Parking

In the parking-v0 environment, the key features include the target parking spot, boundaries, and obstacles. The **target parking spot** is the designated area where the vehicle needs to park, typically represented as a rectangular region within the simulation. This area is crucial as the agent must manoeuvre the vehicle into this spot while avoiding collisions. The **boundaries** are the limits of the parking space and the surrounding area, often defined by solid lines or barriers that the vehicle should not cross. These boundaries help constrain the agent’s movement and provide a clear demarcation of the parking space. Multiple parking spots are available in the environment, and the agent must learn to navigate and park in one without colliding with the boundaries of nearby spots. Together, these features define the environment and the agent's task of parking the vehicle successfully.

A screenshot of a video game

Description automatically generated

A car navigating into a designated parking spot with key elements like the target position and trajectory.

## Methodology:

### 1.1 Algorithms:

| **Hyperparameter** | **SAC** | **TD3** |
| --- | --- | --- |
| **Policy Type** | Stochastic | Deterministic |
| **Action Noise** | Not needed | Gaussian/OU noise |
| **Learning Rate** | 3e-4 | 3e-4 |
| **Batch Size** | 256 | 256 |
| **Buffer Size** | 1e6 | 1e6 |
| **Discount Factor (γ)** | 0.99 | 0.99 |
| **Target Smoothing (τ)** | 0.005 | 0.005 |
| **Policy Delay** | Not applicable | 2 |
| **Entropy Coefficient (α)** | Automatic or fixed | Not applicable |
| **Train Frequency** | (1, "step") | (1, "step") |

Soft Actor-Critic (SAC) and Twin Delayed Deep Deterministic Policy Gradient (TD3) are two popular reinforcement learning algorithms designed for continuous action spaces, each with distinct features. SAC employs a stochastic policy, sampling actions from a probability distribution, which promotes effective exploration and is particularly suited for environments with high-dimensional or sparse-reward structures. In contrast, TD3 uses a deterministic policy, relying on explicitly added Gaussian or Ornstein-Uhlenbeck noise to facilitate exploration. Both algorithms typically use a learning rate of 3e -4, a batch size of 256, and a large replay buffer with a size of 1e6.

A key differentiator is SAC's use of an entropy coefficient (α) to balance exploration and exploitation, which can be automatically tuned or fixed manually. TD3 does not use an entropy term but implements a policy delay mechanism, updating the policy network less frequently than the critic networks (e.g., every two steps) to stabilize training. Both methods employ a target smoothing coefficient (τ) of 0.005 and a discount factor (γ) of 0.99. Training frequency for both is typically configured as one step per environment interaction.

In summary, SAC emphasizes exploration through its stochastic nature and entropy maximization, making it computationally more intensive but robust in complex scenarios. TD3, with its deterministic policy and delayed updates, provides greater efficiency and stability, particularly in tasks requiring precise control.

### 1.2 Training and testing:

* **Training**: Specify the training duration, hardware used, and any specific configurations (e.g., TensorBoard logging).
* **Testing**: Explain the evaluation process over 100 episodes and criteria for success (e.g., parking within a certain distance of the target).

## Result:

### 2.1 Quantitive Performance

**Success Rate**:

* SAC: 79% over 100 episodes
* TD3: 100% over 100 episodes

A graph of a graph of a number of different colored squares

Description automatically generated with medium confidence

**Cumulative Reward**:

A graph of a graph showing a graph

Description automatically generated with medium confidence A graph with blue lines

Description automatically generated

Compare the cumulative reward graphs for both models, highlighting trends and variability.

### 2.2 Training performance

Episode reward mean over timesteps:

A graph of a graph

Description automatically generated with medium confidence A graph with colorful lines

Description automatically generated

Success rate progression over training iterations:

A graph with colored lines

Description automatically generated A graph with different colored lines

Description automatically generated

### 2.3 Observations

**SAC**:

* Strong learning curve but less stable performance during testing.
* Lower final success rate compared to TD3.

**TD3**:

* Consistent performance during testing with a perfect success rate.
* Slightly slower initial learning curve compared to SAC.

## Discussion

**Algorithm Strengths and Weaknesses**:

* SAC: Benefits from entropy tuning but may struggle with local optima in complex tasks like parking.
* TD3: The policy delay and action noise stabilization contribute to robust performance.

**Impact of Hyperparameters**:

* Discuss the role of batch size, learning rate, and tau on training efficiency and stability.

**Environment-Specific Challenges**:

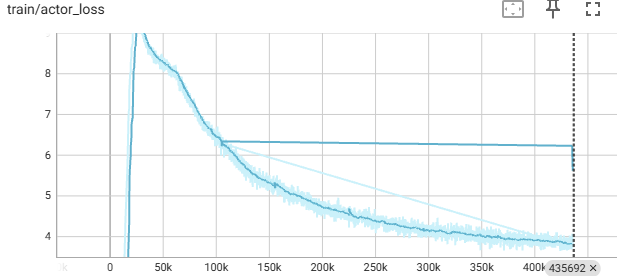
* Highlight challenges like tight parking spaces, collision avoidance, and continuous control.

A screenshot of a video game

Description automatically generated A video game with a green car in the middle of a parking lot

Description automatically generated

# Td3



A graph showing a line of blue and white

Description automatically generated with medium confidence