From Black-Box to Glass-Box Models:
Toward Explainability in Reinforcement Learning

Alexa Jordyn Cole, Erik Bloomquist, Stephen Wormald*, Damon L. Woodard†

Department of Electrical and Computer Engineering, University of Florida

Email: alexa.cole@ufl.edu, erikbloomquist@ufl.edu, *Stephen.wormald@ufl.edu, †dwoodard@ece.ufl.edu

Herbert Wertheim College of Engineering
Florida Institute for National Security (FINS)
UNIVERSITY *of* FLORIDA

CENTER FOR UF UNDERGRADUATE RESEARCH

FINS Florida Institute for National Security

Abstract

Reinforcement learning (RL) is a subcategory of machine learning where an **agent** learns how to act in an **environment** by repeatedly learning what actions and strategy maximize a total cumulative reward. The environment can be complex (large state and action spaces) and hard to interpret, and the relationship between environment characteristics is not well understood. These observations lead toward two research goals:

(1) **Explore and quantify** the relationship between the accuracy of RL algorithms and environment properties. For example, RL algorithms may generally perform better in low-dimensional state spaces.

(2) **Find interpretable policy representations** that maintain agent performance. For example, modeling agent policies using **Markov Decision Processes (MDPs)** could reveal regions in an environment where agents are more prone to certain actions.

History and Challenges

History: For applications in medical, financial, and government fields, "black-box" models may not always be suitable systems for making predictions, regardless of high performance. In these applications, the agent's decision-making process must be transparent to ensure trustworthy and safe autonomy.

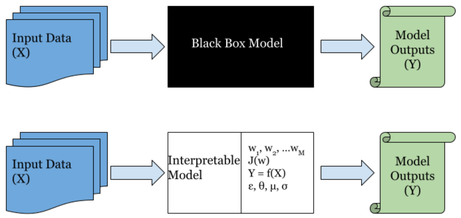


Fig. 1 Comparing black box and interpretable models.

Challenges: Many RL algorithms use black boxes to learn the when to act in environment states – understanding (1) what RL algorithms lean and (2) when to use certain algorithms learned are open challenges. How can we learn when to use certain algorithms and understand the insights leveraged by each model?

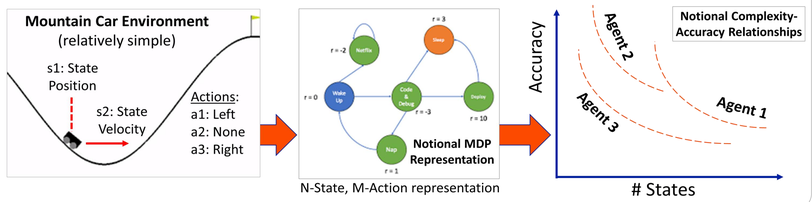
Approach: Our approach is to extract an MDP that represent an algorithm's policy and interpret characteristics of the environment from this MDP that may reveal when certain algorithms perform better in certain environments.

The Theoretical Framework

(1) The behaviors of RL agents acting in environments can be challenging to understand...

(2) Extracting interpretable "policies" could simplify behavior comprehension...

(3) Quantifying algorithm complexity-accuracy relationships could streamline model selection.



Current Work

Finding Interpretable Policy Representations: Studying policies learned by PPO and DQN on the simple environment Mountain Car.

- PPO and DQN's respective policies behave differently in the state-action space
  - PPO's policy has clustered actions based on regions within the state space. Meanwhile, DQN's policy does not show any association between an action and its position in the state space. Due to this, PPO's policy can easily be extracted from the scatterplot when compared to DQN.
- Explored ways to extract human-readable policies from black-box models.

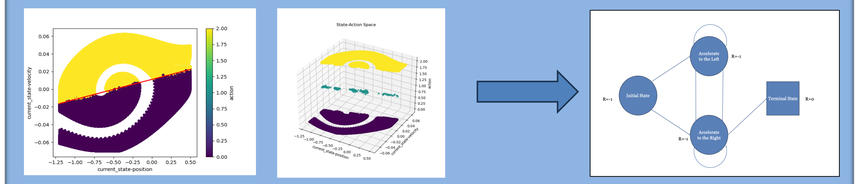


Fig 2. The State-Action Space of Mountain Car for PPO agent. The red line seen within the 2D scatterplot corresponds to the extracted policy for PPO. Due to the ability to easily extract the policy, we can easily represent it as an MDP.

Fig 3. The State-Action Space of Mountain Cart for DQN agent. Due to the inability to extract the policy from the state-action space, an associated MDP could not be represented.

Performance Comparison: Investigated RL algorithm performance across different environments to study variability in training performance.

- With RL's high computational complexity, it may not always be feasible to train a variety of algorithms on the same task before selecting the final model.
- Analyzed how changes in the state-action space affect convergence and policy quality.

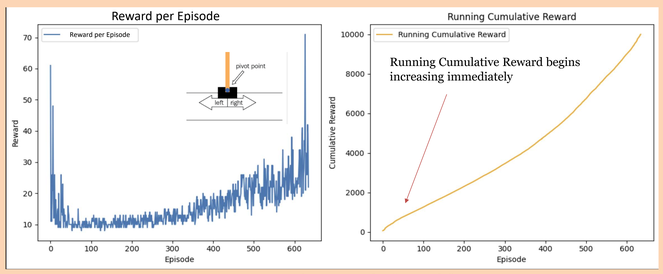


Fig 4. Training curves for DQN under "CartPole" environment.

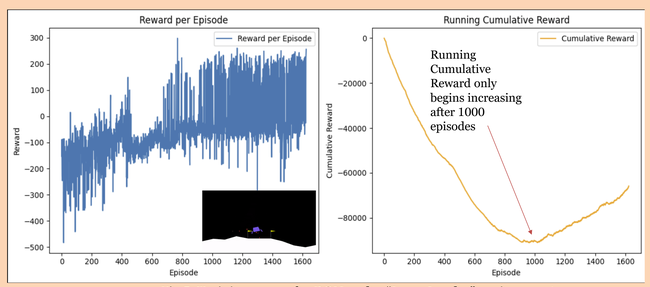


Fig 5. Training curves for DQN under "LunarLander" environment.

Environments Under Test

Cart Pole

Objective: Keep a pole balanced upright on a moving cart only by applying left/right forces.

Characteristics: Low-dimensional state and action spaces (simple task).

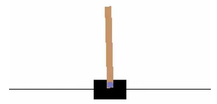


Fig 6. Visual representation of "Cart Pole" environment.

Lunar Lander

Objective: Land a spacecraft softly between flags on a 2D plane.

Characteristics: State and action spaces have medium-sized dimensionality (more complexity).

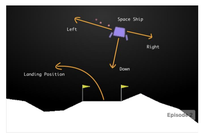


Fig 7. Visual representation of "Lunar Lander" environment.

Mountain Car

Objective: Drive a car up a steep hill, but engine is not powerful enough to propel up the entire hill in one go. Must learn to leverage momentum.

Characteristics: Low-dimensional state and action spaces.

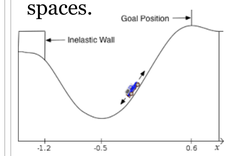


Fig 8. Visual representation of "Mountain Car" environment.

Key Takeaways and Future Work

Key Takeaways:

- By evaluating one algorithm on a variety of environments, we can start to uncover exactly which characteristics make an algorithm optimal for an environment.
- Different algorithms' policies vary in how they interact with the state-action space. Whether this is due to the environment or the algorithm itself needs to be explored.

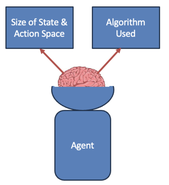


Fig 9. Abstraction of agent performance dependency on both environment characteristics and choice of RL algorithm.

Future Work:

- Explainability
  - Study state abstraction across more complex environments.
  - Evaluate how different representations scale with environment type.
  - Implementing clustering based approach for extracting MDP in the state-action space.
- Defining Environment Types
  - Using the extracted MDP representations, compress and cluster various environments to group environments and discover their similarities.
- Evaluation of Algorithm Types
  - Evaluate how other agent's performance is impacted on more complex environments.

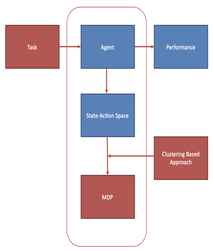


Fig 10. Policy extraction using a clustering-based approach.