ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9747886

# POPO: PESSIMISTIC OFFLINE POLICY OPTIMIZATION

*Qiang He*[⋆†] *Xinwen Hou*[⋆] ✉ *Yu Liu*[⋆]

[⋆] Institute of Automation, Chinese Academy of Sciences
[†] School of Artificial Intelligence, University of Chinese Academy of Sciences
{heqiang2019, xinwen.hou, yu.liu}@ia.ac.cn

## ABSTRACT

Offline reinforcement learning (RL) aims to optimize policy from large pre-recorded datasets without interaction with the environment. This setting offers the promise of utilizing diverse and static datasets to obtain policies without costly, risky, active exploration. However, commonly used off-policy deep RL methods perform poorly when facing arbitrary off-policy datasets. In this work, we show that there exists an estimation gap of value-based deep RL algorithms in the offline setting. To eliminate the estimation gap, we propose a novel offline RL algorithm that we term Pessimistic Offline Policy Optimization (POPO), which learns a pessimistic value function. To demonstrate the effectiveness of POPO, we perform experiments on various quality datasets. And we find that POPO performs surprisingly well and scales to tasks with high-dimensional state and action space, comparing or outperforming tested state-of-the-art offline RL algorithms on benchmark tasks.

***Index Terms***— Reinforcement Learning, Offline Optimization, Out-of-distribution

## 1. INTRODUCTION

One of the main driving factors for the success of mainstream machine learning paradigms is that high capacity function approximators (e.g. deep neural networks) can learn open-world perception ability from large amounts of data [1, 2]. Combined with deep learning, reinforcement learning (RL) has proven its great potential in a wide range of fields such as playing Atari games [3], playing chess, Go, and Shoji [4], etc. However, it turns out that RL is difficult to extend from simulators to the unstructured physical real world because most RL algorithms need to actively collect data due to the nature of sequential decision-making, which is distinct from a typical supervised learning setting. In the physical world, we can usually obtain static data from historical experiences more easily than dynamics data. Learning from static datasets is a crucial requirement for generalizing RL to a system where the data collection procedure is time-consuming, risky, and expensive. In this paper, we study how to utilize RL to solve sequential decision-making problems from pre-collected datasets, i.e., offline RL, which is opposite to the research paradigm of active, interactive learning with the environment.

Off-policy RL algorithms, in general, are considered to be able to leverage any data to learn skills. In practice, these methods, however, still fail when facing arbitrary off-policy data. Specifically, off-policy RL methods suffer from the problem of out-of-distribution (OOD) actions [5, 6] in the offline setting. That means the target of the Bellman backup operator utilizes actions sampled from the learned policy, which may not exist in the datasets. In this paper, we firstly show that even the off-policy RL method would fail given high-quality expert data produced by the same algorithm. This phenomenon goes against our intuition because if given expert data, then exploration, RL's intractable problem, no longer exists. The sensitivity of existing RL algorithms to data limits the broad application of RL. Formally, we show that a catastrophic estimation gap occurs when applying value-based algorithms to completely offline data. That means when we evaluate the value function, the inability to interact with the environment makes it unable to eliminate the estimation gap through the Bellman equation. Secondly, to tackle this issue, we propose a novel offline policy optimization method, termed Pessimistic Offline Policy Optimization (POPO), where the policy utilizes a pessimistic distributional value function to approximate the true value, thus learning a strong policy. Our proposed algorithm can alleviate the estimation gap between the true value function and the estimated value function due to its pessimistic nature. Finally, to demonstrate the effectiveness of POPO, we compare it with state-of-the-art offline RL methods on d4rl benchmarks [7]. The experimental results show that our method compares or outperforms the tested algorithms.

**Related Work**. Imitation learning (IL) [8] studies how to learn a policy by mimicking expert demonstrations. IL has been combined with RL, either by learning from demonstrations [9] or utilizing deep RL extensions [10]. However, IL may still fail when applied on fully offline demonstrations. Because IL either requires interaction with the environment or needs high-quality data. We introduce a generative model into POPO, wherein we get insights from IL. Recently, offline RL algorithms have received significant attention [5, 6, 11, 12]. The BCQ algorithm [5] can ensure itself converges to the

This project was supported by National Defense Basic Scientific Research Program (JCKY2019203C029).

optimal policy under the given consistent datasets. BEAR algorithm [6] utilizes maximum mean discrepancy (MMD) to constrain the support of learned policy close to the behavior policy. CRR [11] can be considered as a form of filtered behavior cloning where data is selected w.r.t. the policy's value function. CQL [12] aims to learn a conservative Q-function such that the expected value of a policy under this Q-function lower-bounds its true value. We get insight from CQL and BCQ. If the agent could directly learn a pessimistic attitude towards the actions out of the support of behavior policy, we can suppress the estimation gap so that the agent can obtain a pessimistic value function to learn a strong policy. To this end, we utilize the quantile regression [13] method when optimizing the value function.

## 2. PESSIMISTIC OFFLINE POLICY OPTIMIZATION

### 2.1. Background

We formalize the standard RL paradigm as a Markov Decision Process (MDP), defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p, \rho_0, \gamma)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, transition probability function $p(s', r|s, a)$, initial state distribution $\rho_0$, and discount factor $\gamma \in [0, 1)$. The agent generates action $a$ w.r.t. policy $\pi$, then receives a new state $s'$ and reward $r$. Through interactions, a trajectory $\tau = \{s_0, a_0, s_1, a_1, \cdots\}$ is produced. The goal of RL is to maximize the expected return $J = \mathbb{E}_\tau[R_0]$, where $R_t = \sum_{i=t}^{T} \gamma^{i-t} r(s_i, a_i)$. The action-value function, a.k.a. Q-function, critic, is defined as $Q(s, a) = \mathbb{E}_\tau[R_0|s_0 = s, a_0 = a]$ which measures the quality of an action $a$ given a state $s$. State-value function, a.k.a. value function, $V$-function, is defined as $V(s) = \mathbb{E}_\tau[R_0|s_0 = s]$, measuring the quality of an individual state $s$. For a given policy $\pi$, the $Q$-function can be estimated recursively by Bellman backup operator [14]. When we apply RL to large space or continuous space, the value function can be approximated by neural networks, which is called Deep $Q$-networks [3]. Sutton et al. [14] introduced the policy gradient method.

### 2.2. Diagnosing Value Function Estimation

Offline RL suffers from OOD actions as we have discussed. Hasselt et al. [15] observed that overestimation occurs in the DQN algorithm. We argue that a similar phenomenon also occurs in offline scenarios but for the different underlying mechanisms. OOD actions and overestimation issues[16, 17, 18] are coupled with each other, making the value function more difficult to learn than online setting. We call the result of coupling these two effects as *estimation gap*. In the standard RL setting, the estimation gap could be eliminated through the agent's exploration to obtain an approximately true action value and then updated by the Bellman backup operator. But for offline settings, the estimation gap cannot be eliminated due to the inability to interact with the environment. Furthermore, due to the backup nature of the Bellman operator, the error would gradually accumulate, which would eventually cause the estimation gap to become larger, leading to the failure of policy learning. Therefore, out-of-distribution actions harm these RL algorithms' performance in offline settings. Formally, we define estimation gap for policy $\pi$ in state $s$ as $\delta_{\text{MDP}}(s) = V^\pi(s) - V^\pi_{\mathcal{D}}(s)$ where $V^\pi(s)$ is true value and $V^\pi_{\mathcal{D}}(s)$ is estimated on datasets $\mathcal{D}$. Given any policy $\pi$ and state $s$, the estimation gap $\delta_{\text{MDP}}(s)$ satisfies the following Bellman-like equation

$$\delta_{\text{MDP}}(s) = \sum_a \pi(a|s) \sum_{s',r} [p(s', r|s, a) - p_{\mathcal{D}}(s', r|s, a)] \big(r + \gamma V^\pi_{\mathcal{D}}(s')\big) + \gamma \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \delta_{\text{MDP}}(s') \tag{1}$$

Equation 1 can be proved by expanding this equation through the definition of the $V$ function. The transition probability function of dataset $\mathcal{D}$ is defined as $p_{\mathcal{D}}(s', r|s, a) = \frac{N(s,a,s',r)}{\sum_{s',r} N(s,a,s',r)}$, where $N(s, a, s', r)$ is the number of the transition observed in data set $\mathcal{D}$. What's more, Equation 1 shows that the estimation gap is a divergence function w.r.t. the transition distributions, which means if the agent carefully chooses actions, the estimation gap can be minimized by visiting regions where the transition probability is similar.

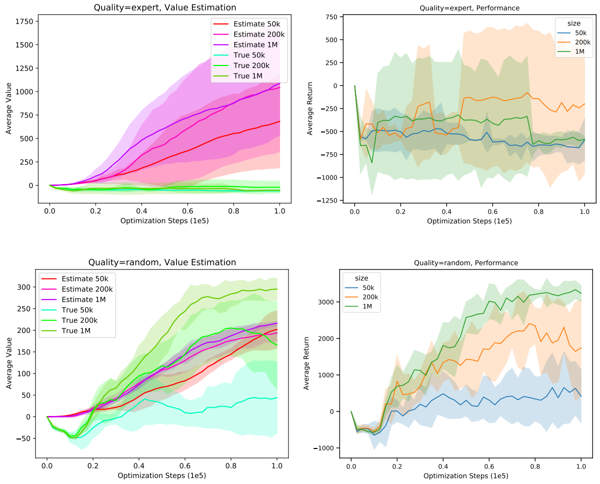

**Fig. 1**. The relationship between data quality, quantity, and corresponding average return. We train the TD3 algorithm on the MuJoCo halfcheetah-v2 environment over five random seeds. 'Estimate 50k' means the curve shows the agent's value estimation on the size=50k dataset. The shaded area represents a standard deviation.

**Does this analysis occur in practice?** We utilize static datasets [7] of different qualities and sizes to verify our analysis. We train TD3 agents in halfcheetah-v2 environment in the offline setting. We show results in Figure 1. Surprisingly, train-

**Algorithm 1** Pessimistic Offline Policy Optimization (POPO)

```
**Initialize**: Dataset $\mathcal{D}$, num of quantiles $N$, target network update rate $\eta$, coefficient $\xi$
**Initialize**: Distortion risk measure $\beta$, random initialized networks and corresponding target networks, parameterized by $\theta_i' \leftarrow \theta_i, \phi' \leftarrow \phi$, VAE $G = \{E(\cdot,\cdot;\omega_1), D(\cdot,\cdot;\omega_2)\}$
  **for** iteration $= 1, 2, \ldots$ **do**
    Sample mini-batch data $(s, a, r, s')$ from data set $\mathcal{D}$
    $\mu, \Sigma = E(a|s;\omega_1)$, $\hat{a} = D(z|s,;\omega_2)$, $z \sim \mathcal{N}(\mu, \Sigma)$
    $\omega \leftarrow \arg\min_\omega \sum (a - \hat{a})^2 + \frac{1}{2} D_{\mathrm{KL}}(\mathcal{N}(\mu, \Sigma) \| \mathcal{N}(0, I))$
    Set Critic loss $\mathcal{L}(\theta)$ (Equation 4)
    $\theta \leftarrow \arg\min_\theta \mathcal{L}(\theta)$.
    Generate $a_{\text{new}}$ from Equation 7
    $\phi \leftarrow \arg\max_\phi Q_\beta(s, a_{\text{new}})$
    $\theta_i' \leftarrow \eta\theta_i + (1-\eta)\theta_i'$, $\phi' \leftarrow \eta\phi + (1-\eta)\phi'$
  **end for**
```

ing on random data gives us a better average return than on expert data. Checking its value function, we find that the estimated value function w.r.t. expert data deviates more and more from the true value as the data set capacity increases, which verifies there does exist an estimation gap. The erroneous estimation of the value function further leads to the failure of policy learning. Why can random data learn better? The analysis above inspires us that if the agent chooses actions similar to the actions in datasets, the estimation gap can be eliminated. The difference in performance suggests that the TD3 agent which is offline trained on random datasets may produces trajectories similar to static random data. Thus, we collect the produced trajectories that are not used to train. And we visualize the distributions of the static datasets and trajectories in Figure 2 through the T-SNE tool [19]. Expert/random trajectory means we train offline TD3 agent with expert/random datasets. We find that the offline training TD3 agent does visit a similar area to the static random datasets. Still, for expert data, the agent visits different areas from expert data, which is consistent with our analysis.

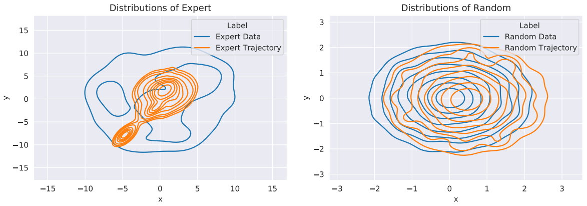

**Fig. 2**. Visualization of data generated by the halfcheetah-v2 environment. Left: expert data, visiting the different areas. Right: random data, visiting a similar area.

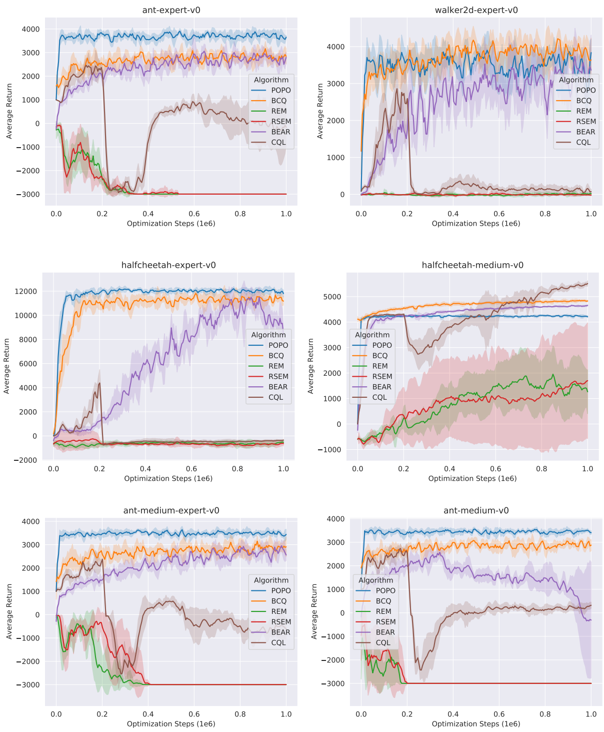

**Fig. 3**. Performance curves for OpenAI gym continuous control. The shaded region represents a standard deviation of the average evaluation over five seeds. The BCQ is stable when tested, but it is not as good as the POPO. BEAR suffers from performance decrease when training too much time. REM almost failed during testing.

### 2.3. Pessimistic Value Function

Our insights are if an agent could maintain a pessimistic attitude towards actions out of the support of behavior policy when learning value function, then we can suppress the estimation gap of the value function outside the datasets so that the algorithm can obtain a more pessimistic value function to learn a strong policy through an actor-critic style algorithm. To capture more information on the value function, we utilize the distributional value function [13], which has proved its superiority in the online settings. The distributional Bellman optimality operator is defined by

$$\mathcal{T} Z^\pi(s, a) := r + \gamma Z^\pi(s', \arg\max_{a' \in \mathcal{A}} \mathbb{E}_{s'} Z(s', a')), \qquad (2)$$

where random return $Z^\pi(s, a) := \sum_{t=0}^{\infty} r(s_t, a_t)$, and $=$ symbol denotes that the left and the right random variable have the same distribution. Let $F_Z^{-1}(\tau)$ be the quantile function at $\tau \in [0, 1]$ for random variable $Z$. We write $Z_\tau := F_Z^{-1}(\tau)$ for simplicity. We model the state-action quantile function as a mapping from state-action to samples from certain distribution, such as $\tau \sim U(0, 1)$ to $Z_\tau(s, a)$. Let $\beta : [0, 1] \to [0, 1]$ be a distortion risk measure [13]. Then the distorted expectation of random variable $Z(s, a)$ induced by $\beta$ is $Q_\beta(s, a; \theta) :=$

$\mathbb{E}_{\tau \sim U(0,1)}[Z_{\beta(\tau)}(s, a; \theta)]$. We also call $Z_\beta$ critic. By choosing different $\beta$, we can obtain various distorted expectations, i.e., different attitudes towards the estimation. To avoid the abuse of symbols, $\tau$ in the following marks $\tau$ acted by $\beta$. For the critic loss function, given two samples, $\tau, \tau' \sim U(0, 1)$, the temporal difference error at time step $t$ is

$$\Delta_t^{\tau,\tau'} = r_t + \gamma Z_{\tau'}(s_{t+1}, \pi_\beta(s_{t+1}); \theta') - Z_\tau(s_t, a_t; \theta). \quad (3)$$

Then the critic loss function of POPO is given by

$$\mathcal{L}(s_t, a_t, r_t, s_{t+1}) = \frac{1}{N'} \sum_{i=1}^{N} \sum_{j=1}^{N'} \rho_{\tau_i}^{\kappa}(\Delta_t^{\tau_i, \tau_j'}), \quad (4)$$

where

$$\rho_\tau^\kappa(x) = |\tau - \mathbb{I}\{x < 0\}| \frac{\mathcal{L}_\kappa(x)}{\kappa}, \quad \text{with} \quad (5)$$

$$\mathcal{L}_\kappa(x) = \begin{cases} \frac{1}{2}x^2, & \text{if } |x| \leq \kappa \\ \kappa(|x| - \frac{1}{2}\kappa), & \text{otherwise} \end{cases}$$

in which $N$ and $N'$ is the number of i.i.d. samples, and $\tau_i, \tau_j'$ are sampled from $U(0, 1)$, respectively. We can recover $Q_\beta(s, a)$ from $Z_{\beta(\tau)}$.

### 2.4. Generating Actions from VAE

To tackle OOD actions, we introduce a generative model, specifically, conditional Variational Auto-Encoder (VAE), consists of Encoder $E(\cdot|\cdot; \omega_1)$ and Decoder $D(\cdot|\cdot; \omega_2)$. Furthermore, VAE could constrain the distance between the actions sampled from the learned policy and that provided by the datasets [5]. VAE reconstructs action on condition state $s$. We call the action produced by the VAE the central action $\hat{a}$. The loss function of VAE is

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{(s,a)\sim\mathcal{D}}\big[(a - \hat{a})^2 + \frac{1}{2} D_{\text{KL}}(\mathcal{N}(\mu, \Sigma) \| \mathcal{N}(0, I))\big]. \quad (6)$$

where $\hat{a} = D(z|s; \omega_2)$. To generate action $a'$ w.r.t. state $s'$, firstly we copy the action $n$ times and send it to VAE to incorporate with policy improvement. Then we feed the actor network with central action $\hat{a_i'} = D(z_i|s'; \omega_2)$ and state $s'$, then the actor network $\pi(\cdots; \phi)$ outputs a new action $\bar{a_i'}$. Combining $\hat{a_i'}$ and $\bar{a_i'}$ with residual style $\tilde{a_i'} = \xi \bar{a_i'} + (1 - \xi)\hat{a_i'}$ with coefficient $\xi$, we get the selected action $\tilde{a_i'}$. We choose action of $n$ outputs with highest value as the final output

$$a'_{\text{new}} = \arg\max_{a_i} Q_\beta(s', \tilde{a_i'}; \theta), \quad (7)$$

where $\{\tilde{a_i'} = (\pi \circ D)(z_i|s')\}_{i=1}^n$. We call this action generation method the residual action generation. We use the DPG method [20] to train actor network $\pi$. For a given state, the generated action can be close to the actions contained in the data set with a similar state. At the same time, residual action generation maintains a large potential for policy improvement. We summarize POPO in Algorithm 1.

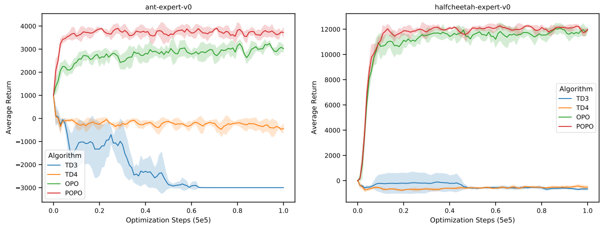

**Fig. 4**. Performance curves for ablation study. The results show that the pessimistic critic does have an improvement over the original TD3 algorithm. VAE makes offline optimization successful because it deals with the OOD action issue.

### 2.5. Experimental Results

To evaluate our algorithm, we utilize the various quality datasets [7, 21] to train our proposed algorithm. We recommend readers check [7] for more details. Given the recent concerns about algorithms reflect the principles that informed its development [22], we implement POPO without any engineering tricks so that POPO works as we originally intended for. We compare our algorithm with the recently proposed SOTA offline RL algorithms BCQ, REM, BEAR, and CQL. And we use the authors' official implementations. The performance curves are graphed in Figure 3, which shows POPO matches or outperforms all compared algorithms.

**Ablation Study.** The main components of POPO are VAE and pessimistic distributional critic. We term the POPO version without distributional critic OPO. And we term the POPO version without VAE TD4. The performance curves are graphed in Figure 4, which shows the pessimistic distributional critic does have a significant performance improvement over TD3 and OPO. Besides, VAE makes the offline RL successful because it solves the OOD actions issue. The combination between VAE and pessimistic critics would produce better results.

## 3. CONCLUSION

In this work, we studied why off-policy RL methods fail to learn in offline settings and proposed a novel offline RL algorithm POPO. Firstly, we showed that the inability to interact with the environment makes offline RL unable to eliminate the estimation gap through the Bellman equation. And we conducted experiments to verify the correctness of our analysis. Secondly, We proposed the POPO algorithm, which learns a pessimistic value function to get a strong policy. Finally, we demonstrated the effectiveness of POPO by comparing it with SOTA offline RL methods on the offline RL datasets. In future works, we can extend the POPO algorithm to discrete control settings and test more hyper-parameters.

## 4. REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., “Language models are few-shot learners,” *arXiv preprint arXiv:2005.14165*, 2020.

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[4] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al., “A general reinforcement learning algorithm that masters chess, shogi, and go through self-play,” *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[5] Scott Fujimoto, David Meger, and Doina Precup, “Off-policy deep reinforcement learning without exploration,” in *International Conference on Machine Learning*, 2019, pp. 2052–2062.

[6] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine, “Stabilizing off-policy q-learning via bootstrapping error reduction,” in *Advances in Neural Information Processing Systems*, 2019, pp. 11784–11794.

[7] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine, “D4rl: Datasets for deep data-driven reinforcement learning,” *arXiv preprint arXiv:2004.07219*, 2020.

[8] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne, “Imitation learning: A survey of learning methods,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[9] Beomjoon Kim, Amir-massoud Farahmand, Joelle Pineau, and Doina Precup, “Learning from limited demonstrations,” in *Advances in Neural Information Processing Systems*, 2013, pp. 2859–2867.

[10] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Gabriel Dulac-Arnold, et al., “Deep q-learning from demonstrations,” *arXiv preprint arXiv:1704.03732*, 2017.

[11] Ziyu Wang, Alexander Novikov, Konrad Żołna, Jost Tobias Springenberg, Scott Reed, Bobak Shahriari, Noah Siegel, Josh Merel, Caglar Gulcehre, Nicolas Heess, et al., “Critic regularized regression,” *arXiv preprint arXiv:2006.15134*, 2020.

[12] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine, “Conservative q-learning for offline reinforcement learning,” *arXiv preprint arXiv:2006.04779*, 2020.

[13] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos, “Implicit quantile networks for distributional reinforcement learning,” *arXiv preprint arXiv:1806.06923*, 2018.

[14] Richard S Sutton and Andrew G Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

[15] Hado Van Hasselt, Arthur Guez, and David Silver, “Deep reinforcement learning with double q-learning,” in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[16] Qiang He and Xinwen Hou, “Wd3: Taming the estimation bias in deep reinforcement learning,” in *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2020, pp. 391–398.

[17] Chen Gong, Qiang He, Yunpeng Bai, Xinwen Hou, Guoliang Fan, and Yu Liu, “Wide-sense stationary policy optimization with bellman residual on video games,” in *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.

[18] Qiang He, Chen Gong, Yuxun Qu, Xiaoyu Chen, Xinwen Hou, and Yu Liu, “Mepg: A minimalist ensemble policy gradient framework for deep reinforcement learning,” *arXiv preprint arXiv:2109.10552*, 2021.

[19] Laurens van der Maaten and Geoffrey Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[20] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller, “Deterministic policy gradient algorithms,” 2014.

[21] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba, “Openai gym,” *arXiv preprint arXiv:1606.01540*, 2016.

[22] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry, “Implementation matters in deep policy gradients: A case study on ppo and trpo,” *arXiv preprint arXiv:2005.12729*, 2020.