Experiment Results of SAC

Jingyi Zhao

New York University Shanghai

Introduction

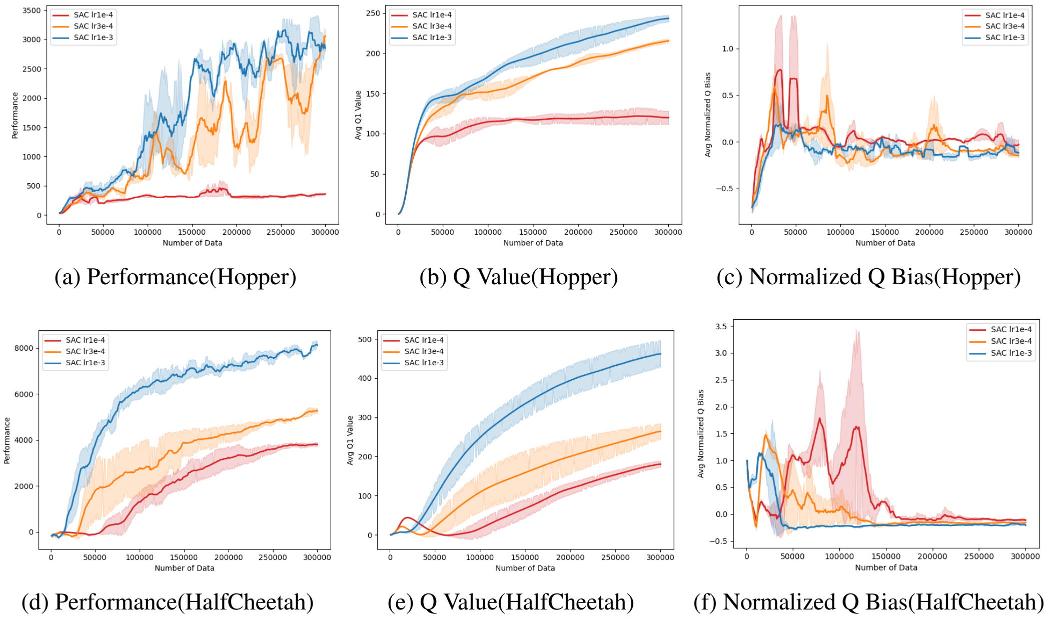
In this report, we study the experiment results of Soft Actor-Critic (SAC) in two MuJoCo environments, namely, Hopper and HalfCheetah. We study the influence of different learning rates (lr), discount values (γ), polyak values (ρ), update-to-date ratios (UTD) and midway network reset. The results are compared in three aspects: average test expected return (performance), average Q_1 value (Q value), and average normalized Q bias (normalized Q bias). For each variant, we train the model for 300 epochs and show the result of 2 independent trials.

In section 1, 2, and 3 we study the influence of learning rate, discount value, and polyak value respectively. In section 4, we study the influence of UTD and modify the training phase by introducing midway reset of network parameters.

1 Study of Learning Rate

Figure 1 shows the experiment results of different choices of learning rates. In all the cases, we fix $\gamma = 0.99$, $\rho = 0.995$, $UTD = 1$ and use no midway resets. As we can see from the figure, the variant with $lr = 0.001$ has the best performance, giving the highest average return and Q value as well as maintaining the lowest bias, while the ones with $lr = 0.0003$ and $lr = 0.0001$ are not so good. The reason is that given limited training epochs, larger learning rate enables faster learning.

Figure 1: Experiment Results of Learning Rate Study (fix $\gamma = 0.99$ and $\rho = 0.995$)



(a) Performance(Hopper)

(b) Q Value(Hopper)

(c) Normalized Q Bias(Hopper)

(d) Performance(HalfCheetah)

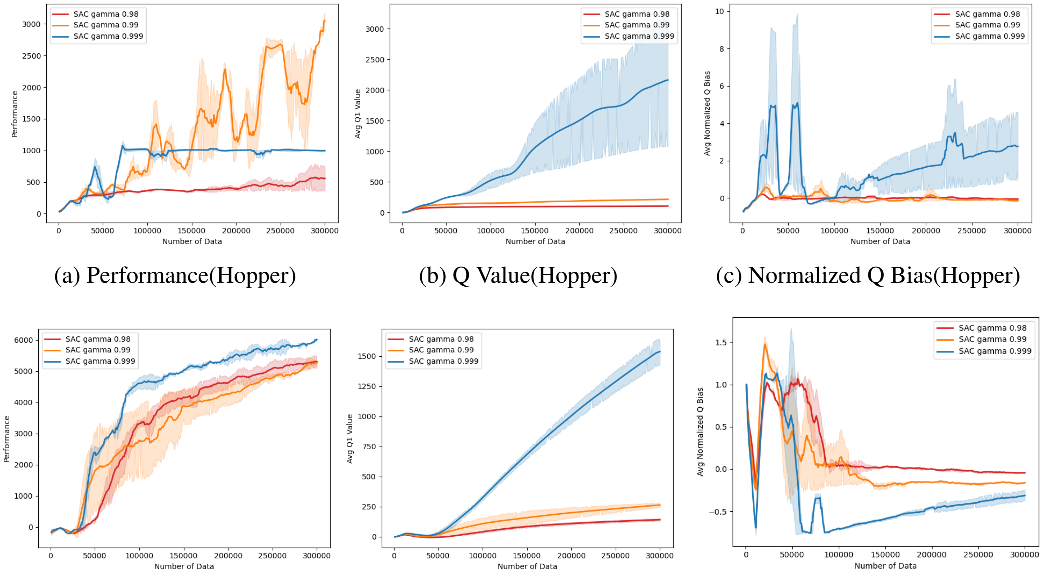
(e) Q Value(HalfCheetah)

(f) Normalized Q Bias(HalfCheetah)

2 Study of Discount Value

Figure 2 shows the experiment results under difference choices of discount values, namely, $\gamma = 0.98, 0.99, 0.999$. In the HalfCheetah environment, the variant with $\gamma = 0.999$ appears to be the best model, giving the highest performance and Q value and the lowest normalized Q bias. However, in the Hopper environment, though the variant with $\gamma = 0.999$ has the highest Q value, it fails to perform well in terms of performance and normalized Q bias. This shows that this model fails to learn well. In fact, our model is learning towards higher Q values, but higher Q values not necessarily lead to higher performance. Since the discount value γ indicates how much we care about previous rewards when calculating the return, the unsatisfactory performance is probably due to overmuch involvement of previous rewards.

Figure 2: Experiment Results of Discount Value Study (fix $lr = 0.0003$ and $\rho = 0.995$)



(a) Performance(Hopper)

(b) Q Value(Hopper)

(c) Normalized Q Bias(Hopper)

(d) Performance(HalfCheetah)

(e) Q Value(HalfCheetah)

(f) Normalized Q Bias(HalfCheetah)

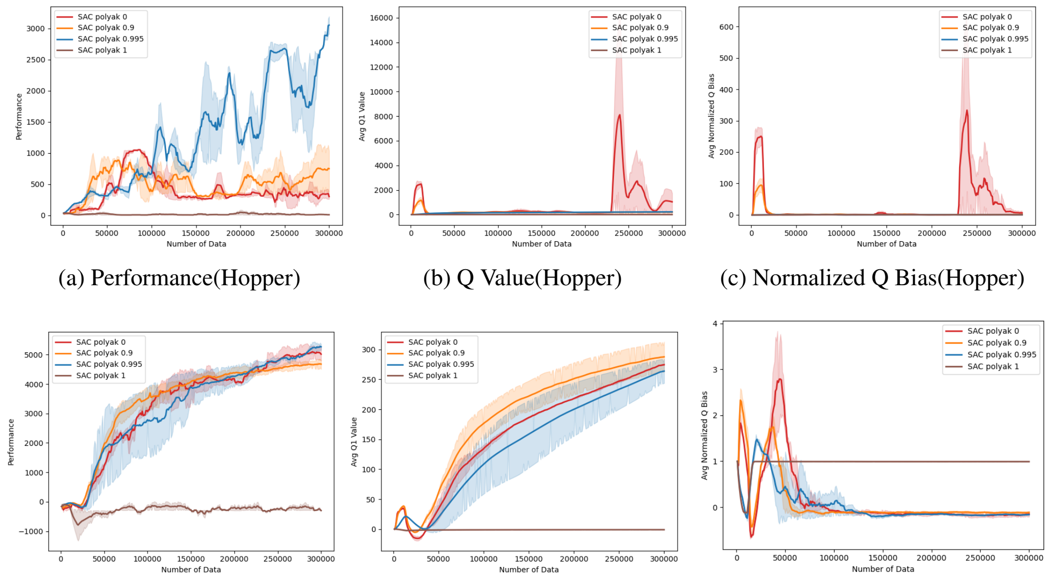
3 Study of Polyak Value

Figure 3 shows the experiment results under different choices of polyak values, namely, $\rho = 0, 0.9, 0.995, 1$. Let's first consider the case when $\rho = 0$ which means we are not employing a target network. This variant sometimes has extremely high Q value and Q bias, and the performance is not very good. The reason is that without separate target networks, the target depends on the same parameters we used to train the Q network, which makes the error minimization stage unstable. Hence, we may sometimes obtain extremely high Q values but result in noneffective learning. For clearer illustration, we then plot the results without considering this variant (i.e.,when $\rho = 0$).

Figure 4 shows the experiment results under polyak value $\rho = 0.9, 0.995, 1$. When $\rho = 1$, which means that the target network is not updated, and we are learning so that our Q function approaches the target Q function which is randomly initialized. Apparently, this training is meaningless. For $\rho = 0.9$ and $\rho = 0.995$, The variant with $\rho = 0.9$ performs better at first, but the one with $\rho = 0.995$ outperforms it as time goes on.

In our algorithm, the target network is updated by $\phi_{tag,i} = \rho\phi_{tag,i} + (1-\rho)\phi_i$. Hence, a lower polyak value means faster change of Q target network. From figure 4, we can see that the variant with $\rho = 0.995$ outperforms that with $\rho = 0.9$, which suggests that slower change of target network leads to a better performance. Such phenomenon is reasonable because time delay in updating the target network reduces the unstability introduced by the involvement of parameters to be trained.

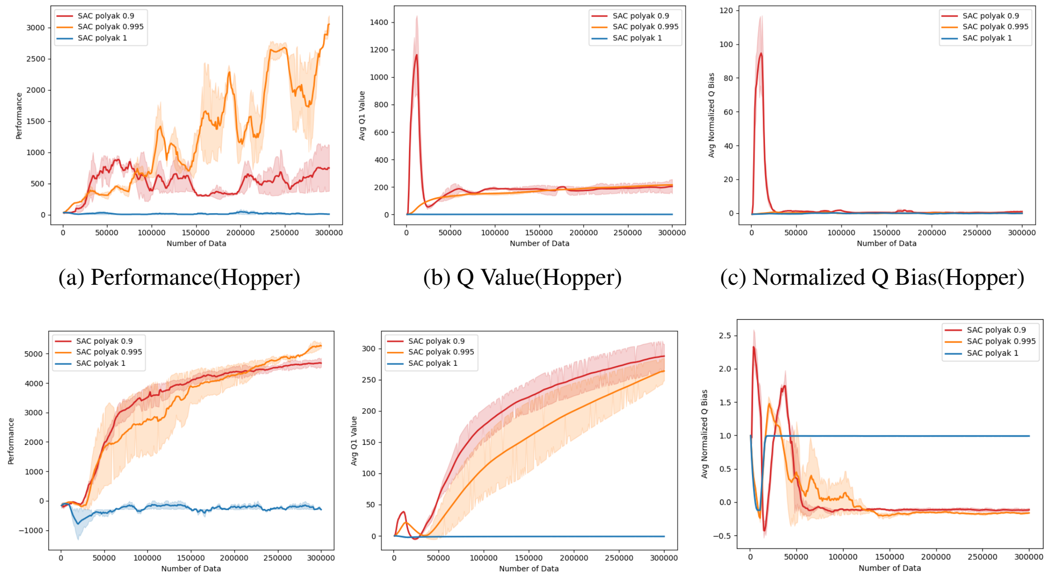
Figure 3: Experiment Results of Polyak Value Study (fix $lr = 0.0003$ and $\gamma = 0.99$)



(a) Performance(Hopper) (b) Q Value(Hopper) (c) Normalized Q Bias(Hopper)

(d) Performance(HalfCheetah) (e) Q Value(HalfCheetah) (f) Normalized Q Bias(HalfCheetah)

Figure 4: Experiment Results of Polyak Value Study (Cont.) (choosing $lr = 0.0003$ and $\gamma = 0.99$)



(a) Performance(Hopper) (b) Q Value(Hopper) (c) Normalized Q Bias(Hopper)

(d) Performance(HalfCheetah) (e) Q Value(HalfCheetah) (f) Normalized Q Bias(HalfCheetah)

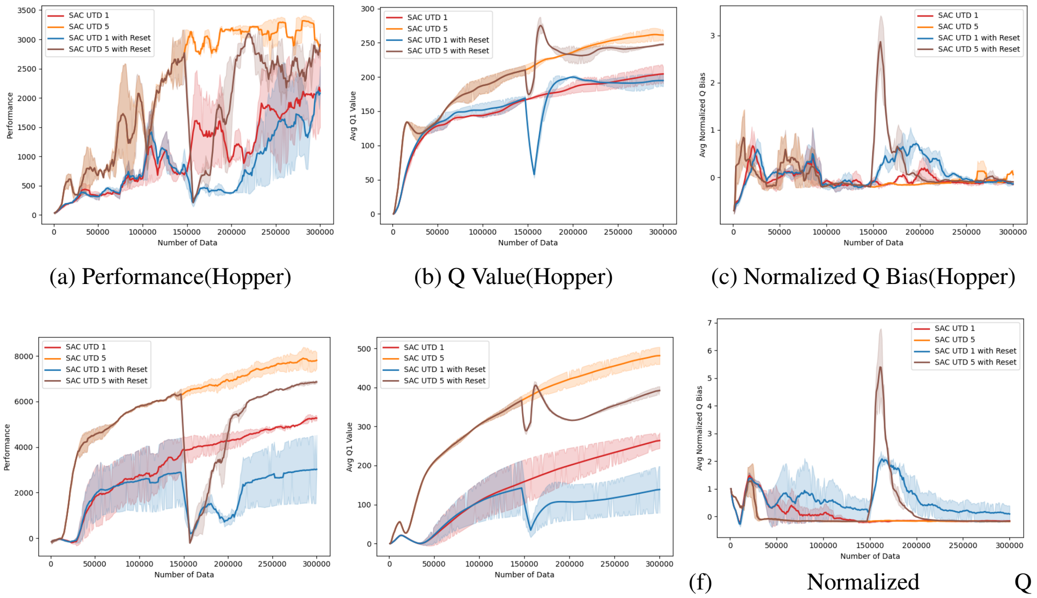
4 Study of UTD Ratio and Network Reset

Figure 5 shows the experiment results under different choices of UTD, namely, $UTD = 1, 5$, with and without midway reset. Note that the midway reset here means that we reinitialize the network parameters for Q and policy networks after 150 epochs (we train for 300 epochs in total).

In every case, the variants with $UTD = 5$ largely outperform those with $UTD = 1$ regarding performance and Q value, which suggests that the improvement in sample efficiency largely improves the performance. In terms of Q bias, the variants with larger UTD values performs at least as good as those with lower UTD values.

After we introducing the midway reset, we observe great drop in performance and Q value and great increase in Q bias immediately when the network is reset. Nonetheless, the performances quickly restore back to where the model was before the network reset. In general, reset models seem to have lower performance and Q values after the midpoint than those without reset, and they have relatively higher Q bias. However, we may notice that we're here comparing the results of training epoch 150 to 300, which means the reset models have only trained for 150 epochs (after reset) which is half the time the non-reset ones have trained. Hence, with the given number of training epochs, we may not directly draw conclusion that the reset models are performing worse. Meanwhile, if we consider the experiment results of reset model for 150-300 epochs and that between 0-150 epochs, we find that the model perform better in 150-300 epochs than in 0-150 epochs. The reason is that although we're resetting the network parameters, we don't clean the replay buffer, and our reset models start training on a existing dataset collected in the first 150 epochs.

Figure 5: Experiment Results of UTD Study (fix $lr = 0.0003$, $\gamma = 0.99$, and $p = 0.995$)



(a) Performance(Hopper)

(b) Q Value(Hopper)

(c) Normalized Q Bias(Hopper)

(d) Performance(HalfCheetah)

(e) Q Value(HalfCheetah)

(f) Normalized Q Bias(HalfCheetah)