Addressing Function Approximation Error in Actor-Critic Methods: Supplementary Material

A. Proof of Convergence of Clipped Double Q-Learning

In a version of Clipped Double Q-learning for a finite MDP setting, we maintain two tabular value estimates Q^A, Q^B. At each time step we select actions $a^* = \text{argmax}_a Q^A(s, a)$ and then perform an update by setting target y:

$$\begin{aligned} a^* &= \operatorname*{argmax}_a Q^A(s', a) \\ y &= r + \gamma \min(Q^A(s', a^*), Q^B(s', a^*)), \end{aligned} \tag{1}$$

and update the value estimates with respect to the target and learning rate $\alpha_t(s, a)$:

$$\begin{aligned} Q^A(s, a) &= Q^A(s, a) + \alpha_t(s, a)(y - Q^A(s, a)) \\ Q^B(s, a) &= Q^B(s, a) + \alpha_t(s, a)(y - Q^B(s, a)). \end{aligned} \tag{2}$$

In a finite MDP setting, Double Q-learning is often used to deal with noise induced by random rewards or state transitions, and so either Q^A or Q^B is updated randomly. However, in a function approximation setting, the interest may be more towards the approximation error and thus we can update both Q^A and Q^B at each iteration. The proof extends naturally to updating either randomly.

The proof borrows heavily from the proof of convergence of SARSA (Singh et al., 2000) as well as Double Q-learning (Van Hasselt, 2010). The proof of lemma 1 can be found in Singh et al. (2000), building on a proposition from Bertsekas (1995).

Lemma 1. Consider a stochastic process $(\zeta_t, \Delta_t, F_t), t \geq 0$ where $\zeta_t, \Delta_t, F_t : X \to \mathbb{R}$ satisfy the equation:

$$\Delta_{t+1}(x_t) = (1 - \zeta_t(x_t))\Delta_t(x_t) + \zeta_t(x_t)F_t(x_t), \tag{3}$$

where $x_t \in X$ and $t = 0, 1, 2, ...$ Let P_t be a sequence of increasing σ-fields such that ζ_0 and Δ_0 are P_0-measurable and ζ_t, Δ_t and F_{t-1} are P_t-measurable, $t = 1, 2, ...$ Assume that the following hold:

1. The set X is finite.
2. $\zeta_t(x_t) \in [0, 1], \sum_t \zeta_t(x_t) = \infty, \sum_t (\zeta_t(x_t))^2 < \infty$ with probability 1 and $\forall x \neq x_t : \zeta(x) = 0$.
3. $||\mathbb{E}[F_t|P_t]|| \leq \kappa||\Delta_t|| + c_t$ where $\kappa \in [0, 1)$ and c_t converges to 0 with probability 1.
4. $\text{Var}[F_t(x_t)|P_t] \leq K(1 + \kappa||\Delta_t||)^2$, where K is some constant

Where $||\cdot||$ denotes the maximum norm. Then Δ_t converges to 0 with probability 1.

Theorem 1. Given the following conditions:

1. Each state action pair is sampled an infinite number of times.
2. The MDP is finite.
3. $\gamma \in [0, 1)$.
4. Q values are stored in a lookup table.
5. Both Q^A and Q^B receive an infinite number of updates.

6. The learning rates satisfy $\alpha_t(s,a) \in [0,1], \sum_t \alpha_t(s,a) = \infty, \sum_t (\alpha_t(s,a))^2 < \infty$ with probability 1 and $\alpha_t(s,a) = 0, \forall (s,a) \neq (s_t, a_t)$.

7. $\text{Var}[r(s,a)] < \infty, \forall s, a$.

Then Clipped Double Q-learning will converge to the optimal value function Q^*, as defined by the Bellman optimality equation, with probability 1.

Proof of Theorem 1. We apply Lemma 1 with $P_t = \{Q_0^A, Q_0^B, s_0, a_0, \alpha_0, r_1, s_1, ..., s_t, a_t\}, X = S \times A, \Delta_t = Q_t^A - Q^*, \zeta_t = \alpha_t$.

First note that condition 1 and 4 of the lemma holds by the conditions 2 and 7 of the theorem respectively. Lemma condition 2 holds by the theorem condition 6 along with our selection of $\zeta_t = \alpha_t$.

Defining $a^* = \text{argmax}_a Q^A(s_{t+1}, a)$ we have

$$\begin{aligned}\Delta_{t+1}(s_t, a_t) &= (1 - \alpha_t(s_t, a_t))(Q_t^A(s_t, a_t) - Q^*(s_t, a_t)) \\ &\quad + \alpha_t(s_t, a_t)(r_t + \gamma \min(Q_t^A(s_{t+1}, a^*), Q_t^B(s_{t+1}, a^*)) - Q^*(s_t, a_t)) \\ &= (1 - \alpha_t(s_t, a_t))\Delta_t(s_t, a_t) + \alpha_t(s_t, a_t)F_t(s_t, a_t)),\end{aligned} \tag{4}$$

where we have defined $F_t(s_t, a_t)$ as:

$$\begin{aligned}F_t(s_t, a_t) &= r_t + \gamma \min(Q_t^A(s_{t+1}, a^*), Q_t^B(s_{t+1}, a^*)) - Q_t^*(s_t, a_t) \\ &= r_t + \gamma \min(Q_t^A(s_{t+1}, a^*), Q_t^B(s_{t+1}, a^*)) - Q_t^*(s_t, a_t) + \gamma Q_t^A(s_{t+1}, a^*) - \gamma Q_t^A(s_{t+1}, a^*) \\ &= F_t^Q(s_t, a_t) + c_t,\end{aligned} \tag{5}$$

where $F_t^Q = r_t + \gamma Q_t^A(s_{t+1}, a^*) - Q_t^*(s_t, a_t)$ denotes the value of F_t under standard Q-learning and $c_t = \gamma \min(Q_t^A(s_{t+1}, a^*), Q_t^B(s_{t+1}, a^*)) - \gamma Q_t^A(s_{t+1}, a^*)$. As $\mathbb{E}\left[F_t^Q | P_t\right] \leq \gamma ||\Delta_t||$ is a well-known result, then condition 3 of lemma 1 holds if it can be shown that c_t converges to 0 with probability 1.

Let $y = r_t + \gamma \min(Q_t^B(s_{t+1}, a^*), Q_t^A(s_{t+1}, a^*))$ and $\Delta_t^{BA}(s_t, a_t) = Q_t^B(s_t, a_t) - Q_t^A(s_t, a_t)$, where c_t converges to 0 if Δ^{BA} converges to 0. The update of Δ_t^{BA} at time t is the sum of updates of Q^A and Q^B:

$$\begin{aligned}\Delta_{t+1}^{BA}(s_t, a_t) &= \Delta_t^{BA}(s_t, a_t) + \alpha_t(s_t, a_t)\left(y - Q_t^B(s_t, a_t) - (y - Q_t^A(s_t, a_t))\right) \\ &= \Delta_t^{BA}(s_t, a_t) + \alpha_t(s_t, a_t)\left(Q_t^A(s_t, a_t) - Q_t^B(s_t, a_t)\right) \\ &= (1 - \alpha_t(s_t, a_t))\Delta_t^{BA}(s_t, a_t).\end{aligned} \tag{6}$$

Clearly Δ_t^{BA} will converge to 0, which then shows we have satisfied condition 3 of lemma 1, implying that $Q^A(s_t, a_t)$ converges to $Q_t^*(s_t, a_t)$. Similarly, we get convergence of $Q^B(s_t, a_t)$ to the optimal vale function by choosing $\Delta_t = Q_t^B - Q^*$ and repeating the same arguments, thus proving theorem 1.

B. Overestimation Bias in Deterministic Policy Gradients

If the gradients from the deterministic policy gradient update are unnormalized, this overestimation is still guaranteed to occur under a slightly stronger condition on the expectation of the value estimate. Assume the approximate value function is equal to the true value function, in expectation over the steady-state distribution, with respect to policy parameters between the original policy and in the direction of the true policy update:

$$\begin{aligned}&\mathbb{E}_{s\sim\pi}\left[Q_\theta(s, \pi_{\text{new}}(s))\right] = \mathbb{E}_{s\sim\pi}\left[Q^\pi(s, \pi_{\text{new}}(s))\right] \\ &\forall \phi_{\text{new}} \in [\phi, \phi + \beta(\phi_{\text{true}} - \phi)] \text{ such that } \beta > 0.\end{aligned} \tag{7}$$

Noting that ϕ_{true} maximizes the rate of change of the true value $\Delta_{\text{true}}^\pi = Q^\pi(s, \pi_{\text{true}}(s)) - Q^\pi(s, \pi_\phi(s))$, $\Delta_{\text{true}}^\pi \geq \Delta_{\text{approx}}^\pi$. By the given condition 7 the maximal rate of change of the approximate value must be at least as great $\Delta_{\text{approx}}^\theta \geq \Delta_{\text{true}}^\pi$. Given $Q_\theta(s, \pi_\phi) = Q^\pi(s, \pi_\phi)$ this implies $Q_\theta(s, \pi_{\text{approx}}(s)) \geq Q^\pi(s, \pi_{\text{true}}(s)) \geq Q^\pi(s, \pi_{\text{approx}}(s))$, showing an overestimation of the value function.

Table 1. A complete comparison of hyper-parameter choices between our DDPG and the OpenAI baselines implementation (Dhariwal et al., 2017).

Hyper-parameter	**Ours**	**DDPG**
Critic Learning Rate	10^{-3}	10^{-3}
Critic Regularization	None	$10^{-2} \cdot \|\|\theta\|\|^2$
Actor Learning Rate	10^{-3}	10^{-4}
Actor Regularization	None	None
Optimizer	Adam	Adam
Target Update Rate (τ)	$5 \cdot 10^{-3}$	10^{-3}
Batch Size	100	64
Iterations per time step	1	1
Discount Factor	0.99	0.99
Reward Scaling	1.0	1.0
Normalized Observations	False	True
Gradient Clipping	False	False
Exploration Policy	$\mathcal{N}(0, 0.1)$	OU, $\theta = 0.15, \mu = 0, \sigma = 0.2$

C. DDPG Network and Hyper-parameter Comparison

DDPG Critic Architecture

```
(state dim, 400)
ReLU
(action dim + 400, 300)
ReLU
(300, 1)
```

DDPG Actor Architecture

```
(state dim, 400)
ReLU
(400, 300)
ReLU
(300, 1)
tanh
```

Our Critic Architecture

```
(state dim + action dim, 400)
ReLU
(action dim + 400, 300)
RelU
(300, 1)
```

Our Actor Architecture

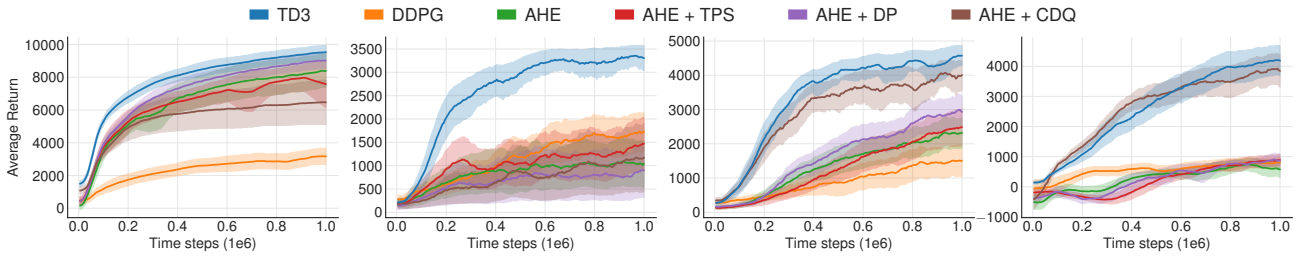
```
(state dim, 400)
ReLU
(400, 300)
RelU
(300, 1)
tanh
```

D. Soft Actor-Critic Implementation Details

For our implementation of Soft Actor-Critic (Haarnoja et al., 2018) we use the code provided by the author (https://github.com/haarnoja/sac), using the hyper-parameters described by the paper. We use a Gaussian mixture policy with 4 Gaussian distributions, except for the Reacher-v1 task, where we use a single Gaussian distribution due to numerical instability issues in the provided implementation. We use the environment-dependent reward scaling as described by the authors, multiplying the rewards by 3 for Walker2d-v1 and Ant-v1, and 1 for all remaining environments.

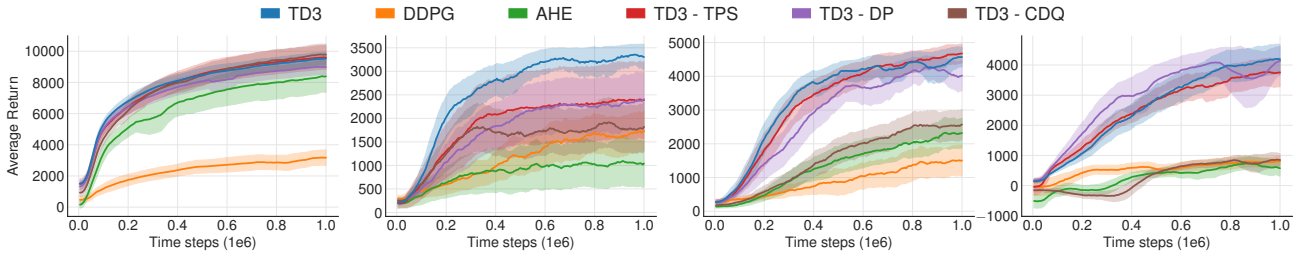
For fair comparison with our method, we train for only 1 iteration per time step, rather than the 4 iterations used by the results reported by the authors. This along with fewer total time steps should explain for the discrepancy in results on some of the environments. Additionally, we note this comparison is against a prior version of Soft Actor-Critic, while the most recent variant uses our Clipped Double Q-learning and produces competitive results to TD3 on most tasks.

E. Learning Curves



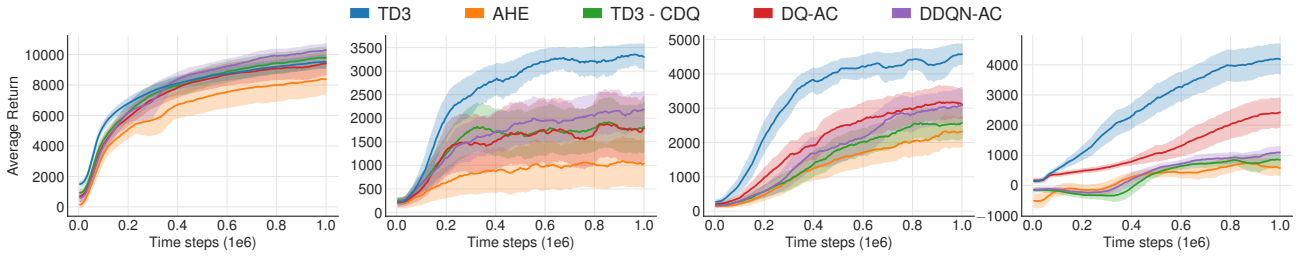
(a) HalfCheetah-v1 (b) Hopper-v1 (c) Walker2d-v1 (d) Ant-v1

Figure 1. Ablation over the varying modifications to our DDPG (AHE), comparing the subtraction of delayed policy updates (TD3 - DP), target policy smoothing (TD3 - TPS) and Clipped Double Q-learning (TD3 - CDQ).



(a) HalfCheetah-v1 (b) Hopper-v1 (c) Walker2d-v1 (d) Ant-v1

Figure 2. Ablation over the varying modifications to our DDPG (AHE), comparing the addition of delayed policy updates (AHE + DP), target policy smoothing (AHE + TPS) and Clipped Double Q-learning (AHE + CDQ).



(a) HalfCheetah-v1 (b) Hopper-v1 (c) Walker2d-v1 (d) Ant-v1

Figure 3. Comparison of TD3 and the Double Q-learning (DQ-AC) and Double DQN (DDQN-AC) actor-critic variants, which also leverage delayed policy updates and target policy smoothing.

References

Bertsekas, D. P. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.

Dhariwal, P., Hesse, C., Plappert, M., Radford, A., Schulman, J., Sidor, S., and Wu, Y. Openai baselines. https://github.com/openai/baselines, 2017.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Singh, S., Jaakkola, T., Littman, M. L., and Szepesvári, C. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine learning*, 38(3):287–308, 2000.

Van Hasselt, H. Double q-learning. In *Advances in Neural Information Processing Systems*, pp. 2613–2621, 2010.