Comparison of Q-Learning (Tabular) vs. Deep Q-Network (DQN)

1. <u>Implementation Details</u>

Feature	Q-Learning (Tabular)	Deep Q-Network (DQN)
State Space	Discretized into 10 bins for position, velocity, pole angle, and pole angular velocity (11x11x11x11)	Continuous (4-dimensional)
Action Space	Discrete (2 actions: left and right)	Discrete (2 actions: left and right)
Learning Rate (α)	0.1	0.001
Epsilon Decay Rate	0.00001	500 (steps)
Neural Network Architecture	No Neural Network model there	Fully connected layers: 128 -> 128 -> n_actions
Target Network Update Frequency	N/A	Every 10 episodes
Scalability	Impractical for large or continuous state spaces due to the curse of dimensionality.	Handle large and continuous state spaces, making it suitable for complex environments.

2. <u>Results</u>

2.1 Learning Curves

Q-Learning (Tabular)

- Typically shows a gradual increase in average reward and a reduction in loss as episodes progress. The curve might be jagged due to the discrete state representation and learning rate adjustments.
- The discrete nature of the state space can lead to abrupt changes in the Q-values, causing the learning curve to be less smooth.
- The reward per episode graph demonstrates how the agent improves its performance over time. Loss values help in understanding how well the Q-values are being estimated.

Deep Q-Network (DQN)

- Shows a more continuous improvement in average rewards and loss. The use of neural networks usually provides smoother learning curves due to better approximation of Q-values.
- The rewards per episode and loss plots generally show a smoother and faster convergence compared to tabular methods. This is due to the efficiency of neural networks in handling continuous state spaces.
- The replay buffer helps to break the correlation between consecutive experiences, improving stability and reducing overfitting.
- The target network helps to prevent overestimation of Q-values, further enhancing stability.

2.2 Average Reward per Episode

Q-Learning (Tabular):

- The average reward per episode tends to increase gradually but may have significant fluctuations, especially in the early episodes. Performance stabilizes as the agent learns and the epsilon value decreases.
- The limited scalability of Q-Learning can result in slower convergence, especially for large or complex environments due to the discrete state space representation.

Deep Q-Network (DQN):

- The average reward per episode usually increases more rapidly and reaches higher levels due to the neural network's ability to generalize better from continuous states.
- Generally, DQN achieves higher average rewards per episode faster compared to Q-Learning.

2.3 Stability

Q-Learning (Tabular)

- The choice of discretization can significantly impact stability. If the discretization is too coarse, it may fail to capture important nuances of the environment, leading to unstable learning.
- Q-Learning's reliance on a discrete state space can make it less robust to changes in the environment or variations in state representations.

Deep Q-Network (DQN)

- Experience replay and target networks help to reduce the correlation between consecutive experiences and prevent overestimation of Q-values, respectively, improving stability.
- DQN's ability to learn a function approximate allows it to generalize better to unseen states, making it more robust to variations in the environment.

3. Discussion

Q-Learning (Tabular)

Strengths:

- Easy to understand and implement, suitable for small and discrete state spaces.
- No need for complex neural network computations.
- Q-Learning's tabular representation makes it easier to interpret and understand the learned policy.
- The Q-table directly represents the optimal action for each state, leading to a deterministic policy.

Weaknesses:

- Struggles with larger or continuous state spaces due to discretization, which can lead to suboptimal performance.
- May require many episodes to converge, especially if the state space is large or poorly discretized.

Deep Q-Network (DQN)

Strengths:

- Efficient in managing continuous state spaces and can approximate complex Q-functions using neural networks.
- Typically converges faster and achieves higher performance due to better generalization from continuous states.
- DQN's use of a neural network allows it to approximate Q-values for any state-action pair, providing flexibility and scalability.
- The neural network's output is typically a Q-value for each action, leading to a stochastic policy (e.g., ε-greedy exploration).

Weaknesses:

- More complex to implement and tune, requires managing neural network training, experience replay, and target networks.
- More computationally intensive, requiring more resources for training and longer training times.

4. Conclusion

Q-Learning (Tabular): Best suited for smaller or simpler environments where state spaces can be effectively discretized. It is straightforward but may struggle with larger or continuous state spaces.

Deep Q-Network (DQN): More appropriate for complex or continuous environments. It generally provides better performance and stability due to its ability to handle continuous state spaces and advanced techniques like experience replay.

While Q-Learning is a valuable algorithm for learning in discrete spaces, DQN offers superior performance and stability for more complex environments, making it a preferred choice for modern reinforcement learning tasks.