# Adaptive Estimation Q-learning with Uncertainty and Familiarity

**Xiaoyu Gong**[1,2], **Shuai Lü**[1,2,3,*], **Jiayu Yu**[1,2], **Sheng Zhu**[1,3] and **Zongze Li**[1,2]
[1]Key Laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Ministry of Education, China
[2]College of Computer Science and Technology, Jilin University, China
[3]College of Software, Jilin University, China
lus@jlu.edu.cn, {gongxy20, yujy19, zhusheng20, zzli20}@mails.jlu.edu.cn

## Abstract

One of the key problems in model-free deep reinforcement learning is how to obtain more accurate value estimations. Current most widely-used off-policy algorithms suffer from over- or underestimation bias which may lead to unstable policy. In this paper, we propose a novel method, Adaptive Estimation Q-learning (AEQ), which uses uncertainty and familiarity to control the value estimation naturally and can adaptively change for specific state-action pair. We theoretically prove the property of our familiarity term which can even keep the expected estimation bias approximate to 0, and experimentally demonstrate our dynamic estimation can improve the performance and prevent the bias continuously increasing. We evaluate AEQ on several continuous control tasks, outperforming state-of-the-art performance. Moreover, AEQ is simple to implement and can be applied in any off-policy actor-critic algorithm.

## 1 Introduction

Off-policy deep reinforcement learning algorithm is widely-used in continuous control tasks. Recent off-policy methods typically utilize actor-critic framework to pursue sampling efficiency, including Deep Deterministic Policy Gradient (DDPG) [Lillicrap *et al.*, 2015], Twin Delayed Deep Deterministic Policy Gradient (TD3) [Fujimoto *et al.*, 2018] and Soft Actor Critic (SAC) [Haarnoja *et al.*, 2018], etc. However, these successful methods usually fail in Q-value estimation. Q-value is essential for reinforcement learning, and it estimates how good a state-action pair is. Moreover, the policy network is trained by directly maximizing the expected Q-value commonly. Therefore, accurate Q-value estimation is critical to training stability and the final performance.

The overestimate bias problem has been widely studied. van Hasselt et al. [van Hasselt *et al.*, 2016] reveal that single Q function estimator may lead to overestimation problem and propose Double DQN algorithm to alleviate it. It introduces another Q network to decouple action selection and value estimation. DDPG follows the similar target of Q function as Double DQN and uses noised deterministic policy gradient with the actor-critic framework to solve continuous control tasks, but unfortunately, DDPG still has overestimation problem. TD3 [Fujimoto *et al.*, 2018] further reduces the overestimation by taking the minimum value over two separate Q-value estimators, but this leads to underestimation issue [Ciosek *et al.*, 2019]. Inspired by TD3, many methods can further address this issue by using other operators including max [van Hasselt *et al.*, 2016], average [Anschel *et al.*, 2017], and softmax [Pan *et al.*, 2020], etc., or ensemble estimator [Agarwal *et al.*, 2020; Chen *et al.*, 2021; Lan *et al.*, 2020].

Recent methods can even maintain the estimation bias within a small range for most of the training time [Chen *et al.*, 2021]. It seems the over- or underestimation bias has been well-studied, and current methods all struggle to get an accurate estimation which is in fact impossible theoretically [Thrun and Schwartz, 1993]. However, none of these methods focuses on how to estimate specific state-action pair properly to improve performance. Both over- and underestimation bias may improve learning performance, depending on the different state or situation [Lan *et al.*, 2020]. In some cases, overestimation can help policy to be more optimistic to explore the high-value regions, and underestimation can prevent the policy from going into risky regions [Ciosek *et al.*, 2019].

In this paper, we propose a novel method called Adaptive Estimation Q-learning (AEQ). Based on a relatively accurate Q-value estimation, we dynamically control Q-values through uncertainty and familiarity to over- or underestimate a specific state-action pair relatively. Uncertainty gives the epistemic uncertainty of state-action pairs, which can naturally serve as an upper or lower bound for ensemble Q-learning. Therefore, it keeps that the estimations of Q-values are close to the real Q-value in AEQ. Familiarity measures the potential novelty of state-action pairs and identifies experiences that may have higher returns. If the familiarity of an experience is low, the novelty of this experience is likely to be high. When low familiarity encounters with worse action, the uncertainty will give a penalty firstly. We can also consider it an optimistic estimate if uncertainty does not work either. Besides, familiarity can dynamically change with the sampling of experiences and the training process of learning, so it can control the estimation of Q-value to be overestimated or

*Corresponding Author

underestimated relatively when combined with uncertainty.

We integrate AEQ to TD3, and evaluate it on a series of continuous control tasks from OpenAI Gym [Brockman *et al.*, 2016]. The results show that AEQ-TD3 outperforms the current state-of-the-art algorithms without tuning environment-specific hyperparameters. Further, we apply AEQ to SAC and set the Update-To-Data (UTD) ratio to 20 as REDQ [Chen *et al.*, 2021], the experiments suggest that AEQ-SAC can exceed REDQ. We also conduct ablations to show our adaptive estimation is effective and robust. To ensure that our results are convincing and reproducible, we will open-source the code.

To sum up, our main contributions are as follow:

- We propose Adaptive Estimation Q-learning, which is the first method that can dynamically control Q-values through uncertainty and familiarity to over- or underestimate a specific state-action pair.
- We prove the property of our familiarity term and the role it plays in controlling the bias.
- We show AEQ is sample efficient and outperforms the state-of-the-art algorithms.
- We demonstrate that AEQ is simple to implement, and is general which can be applied to any off-policy Q-learning algorithm.

## 2 Related Work

**Estimation bias in Q-learning.** Thrun & Schwartz [1993] first investigate and propose the problem of estimation bias in Q-learning. Double Q-learning [Hasselt, 2010] uses two estimators to solve the overestimation issue, and Double DQN [van Hasselt *et al.*, 2016] applies this approach to DQN. TD3 [Fujimoto *et al.*, 2018] and SAC [Haarnoja *et al.*, 2018] improve the performance of DDPG [Lillicrap *et al.*, 2015] significantly by using clipped double Q-learning in continuous action space. Subsequently, some methods [Zhang *et al.*, 2017; Li and Hou, 2019] weight the minimum and maximum estimations of Q-value. SD3 [Pan *et al.*, 2020] applies softmax operator in updating Q-value to help reducing estimation bias. Recently, many works use ensemble to further reduce estimation bias in order to improve the performance. Averaged-DQN [Anschel *et al.*, 2017] uses the average of multiple Q-value estimations to reduce variance. REM [Agarwal *et al.*, 2020] also uses ensemble Q-value estimations but combines with random convex to enhance generalization in the offline setting. Maxmin Q-learning [Lan *et al.*, 2020] controls over- and underestimation by adjusting the number of Q-value estimators. REDQ [Chen *et al.*, 2021] reduces the variance of estimation bias through minimizing a random subset of multiple Q-value estimations, and uses a high UTD ratio to improve performance. Similar to the ensemble, distributional representation [Kuznetsov *et al.*, 2020; Duan *et al.*, 2021] is another way to address this issue.

**Uncertainty with ensemble.** Uncertainty estimation has been widely used in reinforcement learning when combined with the ensemble. Bootstrapped DQN [Osband *et al.*, 2016] utilizes ensemble of Q-value estimator to estimate the uncertainty of Q-value to improve exploration. Multiple Q-value estimations can also enhance exploration by applying the principle of optimism in the face of uncertainty [Ciosek *et al.*, 2019; Chen *et al.*, 2017]. SUNRISE [Lee *et al.*, 2021] uses uncertainty to get the upper confidence bound (UCB) of Q-values to choose action. EDAC [An *et al.*, 2021] leverages uncertainty with diversified Q-ensemble to penalize out-of-distribution data points.

**Novelty exploration.** Most exploration strategies try to approximate the novelty of the visited state in different ways. Count-based methods [Bellemare *et al.*, 2016; Tang *et al.*, 2017] count how many times a state has been encountered probably to generate intrinsic rewards. In this paper, we use a simpler count-based technique to get the familiarity of a state-action pair to adjust the estimation of Q-value. Dynamics model [Pathak *et al.*, 2019] and random network [Burda *et al.*, 2019] can also be used to predict whether similar states have been visited. Recently, some works [Conti *et al.*, 2018; Cideron *et al.*, 2020] of evolutionary reinforcement learning also use the Quality-Diversity algorithms to deal with the exploration-exploitation trade-off.

## 3 Preliminaries

The standard reinforcement problem can be considered as a Markov decision process (MDP), defined as $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma \rangle$, with the state and action space $\mathcal{S}$ and $\mathcal{A}$, the reward function $r$, the transition probability $\mathcal{P}$, and the discount factor $\gamma \in (0, 1]$. The goal of reinforcement learning is to find the optimal policy $\pi^*$ to maximize the expected discounted return $\mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$.

DDPG [Lillicrap *et al.*, 2015] is a widely-used off-policy algorithm based on actor-critic framework. It learns a deterministic policy $\pi_\phi(s)$ which is as effective as stochastic policy in continuous action space. The actor parameter $\phi$ can be learned using Eq.(1).

$$\nabla_\pi J(\phi) = \mathbb{E}_{\mathcal{B},\pi}\left[\nabla_a Q_\theta(s,a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)\right] \quad (1)$$

where $\mathcal{B}$ is the replay buffer. The critic parameter $\theta$ can be learned by minimizing the

$$J(\theta) = \mathbb{E}_{\mathcal{B},\pi}\left[(y - Q_\theta(s,a))^2\right] \quad (2)$$

where $y = r + \gamma Q'_{\theta'}(s', \pi_{\phi'}(s'))$ is the target value. TD3 [Fujimoto *et al.*, 2018] is an improved algorithm of DDPG, it uses clipped double Q-learning with two independent critics to obtain target value $y = r + \gamma \min_{i=1,2} Q'_{\theta'_i}(s', \pi_{\phi'}(s') + \epsilon)$, but still directly applies the mean squared error to optimize like DDPG.

## 4 Adaptive Estimation Q-learning

In this section, we will first introduce the problem of estimation bias in Q-learning, followed by a method using uncertainty to address the estimation bias with ensemble Q-learning. Finally, we present our Adaptive Estimation Q-learning (AEQ) which uses familiarity and uncertainty to obtain an adaptive estimation, and show how to apply AEQ to modern off-policy RL algorithms in practice.

### 4.1 Estimation Bias Problem

Using neural networks as function approximators to estimate the Q function will lead to unavoidable bias due to inaccuracy of the network [Thrun and Schwartz, 1993]. The study shows the $\max$ operator will exaggerate this bias, and this bias will also be accumulated and propagated through temporal difference learning, eventually leads to overestimation [Thrun and Schwartz, 1993; Hasselt, 2010].

Here, we use $Q^\pi(s,a)$ to denote ground truth of Q-functions and assume each $Q_e^i(s,a)$ has a random approximation error $e_{sa}^i$, where each $e_{sa}^i$ is identically distributed for each fixed $(s,a)$ pair [Thrun and Schwartz, 1993].

$$Q_e^i(s,a) = Q^\pi(s,a) + e_{sa}^i \tag{3}$$

Then, we can define the general updated estimation bias $Z_N$ with $N$ estimators for each fixed $(s',a')$ pair as follows:

$$\begin{aligned} Z_N &\triangleq r(s,a) + \gamma f_{op}\left\{Q_e^i(s',a')\right\}_{i=1}^N \\ &\quad - \left(r(s,a) + \gamma Q^\pi(s',a')\right) \\ &= \gamma\big(f_{op}\left\{Q_e^i(s',a')\right\}_{i=1}^N - Q^\pi(s',a')\big) \end{aligned} \tag{4}$$

where $f_{op}$ is the operator that decides how to combine these $N$ Q-values. Under the zero-mean assumption, the expected estimation bias of $Q$ is $\mathbb{E}\left[Q_e^i(s,a) - Q^\pi(s,a)\right] = 0$ [Lan *et al.*, 2020; Thrun and Schwartz, 1993]. Therefore, if $\mathbb{E}[Z_N] > 0$, the Q-value will have a tendency of overestimation; and if $\mathbb{E}[Z_N] < 0$, the Q-value will have a tendency of underestimation.

If $N = 2$ and $f_{op}$ is specific to $\min_{i=1,2}\max_{a'\in\mathcal{A}}$, Eq.(4) can denote the updated estimation bias of clipped double Q-learning. Because

$$\begin{aligned} &\mathbb{E}\left[\min_{i=1,2}\max_{a'\in\mathcal{A}} Q_e^i(s',a')\right] \\ &= \mathbb{E}\left[\min_{i=1,2}\max_{a'\in\mathcal{A}}\left(Q^\pi(s',a') + e_{s'a'}^i\right)\right] \\ &< \mathbb{E}\left[\max_{a'\in\mathcal{A}} Q^\pi(s',a')\right] \end{aligned} \tag{5}$$

$\mathbb{E}[Z_2] < 0$, which explains why TD3 and SAC will have an underestimation tendency.

### 4.2 Ensemble Q-learning with Uncertainty

In the following, we present our AEQ method. AEQ uses multiple estimators of Q functions like Maxmin Q-learning [Lan *et al.*, 2020] and REDQ [Chen *et al.*, 2021], but we use uncertainty and familiarity to get a more accurate and reasonable target.

First, we present how to use uncertainty to penalize the target of ensemble Q-learning which is similar to RAC [Li *et al.*, 2021], and we analyze its motivation and weakness. As mentioned before, the estimation of Q-value is usually biased when using an approximator and the $\max$ operator. It is well known that the overestimation is more harmful, so we need to add a penalty term to Q-value estimation, like $\min$. We assume the estimations of multiple Q-value estimators are obeying the Gaussian distribution for a fixed state-action pair. If we use $\min$ to penalize multiple Q-value estimations, it is equal to taking the lower bound of the Gaussian distribution as the estimation, which will lead to potential underestimation ($\mathbb{E}[Z_N] < 0$). In contrast, if we use the mean of multiple Q-values Eq.(6) as the estimation, we will have a potential risk of overestimation, so we can use the standard deviation of the Gaussian distribution Eq.(7) to measure the uncertainty, and this uncertainty can naturally become the penalty term.

$$\bar{Q}_{\theta'}(s',a') = \frac{1}{N}\sum_{i=1}^N Q_{\theta'}^i(s',a') \tag{6}$$

$$\hat{\sigma}\left(Q_{\theta'}(s',a')\right) = \sqrt{\frac{1}{N-1}\sum_{i=1}^N\left[Q_{\theta'}^i(s',a') - \bar{Q}_{\theta'}^i(s',a')\right]^2} \tag{7}$$

where $\bar{Q}_{\theta'}(s',a')$ is the mean of target Q-value estimations. Therefore, the update target of critics is:

$$y = r(s,a) + \gamma\mathbb{E}_{\mathcal{B},\pi}\left[\bar{Q}_{\theta'}(s',a') - \beta\hat{\sigma}\left(Q_{\theta'}(s',a')\right)\right] \tag{8}$$

We find that using the target above is better than the usual $\min$ function, because we can control the Q-value estimation between overestimation and underestimation by adjusting the $\beta$, which is similar to the weighted DDPG [He and Hou, 2020], but the adjustment range is more flexible than the weighted DDPG, which is more helpful to obtain an accurate Q-value. Further, we can get the updated estimation bias following [Li *et al.*, 2021]:

$$\begin{aligned} Z_N \approx \gamma\Bigg[&\max_{a'\in\mathcal{A}}\left(Q^\pi(s',a') + \bar{e}_{s'a'} - \beta\hat{\sigma}\left(e_{s'a'}\right)\right) \\ &- \max_{a'\in\mathcal{A}} Q^\pi(s',a')\Bigg] \end{aligned} \tag{9}$$

where $\beta$ is the hyperparameter to control the penalty term.

We find that the updated estimation bias is related to $\beta$ which is constant, which leads to a critical problem. In general, the uncertainty of estimations of multiple critics is getting smaller after training the same state-action pair several times, which leads to this penalty term being smaller with the training process gradually. Due to the Eq.(9), this will lead the overestimation. Therefore, we need to add an additional term to neutralize the effect of the standard deviation term, which is the familiarity term we will introduce in the next subsection.

### 4.3 Adaptive Estimation with Familiarity

In this subsection, we will show how familiarity can further adjust the penalty term and over- or underestimate specific state-action pair. In this paper, we use a simple way to calculate familiarity which is similar to count-based methods. We add a count record $c$ in the tuple $(s,a,r,s')$, which is initialized to 0 for each experience when it enters the replay buffer. Every time the experience is sampled, this record value will increase to track the number of times the experience is sampled. In addition, at each sampling, we also record the maximum count value $c_{max}$ through the training and calculate

the familiarity of each experience based on this value. When $v \leq i \leq t < i+N$, $\mathcal{F}^i(t)$ is defined in Eq.(10).

$$\mathcal{F}^i(t) \triangleq \begin{cases} \dfrac{c_i - 1}{max\left\{c_1, ..., c_t\right\}}, & t \leq N \\ \dfrac{c_i - 1}{max\left\{c_{t-N+1}, ..., c_t, c_{max}\right\}}, & t > N \end{cases} \tag{10}$$

where $i$ is the number of time steps when the experience is generated, $t$ is the number of time steps, $v$ is the number of time steps to start sampling from the replay buffer, and $N$ is the capacity of the replay buffer. The calculation of Eq.(10) can be divided into two cases according to whether the replay buffer is full or not. It calculates the ratio of sampling times between experience $i$ and the experience with maximum sampling times, and these two denominators indicate the historical maximum of $c$ in replay buffer for $t \leq N$ and $t > N$ respectively. In addition, we use $c_i - 1$ to make the $\mathcal{F}^i = 0$ when the experience is sampled in the first time.

Then, we combine familiarity with uncertainty, so that they can control the estimation of Q-value jointly:

$$y = r(s,a) + \gamma \mathbb{E}_{\mathcal{B},\pi}\left[\bar{Q}_{\theta'}\left(s', a'\right) - \beta_b \hat{\sigma}_{Q'} - \beta_s \mathcal{F} \hat{\sigma}_{Q'}\right] \tag{11}$$

where the first penalty term only contains uncertainty, and $\beta_b$ is used to control its weight; the second penalty term consists of the product of uncertainty and familiarity, and $\beta_s$ can control its weight.

**Theorem 1.** *For any $s, a, i$, $0 \leq \mathcal{F} < 1$, and the expected $\mathcal{F}^i$ will increase with the number of training steps $t$ grows through its life time.*

*Proof.* see Appendix. □

Based on the property of familiarity above, the familiarity is small when the experience first enters the replay buffer, which means the penalty term will be small, making the experience be overestimated. As the number of sampling times of experience gradually increases, the familiarity will also increase, making the experience receive an appropriate underestimation.

According to Section 4.1, we can conclude the updated estimation bias when combining familiarity and uncertainty:

$$\begin{aligned} Z_N &\triangleq r\left(s,a\right) + \gamma \max_{a' \in \mathcal{A}} \left(\bar{Q}_e\left(s', a'\right) - \beta_b \hat{\sigma}_{Q_e} - \beta_s \mathcal{F} \hat{\sigma}_{Q_e}\right) \\ &\quad - \left(r\left(s,a\right) + \gamma \max_{a' \in \mathcal{A}} Q^{\pi}\left(s', a'\right)\right) \\ &= \gamma \left(\max_{a' \in \mathcal{A}} \left(\bar{Q}_e\left(s', a'\right) - \beta_b \hat{\sigma}_{Q_e} - \beta_s \mathcal{F} \hat{\sigma}_{Q_e}\right)\right. \\ &\quad \left. - \max_{a' \in \mathcal{A}} Q^{\pi}\left(s', a'\right)\right) \end{aligned} \tag{12}$$

Based on Eq.(12), we prove that our method can reduce the bias of Q-learning to 0 under specific conditions.

**Theorem 2.** *For any $s', a'$, there exists a $\mathcal{F}_0$ satisfying Eq.*(13)*, $\mathbb{E}\left[Z_N\right] \approx 0$.*

**Algorithm 1** AEQ-TD3

1: Initialize actor network $\pi_\phi$ and with parameter $\phi$, $N$ critic network $Q^i_\theta$ with parameter $\theta_i$, where $i \in 1, ..., N$
2: Initialize target actor network $\pi_{\phi'}$ with parameter $\phi' \leftarrow \phi$, $N$ target critic network $Q^i_{\theta'}$ with parameter $\theta'_i \leftarrow \theta_i$, where $i \in 1, ..., N$
3: Initialize experience replay buffer $\mathcal{B}$
4: **for** $t = 1$ **to** $T$ **do**
5: Select action with exploration noise $a_t \sim \pi_\phi(s_t) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma)$, and observe reward $r_t$ and new state $s_{t+1}$
6: Store transition tuple $(s_t, a_t, r_t, s_{t+1}, 0)$ in $\mathcal{B}$
7: Sample mini-batch of $N$ transitions $(s, a, r, s', c)$ from $\mathcal{B}$
8: Update the corresponding $c \leftarrow c + 1$ for each sampled experience
9: Compute familiarity $\mathcal{F}$ for each sampled experience using Eq.(10)
10: Compute the Q target $y$ using Eq.(11)
11: **for** $i = 1$ **to** $N$ **do**
12: Update critics by minimizing Eq.(2)
13: **end for**
14: **if** $t$ mod $d$ **then**
15: Update actor using Eq.(14)
16: Update target networks: $\theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i$, $\phi' \leftarrow \tau\phi + (1-\tau)\phi'$
17: **end if**
18: **end for**

$$\mathcal{F}_0 \approx \frac{\bar{e}_{s'a'} - \beta_b \hat{\sigma}\left(e_{s'a'}\right)}{\beta_s \hat{\sigma}\left(e_{s'a'}\right)} \tag{13}$$

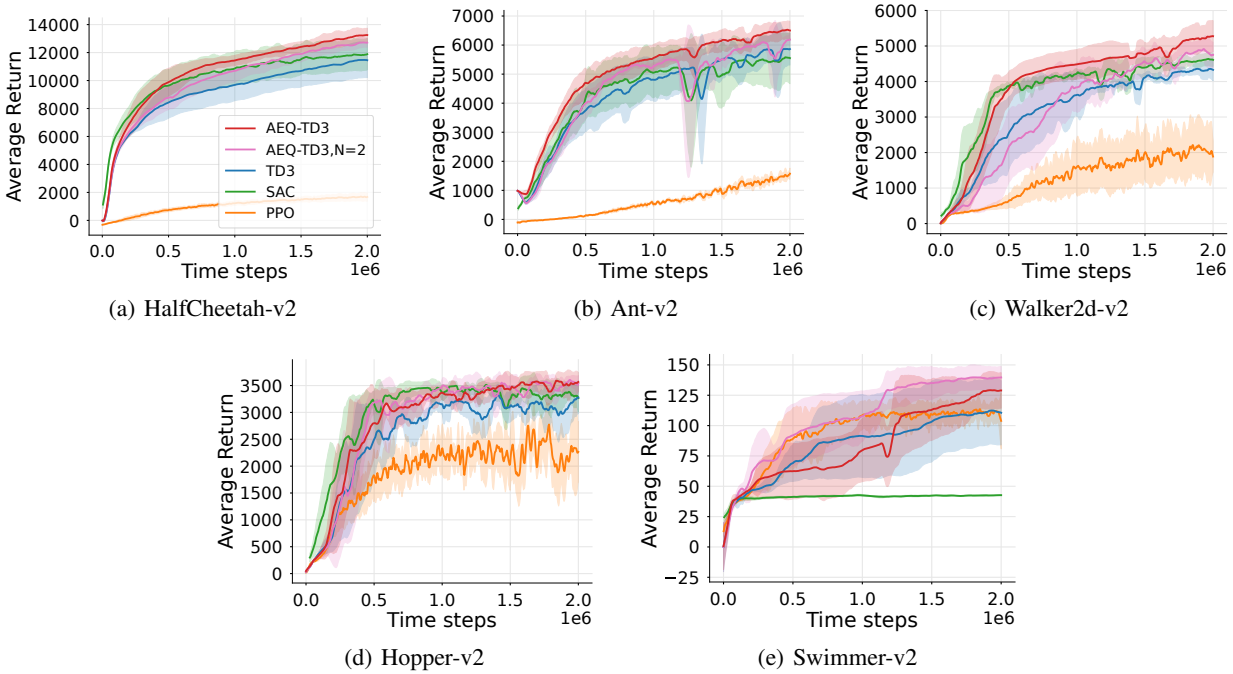
*Proof.* see Appendix. □

The approximation sign of Eq.(13) is due to the sample based mean and variance. Although the conditions above can not always meet usually, our method can still give an appropriate estimation of Q-value for specific state-action pair. When the critics have a large uncertainty about a new experience, the overestimation can improve the exploration; when an old experience has a large uncertainty, we believe that the state-action pair of this experience contains high risk, the underestimation can prevent the agent from entering an unstable state and improve the robustness.

### 4.4 Applying AEQ to TD3 and SAC

We apply AEQ to TD3 [Fujimoto *et al.*, 2018] called AEQ-TD3 and it uses actor-critic framework but with $N$ critics. These $N$ critics are initialized differently but are trained with the same target value Eq.(11). The actor is trained by the deterministic policy gradient with the average Q-value of $N$ critics:

$$\nabla_\pi J(\phi) = \mathbb{E}_{\mathcal{B},\pi}\left[\nabla_a \frac{1}{N} \sum_{i=1}^{N} Q^i_\theta(s,a)\Big|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)\right] \tag{14}$$




(a) HalfCheetah-v2 (b) Ant-v2 (c) Walker2d-v2

(d) Hopper-v2 (e) Swimmer-v2

Figure 1: Average performances comparison on MuJoCo environments.

Besides, we do not modify any other part of TD3, the procedure of AEQ-TD3 is summarized in Algorithm 1.

We also apply AEQ to SAC [Haarnoja *et al.*, 2018] called AEQ-SAC. Like REDQ [Chen *et al.*, 2021], AEQ-SAC employs a UTD = 20 to improve sample efficiency during training. Instead of using two randomly selected critics to calculate the target, AEQ-SAC uses Eq.(11) to be the target of $N$ critics. The pseudo-code of AEQ-SAC is shown in Appendix.

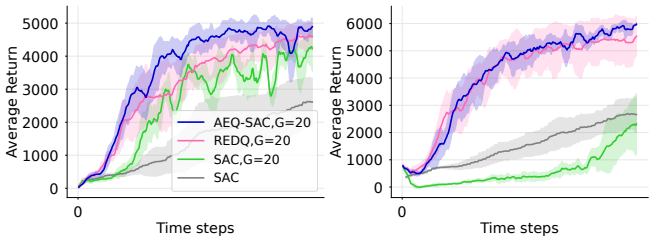
## 5 Experiment

We evaluate our method on a range of MuJoCo [Todorov *et al.*, 2012] continuous control tasks from OpenAI Gym [Brockman *et al.*, 2016]. We implement our methods on TD3 [Fujimoto *et al.*, 2018] and SAC [Haarnoja *et al.*, 2018] as AEQ-TD3 and AEQ-SAC respectively[1]. For AEQ-TD3, we use $N = 2$ and $N = 10$ critics with three hidden layers, $\beta_b = 0.5, \beta_s = 0.5$ for every tasks, and UTD = 1 for fair comparison. For AEQ-SAC, we use $N = 10$ and UTD = 20 to compare with REDQ [Chen *et al.*, 2021]. For simplicity, we will use $G$ instead of UTD ratio in subsequent. The plots of experimental results are generated by rl-plotter [2].

The details of the experimental setup and additional results can be found in Appendix.

### 5.1 Comparative Evaluation

We compare our methods with the state-of-the-art algorithms.




(a) Walker2d-v2 (b) Ant-v2

Figure 2: Average performances comparison on MuJoCo environments when $G = 20$.

For AEQ-TD3, we compare it with PPO [Schulman *et al.*, 2017], TD3, and SAC on five continuous control tasks: HalfCheetah, Ant, Walker2d, Hopper, and Swimmer. The time steps of each algorithm on each task is $2 \times 10^6$.

For AEQ-SAC, we compare it with SAC, SAC20 [Chen *et al.*, 2021], REDQ, and TQC20 [Kuznetsov *et al.*, 2020; Li *et al.*, 2021] on two challenging continuous control tasks: Ant and Walker2d. The time steps of each algorithm on each task is $3 \times 10^5$ following REDQ's setting.

The learning curves are shown in Figure 1 and Figure 2. Each curve is the average result of 5 random seeds with the shaded area of the standard deviation. We evaluate the performance of each algorithm every 5000 steps, and each evaluation is the average of 10 episodes. In Table 1 and Table 2, we also report the average and the standard deviation of last 10 evaluations with 5 random seeds each algorithm. As the results shown, it is obvious that our AEQ-TD3 achieve

[1]Implementations and appendix are available at: https://github.com/gxywy/AEQ

[2]https://github.com/gxywy/rl-plotter

| Environment | PPO | SAC | TD3 | AEQ-TD3,N=2 | AEQ-TD3 |
|---|---|---|---|---|---|
| HalfCheetah-v2 | 1655.91±255.82 | 12002.23±1256.65 | 11521.06±1284.37 | 12845.54±361.09 | **13212.75±672.51** |
| Ant-v2 | 1679.47±652.63 | 5717.51±708.97 | 6011.47±592.43 | 6315.22±269.17 | **6592.58±272.50** |
| Walker2d-v2 | 1816.40±1238.28 | 4563.80±294.28 | 4266.04±695.25 | 4685.51±488.45 | **5236.13±518.57** |
| Hopper-v2 | 2097.39±1250.00 | 3509.85±80.89 | 3506.63±180.53 | **3512.14±243.53** | 3439.55±328.17 |
| Swimmer-v2 | 90.65±46.36 | 42.85±0.69 | 110.51±26.89 | **137.82±8.74** | 128.46±14.11 |

Table 1: Numerical performance comparison of 2M time steps on final score over 5 seeds. The best results are in bold.

| Environment | SAC | SAC20 | REDQ | TQC20 | AEQ-SAC |
|---|---|---|---|---|---|
| Walker2d-v2 | 3220±566 | 5090±365 | 4741±310 | 4833±296 | **5101±316** |
| Ant-v2 | 2785±947 | 2603±1348 | 5561±767 | 4722±567 | **6005±103** |

Table 2: Numerical performance comparison of 0.3M time steps on final score over 5 seeds when $G = 20$. The best results are in bold.

| Variant | Target |
|---|---|
| TD3-Min-10 | $\min_{i=1,\ldots,N} Q'_{\theta'_i}(s', \pi_{\phi'}(s'))$ |
| TD3-Mean-10 | $\underset{i=1,\ldots,N}{\text{average}} Q'_{\theta'_i}(s', \pi_{\phi'}(s'))$ |
| TD3-REDQ-10 | $\min_{i\in\mathcal{M}} Q'_{\theta'_i}(s', \pi_{\phi'}(s'))$ |
| AEQ-TD3-RF | $\bar{Q}_{\theta'}(s', a') - \beta_b \hat{\sigma}_{Q'} - \beta_s \mathcal{F}_r \hat{\sigma}_{Q'}$, random $\mathcal{F}_r \in (0, 1]$ |
| AEQ-TD3-NF | $\bar{Q}_{\theta'}(s', a') - \beta_b \hat{\sigma}_{Q'}$ |

Table 3: The target of 5 different variants.

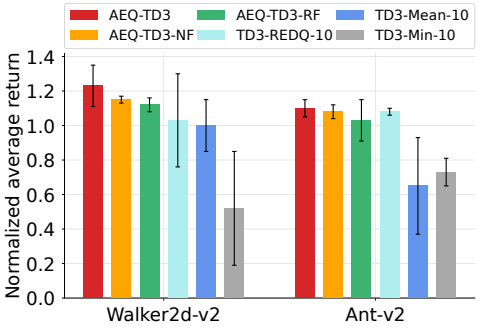



Figure 3: Normalized average final performance of 5 variants over 5 seeds.

the best performance. Even if we set $N = 2$, our results can still outperform TD3 on every task. When $G = 20$, our AEQ-SAC also can achieve better sample efficiency than the state-of-the-art algorithm REDQ.

## 5.2 Ablation Study

We perform ablation experiments on Ant and Walker2d tasks to further analyze the effectiveness of our AEQ target. We build five variants based on TD3 but trained with different target of critic which is shown in Table 3. For all variants, we use the same network structure and $N = 10$ critics, and for TD3-REDQ-10 variant, we use $G = 1$ for fair comparison.

The final performance of different variances is shown in Figure 3 which is normalized using the average final performance of TD3. It suggests that the minimum variants and

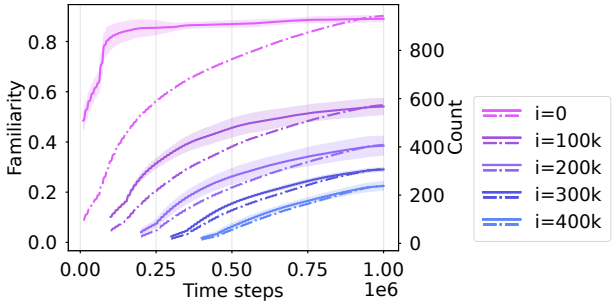



Figure 4: Familiarity and the number of sampling time of experiences in 1M time steps on Ant-v2. The solid line is the familiarity of experiences. The dash dotted line is the number of sampling time of experiences.

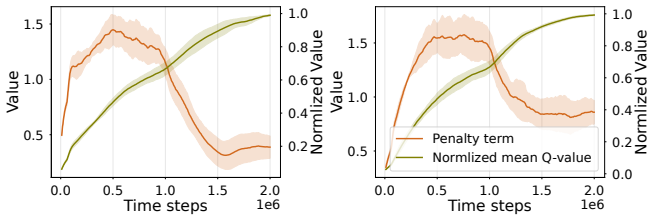
the mean variants perform poorly when increasing the number of critics $N$. Although REDQ can reduces the Std. of bias, it does not bring significant increase in sampling efficiency when applied REDQ's target to TD3. However, the result of AEQ-TD3-NF shows that the target with uncertainty penalty performs better. Moreover, when comparing AEQ-TD3 with AEQ-TD3-RF and AEQ-TD3-NF, it shows our familiarity term can improve the performance on both Ant and Walker2d tasks.

## 5.3 Effect of Familiarity

In order to study the effect of our familiarity term further, we first select some experiences with the indexes: 0, $10^5$, $2\times10^5$, $3\times10^5$, $4\times10^5$ to track the familiarity $\mathcal{F}$ and the number of sampling times $c$.

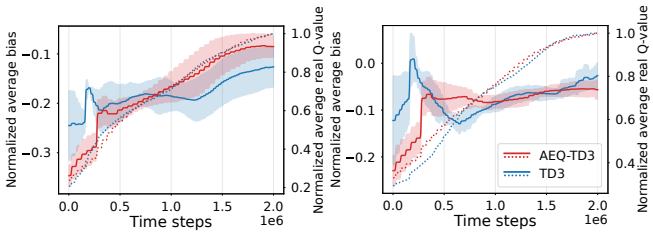
In Figure 4, we find all $\mathcal{F}$ is increasing with $c$, and this conclusion is consistent with Theorem 1. The result also suggests the experience that enters the replay buffer first will increase faster in familiarity than the experience that enters the replay buffer later, and the former will also have a larger familiarity in the end. This phenomenon implies that our familiarity will pay less attention to newer experiences and tend to give them less punishment in Q-value estimation.

Then, we study the tendency of our penalty term $\beta_b \hat{\sigma}_{Q'} + \beta_s \mathcal{F} \hat{\sigma}_{Q'}$ during the training, the results are shown in Figure




(a) Walker2d-v2 (b) Ant-v2

Figure 5: Penalty term and normalized mean Q-value of batches during the training.




(a) Walker2d-v2 (b) Hopper-v2

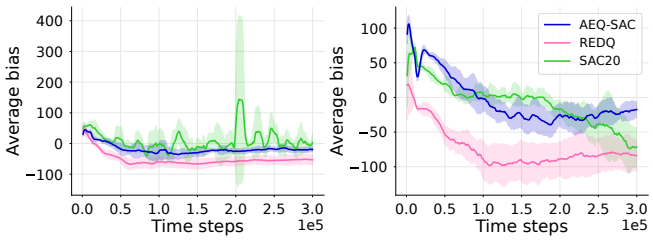
Figure 6: Comparison of the bias of Q-value estimations of TD3 and AEQ-TD3. The solid line is the normalized average bias. The dotted line is the normalized average real Q-values.

5. Overall, our penalty term will increase because the uncertainty in the initial training process of the network, and then decreases due to the neutralization effect of familiarity, and eventually remain almost constant after the training is stable. However, it is just the tendency of the average sampled experiences, which does not mean all experiences follow the same rules, and each experience will have its own tendency in our algorithm. In addition, although our penalty term is changing, the average Q-value keeps increasing, which suggests that our adaptive term does not disturb the training process.

### 5.4 Estimation Bias

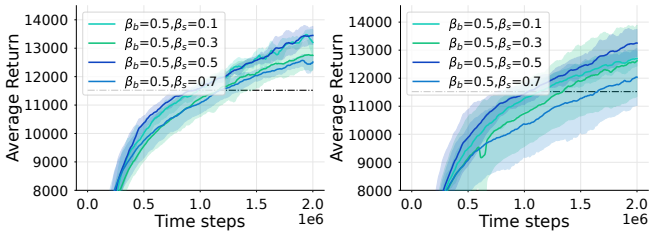
In order to figure out how AEQ estimates in practice, we measure the estimation bias of both AEQ-TD3 and AEQ-SAC. For AEQ-TD3, the Q-value estimations are averaged over 1000 states sampled from the replay buffer every 50000 time steps. The true Q-values are estimated by averaging the discounted long-term rewards obtained by rolling out the current policy starting from the sampled states every 50000 time steps. The setting above is basically same with the original paper of TD3. The results in Figure 6 show that AEQ-TD3 will have a large estimation bias in the beginning, but will reduce gradually through training achieving better performance.

For AEQ-SAC, we follow the REDQ's setting of estimating the bias for fair comparison, which the states is not sampled from the replay buffer. The results in Figure 7 show that AEQ-SAC controls the estimation bias better and is close to 0. Moreover, AEQ-SAC can adjust the estimation bias dynamically.




(a) Walker2d-v2 (b) Ant-v2

Figure 7: Comparison of the bias of Q-value estimations of REDQ and AEQ-SAC.




(a) The sensitivity of $\beta_b$ (b) The sensitivity of $\beta_s$

Figure 8: The hyperparameter sensitivity of $\beta_b$ and $\beta_s$. The dash dotted line is the average final performance of TD3.

### 5.5 Hyperparameter Sensitivity

The hyperparameter $\beta_b$ and $\beta_s$ will directly affect the weight of familiarity term and the estimation of Q-values. Therefore, we study the sensitivity of $\beta_b$ and $\beta_s$ on HalfCheetah task and we choose them from $[0.1, 0.3, 0.5, 0.7]$. The results are shown in Figure 8, which indicates the performance of AEQ-TD3 always better than TD3 in 4 tested $\beta_b$ and $\beta_s$. The results also suggest that $\beta_b$ and $\beta_s$ should not too large or too small to keep the penalty term and adaptive term in a certain range. Besides, the sensitivity to $\beta_b$ and $\beta_s$ indicates in part that the uncertainty and familiarity term we proposed is effective.

## 6 Conclusion

In this paper, we present AEQ that controls the over- and underestimation bias for specific state-action pair adaptively using uncertainty and familiarity. Our method is simple to implement on any off-policy actor-critic RL algorithm, including the most commonly used TD3 and SAC. We not only analyze the property and the effect of familiarity theoretically, but also perform the ablation experiments to demonstrate it can improve the performance with the uncertainty. The results on continuous control tasks suggest that our AEQ can be useful in controlling the estimation bias and can outperform the state-of-the-art performance on sample efficiency.

We think future work should combine familiarity with the density model and focus on investigating how to find a more appropriate metric to over- and underestimate for specific state-action pair.

## Acknowledgments

We sincerely thank the anonymous reviewers for their careful work and thoughtful suggestions, which have greatly improved this article. This work was supported by the Natural Science Research Foundation of Jilin Province of China under Grant Nos. 20220101106JC and YDZJ202201ZYTS423, the National Natural Science Foundation of China under Grant No. 61300049, the Fundamental Research Funds for the Central Universities (Jilin University) under Grant No. 93K172022K10, the Fundamental Research Funds for the Central Universities (Northeast Normal University) under Grant No. 2412022QD040, and the National Key R&D Program of China under Grant No. 2017YFB1003103.

## References

[Agarwal *et al.*, 2020] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, pages 104–114, 2020.

[An *et al.*, 2021] Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified Q-ensemble. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, Sydney, Australia, 2021.

[Anschel *et al.*, 2017] Oron Anschel, Nir Baram, and Nahum Shimkin. Averaged-DQN: Variance reduction and stabilization for deep reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, pages 176–185, Sydney, NSW, Australia, 2017.

[Bellemare *et al.*, 2016] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Proceedings of the 30st Conference on Neural Information Processing Systems (NIPS 2016)*, pages 1471–1479, Barcelona, Spain, 2016.

[Brockman *et al.*, 2016] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

[Burda *et al.*, 2019] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*, New Orleans, LA, USA, 2019.

[Chen *et al.*, 2017] Richard Y Chen, Szymon Sidor, Pieter Abbeel, and John Schulman. UCB exploration via Q-ensembles. *arXiv preprint arXiv:1706.01502*, 2017.

[Chen *et al.*, 2021] Xinyue Chen, Che Wang, Zijian Zhou, and Keith Ross. Randomized ensembled double Q-learning: Learning fast without a model. In *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)*, 2021.

[Cideron *et al.*, 2020] Geoffrey Cideron, Thomas Pierrot, Nicolas Perrin, Karim Beguir, and Olivier Sigaud. QD-RL: Efficient mixing of quality and diversity in reinforcement learning. *arXiv preprint arXiv:2006.08505*, pages 28–73, 2020.

[Ciosek *et al.*, 2019] Kamil Ciosek, Quan Vuong, Robert Loftin, and Katja Hofmann. Better exploration with optimistic actor-critic. In *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2019)*, pages 1785–1796, Vancouver, BC, Canada, 2019.

[Conti *et al.*, 2018] Edoardo Conti, Vashisht Madhavan, Felipe Petroski Such, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS 2018)*, pages 5032–5043, Montréal, Canada, 2018.

[Duan *et al.*, 2021] Jingliang Duan, Yang Guan, Shengbo Eben Li, Yangang Ren, Qi Sun, and Bo Cheng. Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[Fujimoto *et al.*, 2018] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, pages 1582–1591, Stockholmsmässan, Stockholm, Sweden, 2018.

[Haarnoja *et al.*, 2018] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, pages 1856–1865, Stockholmsmässan, Stockholm, Sweden, 2018.

[Hasselt, 2010] Hado Hasselt. Double Q-learning. In *Proceedings of the 24th Conference on Neural Information Processing Systems (NIPS 2010)*, pages 2613–2621, Vancouver, British Columbia, Canada, 2010.

[He and Hou, 2020] Qiang He and Xinwen Hou. WD3: Taming the estimation bias in deep reinforcement learning. In *32nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2020)*, pages 391–398, Baltimore, MD, USA, 2020.

[Kuznetsov *et al.*, 2020] Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, pages 5556–5566, 2020.

[Lan *et al.*, 2020] Qingfeng Lan, Yangchen Pan, Alona Fyshe, and Martha White. Maxmin Q-learning: Controlling the estimation bias of Q-learning. In *Proceedings of*

*the 8th International Conference on Learning Representations (ICLR 2020)*, Addis Ababa, Ethiopia, 2020.

[Lee *et al.*, 2021] Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. SUNRISE: A simple unified framework for ensemble learning in deep reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, pages 6131–6141, 2021.

[Li and Hou, 2019] Zhunan Li and Xinwen Hou. Mixing update Q-value for deep reinforcement learning. In *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN 2019)*, pages 1–6, Budapest, Hungary, 2019.

[Li *et al.*, 2021] Sicen Li, Gang Wang, Qinyun Tang, and Liquan Wang. Balancing value underestimation and overestimation with realistic actor-critic. *arXiv preprint arXiv:2110.09712*, 2021.

[Lillicrap *et al.*, 2015] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[Osband *et al.*, 2016] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. In *Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS 2016)*, pages 4026–4034, Barcelona, Spain, 2016.

[Pan *et al.*, 2020] Ling Pan, Qingpeng Cai, and Longbo Huang. Softmax deep double deterministic policy gradients. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[Pathak *et al.*, 2019] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, pages 5062–5071, Long Beach, California, USA, 2019.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Tang *et al.*, 2017] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, pages 2753–2762, Long Beach, CA, USA, 2017.

[Thrun and Schwartz, 1993] Sebastian Thrun and Anton Schwartz. Issues in using function approximation for reinforcement learning. In *Proceedings of the 4th Connectionist Models Summer School*, pages 255–263, 1993.

[Todorov *et al.*, 2012] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[van Hasselt *et al.*, 2016] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-learning. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016)*, pages 2094–2100, Phoenix, Arizona, USA, 2016.

[Zhang *et al.*, 2017] Zongzhang Zhang, Zhiyuan Pan, and Mykel J Kochenderfer. Weighted double Q-learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017)*, pages 3455–3461, Melbourne, Australia, 2017.