

Smart Sampling: Self-Attention and Bootstrapping for Improved Ensembled Q-Learning

Muhammad Junaid Khan

Dept. of Computer Science
University of Central Florida
Orlando, USA
muhammad.junaid.khan@ucf.edu

Syed Hammad Ahmed

Dept. of Computer Science
University of Central Florida
Orlando, USA
syed.hammad.ahmed@ucf.edu

Gita Sukthankar

Dept. of Computer Science
University of Central Florida
Orlando, USA
gita.sukthankar@ucf.edu

Abstract

We present a novel method aimed at enhancing the sample efficiency of ensemble Q learning. Our proposed approach integrates multi-head self-attention into the ensembled Q networks while bootstrapping the state-action pairs ingested by the ensemble. This not only results in performance improvements over the original REDQ (Chen et al. 2021) and its variant DroQ (Hiraoka et al. 2022), thereby enhancing Q predictions, but also effectively reduces both the average normalized bias and standard deviation of normalized bias within Q-function ensembles. Importantly, our method also performs well even in scenarios with a low update-to-data (UTD) ratio. Notably, the implementation of our proposed method is straightforward, requiring minimal modifications to the base model.

Introduction

Designing reinforcement learning algorithms that learn rapidly with limited data remains a significant research challenge. While previous approaches have solved complex control tasks, they require huge amount of samples for training and solving the task (Andrychowicz et al. 2020; Mendonca et al. 2019). Newer techniques focus on achieving a high update-to-data (UTD) ratio which ensures fewer environmental interactions but high sample efficiency. RL methods such as Model-based Policy Optimization (MBPO) (Janner et al. 2019) achieve a high UTD ratio by using both real data and fake data generated by the model to achieve sample efficiency. In contrast, the model-free Soft Actor Critic (SAC) (Haarnoja et al. 2018) approach has a relatively low UTD ratio of 1.

Two recent model-free ensemble methods use a very high UTD ratio to achieve greater sample efficiency: REDQ (Chen et al. 2021) and DroQ (Hiraoka et al. 2022). Having a high UTD ratio risks creating estimation bias since many model updates are made after a small number of environmental interactions. To prevent this, REDQ employs a Q-function ensemble and reduces the estimation bias by employing in-target minimization over a subset of the ensemble. Similarly, DroQ, which is based on REDQ, in-

jects model uncertainty into a smaller ensemble by using Q-learners with dropout and layer normalization to minimize estimation bias.

Our proposed method uses bootstrapping (Efron 1992) to generate the samples used by the Q-learning ensemble. Several recent deep RL algorithms have leveraged bootstrapping to improve sample complexity but not within ensemble learners. Rahman and Xue (2023) bootstrap experience trajectory clusters to improve agent generalization. Gelada and Bellemare (2019) tackle the problem of off-policy learning with covariate or distribution shift by using bootstrapping to reweight and resample the data from the behavior policy in order to align it with target policy while reducing the bias introduced due to importance sampling.

Here we use multi-head self attention within the individual Q-learners. Self-attention was first introduced by Vaswani et al. (2017) and has been effectively employed in all transformer-based approaches in natural language tasks. Recently, many deep RL approaches have showed its effectiveness in both single-agent and multi-agent problems (Zheng, Zhang, and Grover 2022; Hu et al. 2021; Upadhyay et al. 2019; Khan, Ahmed, and Sukthankar 2022).

This paper introduces a variant of the REDQ and DroQ methods which, in addition to an ensemble of Q-functions, dropout layer, and layer normalization, exploits multi-head self-attention (referred as *MHA*), identity connections, and bootstrapping mechanisms to further improve performance and reduce estimation bias. Our approach provides the following benefits:

- With bootstrapping we create multiple sub-samples of the agent’s experiences, thus helping the agent to utilize its experiences effectively which leads to better state space exploration.
- In addition, MHA effectively captures the temporal relationship between sub-sampled state-action pairs while taking into account the future states and rewards, and learning a variety of dependencies amongst state-action pairs.
- Experiments show that our method effectively improves sample efficiency while reducing the estimation error. We achieve comparable performance to REDQ even with low UTD settings.

We demonstrate the performance of our algorithm vs. REDQ and DroQ in four challenging OpenAI Gym environments:

Ant-v2, Hopper-v2, Humanoid-v2, and Walker2d-v2.

Related Work

Model-free deep RL approaches such as TRPO (Schulman et al. 2015), PPO (Schulman et al. 2017) and A3C (Mnih et al. 2016) have been applied to a variety of decision making and control tasks. While these approaches provide reasonable performance, they suffer from poor sample efficiency due to on-policy learning, requiring new sampling at each step. To tackle these challenges, Soft Actor Critic (Haarnoja et al. 2018), an off-policy learning method, was proposed. SAC achieves a higher sample efficiency, but still uses a very low update-to-data (UTD) ratio which somewhat limits the potential sample efficiency.

In contrast, MBPO (Janner et al. 2019) represents a model-based approach that manages the trade-offs between leveraging a model to generate data and the risks of using an inaccurate model. It integrates both real data from the environment and synthetic data generated from the model, and utilizes higher UTD ratio, usually 20 to 40, to achieve better sample efficiency.

Given the success of MBPO with a high UTD ratio, many recent deep RL methods use a high UTD ratio to achieve better sample complexity than model-based deep RL methods (Shen et al. 2020; Lai et al. 2020). However, the higher UTD ratio comes at the cost of overestimation bias.

In the ongoing quest to improve the performance of deep RL methods, researchers have proposed diverse approaches. Notable strategies include ensembles of Q-functions (Faußer and Schwenker 2015; Osband et al. 2016; Lee et al. 2021), integration of dropout transition models (Gal and Ghahramani 2016; He et al. 2022), and application of normalization techniques such as batch normalization (Ioffe and Szegedy 2015) and layer normalization (Ba, Kiros, and Hinton 2016). While previous methods employed ensembles to capture model uncertainty in both target calculation and policy optimization, they did not specifically focus on the issue of overestimation.

To address the overestimation problem caused by the higher UTD ratio, Chen et al. (2021) proposed Randomized Ensembled Double Q-learning (REDQ). This model-free approach attains superior performance and similar sample efficiency to MBPO, mainly by using a higher UTD ratio. It mitigates the overestimation bias resulting from the increased UTD ratio by employing a large ensemble of Q-functions and then choosing a random subset of the Q-function ensembles for in-target minimization.

Hiraoka et al. (2022) were able to improve the computational performance of REDQ. To achieve this improvement, DroQ incorporates the techniques such as dropout (Srivastava et al. 2014) and layer normalization (Ba, Kiros, and Hinton 2016). Moreover, it aims to discover a policy capable of maximizing the expected return while incorporating an entropy bonus.

Method

Our method incorporates multi-head self-attention, identity connections, and bootstrapping into the REDQ architecture.

Our modifications are shown in Figure 1 while the pseudocode is presented in the Algorithm 1. Differences between the algorithms are highlighted in red.

Algorithm 1 Modified REDQ

- 1: Initialize policy parameters θ , N Q-function parameters $\phi_i, i = 1, \dots, N$, empty replay buffer \mathcal{D} .
- 2: Set target parameters $\phi_{tar,i} \leftarrow \phi_i$, for $i = 1, 2, \dots, N$
- 3: **repeat**
- 4: Take an action $a_t \sim \pi_\theta(\cdot|s_t)$ and observe reward r_t , next state s_{t+1}
- 5: Update buffer $\mathcal{D} \leftarrow \mathcal{D} \cup (s_t, a_t, r_t, s_{t+1})$
- 6: **for** G updates **do**
- 7: Sample a mini-batch $\mathcal{B} = (s, a, r, s')$ from \mathcal{D}
- 8: **for** b_i in \mathcal{B} **do**
- 9: Select a sub-sample b^* from mini-batch \mathcal{B} with replacement
- 10: Apply multi-head self-attention to bootstrapped sample b^*

$$\text{Attention}(q, k, v) = \text{softmax}\left(\frac{qk^T}{\sqrt{d_k}}\right)v$$

- 11: Calculate Q for all \mathcal{M} of M distinct indices from $1, 2, \dots, N$ with bootstrapped sample b^*
- 12: Calculate target y as:

$$y = r + \gamma \left(\min_{i \in \mathcal{M}} Q_{\phi_{tar,i}}(s', \tilde{a}') - \alpha \log \pi_\theta(\tilde{a}'|s') \right),$$

$$\tilde{a}' \sim \pi_\theta(\cdot|s')$$

- 13: **end for**
- 14: **for** $i = 1, \dots, N$ **do**
- 15: Update ϕ_i with gradient descent using

$$\nabla_\phi \frac{1}{|\mathcal{B}|} \sum_{(s,a,r,s')} (Q_{\phi_i}(s, a) - y)^2$$

- 16: Update target networks with

$$\phi_{tar,i} \leftarrow \rho \phi_{tar,i} + (1 - \rho) \phi_i$$

- 17: **end for**
- 18: **end for**
- 19: Update policy parameters θ with gradient ascent using

$$\nabla_\theta \frac{1}{|\mathcal{B}|} \sum_{(s \in \mathcal{B})} \left(\frac{1}{N} \sum_{i=1}^N Q_{\phi_i}(s, \tilde{a}_\theta(s)) - \alpha \log \pi_\theta(\tilde{a}_\theta(s)|s) \right),$$

$$\tilde{a}_\theta(s) \sim \pi_\theta(\cdot|s)$$

- 20: **until** done
-

Similar to both REDQ and DroQ, our approach involves utilizing N ensembles of Q-functions, and we select a subset \mathcal{M} out of N for in-target minimization. Each Q-network is constructed using the original implementation of the REDQ

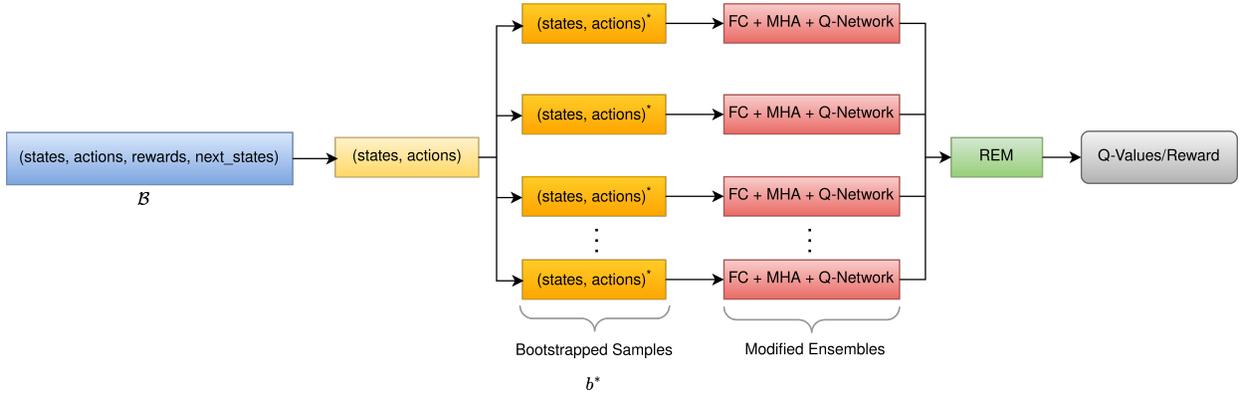


Figure 1: Our modified REDQ approach incorporates both the bootstrapping and MHA mechanisms. Both the states and the actions are first concatenated, and then multiple bootstrapped samples are drawn from the replay buffer for the ensemble of Q-learners. Individual Q-learners incorporate a fully connected layer and multi-head attention on top of a Q-network. The Q-network integrates elements from both the REDQ and DroQ implementations.

method. We also integrate the key modifications inspired by the DroQ, incorporating a dropout layer and layer normalization into each Q-network.

Moving forward, as depicted in Figure 1, we introduce our modifications to each Q-network described above. Each Q-network now boasts a fully connected layer and a multi-head self-attention layer, carefully positioned before the original Q-network implementation. This augmentation ensures the model is capable of capturing even richer dependencies and improves the overall expressive power of the model. During the training process, at each step, we draw a mini-batch \mathcal{B} from the replay buffer \mathcal{D} and generate multiple bootstrapped samples b^* from \mathcal{B} . We evaluated different sizes of the bootstrap sample, $|b^*| = \{2, 4, 8\}$; results are reported for $|b^*| = 4$. Each sample is comprised of concatenated state-action pairs. The incorporation of bootstrapped samples serves several crucial purposes: 1) these samples help agent learn from both its own prediction and those generated by the Q-network; 2) the agent updates its policy based on information it has while reducing the need for extensive exploration to gather new experiences and hence reducing environment interactions; 3) as the agent potentially encounters some state-action pairs multiple times, this repetition significantly amplifies the agent’s predictive capacity; 4) bootstrapping also reduces the variance.

Each bootstrapped sample b^* then undergoes transformation through a fully connected layer before being fed to the multi-head self-attention layer. This transformation ensures that all samples are in the same representation space and share a consistent embedding dimension, necessary to apply self-attention. The multi-head self-attention is calculated as given by Equations 1 and 2.

$$head_i = \text{Attention}(q, k, v) = \text{softmax}\left(\frac{qk^T}{\sqrt{d_k}}\right)v \quad (1)$$

$$\text{MultiHead}(q, k, v) = \text{concatenate}(head_i, \dots, head_H)w \quad (2)$$

where the matrices \mathbf{k} , \mathbf{v} , and $\mathbf{q} \in \mathbb{R}^{\mathcal{N} \times d}$ are calculated from

each b^* sub-sample. The d_k is the embedded dimension of the model set to either $d \in \mathbb{R}^{256}$ or $d \in \mathbb{R}^{512}$, depending upon the environment. H is the number of heads which is set to 8 for most of the environments.

The MHA provides significant advantages: 1) by applying attention to state-action pairs, the agent gains a better understanding of the relationship between states and actions, thereby enhancing temporal modeling; 2) it also aids the agent in optimizing its policy and making informed decisions while taking actions; 3) leveraging MHA, each head has the capacity to learn a distinct and meaningful relationships for a state-action pair; 4) and lastly, the MHA ensures that the state-action pairs are invariant to permutations.

While the combination of MHA and bootstrapping contributes to performance improvements over REDQ, in the case of DroQ, MHA typically does not enhance performance. Instead, it often leads to a declined performance. This discrepancy can be attributed to the simplicity of the DroQ network, where the inclusion of attention mechanisms adds unnecessary complexity, adversely affecting performance.

To address this issue, we propose a solution by incorporating identity connections, inspired by the concept introduced in ResNet (He et al. 2016). By removing MHA and integrating identity connections while retaining the bootstrapping approach, we effectively streamline the network architecture and mitigate the performance issues associated with unnecessary complexity.

In the REDQ framework, the ensemble size is typically set to $N = 10$, with a fixed random subset size of $M = 2$ and UTD is set to 20. To ensure a fair comparison in our initial experiments, we adopt the same configuration as outlined by the REDQ implementation, aligning our setup with their established parameters.

However, for our subsequent experiments, we diverge from the conventional REDQ configuration. We choose an ensemble size of $N = 5$, while still maintaining the subset size at $M = 2$ and reducing the UTD from 20 to 10. In both

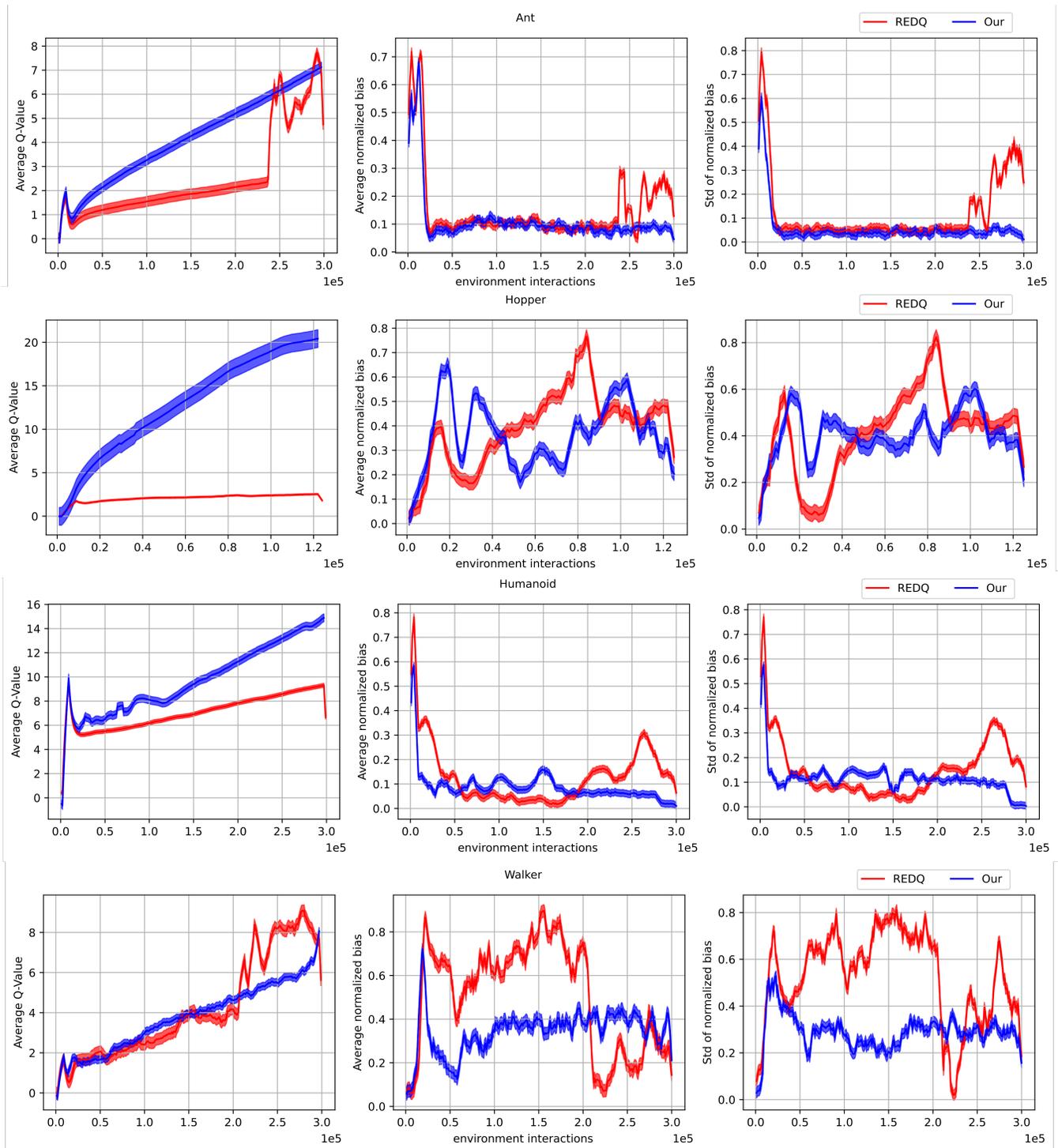


Figure 2: An evaluation of Q-value prediction, average normalized Q-bias, and standard deviation of normalized Q-bias: REDQ vs. our proposed approach. Results are based on three separate runs for each environment. Our method demonstrates enhanced Q-value predictions while effectively managing estimation bias at a level comparable to REDQ.

configurations, each individual Q-network is initialized randomly, but they all undergo updates with the same target. In each case, the target value is determined as follows:

$$\mathbb{E}_{\bar{\phi}_1, \dots, \bar{\phi}_M} \left[\min_{i=1, \dots, M} Q_{\phi_{tar, N}}(s', a') \right] \approx \min_{i=1, \dots, N} Q_{\phi_{tar, i}}(s', a')$$

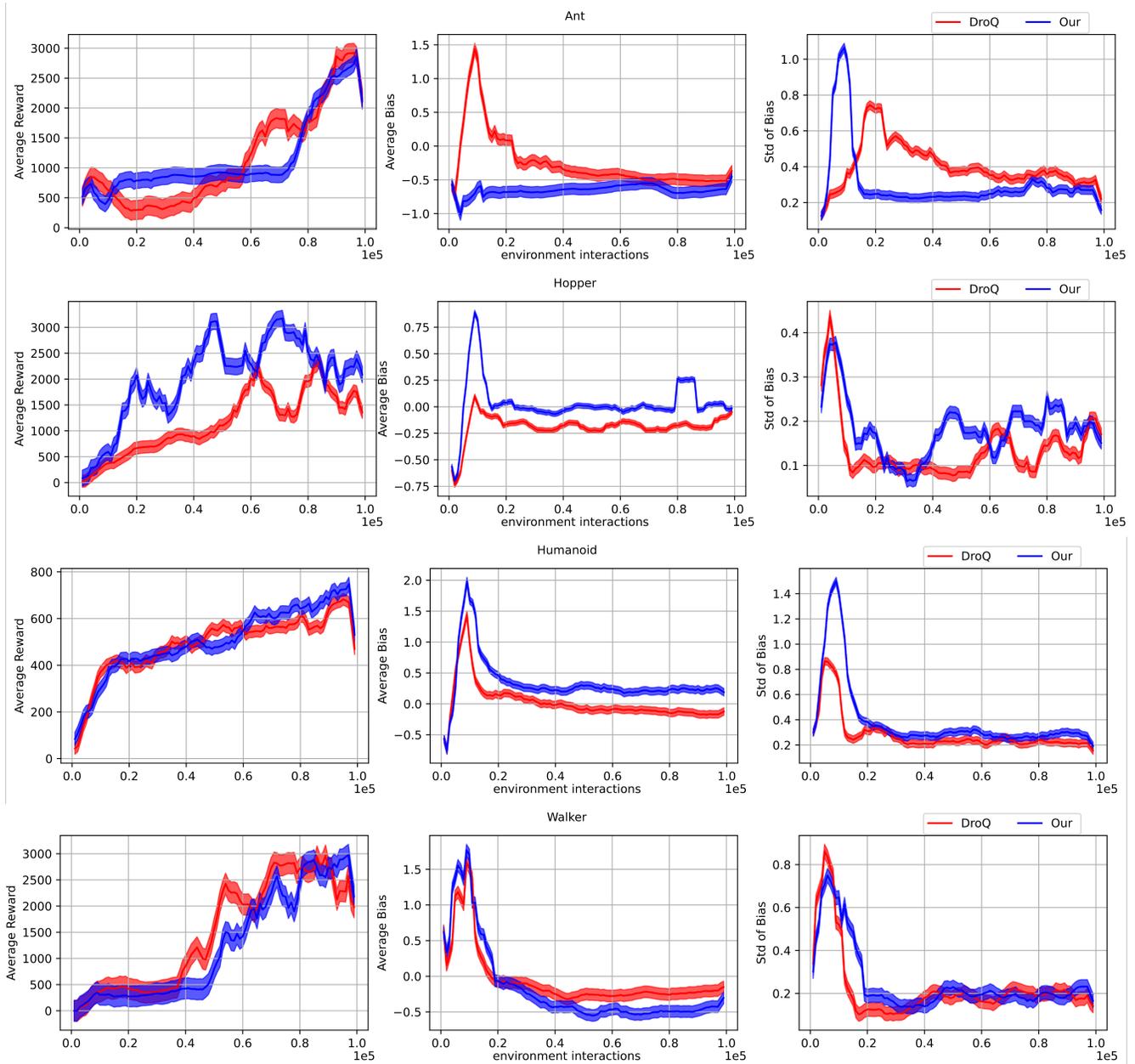


Figure 3: Comparison of our method with identity connections and bootstrapping vs. DroQ. Our approach improves over DroQ in all the environments. Results were obtained from three distinct runs.

where each Q-function of N is independently initialized with parameters $\bar{\phi}_i, \dots, \bar{\phi}_N$ which is given by line 12 of the algorithm. Our approach balances the dual aims of providing the model with ample uncertainty without being excessively computation intensive. It delivers consistent performance, even when the UTD is reduced ($UTD < 20$) (Figure 4).

Experimental Results

We showcase our findings across four challenging OpenAI Gym environments: Ant-v2, Hopper-v2, Humanoid-v2, and Walker2d-v2. In addition to analyzing the rewards, we also

look at average estimation bias and the standard deviation of estimated bias. The estimation bias is calculated as:

$$Q_\phi(s, a) - R^\pi(s, a)$$

where $R^\pi(s, a)$ is the Monte Carlo return while $Q_\phi(s, a)$ is the average return of randomly selected subset of Q-learners. To better predict the estimation bias, normalized estimation bias is also used as a metric and is calculated as:

$$b(s, a) = (Q_\phi(s, a) - R^\pi(s, a)) / \mathbb{E}_{s, a \sim \pi} [R^\pi(s, a)]$$

Both the mean and standard deviation of estimation bias are crucial metrics in assessing the performance. A high ab-

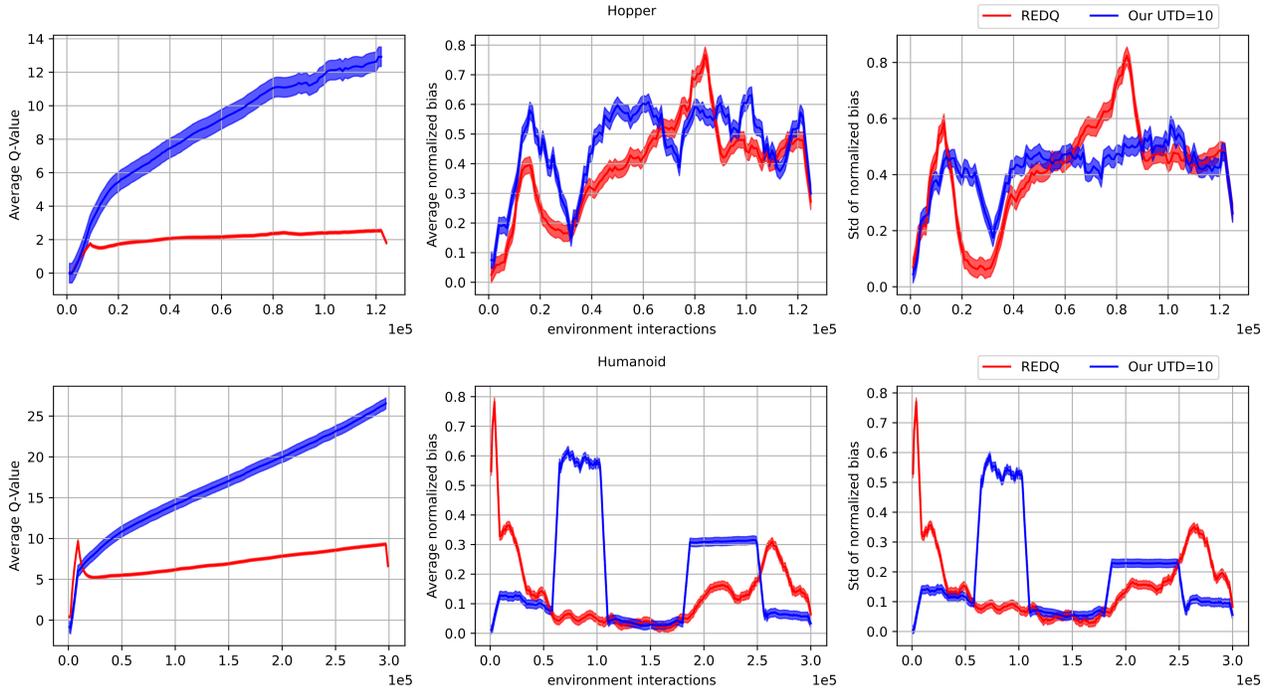


Figure 4: Comparison of Q-value prediction, average normalized bias, and standard deviation of normalized bias between REDQ ($G = 20, N = 10$) and our approach when UTD is set to $G = 10$ while $N = 5$ for our approach. Our method still managed to achieve better performance than REDQ. Results were obtained from three distinct runs for each environment.

solute mean bias indicates that the estimates are consistently inaccurate. On the other hand, a high standard deviation of bias suggests that the estimation errors are not uniform across different states or actions. Our aim is to have the average normalized bias fall close to zero with a small standard deviation.

These experiments are conducted using the optimal parameters of the REDQ. In this comparison, following the REDQ approach, we choose $N = 10$ and $M = 2$, while the UTD is set to $G = 20$. The results of this comparison are illustrated in Figure 2. Under this configuration, our method achieves superior average Q-value predictions while simultaneously keeping estimation bias at a minimum. Furthermore, our approach outperforms REDQ by maintaining better average normalized bias and standard deviation of normalized bias. These results affirm that our method is not only proficient in handling overestimation and underestimation but is, in fact, at least as effective as REDQ.

We conducted another set of experiments using the optimal configuration for both DroQ and our method, incorporating identity connections, with parameters set to $N = M = 2$ and $G = 20$. The results are illustrated in Figure 3. Our method, enhanced by bootstrapping and identity connection, consistently achieves superior rewards across all environments, effectively mitigating estimation bias by keeping both average bias and standard deviation of average bias close to zero.

A higher UTD ratio has substantial challenges including overfitting to specific experiences, constraint exploration, re-

ducing the generalization and increasing the computational cost. One of the straightforward ways to address these challenges is to reduce the UTD ratio. In our subsequent set of comparisons, we opt for a more aggressive configuration by choosing $N = 5$ and $M = 2$ while reducing the UTD $G = 20$ to $G = 10$. For REDQ, we leave the values of N and M unchanged. These results are presented in Figure 4. In this set of experiments, the results exhibit a somewhat mixed nature. Although the average Q-values prediction is reduced for both methods, our approach still manages to perform equally well in terms of average normalized Q bias and standard deviation of normalized Q bias.

Conclusion

This paper introduces an enhanced variant of the REDQ method that incorporates bootstrapping and multi-head self-attention into an ensemble of Q-functions. Our method demonstrates enhanced Q-value predictions while effectively managing estimation bias at a level comparable to REDQ, even when reducing the UTD ratio and ensemble size. A version of our model outperforms the optimal configuration for DroQ. Our results demonstrate the benefits of our proposed approach towards improving both sample efficiency and performance across multiple OpenAI Gym environments.

References

- Andrychowicz, O. M.; Baker, B.; Chociej, M.; Józefowicz, R.; McGrew, B.; Pachocki, J.; Petron, A.; Plappert, M.; Powell, G.; Ray, A.; Schneider, J.; Sidor, S.; Tobin, J.; Welinder, P.; Weng, L.; and Zaremba, W. 2020. Learning dexterous in-hand manipulation. *Int. J. Rob. Res.* 39(1):3–20.
- Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization.
- Chen, X.; Wang, C.; Zhou, Z.; and Ross, K. W. 2021. Randomized Ensembled Double Q-Learning: Learning fast without a model. In *International Conference on Learning Representations*.
- Efron, B. 1992. *Bootstrap Methods: Another Look at the Jackknife*. New York, NY: Springer New York. 569–593.
- Faußer, S., and Schwenker, F. 2015. Selective neural network ensembles in reinforcement learning: Taking the advantage of many agents. *Neurocomputing* 169:350–357.
- Gal, Y., and Ghahramani, Z. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In Balcan, M. F., and Weinberger, K. Q., eds., *Proceedings of the International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, 1050–1059. New York, New York, USA: PMLR.
- Gelada, C., and Bellemare, M. G. 2019. Off-policy deep reinforcement learning by bootstrapping the covariate shift. *Proceedings of the AAAI Conference on Artificial Intelligence* 33(01):3647–3655.
- Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Dy, J., and Krause, A., eds., *Proceedings of the International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 1861–1870. PMLR.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.
- He, Q.; Su, H.; Gong, C.; and Hou, X. 2022. MEPG: A minimalist ensemble policy gradient framework for deep reinforcement learning.
- Hiraoka, T.; Imagawa, T.; Hashimoto, T.; Onishi, T.; and Tsuruoka, Y. 2022. Dropout Q-Functions for doubly efficient reinforcement learning. In *International Conference on Learning Representations*.
- Hu, S.; Zhu, F.; Chang, X.; and Liang, X. 2021. Updet: Universal multi-agent reinforcement learning via policy decoupling with transformers. *arXiv preprint arXiv:2101.08001*.
- Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on International Conference on Machine Learning - Volume 37*, ICML’15, 448–456. JMLR.org.
- Janner, M.; Fu, J.; Zhang, M.; and Levine, S. 2019. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*.
- Khan, M. J.; Ahmed, S. H.; and Sukthankar, G. 2022. Transformer-based value function decomposition for cooperative multi-agent reinforcement learning in starcraft. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 18:113–119.
- Lai, H.; Shen, J.; Zhang, W.; and Yu, Y. 2020. Bidirectional model-based policy optimization. In III, H. D., and Singh, A., eds., *Proceedings of the International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 5618–5627. PMLR.
- Lee, K.; Laskin, M.; Srinivas, A.; and Abbeel, P. 2021. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In Meila, M., and Zhang, T., eds., *Proceedings of the International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 6131–6141. PMLR.
- Mendonca, R.; Gupta, A.; Kravev, R.; Abbeel, P.; Levine, S.; and Finn, C. 2019. Guided meta-policy search. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Harley, T.; Lillicrap, T. P.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of the International Conference on International Conference on Machine Learning - Volume 48*, ICML’16, 1928–1937. JMLR.org.
- Osband, I.; Blundell, C.; Pritzel, A.; and Roy, B. V. 2016. Deep exploration via bootstrapped DQN. In *Proceedings of the International Conference on Neural Information Processing Systems*, NIPS’16, 4033–4041. Red Hook, NY, USA: Curran Associates Inc.
- Rahman, M. M., and Xue, Y. 2023. Bootstrap state representation using style transfer for better generalization in deep reinforcement learning. In Amini, M.-R.; Canu, S.; Fischer, A.; Guns, T.; Kralj Novak, P.; and Tsoumakas, G., eds., *Machine Learning and Knowledge Discovery in Databases*, 100–115. Cham: Springer Nature Switzerland.
- Schulman, J.; Levine, S.; Moritz, P.; Jordan, M.; and Abbeel, P. 2015. Trust region policy optimization. In *Proceedings of the International Conference on International Conference on Machine Learning - Volume 37*, ICML’15, 1889–1897. JMLR.org.
- Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *CoRR* abs/1707.06347.
- Shen, J.; Zhao, H.; Zhang, W.; and Yu, Y. 2020. Model-based policy optimization with unsupervised model adaptation. In *Proceedings of the International Conference on Neural Information Processing Systems*, NIPS’20. Red Hook, NY, USA: Curran Associates Inc.
- Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15(1):1929–1958.
- Upadhyay, U.; Shah, N.; Ravikanti, S.; and Medhe, M. 2019.

Transformer based reinforcement learning for games. *arXiv preprint arXiv:1912.03918*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30.

Zheng, Q.; Zhang, A.; and Grover, A. 2022. Online decision transformer. *arXiv preprint arXiv:2202.05607*.