Analysis of RL Algorithms for a Simulated Hill Climb Racing Agent

July 28, 2025

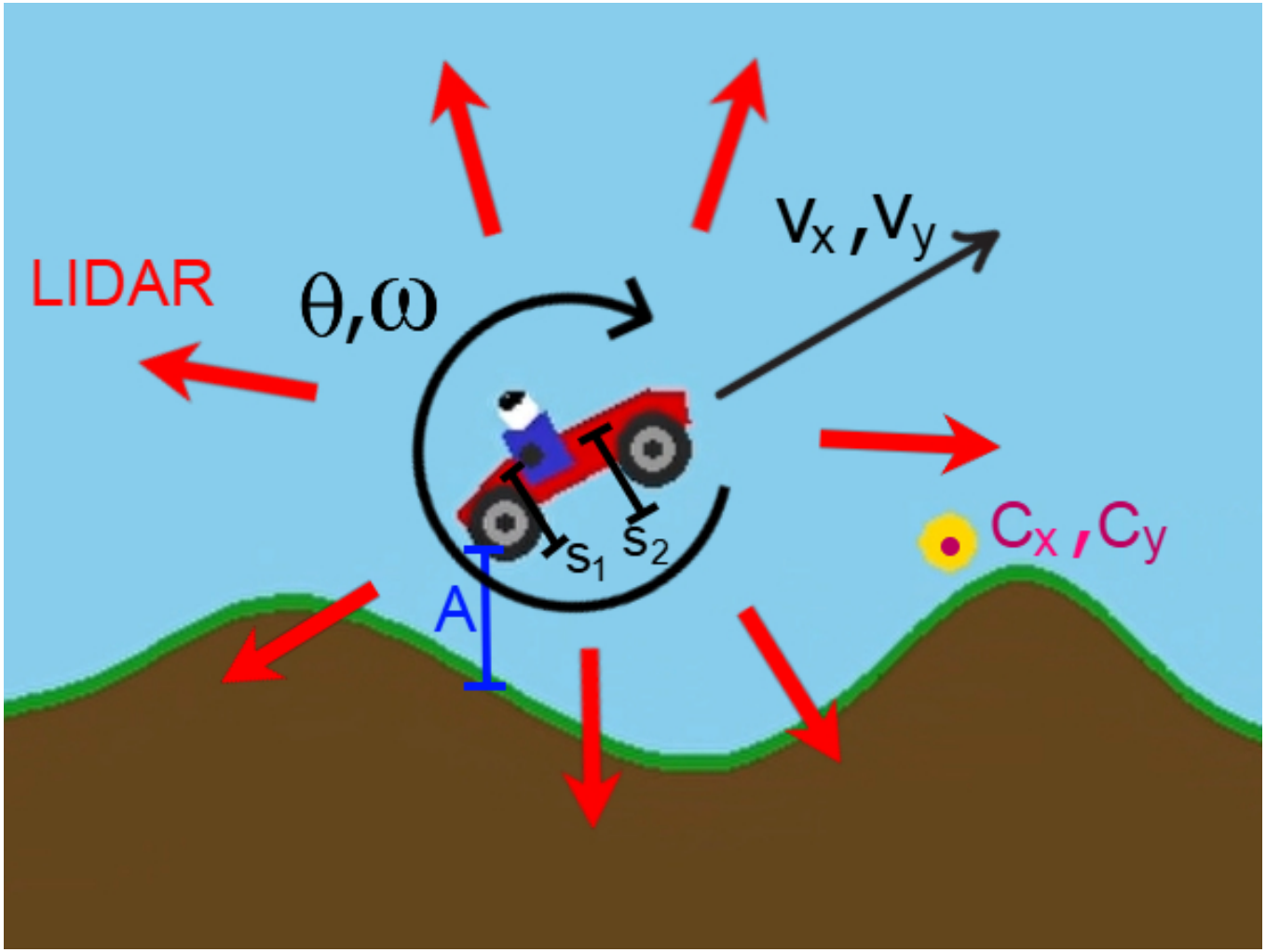
Markov Decision Process

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

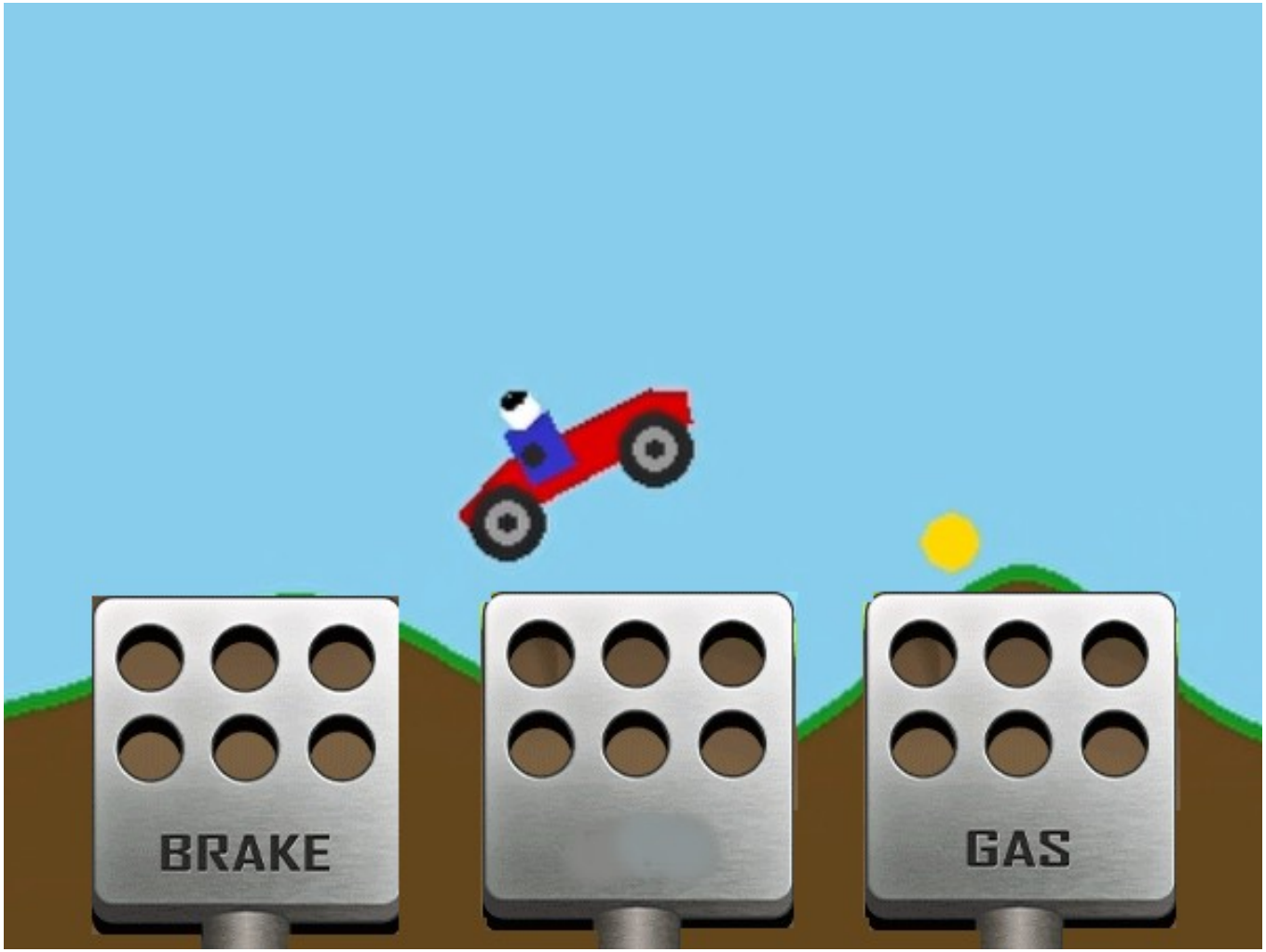
$$(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$$

.

State Space ($\mathcal{S}$)



Action Space ($\mathcal{A}$)



Transition Dynamics ($\mathcal{P}$)

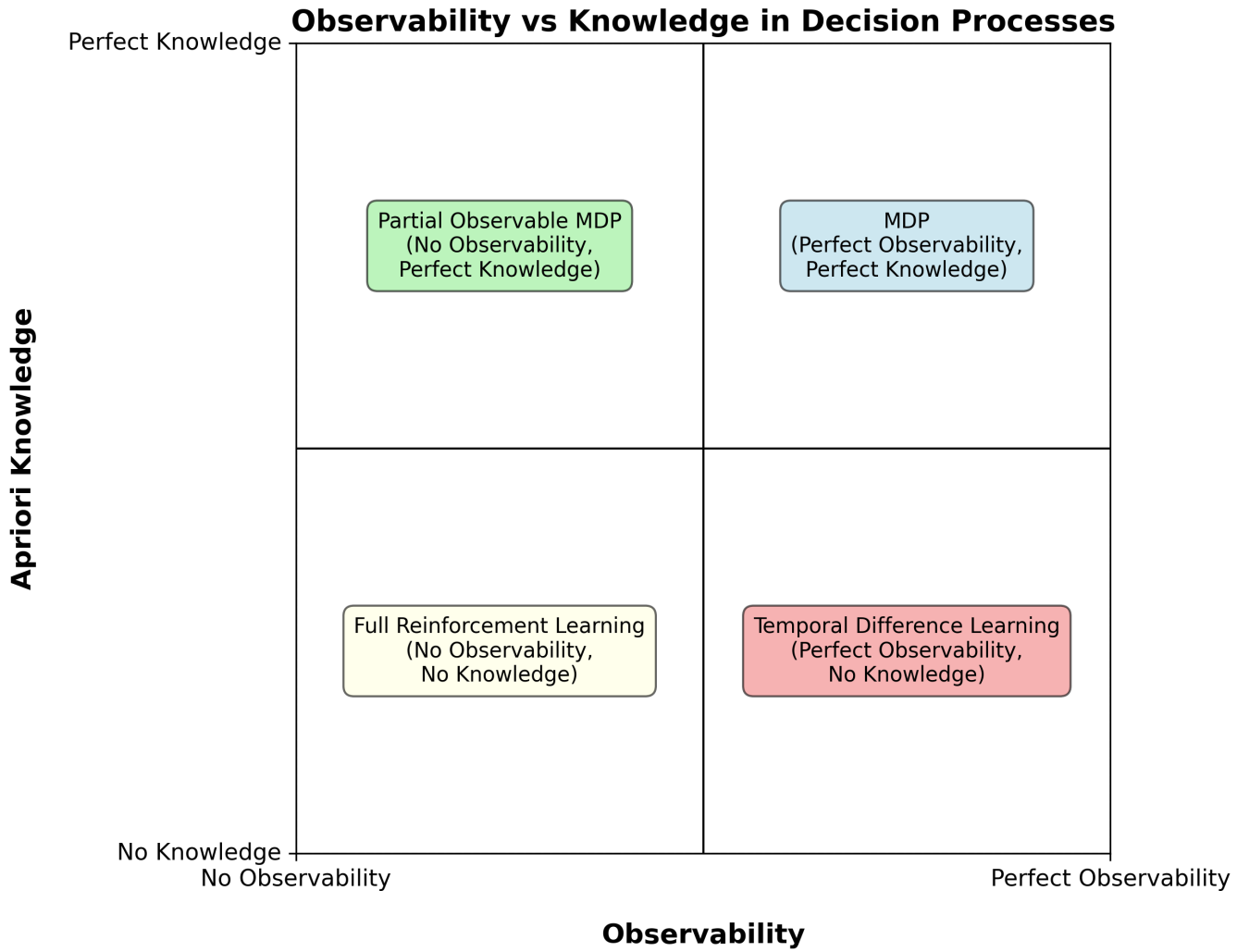
Reward Function ($\mathcal{R}$)

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 (γ)

policy (π)

Problem Classification

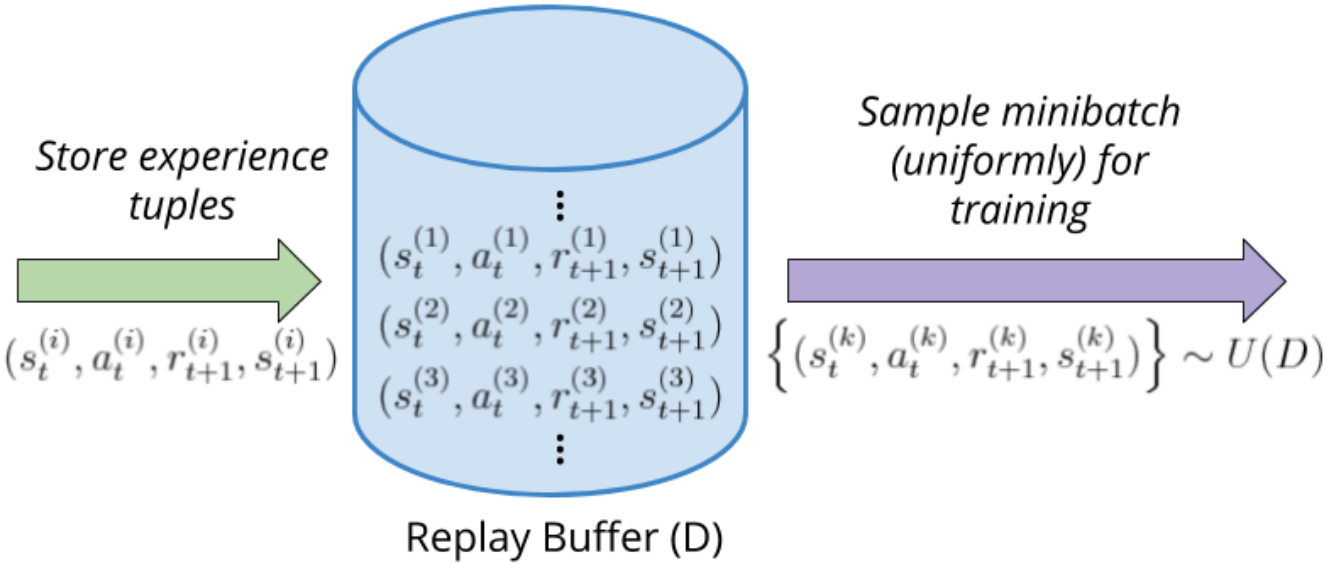


Characteristics of DQN

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

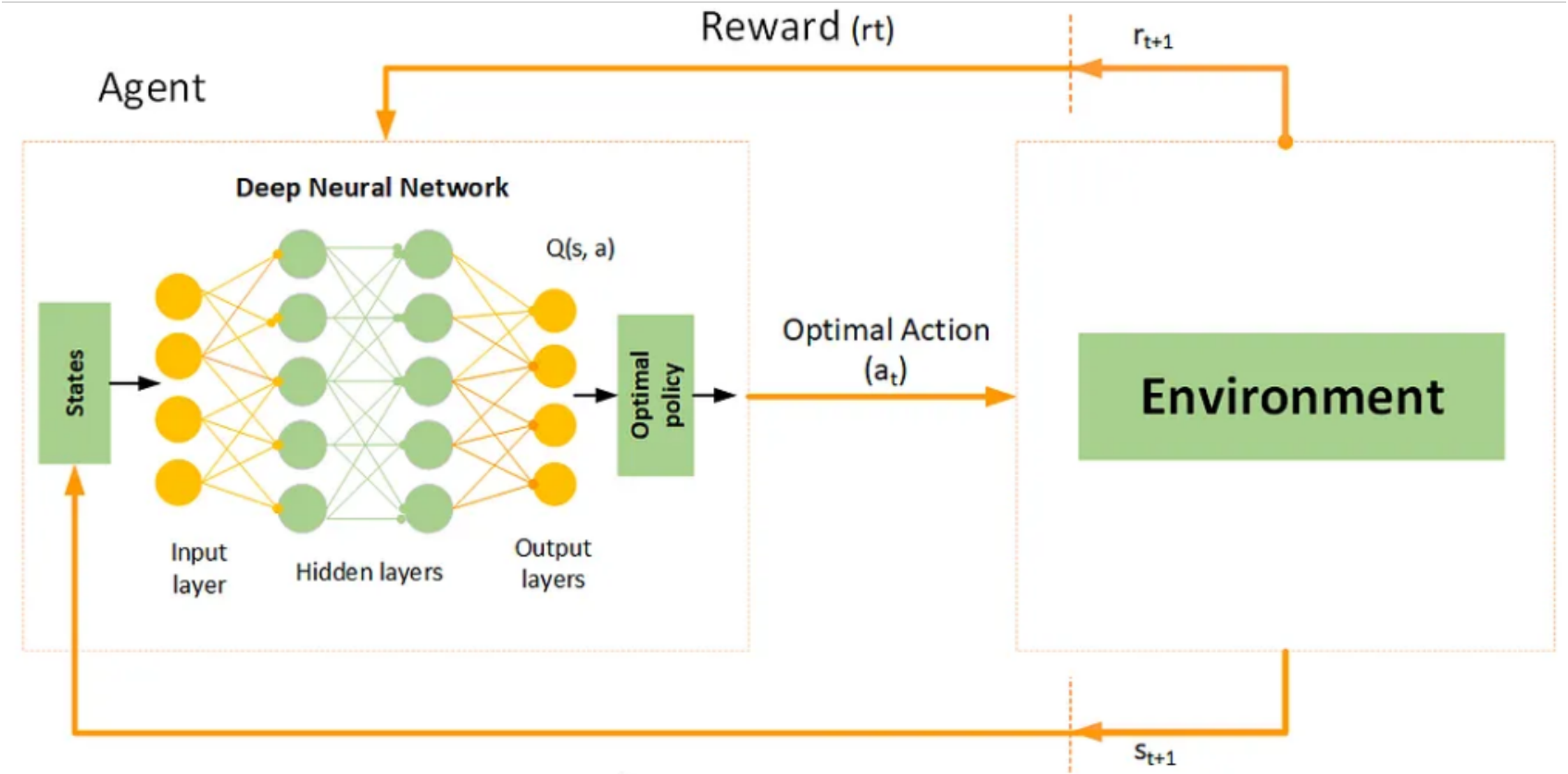
- **Off-policy:** learning from actions taken by different policies.
- **Offline:** it collects a batch of experiences.

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



Characteristics of Expected SARSA

Expected SARSA combines the principles of statistical expectation with the on-policy learning algorithm SARSA.

$$(s, a, r, s', a')$$

- **On-policy:** the update is based on the expected value according to the policy being followed.
- **Online:** An update after each single step.

From SARSA to Expected SARSA

- **SARSA**

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma Q(s',a') - Q(s,a) \right]$$

- **Expected SARSA**

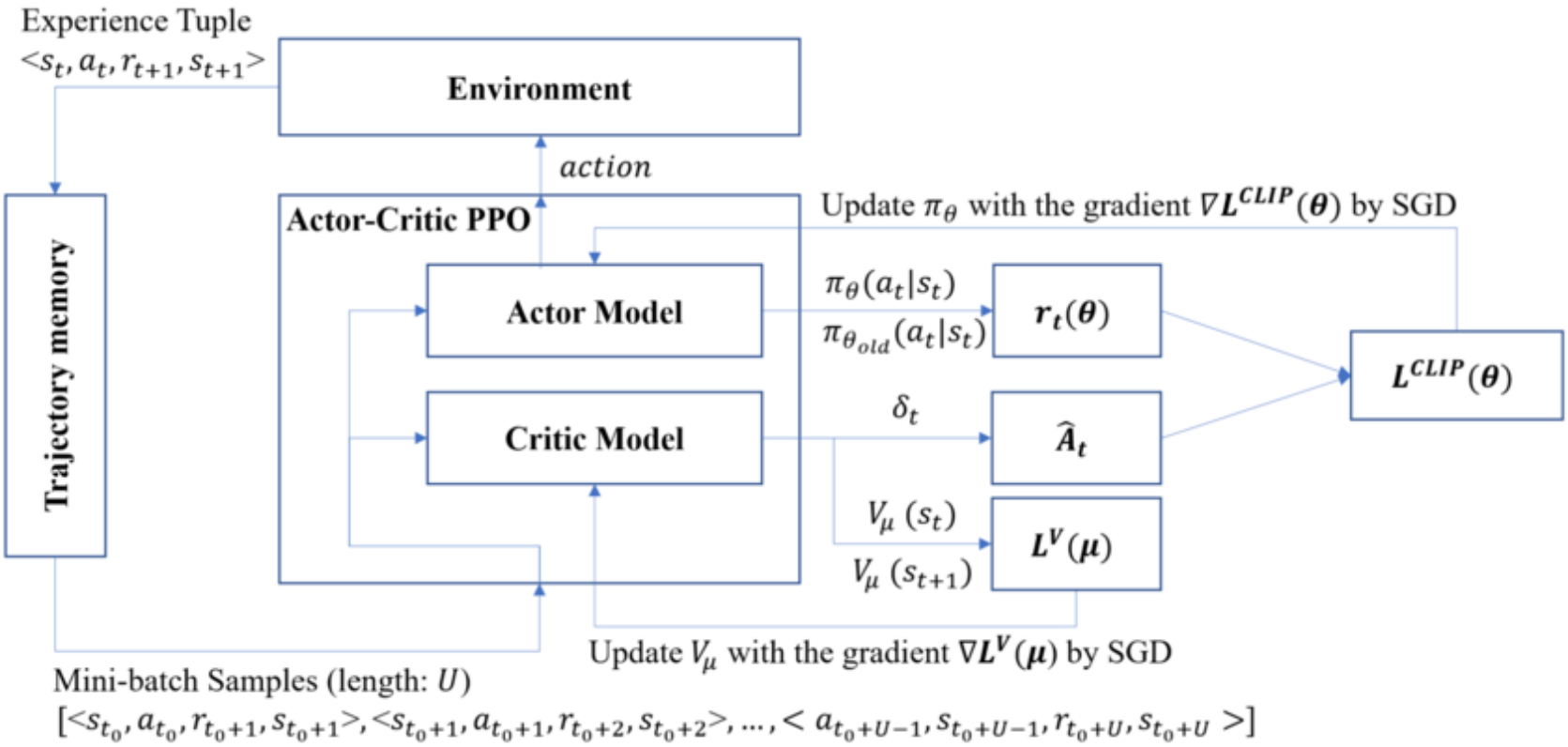
$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \mathbb{E}[Q(s',a')] - Q(s,a) \right]$$

$$\underbrace{\mathbb{E}[Q(s',a')]}_{\text{Expected value over all possible next actions}} = \sum_{a' \in \mathcal{A}} \pi(a'|s') Q(s',a')$$

Loss

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

PPO Algorithm



Thank you!