



UNIVERSITÀ  
DEGLI STUDI  
DI TRIESTE

## Analysis of RL Algorithms for a Simulated Hill Climb Racing Agent

July 28, 2025

- ① Problem Definition
- ② Deep Q-Network
- ③ Expected SARSA
- ④ Proximal Policy Optimization
- ⑤ Results

# 1 Problem Definition

## 2 Deep Q-Network

## 3 Expected SARSA

## 4 Proximal Policy Optimization

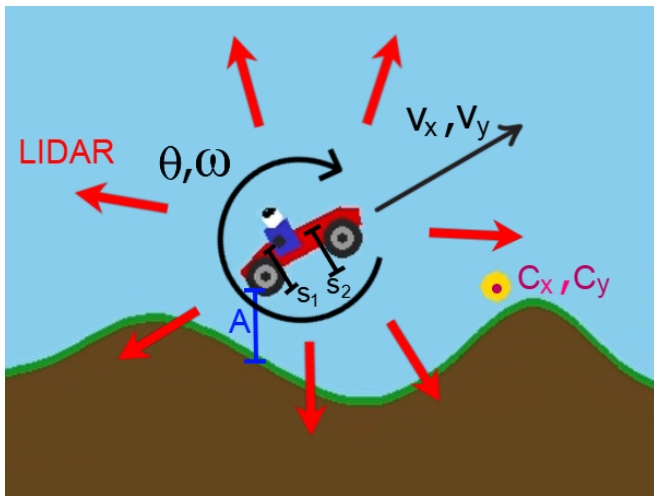
## 5 Results

# Markov Decision Process

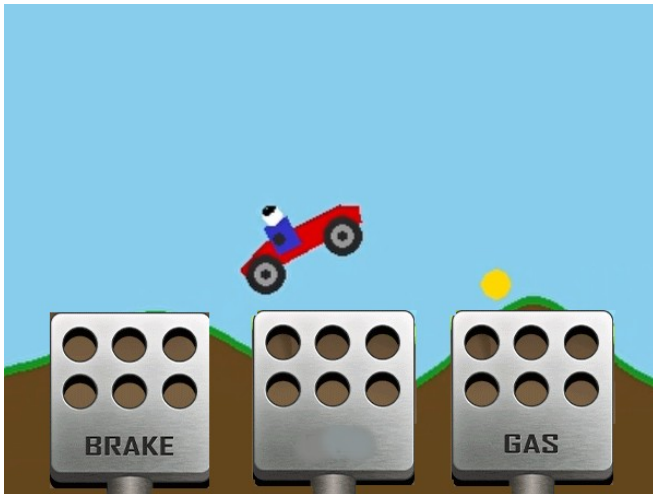
A MDP is a **stochastic model for sequential decision making** defined by a tuple:

$$(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$$

# State Space ( $\mathcal{S}$ )



# Action Space ( $\mathcal{A}$ )



# Transition Dynamics ( $\mathcal{P}$ )

# Reward Function ( $\mathcal{R}$ )

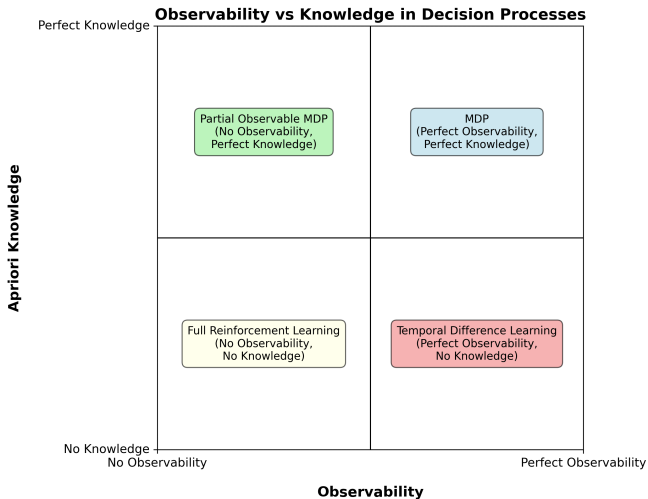
Event	Value
Forward Progress (per meter)	+5.0
Coin Collection	+20.0
Air Time (per second)	+5.0
Time Penalty (per step)	-0.1
Crash (Episode End)	-50.0



# Discount Factor ( $\gamma$ )

policy ( $\pi$ )

# Problem Classification



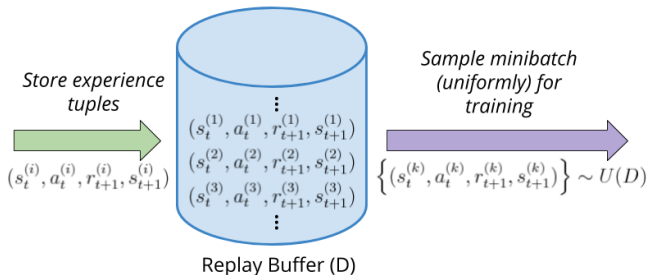
- 1 Problem Definition
- 2 Deep Q-Network
- 3 Expected SARSA
- 4 Proximal Policy Optimization
- 5 Results

# Characteristics of DQN

DQN combines the principles of **deep neural networks** with **Q-learning**.

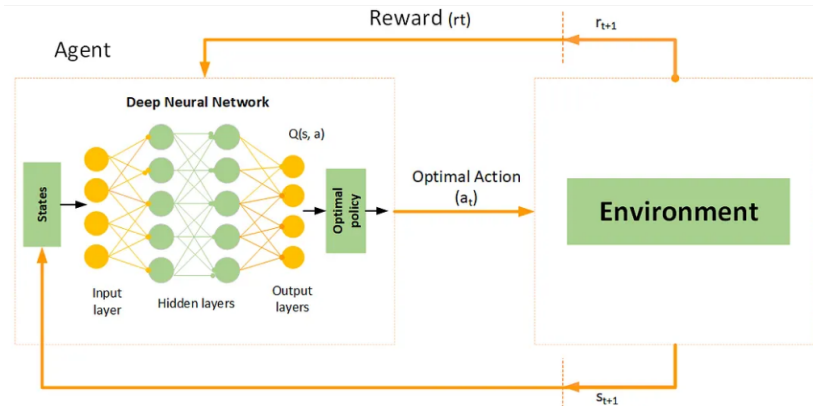
- **Off-policy:** learning from actions taken by different policies.
- **Offline:** it collects a batch of experiences.

# DQN Training



$$L(\theta) = \left( \underbrace{r + \gamma \max_{a'} Q(s', a'; \theta)}_{\text{Target}} - \underbrace{Q(s, a; \theta)}_{\text{Prediction}} \right)^2$$

# DQN Algorithm



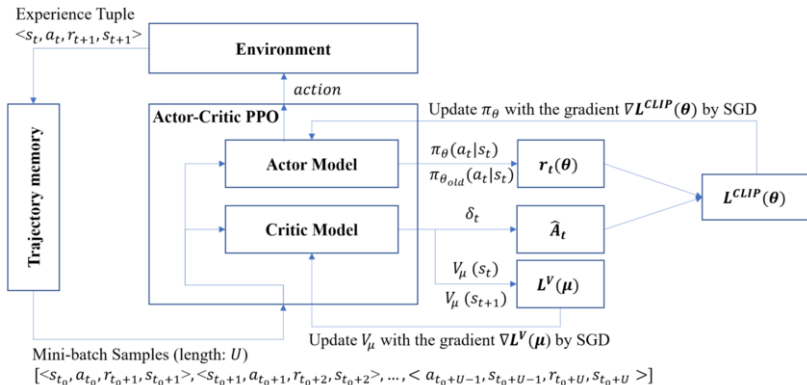
- 1 Problem Definition
- 2 Deep Q-Network
- 3 Expected SARSA**
- 4 Proximal Policy Optimization
- 5 Results



# Expected SARSA Algorithm

- 1 Problem Definition
- 2 Deep Q-Network
- 3 Expected SARSA
- 4 Proximal Policy Optimization**
- 5 Results

# PPO Algorithm



- 1 Problem Definition
- 2 Deep Q-Network
- 3 Expected SARSA
- 4 Proximal Policy Optimization
- 5 Results**

Thank you!