1.1 TD-Learning Bias (2 points)

We say an estimator $f_{\mathcal{D}}$ of f constructed using data $\mathcal{D}$ sampled from process P is *unbiased* when $\mathbb{E}_{\mathcal{D}\sim P}[f_{\mathcal{D}}(x) - f(x)] = 0$ at each x.

Assume $\hat{Q}$ is a noisy (but unbiased) estimate for Q. Is the Bellman backup $\mathcal{B}\hat{Q} = r(s,a) + \gamma \max_{a'} \hat{Q}(s',a')$ an unbiased estimate of $\mathcal{B}Q$?

☐ Yes
☐ No

NO

E[r(s, a)] = r(s, a) so r(s, a) doesn't introduce bias
check if E[BQ^] = BQ. E[max Q^] != max Q since Q^ is a noisy estimate

Indicate which of the statements I-III always hold in the following cases. No justification is required.

	I.	II.	III.
1. $N = 1$ and ...			
(a) on-policy in tabular setting	☐	☐	☐
(b) off-policy in tabular setting	☐	☐	☐
2. $N > 1$ and ...			
(a) on-policy in tabular setting	☐	☐	☐
(b) off-policy in tabular setting	☐	☐	☐
3. In the limit as $N \to \infty$ (no bootstrapping) ...			
(a) on-policy in tabular setting	☐	☐	☐
(b) off-policy in tabular setting	☐	☐	☐

1.3 Variance of Q Estimate (2 points)

Which of the three cases ($N = 1$, $N > 1$, $N \to \infty$) would you expect to have the highest-variance estimate of Q for fixed dataset size B in the limit of infinite iterations k? Lowest-variance?

Highest variance:
- ☐ $N = 1$
- ☐ $N > 1$
- ☐ $N \to \infty$

Lowest variance:
- ☐ $N = 1$
- ☐ $N > 1$
- ☐ $N \to \infty$

Figure 1: Enter Caption

Figure 1: Enter Caption

1.4 Function Approximation (2 points)

Now say we want to represent Q via function approximation rather than with a tabular representation. Assume that for any deterministic policy π (including the optimal policy π^*), function approximation can represent the true Q^π exactly. Which of the following statements are true?

- ☐ When $N = 1$, $Q_{\phi_{k+1}}$ is an unbiased estimate of the Q-function of the last policy Q^{π_k}.
- ☐ When $N = 1$ and in the limit as $B \to \infty$, $k \to \infty$, Q_{ϕ_k} converges to Q^*.
- ☐ When $N > 1$ (but finite) and in the limit as $B \to \infty$, $k \to \infty$, Q_{ϕ_k} converges to Q^*.
- ☐ When $N \to \infty$ and in the limit as $B \to \infty$, $k \to \infty$, Q_{ϕ_k} converges to Q^*.

Figure 2: Enter Caption

Figure 2: Enter Caption

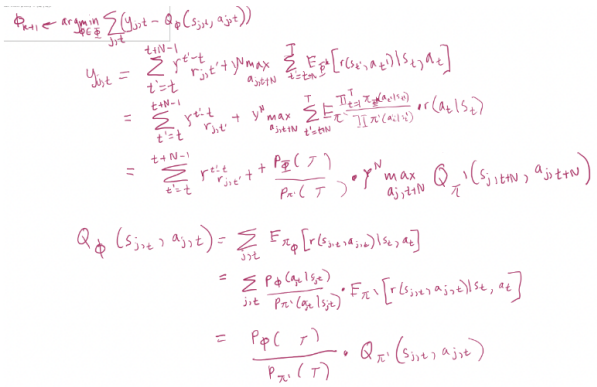

Figure 3: 1.5

Figure 3: Enter Caption

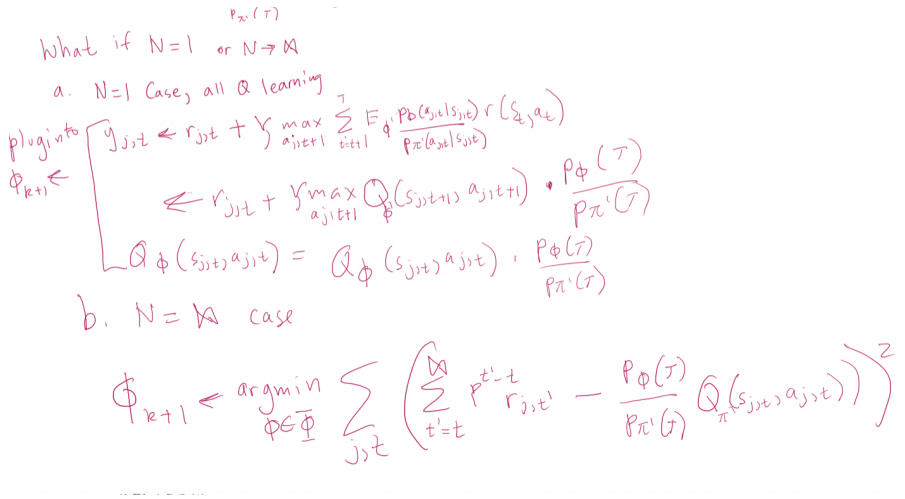

Figure 4: Enter Caption

1 2.4

```
rm -rf ./data/*
python cs285/scripts/run_hw3_dqn.py -cfg experiments/dqn/cartpole.yaml
python cs285/scripts/graph_results.py
```

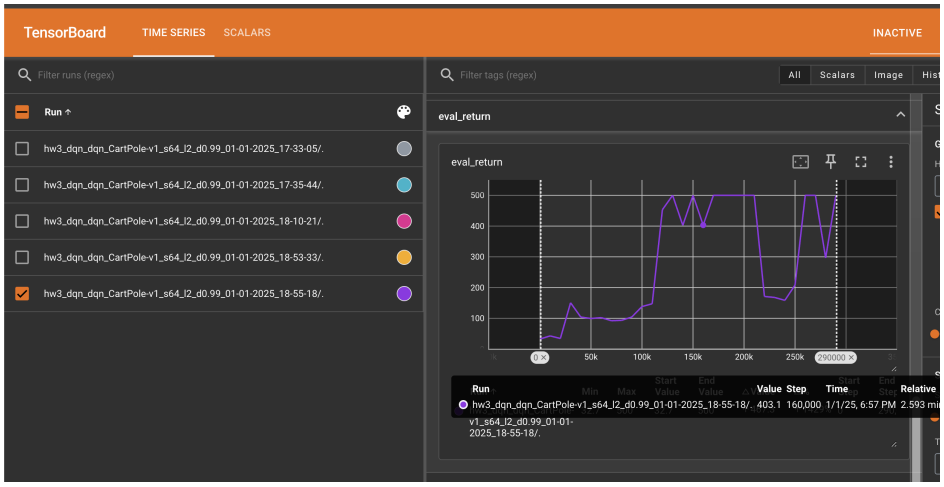

Figure 5: Enter Caption

Note: Purple is seed 3, blue is seed 2, grey is seed 1.

Figure 6: Enter Caption

Change the learning rate to 0.05 in the YAML config file. Analyze the impact on:

1. **Predicted Q-values:** Predicted Q-values increased but became more volatile.
2. **Critic error:** Critic error increased.

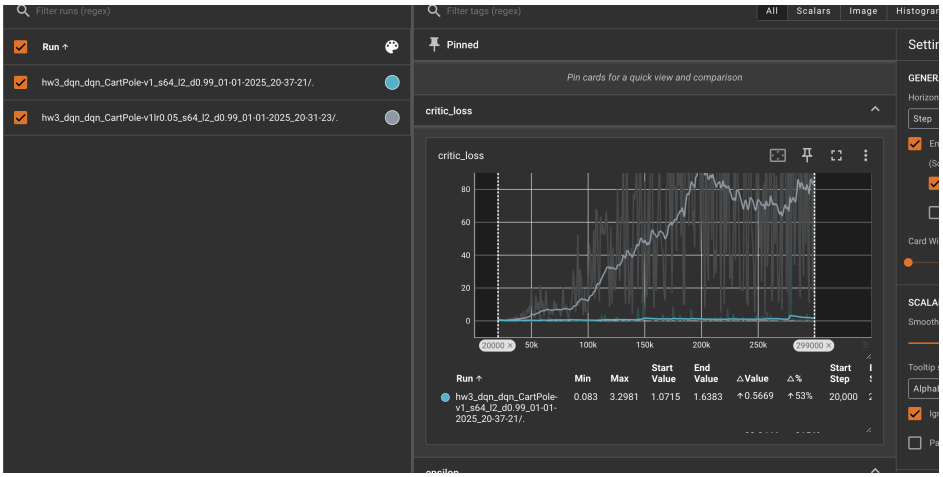

Figure 7: Increased learning rate and critic loss

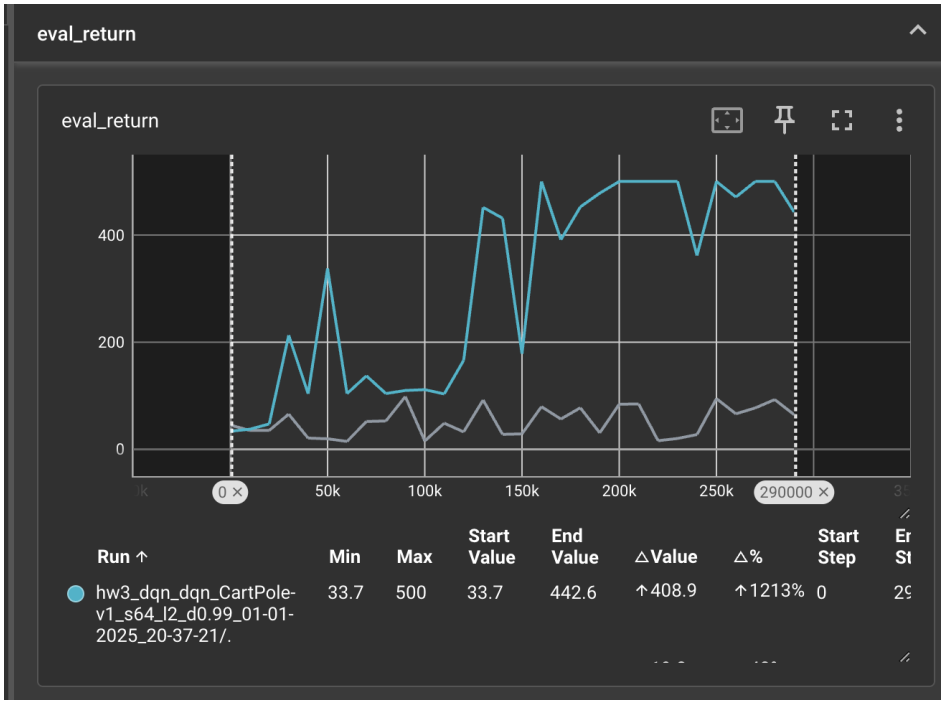

Figure 8: $lr = 0.05$ and eval return

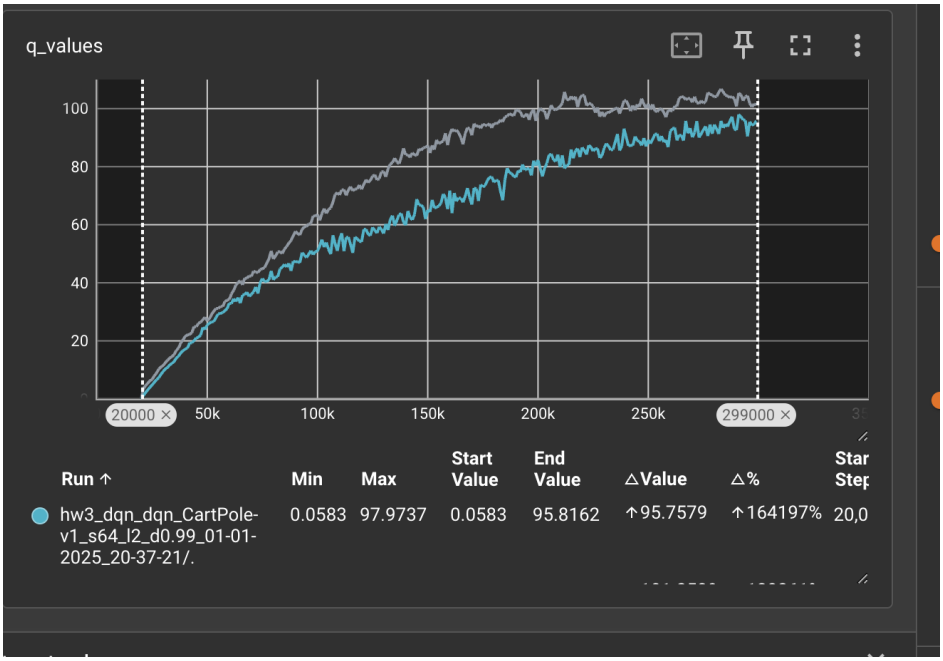

Figure 9: $lr = 0.05$ and q values

Explanation: The predicted Q-values increased because the higher learning rate led to more aggressive updates via the gradient function, trending toward favoring higher future rewards. However, the increased volatility caused higher variance in the predicted Q-values.

Since the Q-values became noisier, the Bellman error also increased. This is because the Bellman error depends on the estimation of future rewards using the Q-function for the target.

Higher variance in Q-values contributes to higher variance in the Bellman error. A high learning rate causes the neural network to overshoot the optimal values of the target critic Q-values, leading to increased critic error.

2 2.5

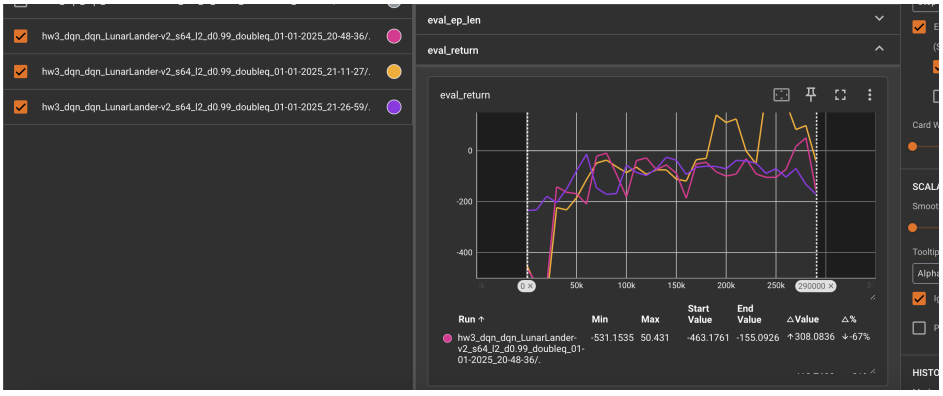

Figure 10: DQN and eval return

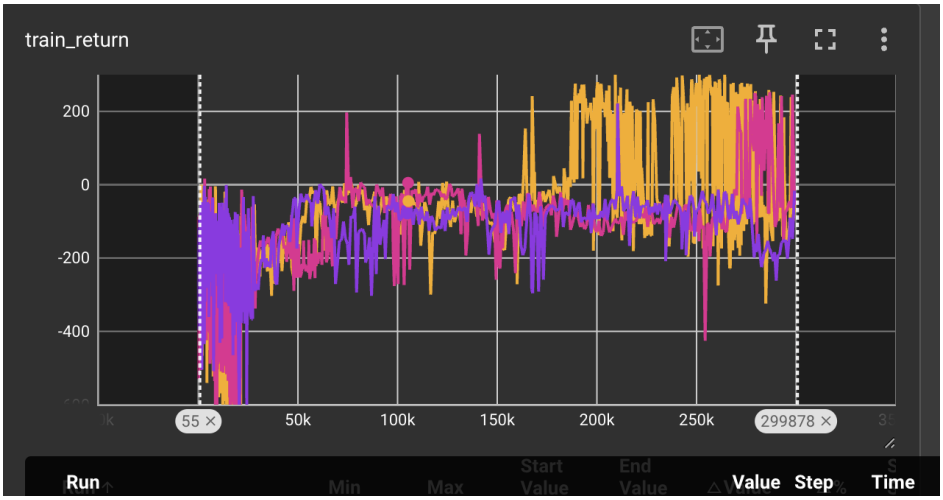

Figure 11: $lr = 0.05$ and q values

Here the double q learning seems to take longer to reach eval return around 200 compared to the vanilla approach since it could be possible that our two critics take longer to be trained to predict Q values compared to only needing to train one critic for both choosing actions and evaluating the q value for the selected action.

3 2.6

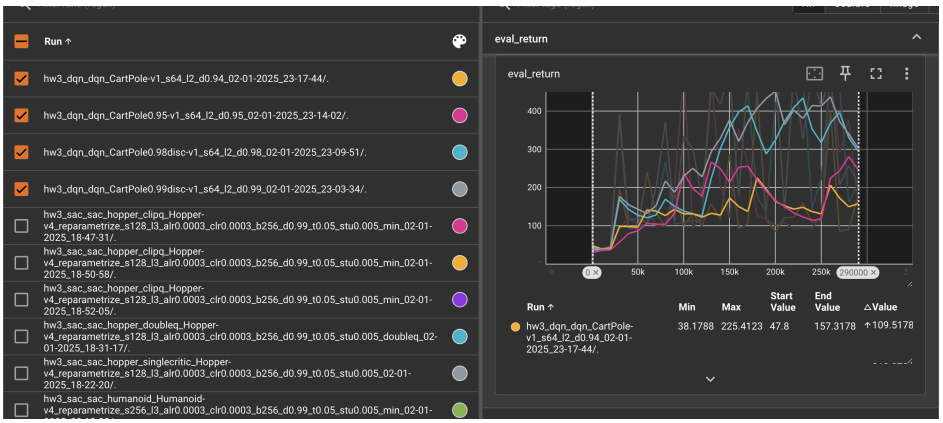

Figure 12: Discount factor vs eval return for cartpole

Impact of Discount Factor

The discount factor determines the extent to which the DQN agent prioritizes future rewards over present rewards, i.e., it affects the Q-value. A higher discount factor results in a higher evaluation return, as the DQN agent assigns more importance to future rewards, weighing them almost equally with current rewards.

By weighing future rewards higher, Q-value estimations align more closely with the target values, improving performance in tasks where long-term outcomes are critical. This leads to better policy learning that balances immediate and delayed rewards effectively.

4 3.1.1

```
python cs285/scripts/run_hw3_sac.py -cfg experiments/sac/sanity_pendulum.yaml
```

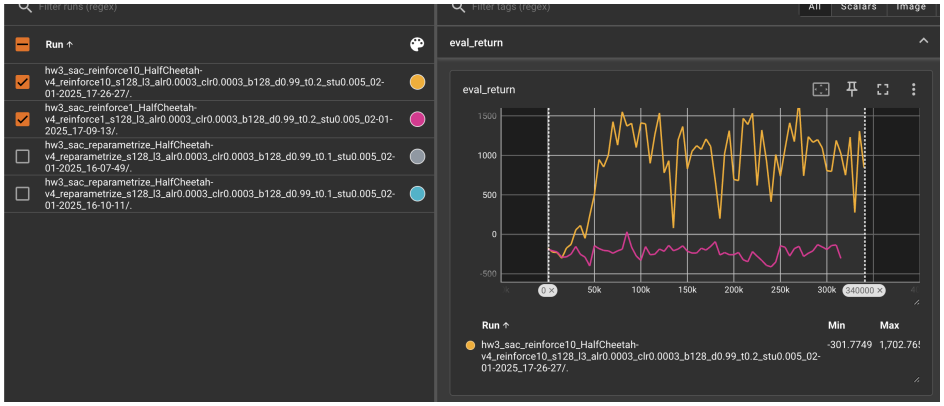

Figure 13: $lr = 0.05$ and q values

5 3.1.3

Figure 14: $lr = 0.05$ and q values

Return is significantly higher when using `Reinforce10` (yellow graph). Using only one sample to train the gradient introduces bias toward the training sample, causing the Q-values to deviate further from their expected values.

6 3.1.4

```
python cs285/scripts/run_hw3_sac.py -cfg experiments/sac/halfcheetah_reparametrize.yaml
python cs285/scripts/run_hw3_sac.py -cfg experiments/sac/sanity_invertedpendulum_reinforce.yaml
python cs285/scripts/run_hw3_sac.py -cfg experiments/sac/halfcheetah_reinforce1.yaml
```

Figure 15: $lr = 0.05$ and q values