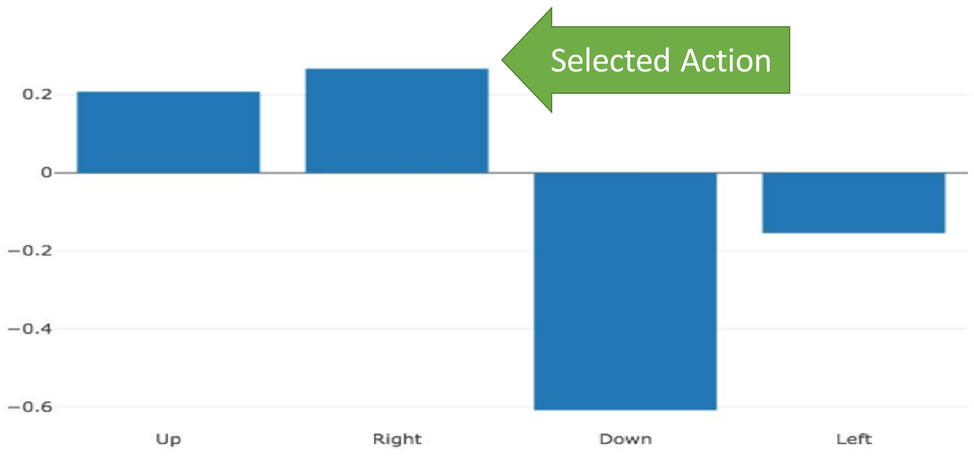
(a) (b)
State Frames without (a) and With (b) lightning



Lightning Occurrence Probability of Cells

3. Explanation

- Why choice A was selected over other choices?
- Selected choice has the maximum expected future award given by $Q(s, A)$
- Each adaptive variable is associated with a Q-function



Q-values for Frame (a)

- Insufficient Explanation by single Q-value
- What factors were responsible for the Q-value? Fruits? Lightning?

4. Explanation via Reward Decomposition

- Reward can be broken down into multiple reward types, each corresponding to semantically distinct ways of acquiring reward
- Total Reward = Each fruit reward + Lightning cost
- Distinct Q-function is learned for each reward type

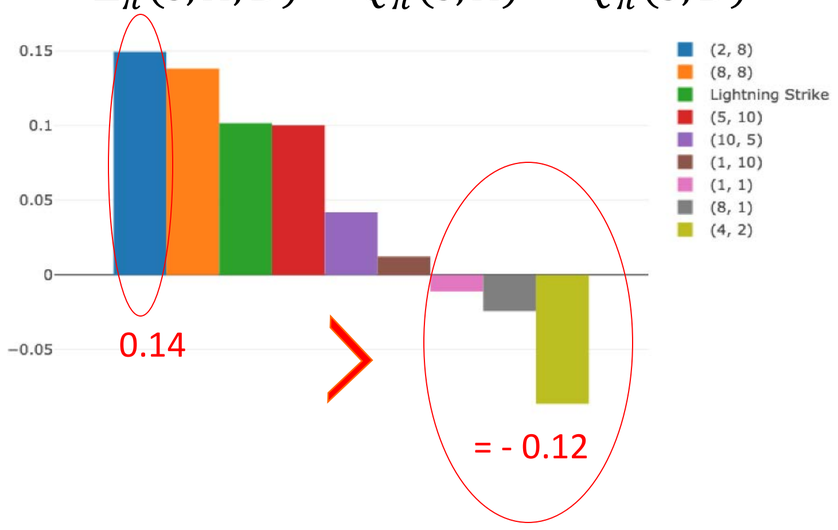
```
state = env.reset()
move = Adaptive(choices = [UP, DOWN, LEFT, RIGHT])
while not done:
   direction = move.choose(state)
   state, rewards, done = env.step(direction)
   for typedReward in rewards:
      move.adapt(typedReward)
```

Decomposed Q-values for each reward type in Frame (a)

5. Reward Difference Explanation (RDX)

- Why choice A was selected over choice B ?
- Reward-type-indexed difference between the decomposed rewards of two actions

$$\Delta_\pi(s, A, B) = Q_\pi(s, A) - Q_\pi(s, B)$$



$\Delta_\pi(s, Right, Left)$

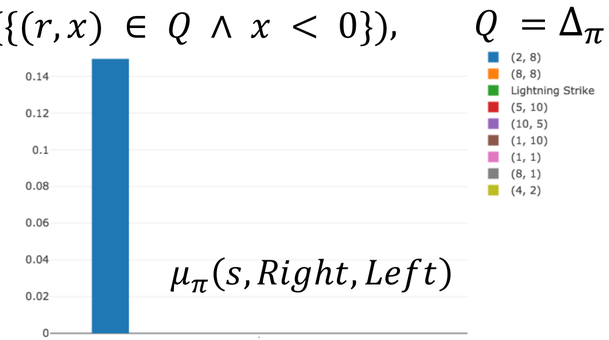
6. Minimum Sufficient Explanation(MSX)

- The minimal subset of $\Delta_\pi(s, A, B)$ whose sum of rewards exceeds the sum of all negative rewards from $\Delta_\pi(s, A, B)$.

$$\mu_\pi(s, A, B) = S \in Sub^+(Q) : \Sigma(S) > R^-(Q) \wedge |S| \text{ is minimal}$$

$$where, Sub^+(Q) = \{S \subseteq Q \mid (r, x) \in S \Rightarrow x > 0\}$$

$$R^-(Q) = \Sigma(\{(r, x) \in Q \wedge x < 0\}), \quad Q = \Delta_\pi(s, A, B)$$



$\mu_\pi(s, Right, Left)$