Carnegie Mellon University

Domain Informed Oracle (DIO) for Reinforcement Learning

Presentation Overview

- Reinforcement Learning (RL)
    - Reinforcement Learning Routine
    - Challenges in Designing a Reward Function
    - Systematic Method for Reward Shaping Outcomes
- Scenario : Traffic Simulation
- Proposed solution : Domain Informed Oracle (DIO)
    - Overview of the Proposed Solution
    - World from RL vs. DIO's perspective
    - Architecture
- Progress Report
    - Evaluation metrics
    - Reinforcement Learning w/o DIO
    - ProbLog

Carnegie Mellon University

Reinforcement Learning (RL)

Reinforcement Learning Routine

- Method of Learning that maps situations to actions in order to maximize its rewards [1].
- The above mapping is a policy.
- Reward function: (S x A) → R
- Reward is domain dependent and crucial in training.
- Bad reward => Suboptimal training time

  Suboptimal policy

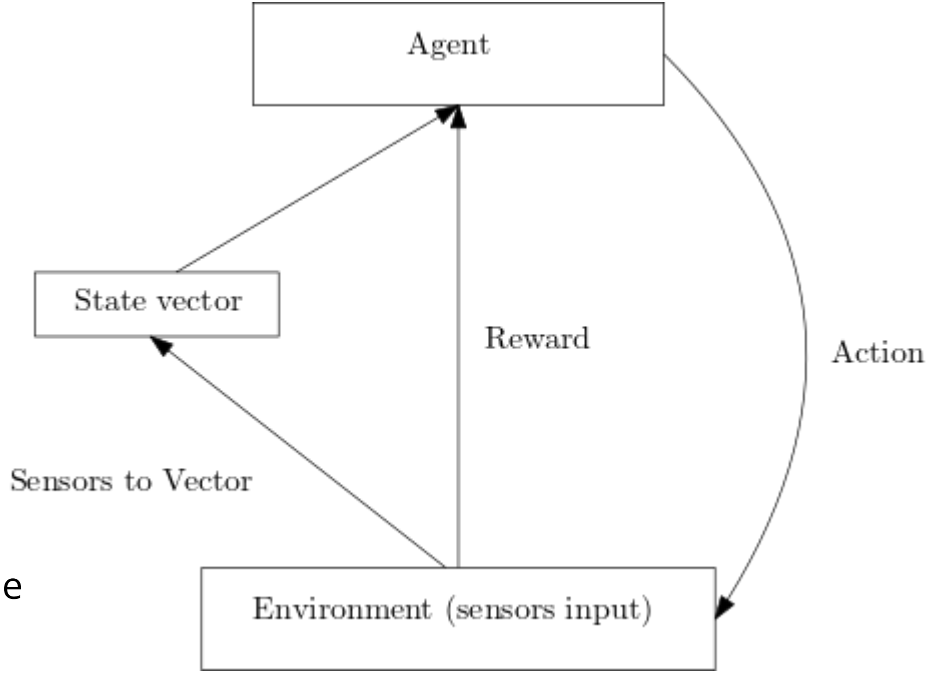


Figure 1 : RL Routine

Challenges in Designing a Reward Function

Lack of systematic methods to design a reward function that ensures fast and efficient learning [2].

Currently dealt with using *standard practices*.

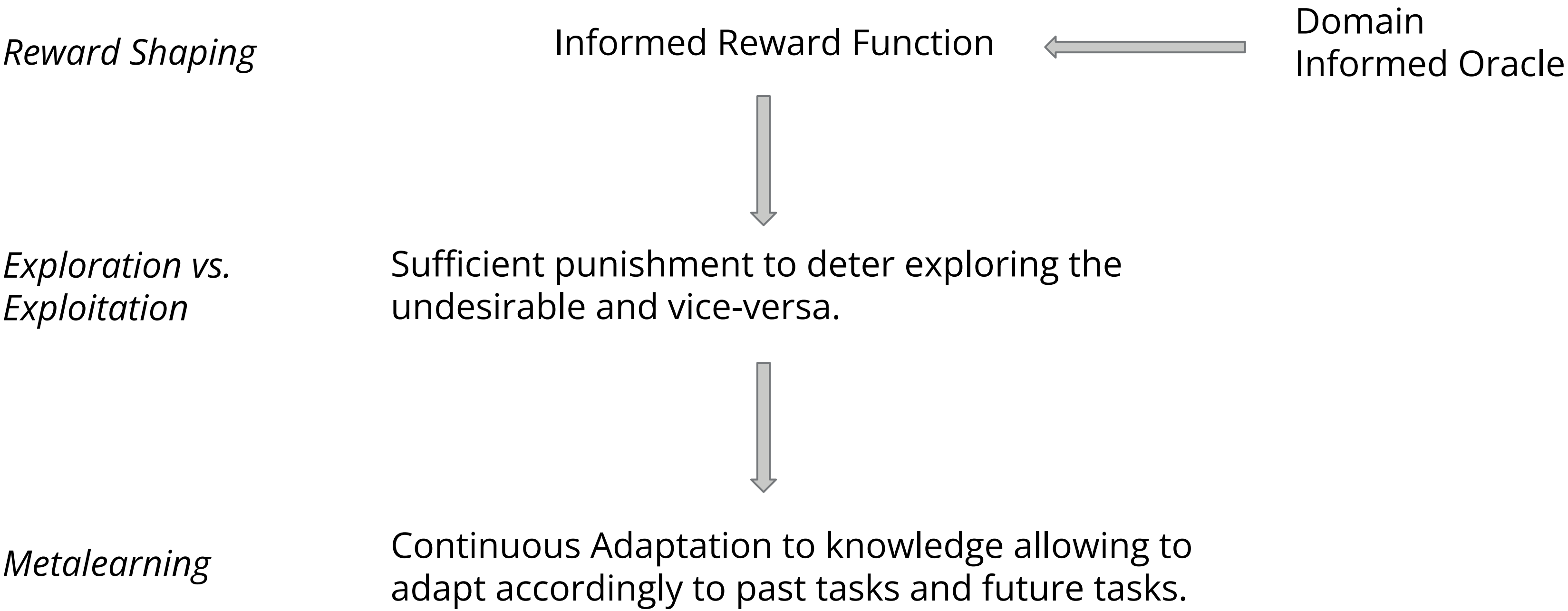
Continuously harvested rewards, collection of +/- rewards, and polynomial differential function if continuous state.

Given solution.

Reward Shaping

Technique for incorporating domain knowledge into Reinforcement Learning. Seen as r’ = r + F where r is the original reward [human bias] and F the shaping reward function.

Systematic Method for Reward Shaping Outcomes



Reward Shaping

Informed Reward Function

Domain
Informed Oracle

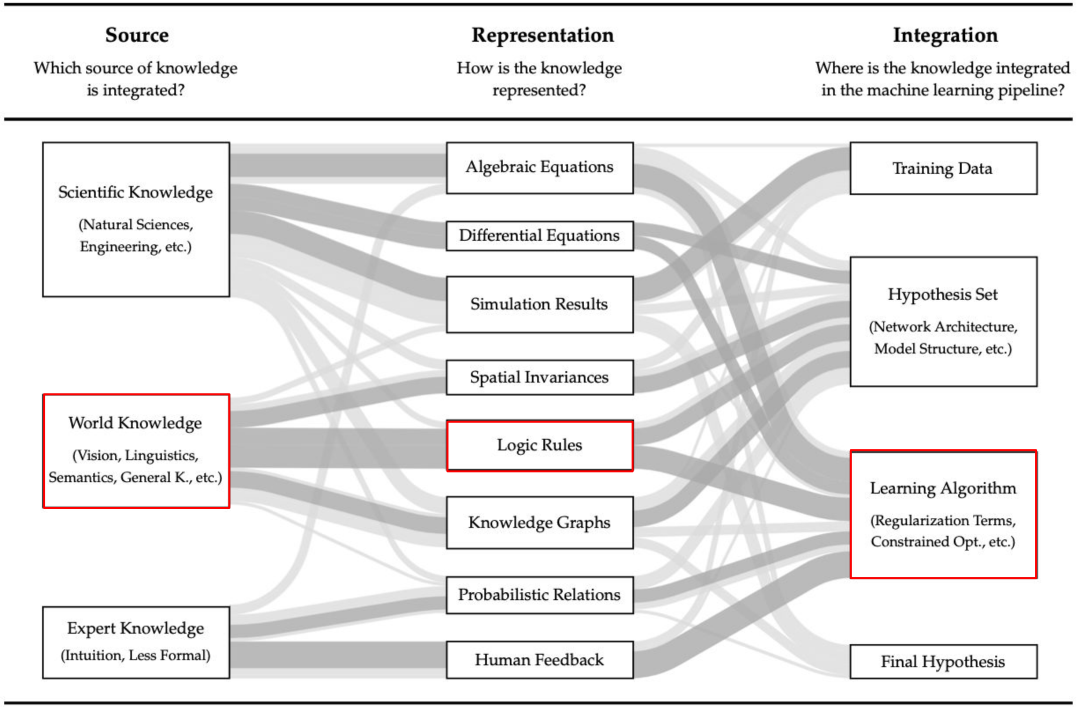
Exploration vs. Exploitation

Sufficient punishment to deter exploring the undesirable and vice-versa.

Metalearning

Continuous Adaptation to knowledge allowing to adapt accordingly to past tasks and future tasks.

Domain Specific Reinforcement Learning (www.)


Source
Which source of knowledge is integrated?
Representation
How is the knowledge represented?
Integration
Where is the knowledge integrated in the machine learning pipeline?
Scientific Knowledge
(Natural Sciences, Engineering, etc.)
World Knowledge
(Vision, Linguistics, Semantics, General K., etc.)
Expert Knowledge
(Intuition, Less Formal)
Algebraic Equations
Differential Equations
Simulation Results
Spatial Invariances
Logic Rules
Knowledge Graphs
Probabilistic Relations
Human Feedback
Training Data
Hypothesis Set
(Network Architecture, Model Structure, etc.)
Learning Algorithm
(Regularization Terms, Constrained Opt., etc.)
Final Hypothesis

World Knowledge -> Logic Rules -> Learning Algorithm

Machine learning limited by its insufficient training data. Under prior knowledge, less time to train needed.

Define knowledge as relational information between entities of the environment.

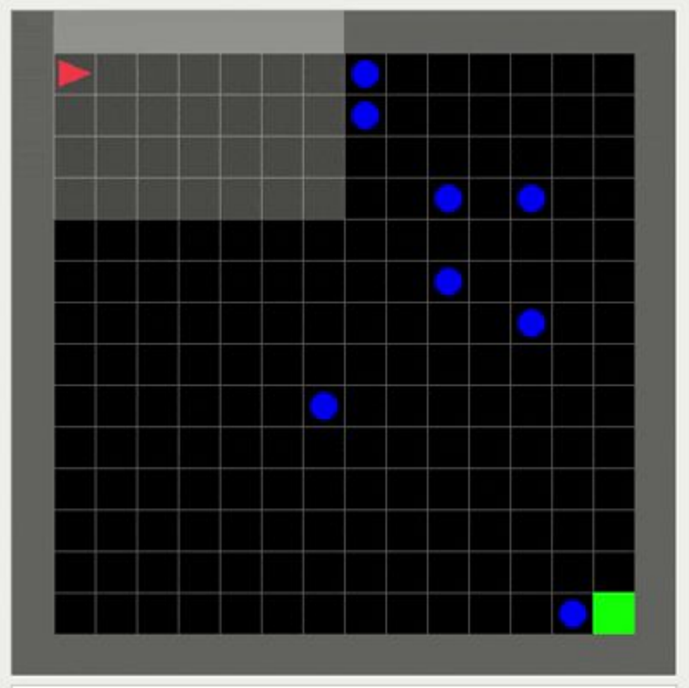
Three ways of looking at the problem: Source / Representation / Integration.

e.g. Integration: from rules to continuous via the t-norm.

Carnegie Mellon University

Traffic Simulation

Traffic Simulation in a Gridworld



Vehicle : Red Arrow

Blue circles : Dynamic Obstacles

Green Square : Goal.

The goal of the agent is to reach point B (the goal) from point A (0,0) in the shortest time possible without colliding with the dynamic obstacles.

Carnegie Mellon University

Proposed Solution

Overview of the Proposed Solution

- Rewards are domain dependent
- The world is governed by rules which are either deterministic or stochastic
- A logic programming module is able to search the space and send feedback to RL
- RL can adapt its reward scheme given this feedback

World from RL perspective

- Has no knowledge of the effect of an action
- Evolves its policy given bayesian rule
- Dependent on the reward scheme
- World represented as a state vector
- Is able to label given states as desirable/undesirable.

World from DIO perspective

- DIO should only be aware of how the world evolves, not what is desirable or undesirable.
- DIO is as good as we make it to be, i.e. how deep the world goes.
- Simplified view of the world that’s type specific, i.e. vehicle vs traffic light vs pedestrian
- Considers all possible worlds [outcomes given stochasticity range of given variables].
- From state can generate multiple set of facts.
- An action is equivalent to a rule.
- From stochastic to deterministic probability.

(Cont.) Overview of the Proposed Solution

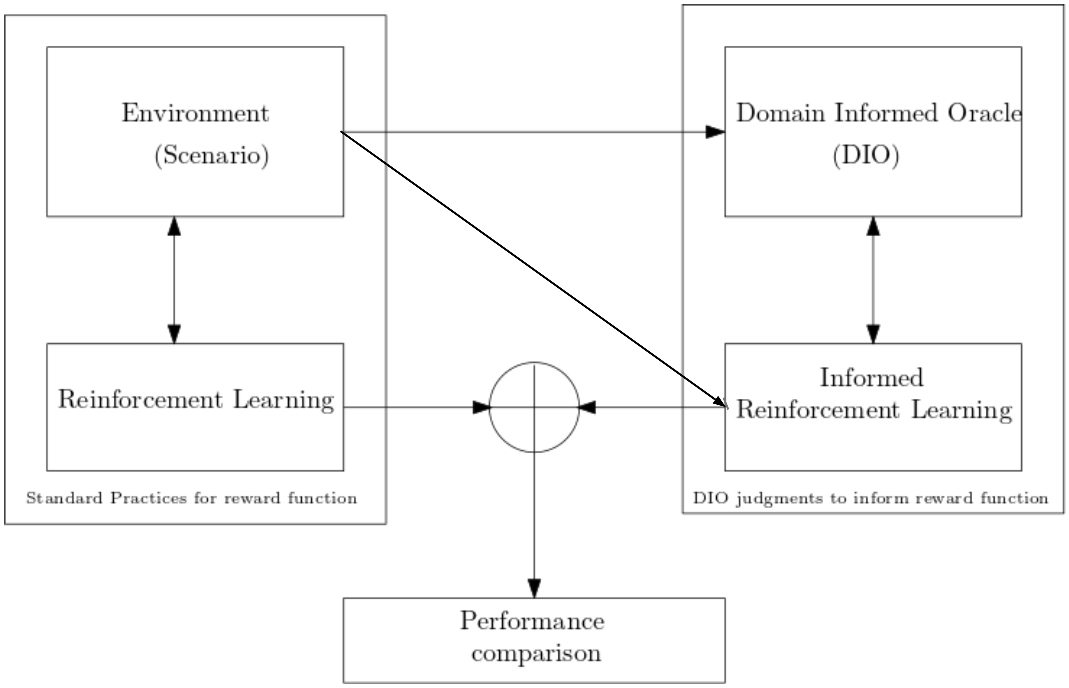


Figure 2 : RL vs. DIO+RL

We define metrics as:

1. Time to train
2. Number of negatives states when deployed in original setting vs. new setting
3. Shortest time to reach B from A

Architecture

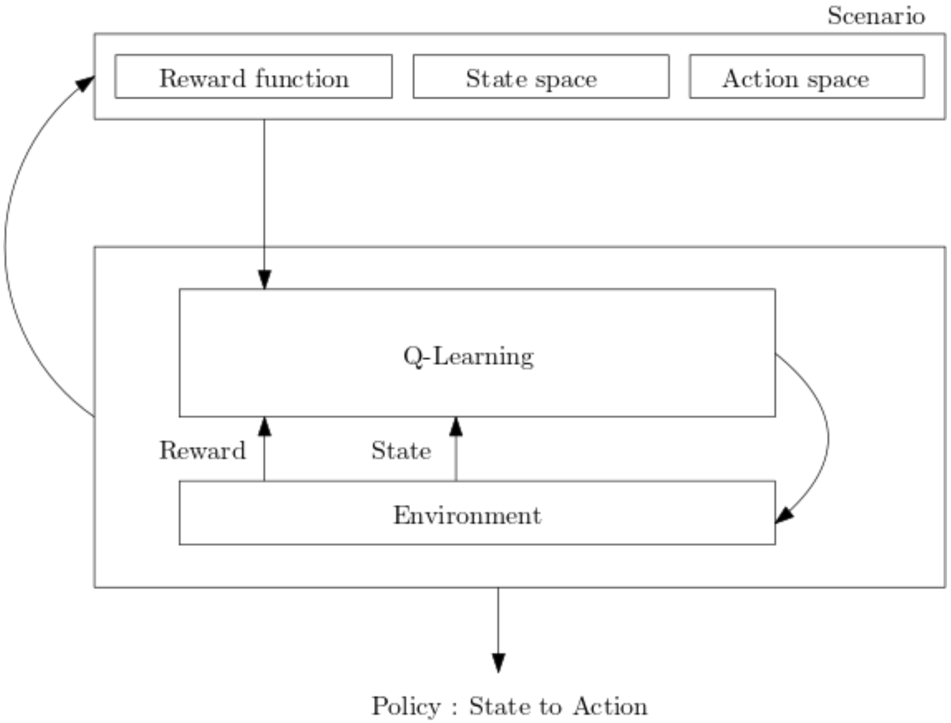


Figure 3 : Reinforcement Learning Architecture

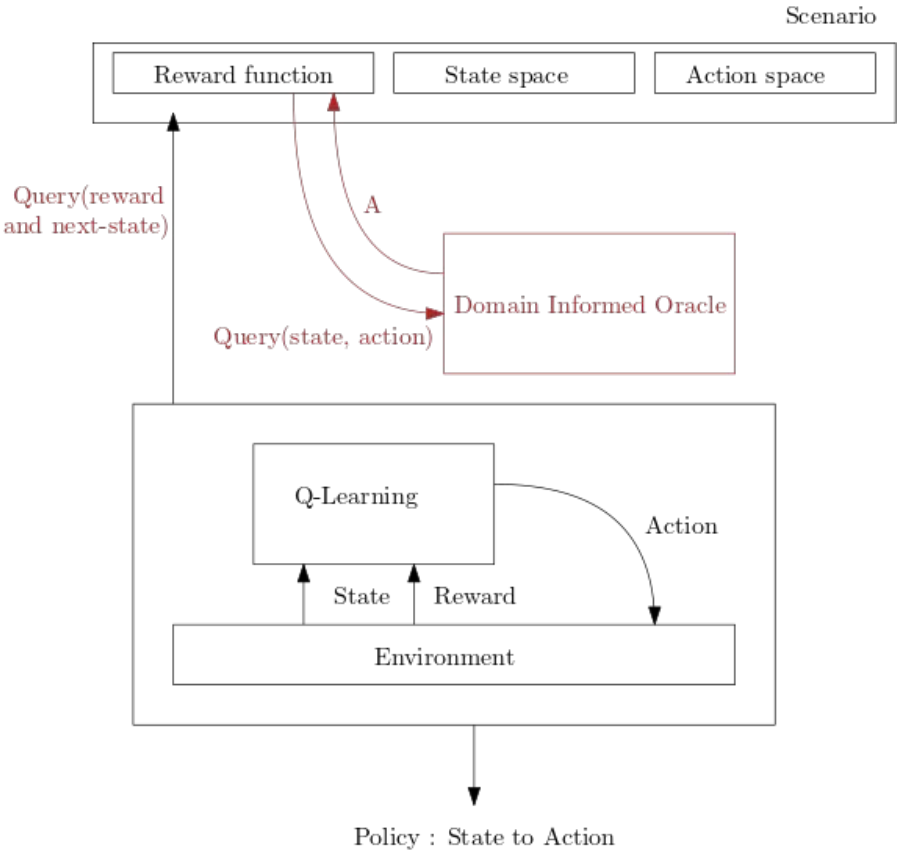


Figure 4 : DIO+RL architecture

Carnegie Mellon University

Carnegie Mellon University

Progress Report

Evaluation Metrics

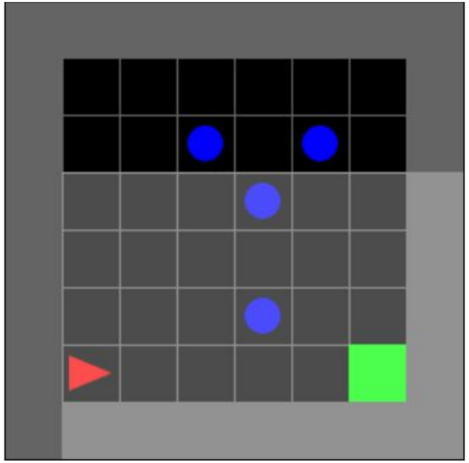
We will evaluate the two implementation (RL w/o DIO) and (RL w/ DIO) using the following metrics.

1. Time to convergence [speed of training]
2. Time to reach the goal from the resulting policy [optimality of solution]
3. Number of negative terminal states of the resulting policy [safety of solution]

(2) and (3) are further tested on a new environment outside of the training environment to assess how well the policy adapts to new scenarios [meta-learning].

Reinforcement Learning w/o DIO

- Gridworld && torch-ac based implementation.
- Proximal Policy Optimization.



8x8 Dynamic Obstacles Grid
Training over 1M frames
Resulting policy does not reach the goal in 4% of the episodes.
When it does reach the goal, the average steps in 30 frames.

ProbLog

To encode real-world uncertainties in Prolog programs.

- Probabilistic facts defined as **p::fact.** where p is the uncertainty.
- **AND** => two events happening simultaneously.
- **OR** => two events happening independently of each other, ‘at least’.
- To ask about the probability of a fact/rule, we **query** it.
- Usage of **evidence(Fact, True/False)** for conditional probabilities.
- Can use CSV to store knowledge base and call ProbLog as a Python library

In a RL context, we consider (1) hardcoded probability distribution over the traffic scenario vs (2) learned rules from set of observations fed into DIO

(Cont.) ProbLog

Knowledge Base

```
1.0 :: atPos(0, 0).
1.0 :: speed(0).
1.0 :: acc(0).
1.0 :: time(0).
1.0 :: timestep(1).
1.0 :: direction(right).
1.0 :: obs(0,1,1,0,0).
```

(deterministic)

Domain Specific Rules

```
atPos(X, Y) :- atPos(X1, Y1),
          speed(V),
          timestep(T1),
          direction(left),
          X is (X1 - V*T1),
          Y is (Y1).
```

Carnegie Mellon University

Next

todo.

- Finalize the translation from the ProbLog module to the RL module.
- Test the resulting architecture on the metrics described before.
- Further the complexity of the scenario by making use of Carla.

References

[1] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. MIT press, 2018.

[2] Adam Laud. Theory and application of reward shaping in reinforcement learning. 04 2011.

[3] Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-Reinforcement Learning of Structured Exploration Strategies. In Conference and Workshop on Neural Information Processing Systems (NeurIPS), page 10, 2018.

[4] Nicolas Schweighofer and Kenji Doya. Meta-learning in Reinforcement Learning. Neural Networks, 16(1): 5–9, January 2003.

Carnegie Mellon University

Appendix

Previous Work in Reward Shaping

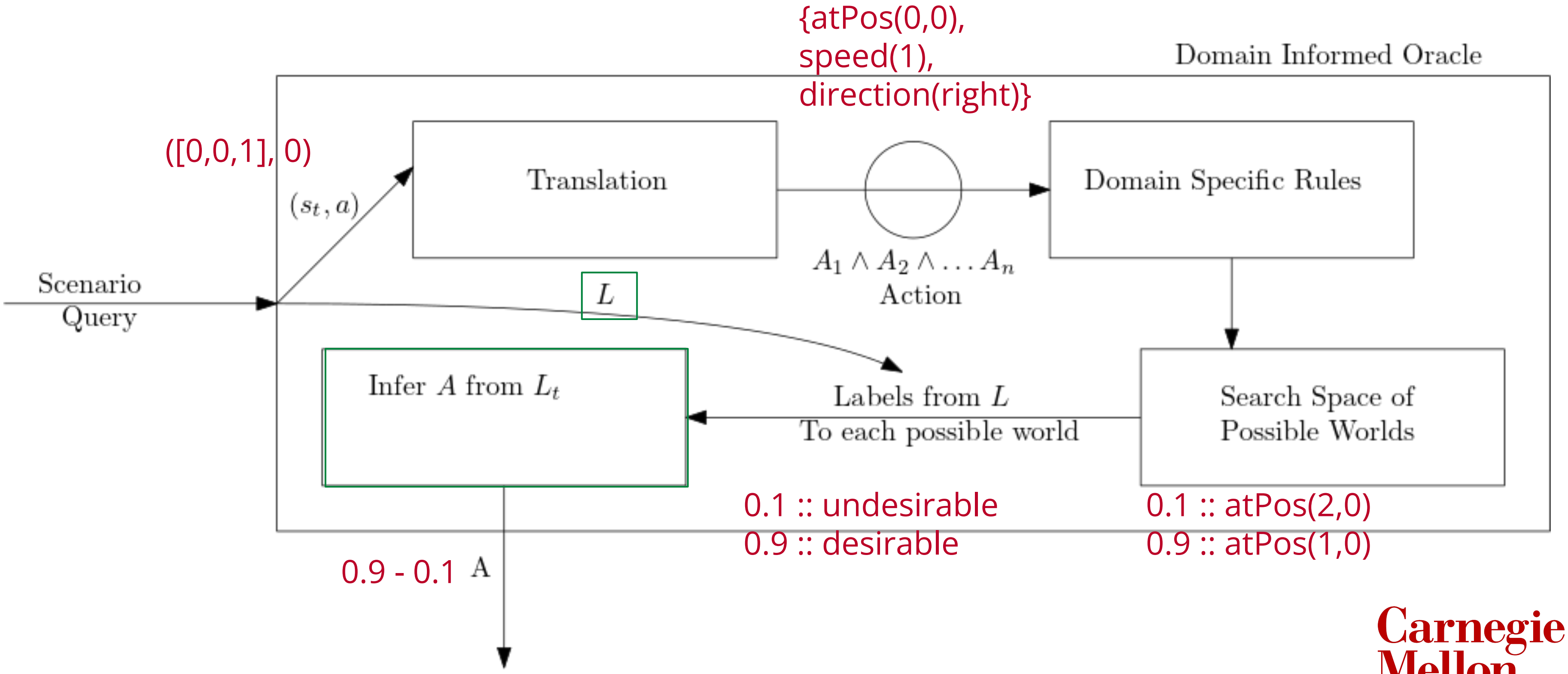
Technique for incorporating domain knowledge into Reinforcement Learning. Seen as r’ = r + F where r is the original reward [human bias] and F the shaping reward function.

1. Parametrized reward shaping. Adaptive approach. (1) Optimize policy using human bias and (2) optimize parametrized shaping weight function. www.
2. Potential-based shaping. Shaping reward define as the discounted change from state 1 to state 2. The potential is added to the natural reward. www.
3. Belief reward shaping. Augments the reward distribution with prior beliefs that decay with experience. Independent implementation from the reinforcement learning algorithm. www.

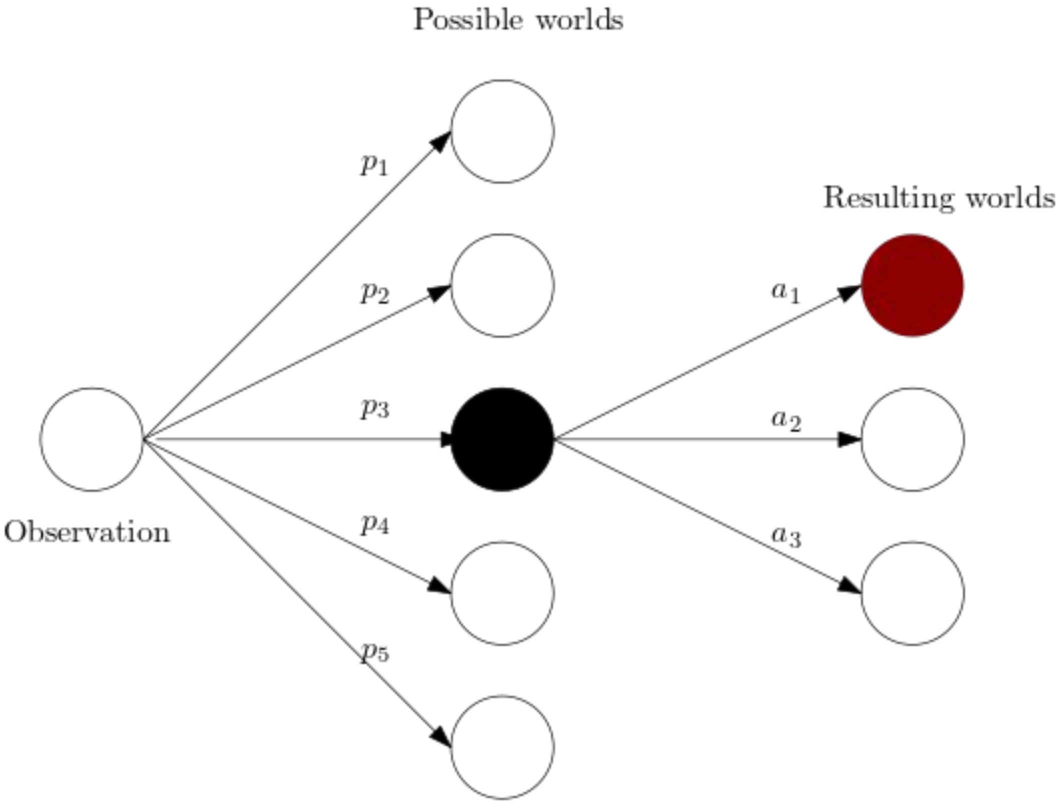
Most work does not define a novel architecture, rather updates the policy function to incorporate reward shaping.

Overall, lack of modularity and making use of modules that already define domain knowledge to specific problems.

Specifications of DIO



Possible Worlds



Method to consider stochasticity of the environment and consider its associated probabilities.

Reinforcement Learning Results

```
F 2218.0 | FPS 1007 | D 2 | R:μσmM 0.84 0.38 -1.00 0.96 | F:μσmM 22.2 6.9 11.0 45.0
10 worst episodes:
- episode 1: R=-1.0, F=13.0
- episode 29: R=-1.0, F=18.0
- episode 50: R=-1.0, F=11.0
- episode 53: R=-1.0, F=28.0
- episode 41: R=0.841796875, F=45.0
- episode 46: R=0.841796875, F=45.0
- episode 83: R=0.8628906011581421, F=39.0
- episode 66: R=0.883984386920929, F=33.0
- episode 79: R=0.883984386920929, F=33.0
- episode 81: R=0.883984386920929, F=33.0
```

Carnegie Mellon University