Bagged Critic for Continuous Control

Payal Bawa [1] **Fabio Ramos** [1 2]

Abstract

Actor-critic methods have been successfully applied to several high dimensional continuous control tasks. Despite their success, they are prone to overestimation bias that leads to sub-optimal policies and divergent behaviour. Algorithms like TD3 and Soft Actor Critic (SAC) address overestimation bias by employing twin Q functions and optimizing the policy with respect to the lower bound of the two Q functions. Although this resolves the issue of overestimation bias, it inadvertently introduces underestimation bias. Underestimation bias, though not as problematic as overestimation bias, affects the asymptotic performance of the algorithms. To address both overestimation and underestimation bias in actor critic methods, we propose Bagged Critic for Continuous Control (BC3). BC3 uses an ensemble of independently trained Q functions as critic to address estimation biases. We present theoretical bounds on the biases and convergence analysis of our method demonstrating its benefits. The empirical results on several challenging reinforcement learning benchmarks substantiate our theoretical analysis and demonstrate reduction in biases with overall more robust policies.

1. Introduction

Reinforcement Learning (RL) enables an agent to learn policies for sequential decision problems (Sutton and Barto, 2018) by optimising the expected long-term returns. It does so by learning optimal actions for each state of the environment typically encoded in policy. Over the years several RL algorithms have been proposed to find optimal policies. Policy gradient algorithms find widespread applications in continuous control tasks. Deterministic policy gradient (DPG) (Silver et al., 2014) models a deterministic policy for learning within an actor critic framework. DDPG (Deep deterministic policy gradient) (Lillicrap et al., 2016) extends DPG to high dimensional continuous action spaces by using neural networks to model the policy (actor) and the Q function (critic) respectively. Even though DDPG works well in continuous control settings, the issue of overestimation persists.

Twin Delayed Deep Deterministic policy gradient algorithm (TD3) (Fujimoto et al., 2018) addresses the overestimation bias in DDPG. Similar to DDPG, it has a single neural network for actor. For critic, TD3 employs a pair of Q functions to estimate the Q value. The target update for learning is the minimum of the two Q functions. Taking the minimum prevents introduction of additional overestimation over the standard Q-learning target but introduces underestimation. Unlike overestimation, underestimation does not propagate via TD updates. But consistent underestimation affects the asymptotic performance of the policies especially in high dimensional continuous action spaces.

An extension of policy gradient algorithms are Maximum Entropy Reinforcement Learning algorithms (MERL) (Ziebart, 2010; Haarnoja et al., 2018). MERL algorithms augment the standard reinforcement learning objective of maximum reward with a maximum entropy objective. A state-of-the-art member of maximum entropy RL algorithms family is Soft Actor Critic (SAC) (Haarnoja et al., 2018). SAC is well known to be extremely sample efficient and performs better than TD3 on challenging high dimensional tasks. Like TD3, the implementation of SAC addresses overestimation bias by introducing underestimation.

We address both overestimation and underestimation biases in our method Bagged Critic for Continuous Control (BC3) by using an ensemble of randomly initialized critics. Each member of the ensemble is trained independently on a different subset of experience samples and their outputs are aggregated (bagging). The combined approach of bagging and random initialization gives an extremely diverse set of Q-functions such that the averaged prediction offsets any overestimation and underestimation bias in the actor critic setting. We summarise our contributions as follows:

- A novel extension to actor critic methods that stabilises training, mitigates estimation biases and improves per-

[1]School of Computer Science, University of Sydney, Sydney, Australia [2]NVIDIA, Seattle, USA. Correspondence to: Payal Bawa <pbaw4818@uni.sydney.edu.au>.

Presented at the ICML 2021 Workshop on Theoretical Foundations of Reinforcement Learning.

formance by using an ensemble of critics;

- Theoretical upper and lower bounds on the estimation biases validated by empirical results;
- Experimental results on OpenAI benchmark suites that demonstrate improved performance with robust policies.

2. Background

2.1. Estimation biases in Actor Critic methods

To motivate our method, we start with the theoretical analysis of estimation bias in actor critic methods. We reiterate the overestimation bias analysis in Fujimoto et al. (2018) and extend it to present the underestimation bias analysis. In actor critic methods we use function approximators for both Q function and the policy. We consider a parameterized critic $Q_\theta(s_t, a_t)$ with parameters θ and a parameterized policy $\pi_\phi(a_t \mid s_t)$ with parameters ϕ. The policy is updated with respect to the estimates of the Q function. We assume that the policy is updated using deterministic policy gradient. Given the current policy with parameters ϕ, let the updated parameters of the policy be ϕ_{approx} where ϕ_{approx} are obtained by updating with respect to the approximate critic. Let ϕ_{true} be the parameters of the policy update with respect to true Q values Q^{true}(s,a). Then the update rules are:

$$\phi_{approx} = \phi + \frac{\alpha}{Z_1}\mathbb{E}_{s\sim p_\pi}[\nabla_\phi \pi_\phi(s)\nabla_a Q_\theta(s,a) \mid_{a=\pi_\phi(s)}]$$
$$\phi_{true} = \phi + \frac{\alpha}{Z_2}\mathbb{E}_{s\sim p_\pi}[\nabla_\phi \pi_\phi(s)\nabla_a Q^{true}(s,a) \mid_{a=\pi_\phi(s)}], \tag{9}$$

where Z_1 and Z_2 are normalization constants for the gradients. Since the gradient direction is a local maximizer there exists an $\epsilon_1 > 0$ such that for $\alpha \leq \epsilon_1$:

$$\mathbb{E}[Q_\theta(s, \pi_{approx}(s))] \geq \mathbb{E}[Q_\theta(s, \pi_{true}(s))], \tag{10}$$

where π_{approx} and π_{true} are policies with parameters ϕ_{approx} and ϕ_{true} respectively. Conversely, there exists an $\epsilon_2 > 0$ such that for $\alpha \leq \epsilon_2$, the true value of π_{approx} is bounded above by the true value of π_{true},

$$\mathbb{E}[Q^{true}(s, \pi_{true}(s))] \geq \mathbb{E}[Q^{true}(s, \pi_{approx}(s))]. \tag{11}$$

Therefore if $\mathbb{E}[Q_\theta(s, \pi_{true}(s))] \geq \mathbb{E}[Q^{true}(s, \pi_{true}(s))]$ and $\alpha \leq min(\epsilon_1, \epsilon_2)$, then equations (10) and (11) imply that the Q values will be overestimated.

Similarly for underestimation bias analysis, let $Q_{ac}(s,a)$ be the estimate of the Q function where $Q_{ac}(s,a) = \min_{i=1,2} Q_{\theta^i}(s,a)$ (similar to Q-function estimate in TD3 and SAC). Let ϕ_{ac} be the parameters of the policy updated with respect to $Q_{ac}(s,a)$. Then the update rule for ϕ_{ac} is:

$$\phi_{ac} = \phi + \frac{\alpha}{Z_3}\mathbb{E}_{s\sim p_\pi}[\nabla_\phi \pi_\phi(s)\nabla_a Q_{ac}(s,a) \mid_{a=\pi\phi(s)}] \tag{12}$$

where Z_3 is the normalization constant for the gradient. The gradient direction is a local maximizer of the minimum of the two Q function estimates. The policy ϕ_{ac} is optimized with respect to the lowerbound of the dual estimates of Q function. Therefore the true Q value of an action under policy ϕ_{ac} is atleast as large as the approximate one,

$$\mathbb{E}[Q^{true}(s, \pi_{ac}(s))] \geq \mathbb{E}[Q_{ac}(s, \pi_{ac}(s))]. \tag{13}$$

Then equation (13) implies that the Q values will be consistently underestimated.

3. Bagged Critics for Continuous Control

TD3 and SAC address overestimation by iteratively maximizing the lower confidence bound on the Q-function. This in return induces underestimation. Unlike overestimation bias, underestimation bias is not propagated through policy updates but it ultimately hurts the performance in practice. We introduce an ensemble based approach to learning in Bagged Critics for Continuous Control (BC3) to mitigate both overestimation and underestimation biases. BC3 replaces the lower bound on the Q function in TD3 and SAC with an ensemble of critics. BC3 learns the ensemble of critics by first randomly initializing each neural network. Each member is then trained independently on different set of experience samples from the buffer $\mathcal{B}$ and the aggregated output of the ensemble is used as target value in the modified Bellman backup operator. The combination of random initialization and bagging (Breiman, 1996) reduces the variance in the action value estimates and offsets the estimation biases.

We constructed BC3 using the SAC framework but it can be extended to any off-policy actor critic algorithm. Similar to SAC, the critics represent estimates of the soft Q-function. We consider a parameterized ensemble of critics Q_{BC3} :

$$Q_{BC3}(s_t, a_t) = \frac{1}{K}\sum_{k=1}^{K} Q(s_t, a_t; \theta_k). \tag{15}$$

The Q-functions are learned by minimizing the individual Bellman residual $J(\theta_k) = \frac{1}{2}(Q_{\theta_k}(s_t,a_t) - (r(s_t,a_t) + \gamma\ \mathbb{E}_{s_{t+1}}[Q_{\overline{BC3}}(s_t,a_t)]))$. The parameters of each Q-function are optimized using stochastic gradient descent:

$$\hat{\nabla}_{\theta_k} J(\theta_k) = \nabla_{\theta_k} Q_{\theta_k}(s_t,a_t)(Q_{\theta_k}(s_t,a_t) - (r(s_t,a_t) + \gamma\ (Q_{\overline{BC3}}(s_{t+1},a_{t+1}) - \alpha \log \pi_\phi(a_{t+1} \mid s_{t+1})))), \tag{16}$$

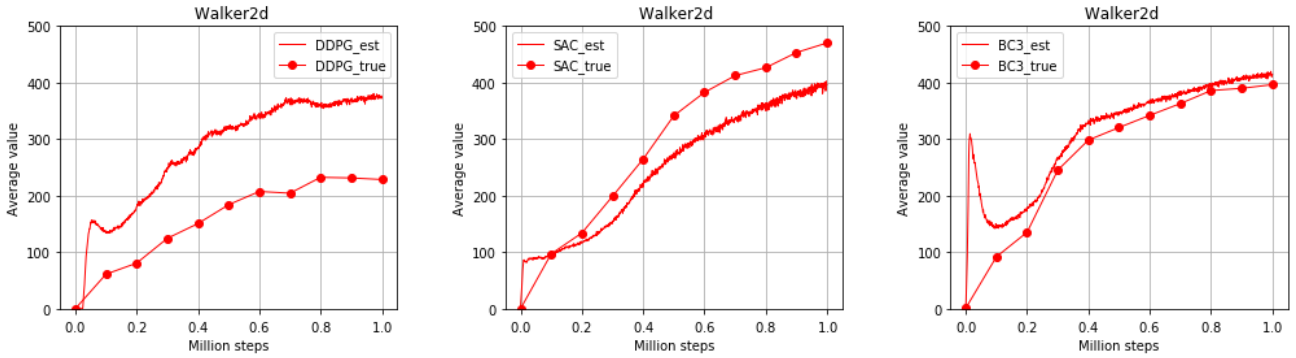

Figure 1. Comparison of estimation values of DDPG, SAC and BC3 on Walker2d-v2 environment. DDPG leads to overestimation. SAC leads to an underestimation. BC3 estimates are closer to the true values.The absolute bias of BC3 is less than the absolute bias in DDPG and SAC.

where $Q_{\overline{BC3}}(s_t, a_t)$ is an ensemble of target soft Q-functions with parameters $\overline{\theta}_1, \cdots, \overline{\theta}_K$. The parameters $\overline{\theta}_1, \cdots, \overline{\theta}_K$ are obtained from the exponentially moving average of the corresponding soft Q-function weights. Similar to SAC, we use a tractable policy and its parameters are optimized by minimizing the KL divergence $J_\phi = \mathbb{E}_{s_t \sim \mathcal{B}}[\mathbb{E}_{a_t \sim \phi_\phi}[\alpha \log \pi_\phi(a_t \mid s_t) - Q_{BC3}(s_t, a_t)]]$. Applying the reparameterization trick, the parameters are learned using the gradient:

$$\hat{\nabla}_\phi J(\phi) = \nabla_\phi \alpha \log \pi_\phi(a_t \mid s_t) + (\nabla_{a_t} \log \pi_\phi(a_t \mid s_t) - \nabla_{a_t} Q_{BC3}(s_t, a_t)) \nabla_\phi f_\phi(\epsilon_t; s_t), \tag{17}$$

where ϵ_t is the noise sampled from a Gaussian distribution and $f_\phi(\epsilon_t; s_t)$ is the reparamaterization using neural network transformation for the policy. The practical BC3 algorithm is presented in Algorithm 1 in appendix A.

3.1. Estimation bias reduction in BC3

We next prove that BC3 reduces the overestimation bias and improves the underestimation bias in critics. The following proofs follow the format presented in Pan et al. (2020).

Theorem 3.1. *Let $\mathcal{T}_{DDPG}$ and $\mathcal{T}_{BC3}$ be the state action value estimates for critics in algorithms DDPG and BC3 respectively. Let bias in state-action value estimates be: $bias(\mathcal{T}) = \mathbb{E}[Q(s_{t+1}, a_{t+1}; \theta)] - \mathbb{E}[Q(s_{t+1}, \pi_(s_{t+1}; \phi); \theta^{true})$ where θ^{true} are the parameters of true value function. Assume that the actor is a local maximizer with respect to the critic with a clipped action space A_l, then $bias(\mathcal{T}_{DDPG}) \geq bias(\mathcal{T}_{BC3})$.*

Proof. Proof in Appendix A □

Theorem 3.2. *Let $\mathcal{T}_{SAC}$ and $\mathcal{T}_{BC3}$ be the state action value estimates for critics in algorithms SAC and BCCC respectively. Assume that the actor is a local maximizer with respect to the critic, then $bias(\mathcal{T}_{BC3}) \geq bias(\mathcal{T}_{SAC})$.*

Proof. Proof in Appendix A □

We next present empirical results to validate our theoretical proofs. We train DDPG, SAC and BC3 on the Walker2d environment. At every 100000 time-steps we sample 1000 state-action pairs from the experience buffer and compute their value estimate using the critic. We approximate the true values for each of these 1000 state-action pair by rolling out the current policy 10 times starting from the pair and average the observed discounted return. Figure 1 shows the estimated and true Q values for each algorithm. DDPG has an overestimation bias with value estimates significantly higher than the true values. SAC has an underestimation bias with value estimates lower than the true values. BC3 estimates on the other hand are more accurate compared to DDPG and SAC.

4. Experiments

To evaluate the algorithm, we measure its performance on a suite of MuJoCo continuous control tasks (Todorov et al., 2012), interfaced through OpenAI Gym (Brockman et al., 2016). We use the original set of tasks without any modifications. BC3 is built on top of SAC. We use the same hyperparameters as the author's implementation of SAC. The number of networks used in the ensemble is fixed to 5. We compare our method to other off-policy deep learning algorithms like DDPG, TD3 and the state of art off-policy algorithm SAC. We compare the results using author-provided implementations. The temperature hyperparameter is fixed to $\alpha = 0.2$ and the gradient step is fixed to 1 for both BC3 and SAC. We train each algorithm with five different random seeds. We evaluate the algorithms every 1000 envi-

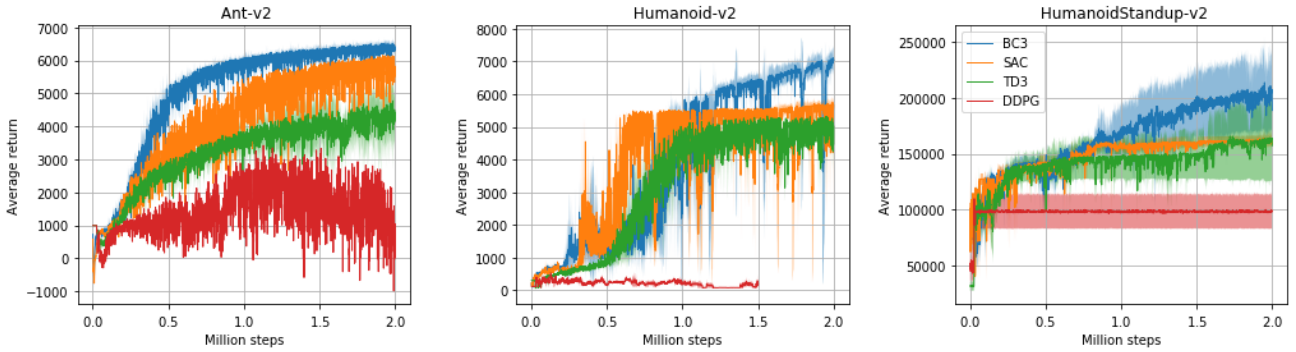

Figure 2. Learning curves for high dimensional continuous control benchmarks. Solid curves correspond to mean and shaded region correspond to one standard deviation. BC3 outperforms other methods in the most challenging tasks.

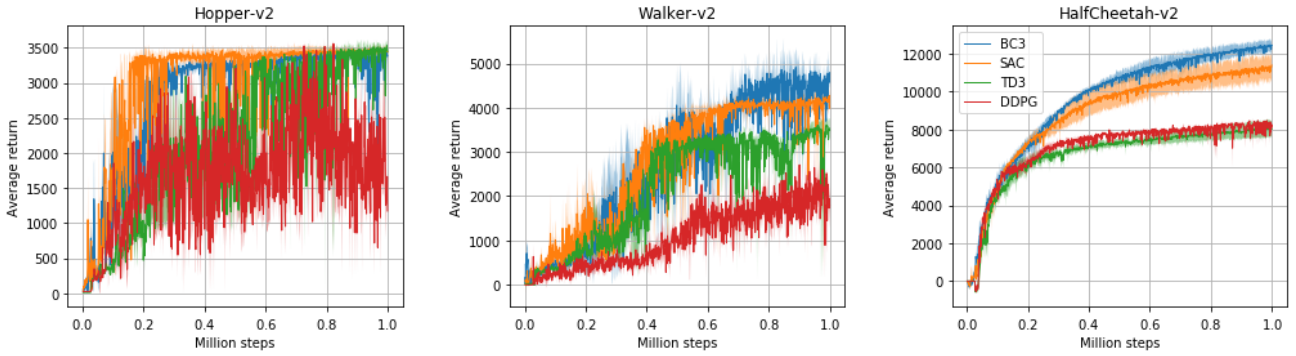

Figure 3. Learning curves for low dimensional continuous control benchmarks. Solid curves correspond to mean and shaded region correspond to one standard deviation. BC3 performs consistently across the tasks and outperforms on HalfCheetah-v2.

ronment steps as an average of 10 rollouts. The solid curves correspond to the mean and the shaded region corresponds to one standard deviation over the five trials.

Figure 2 shows learning curves for various algorithms on complex tasks. The results show that BC3 outperforms other algorithms on high dimensional tasks: Humanoid-v2 (action space dimensionality: 17, state space dimensionality: 376), HumanoidStandup-v2 (action space dimensionality: 17, state space dimensionality: 376) and Ant-v2 (action space dimensionality: 8, state space dimensionality: 111). BC3 performs better both in terms of sample efficiency and average rewards. On the other hand, DDPG fails to make any progress on Humanoid. Figure 3 shows the learning curves for various algorithms on low to medium dimensional tasks. The results show that BC3 performs on par with other algorithms on simple tasks: Hopper-v2 (action space dimensionality: 3, state space dimensionality: 11), Walker2d-v2 (action space dimensionality: 6, state space dimensionality: 17), HalfCheetah-v2 (action space dimensionality: 6, state space dimensionality: 17).

4.1. Bagging+random initialization vs random initialization

In most off-policy deep reinforcement learning algorithms, the learners are trained after each time step using a subset of samples from the experience buffer, unlike regression and classification tasks where the entire training dataset is used. Due to the small training dataset used at each step, random initialization of Q functions is not enough to mitigate the estimation biases. Combining random initialization with bagging on the other hand helps inject additional randomization. The combined approach encourages diversity among the neural networks so that the averaged prediction improves over the individual outputs. Figure 4 compares the two approaches on complex Humanoid task. Both approaches use $n = 5$ neural networks. The original BC3 algorithm with random initialization and bagging outperforms BC3 with just random initialization of the members of the ensemble.

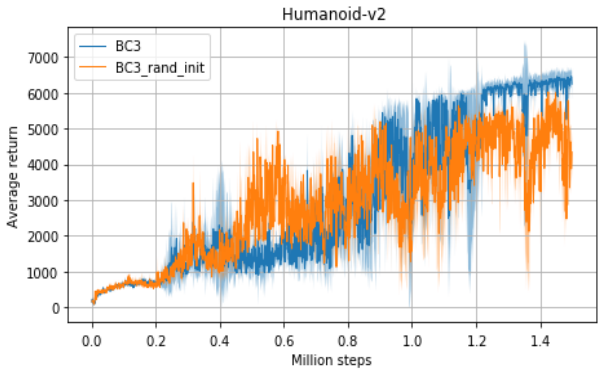

Figure 4. Learning curves for BC3 and BC3 with only random initialization. Solid curves correspond to mean and shaded region correspond to one standard deviation.

4.2. Number of neural networks

We analyze the effect of the number of neural networks in the ensemble on complex tasks. We experimented with $n \in \{2, 5, 7\}$. The results in Figure 5 show that larger ensembles are better than smaller ensembles especially for high dimensional tasks. The performance improvement saturates at approximately $n = 5$.

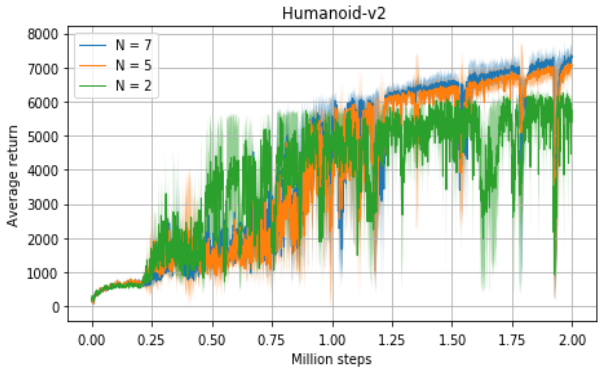

Figure 5. Learning curves for BC3 with different number of neural networks in the ensemble. Solid curves correspond to mean and shaded region correspond to one standard deviation.

5. Convergence Analysis

We next prove convergence of our method similar to the proof of convergence for SARSA (Singh et al., 2000), Double Q-learning (Van Hasselt, 2010) and TD3. We consider a version of our method for finite MDP setting, with tabular value estimates $Q^1, Q^2, \ldots, Q^K$. The action selected at any time-step is $a^* = \arg\max_a Q^{BCCC}(s_t, a)$ and is then used to perform updates by setting target y:

$$a^* = \arg\max_a Q^{BCCC}(s_{t+1}, a)$$
$$y = r_t + \gamma\, Q^{BCCC}(s_{t+1}, a^*),$$

where $r_t = r(s_t, a_t) + \arg\max_a \mathcal{H}(\cdot \mid s_{t+1})$ is the entropy augmented reward. We start with lemma 5.1. The proof for this lemma can be found in Singh et al. (2000):

Lemma 5.1. *Consider a stochastic process $(\xi_t, \delta_t, F_t), t \geq 0$ where $\xi_t, F_t : X \to \mathbb{R}$ satisfy the following equation: $\Delta_{t+1}(x_t) = (1-\xi_t(x_t))\Delta_t(x_t)+\xi_t(x_t)F_t(x_t)$, where $x_t \in X$ and $t = 1, 2, \ldots$. Let P_t be the sequence of increasing sigma-fields such that ξ_0 and Δ_0 are P_0 measurable and ξ_t, Δ_t and F_t are P_t measurable, $t = 0, 1, 2, \ldots$. Assume the following hold: 1) The set X is finite. 2) $\xi_t(x_t) \in [0, 1], \sum_t \xi_t(x_t) = \infty, \sum_t (\xi_t(x_t))^2 < \infty$ with probability 1 and $\forall x \neq x_t : \xi_t = 0$. 3) $\|\mathbb{E}[F_t \mid P_t]\| \leq \kappa \|\Delta_t\| + c_t$ where $\kappa \in [0, 1)$ and c_t converges to 0 with probability 1. 4) $Var[F_t(x_t) \mid P_t] \leq K(1 + \kappa \|\Delta_t\|^2)$ where K is some constant. Here $\|\cdot\|$ denotes the maximum norm. Then Δ_t converges to 0 with probability 1.*

We use the above lemma to prove convergence of BCCC.

Theorem 5.2. *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) All $Q^k, k = 1, \ldots k$ receive an infinite number of updates. 6) The learning rate 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) $Var[r(s, a)] < \infty, \forall s, a$. Then BCCC converges to the optimal value function Q^* with probability 1.*

Proof. Proof in Appendix B □

6. Conclusion

Overestimation bias is a major impediment for value based actor critic algorithms. Current solutions overcome overestimation bias by introducing underestimation bias. However, both forms of estimation biases result in sub-optimal policies and sub-par performances especially in high dimensional action spaces. In this work, we propose an ensemble-based actor critic algorithm BC3, which uses a combined approach of random intialization and bagging to introduce diversity in the ensemble. Our theoretical results show that the combined approach mitigates both overestimation and underestimation bias and converges to the optimal policy. The theoretical results are supported by empirical results where BC3 outperforms state-of-the-art SAC on challenging RL tasks. Our results suggest exploring methods that increase diversity in ensembles with quantitative bounds on bias are an exciting avenue for future work.

References

L. Breiman. Bagging predictors. In *Machine learning*, page 123–140, 1996.

G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W Zaremba. Openai gym, 2016.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, page 1587–1596, 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 *International Conference on Machine Learning*, 2018.

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. In *ICLR*, 2016.

Ling Pan, Qingpeng Cai, and Longbo Huang. Softmax deep double deterministic policy gradients. In *NeurIPS*, 2020.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *ICML*, 2014.

Satinder Singh, Tommi Jaakkola, Michael Littman, and Csaba Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 38:287–308, 03 2000.

E. Smith and R. L. Winkler. The optimizer's curse: Skepticism and post decision surprise indecision analysis. In *Management Science*, page 311–322, 2006.

Richard S Sutton and Andrew G Barto, editors. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, 2018.

E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS 2012)*, page 5026–5033, 2012.

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

B. D. Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. PhD thesis, Carnegie Mellon University, 2010.

A. Theoretical proofs for estimation bias reduction in BC3

Theorem A.1. *Let $\mathcal{T}_{DDPG}$ and $\mathcal{T}_{BC3}$ be the state action value estimates for critics in algorithms DDPG and BC3 respectively. Let bias in state-action value estimates be: $bias(\mathcal{T}) = \mathbb{E}[Q(s_{t+1}, a_{t+1}; \theta)] - \mathbb{E}[Q(s_{t+1}, \pi_(s_{t+1}; \phi); \theta^{true})$ where θ^{true} are the parameters of true value function. Assume that the actor is a local maximizer with respect to the critic with a clipped action space A_l, then $bias(\mathcal{T}_{DDPG}) \geq bias(\mathcal{T}_{BC3})$.*

Proof. By definition,we have

$$\mathcal{T}_{\text{DDPG}}(s_{t+1}) = Q(s_{t+1}, \pi(s_{t+1}; \phi); \theta)$$
$$\mathcal{T}_{\text{BC3}}(s_{t+1}) = \frac{1}{K}\sum_{k=1}^{K} Q(s_{t+1}, \pi(s_{t+1}; \phi); \theta_k)$$

The actor is a local maximizer with respect to the critic. Hence the action selected by policy $\pi(s_{t+1}; \phi)$ is a local maximimum of $Q(s_{t+1}, \cdot; \theta)$ of any state s_{t+1} for DDPG and a local maximum of the ensemble for BC3. Therefore

$$\mathcal{T}_{\text{DDPG}}(s_{t+1}) = \max_{a \in A_l} Q(s_{t+1}, a; \theta)$$
$$\mathcal{T}_{\text{BC3}}(s_{t+1}) = \max_{a \in A_l} \frac{1}{K}\sum_{k=1}^{K} Q(s_{t+1}, a; \theta_k).$$

Since each critic in the ensemble for BC3 is trained independently using different set of samples from experience buffer, the action selected by policy $\pi(s_{t+1}; \phi)$ is a local maximum of the ensemble and not of individual critic. Hence $\frac{1}{K}\sum_{k=1}^{K} Q(s_{t+1}, a; \theta_k)$ is an unbiased estimator of $E[Q(s_{t+1}, a; \theta)]$ (Van Hasselt, 2010). On the other hand, $\max_{a \in A_l} Q(s_{t+1}, a; \theta)$ is an unbiased estimator of $\mathbb{E}[\max_{a \in A_l} Q(s_{t+1}, a; \theta)]$ but a biased estimator of $\max_{a \in A_l} \mathbb{E}[Q(s_{t+1}, a; \theta)]$ (Smith and Winkler, 2006; Van Hasselt, 2010). Therefore:

$$\begin{aligned}
\mathcal{T}_{\text{DDPG}}(s_{t+1}) &= \max_{a \in A_l} Q(s_{t+1}, a; \theta) \\
&\approx \mathbb{E}[\max_{a \in A_l} Q(s_{t+1}, a; \theta)] \\
&\geq \max_{a \in A_l} \mathbb{E}[Q(s_{t+1}, a; \theta)] \\
&\approx \max_{a \in A_l} \frac{1}{K}\sum_{k=1}^{K} Q_k(s_{t+1}, a; \theta_k) \\
&= \mathcal{T}_{\text{BC3}}(s_{t+1}).
\end{aligned} \tag{18}$$

By definition, $bias(\mathcal{T}_{\text{DDPG}}) = \mathcal{T}_{\text{DDPG}}(s_{t+1}) - \mathbb{E}[Q(s_{t+1}, \pi_(s_{t+1}; \phi); \theta^{true})$ and $bias(\mathcal{T}_{\text{BC3}}) = \mathcal{T}_{\text{BC3}}(s_{t+1}) - \mathbb{E}[Q(s_{t+1}, \pi_(s_{t+1}; \phi); \theta^{true})$. Hence, from equation (18) we have $bias(\mathcal{T}_{\text{DDPG}}) \geq bias(\mathcal{T}_{\text{BC3}})$. □

Theorem A.2. *Let $\mathcal{T}_{SAC}$ and $\mathcal{T}_{BC3}$ be the state action value estimates for critics in algorithms SAC and BC3 respectively. Assume that the actor is a local maximizer with respect to the critic, then $bias(\mathcal{T}_{BC3}) \geq bias(\mathcal{T}_{SAC})$.*

Proof. By definition,we have

$$\mathcal{T}_{\text{SAC}}(s_{t+1}) = \min_{i=1,2}(Q_i(s_{t+1}, \pi(s_{t+1}; \phi); \theta_i).$$

The state action value estimate is the minimum of the two Q functions. The action a selected by policy $\pi(s_{t+1}; \phi)$ of SAC for any state s_{t+1} is a local maximum of the min of the two Q functions. Therefore $\mathcal{T}_{\text{SAC}}(s_{t+1}) = \min_{i=1,2}(\max_{a \in A_l} Q_i(s_{t+1}, a; \theta_i))$. On the other hand, the action selected by policy $\pi(s_{t+1}; \phi)$ for BC3 is a local

Algorithm 1 Bagged Critic for Continuous Control

Initialize $K \in \mathbb{N}_+$ critic networks $Q_{s,a \mid \theta_k}$ with random parameters θ_k for $k \in 1, 2. \ldots, k$ and actor network $\pi_\phi(s)$ with parameters ϕ
Initialize $K \in \mathbb{N}_+$ corresponding target networks with parameters $\theta_k^{'} \leftarrow \theta_1$,
Initialize experience replay buffer $\mathcal{B}$
for each iteration **do**
  for each environment step **do**
    Sample action from policy $a_t \sim \pi_\phi(a_t \mid s_t)$
    Sample transition from the environment $s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)$
    Store transition in replay buffer $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s_t, a_t, r(s_t, a_t,), s_{t+1})\}$
  end for
  for each gradient step **do**
    for $k \in 1, 2. \ldots, K$ **do**
      Sample mini-batches of N transitions (s, a, r, s') from $\mathcal{B}$
      Update Q function parameters $\theta_k \leftarrow \theta_k - \lambda_Q \hat{\nabla}_{\theta_k} J_Q(\theta)_k$
      Update target network parameters $\bar{\theta}_k \leftarrow \tau\theta_k + (1-\tau)\bar{\theta}_k$
    end for
    Update policy parameters $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
  end for
end for

maximum of the ensemble. Therefore:

$$\begin{aligned}\mathcal{T}_{\text{BC3}}(s_{t+1}) &= \max_{a \in A_l} \frac{1}{K} \sum_{k=1}^{K} Q_k(s_{t+1}, a; \theta_k) \\ &\geq \min_{i=1,2} (\max_{a \in A_l} Q_i(s_{t+1}, a; \theta_i)) \\ &= \mathcal{T}_{\text{SAC}}(s_{t+1}). \end{aligned} \tag{19}$$

By definition, $bias(\mathcal{T}_{\text{SAC}}) = \mathcal{T}_{\text{SAC}}(s_{t+1}) - \mathbb{E}[Q(s_{t+1}, \pi_(s_{t+1}; \phi); \theta^{true})$ and $bias(\mathcal{T}_{\text{BC3}}) = \mathcal{T}_{\text{BC3}}(s_{t+1}) - \mathbb{E}[Q(s_{t+1}, \pi_(s_{t+1}; \phi); \theta^{true})$. Hence $bias(\mathcal{T}_{\text{BC3}}) \geq bias(\mathcal{T}_{\text{SAC}})$. □

Thus BC3 reduces the overestimation bias (DDPG) and improves the underestimation bias (SAC) in critics.

B. Proof of Convergence

Theorem B.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) All* $Q^k, k = 1, \dots k$ *receive an infinite number of updates. 6) The learning rate 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)* $Var[r(s, a)] < \infty, \forall s, a$. *Then BC3 converges to the optimal value function* Q^* *with probability 1.*

Proof. We apply lemma 5.1 with $P_t = \{Q_0^1, \dots, Q_0^k, s_0, a_0, r_1, s_1, \dots, s_t, a_t\}$, $X = S \times A, \Delta_t = Q_t^1 - Q^*, \xi_t = a_t$. Condition 1 and 4 of the lemma holds by condition 2 and 7 of the theorem respectively. Lemma condition 6 holds by theorem condition 6 and with our selection of $\xi_t = a_t$. Defining $a^* = \arg\max_a Q^{BC3}(s_{t+1}, a)$ we have:

$$\begin{aligned}
\Delta_{t+1}(s_{t+1}, a_{t+1}) &= (1 - \alpha_t(s_t, a_t)) \times (Q^1(s_t, a_t) - Q^*(s_t, a_t)) + \alpha_t(s_t, a_t)(r_t + \gamma Q^{BC3}(s_t, a_t) - Q^*(s_t, a_t)) \\
&= (1 - \alpha_t(s_t, a_t))(Q^1(s_t, a_t) - Q^*(s_t, a_t)) + \alpha_t(s_t, a_t)(r_t + \gamma \frac{1}{K} \sum_{k=1}^{K} Q^k(s_t, a_t) - Q^*(s_t, a_t)) \\
&= (1 - \alpha_t(s_t, a_t))(Q^1(s_t, a_t) - Q^*(s_t, a_t)) + \frac{\alpha_t(s_t, a_t)}{K}(Kr_t + \gamma \sum_{k=1}^{K} Q^k(s_t, a_t) - KQ^*(s_t, a_t)) \\
&= (1 - \alpha_t(s_t, a_t))\Delta_t(s_t, a_t) + \frac{\alpha_t(s_t, a_t)}{K} F_t(s_t, a_t), \qquad (20)
\end{aligned}$$

where $F_t(s_t, a_t)$ is

$$\begin{aligned}
F_t(s_t, a_t) &= Kr_t + \gamma \sum_{k=1}^{K} Q^k(s_t, a_t) - KQ^*(s_t, a_t) \\
&= Kr_t + \gamma \sum_{k=1}^{K} Q^k(s_t, a_t) - KQ^*(s_t, a_t) + KQ^1(s_t, a_t) - KQ^1(s_t, a_t) \\
&= KF_t^Q(s_t, a_t) + c_t, \qquad (21)
\end{aligned}$$

where $F_t^Q(s_t, a_t) = Q^1(s_t, a_t) - Q^*(s_t, a_t)$ is value of F_t under standard Q-learning and $c_t = \sum_{k=2}^{K} Q^k(s_t, a_t) - (K - 1)Q^1(s_t, a_t)$. As $\mathbb{E}[F_t^Q \mid P_t] \leq \gamma \left\|\Delta\right\|_t$ is a known result, the condition 3 of lemma 5.1 holds if it can be shown that c_t converges to 0 with probability 1.

Let $y = r_t + \gamma Q^{BC3}(s_t, a_t)$ and $\Delta^{k1}(s_t, a_t) = Q^k(s_t, a_t) - Q^1(s_t, a_t)$ where $k = 2, \dots, K$. Then c_t converges to 0 if all Δ^{k1}, $k = 2...K$ converge to zero. The update for any Δ^{k1} at time t is sum of updates of $Q^1, \dots, Q^K$. The update Δ^{k1} is :

$$\begin{aligned}
\Delta_{t+1}^{k1} &= \Delta_t^{k1}(s_t, a_t) + \alpha_t(s_t, a_t)((y - Q_t^k(s_t, a_t)) - (y - Q_t^1(s_t, a_t))) \\
&= \Delta_t^{k1}(s_t, a_t) + \alpha_t(s_t, a_t)(Q_t^1(s_t, a_t) - Q_t^k(s_t, a_t))) \\
&= (1 - \alpha_t(s_t, a_t))\Delta_t^{k1}(s_t, a_t). \qquad (22)
\end{aligned}$$

As all Δ_t^{k1}, $k = 2, \dots, K$ converge to 0, we have satisfied condition 3 of the lemma, implying $Q^1(s_t, a_t)$ converges to $Q^*(s_t, a_t)$. Using the same arguments we can prove that any $Q^k(s_t, a_t)$ converges to $Q^*(s_t, a_t)$ by choosing $\Delta_t = Q^k(s_t, a_t) - Q^*(s_t, a_t)$ thus proving the theorem. □

C. Hyperparameters

The table 1 lists the parameters used to implement BC3.

Table 1. Hyperparameters for BC3.

Hyperparameter	BC3
Optimizer	Adam
Learning rate	10^{-4}
Discount γ	0.99
Replay buffer size	10^6
Number of critics	5
Number of hidden layers (all networks)	2
Number of hidden units per layer	256
Minibatch size	256
Nonlinearity	ReLU
Target smoothing coefficient τ	0.005
Target update interval	1
Gradient steps per iteration	1
Environment steps per iteration	1
Temperature coefficient α	0.02