Proceedings of The 6th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/2026.31513

Improvement of Offline Reinforcement Learning Based on Prior Sampling Mechanism

Yuanhang Yang

School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China
ts23060113A31@cumt.edu.cn

Abstract. Offline reinforcement learning (RL) circumvents real-time environmental interaction by leveraging pre-collected historical data, yet it faces critical challenges of distributional shift and suboptimal data distribution in experience replay buffers. To enhance agent training efficacy, this study proposes two novel prioritized sampling mechanisms: one anchored in temporal-difference (TD) error estimation and the other grounded in martingale theory. The TD error-based mechanism captures supplementary experiential data to mitigate out-of-distribution (OOD) state risks, while the martingale-theoretic counterpart prioritizes positive samples for policy optimization and suppresses negative sample interference. Integrating these mechanisms with Batch-Constrained Deep Q-Learning (BCQ) yields two enhanced algorithms: TD-PBCQ and M-PBCQ. Extensive experiments on D4RL (Ant, HalfCheetah, Hopper, Walker2d) and Torcs datasets demonstrate that the proposed sampling models selectively collect valuable experiential data, achieving significantly improved reward performance compared to the baseline BCQ. Comparative analysis shows TD-PBCQ excels in tasks with fast-converging behavior policies and low-error datasets, while M-PBCQ adapts better to high-error offline buffers. This work provides a data-driven solution for advancing offline RL performance via efficient sampling design.

Keywords: Offline reinforcement learning frameworks, Prioritized sampling strategies, Temporal difference error estimation, Martingale theory

1. Introduction

Deep reinforcement learning (DRL) is a viable solution for complex environmental perception and decision-making challenges due to its strengths in decision-making and deep learning representation. In recent years, it has been applied in fields such as robot control and power system optimization.

As the theory and methodology advance, researchers employ agents for learning tasks that involve difficulties in data acquisition and threats to hardware security. Before 2020, scholars proposed batch reinforcement learning by referring to batch learning. After 2020, Levine et al. renamed it offline reinforcement learning.

Offline reinforcement learning has a fixed-size experience buffer, which avoids noise and risky behaviors in online exploration. It inherits the merits of classical online reinforcement learning

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.31513

algorithms and introduces imitation learning to reduce the barriers with other machine learning methods. However, it still faces challenges such as agent errors in evaluating out-of-distribution states [1].

To address the challenge, researchers proposed multiple solutions. Kumar et al. used variational autoencoders for action generation and fine-tuning. However, BCQ had limited improvement in policy performance when dealing with low-quality offline data.

Then, Kumar et al. proposed the Bootstrap Error Accumulation Reduction (BEAR) algorithm to prevent error accumulation and distribution drift. Nevertheless, it had high computational costs. Jaques et al. reduced the KL divergence between the learned and behavioral policies, and Maran et al. incorporated the Wasserstein distance into the optimization objectives [2].

2. Basic assumptions

To evaluate the importance of regularization terms for different behavioral policies, Wu et al. developed a universal algorithm framework that included BCQ and BEAR. They proposed two regularization methods: BRAC - v (value function regularization) and BRAC - p (policy regularization). Value function regularization improves the accuracy of out-of-distribution (OOD) state evaluation but increases noise and hinders policy convergence. On the other hand, policy regularization reduces distribution drift and enhances stability but raises the probability of local optima. The aforementioned offline reinforcement learning methods focus on reducing distribution drift and improving policy quality but overlook the quality of the offline dataset. In online reinforcement learning, the quality of experience is crucial for agent training, and efficient sampling can enhance algorithm performance. For example, Schaul et al. utilized prior experience replay techniques, and Horgan et al. proposed distributed experience pools.

Influence upon the procedural unfolding of offline reinforcement learning exerted by experience caching in the non-online realm manifests dichotomously: firstly,discernible within agent behavior becomes the phenomenon wherein experiences indicative of erroneous execution are subjected to omission or neglect; secondly,evidence emerges of a situation whereby an overabundance is found of negative-sample extraction,whose presence might produce degradation traceable within policy quality. Derivable from such observations must be the necessity that,for processes situated under the umbrella term "offline reinforcement learning," attention to and refinement regarding sampling model configuration persists as imperative [3].

Experiments on D4RL and Torcs datasets show that TD - PBCQ is suitable for tasks with quickly - converging behavioral policies and few erroneous experiences in the offline cache, while M - PBCQ is more appropriate for tasks with more erroneous experiences.

$$\mu^{\mathrm{BCQ}}(s)=arg\max_{a_i+\xi_\phi(s,a_i,\Phi)}Q_\theta\left(s,a_i+\xi_\phi\left(s,a_i,\Phi\right)\right) \tag{1}$$

$$Cov\left(X,Y\right)=\frac{1}{n}\sum_{i=1}^{n}\left(X_i-\bar{X}\right)\left(Y_i-\bar{Y}\right) \tag{2}$$

DOI: 10.54254/2755-2721/2026.31513

$$t = r \cdot \sqrt{\frac{n-2}{1-r^2}} \tag{3}$$

3. Stochastic sampling model based on stochastic process

Martingale theory is a cornerstone of modern probability theory and a vital tool for stochastic process and mathematical statistics research [4]. Reinforcement learning algorithms have a deep connection with martingale theory, and many martingale - based methods are used to validate their effectiveness. To clarify the relationship between martingales and policy optimization, we use a grid world environment as a case study. In the schematic, the agent starts an episode from the starting point, ends it upon reaching the target. The heat map of the optimal value function shows that as the episode ends at the endpoint, the state value there remains unchanged during iteration, leading to a relatively low value [5]. In the experiment, Q - learning based on linear function approximation was used to train 300 steps in the maze, and a heat map of the value function was plotted every 50 iterations. Two training batches were carried out, and the iterative update process of the value function is shown in the figure. The above content is shown in Figures 1 and 2.

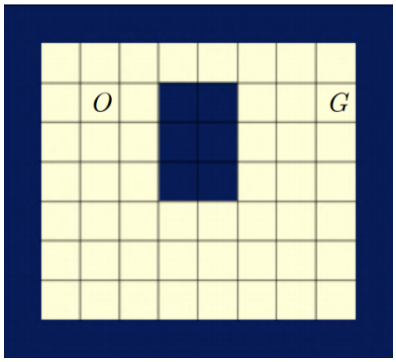

Figure 1. Schematic diagram of the environment



Figure 2. Heatmap of the optimal value function

Two offline reinforcement learning methods, TD-PBCQ and M-PBCQ, are developed by combining the sampling model based on temporal difference error with the BCQ, and the M-PBCQ with the BCQ, respectively. For convenience, the pseudocode of BCQ is given in Algorithm 1.

DOI: 10.54254/2755-2721/2026.31513

3.1. Algorithm 1.BCQ

input. Sampling batch size m , Maximum training sessions T , soft update parameter $\mathscr{T}$,step length n , maximum perturbation parameter Φ , Target value calculation parameters λ , Number of sampling actions n , Offline Experience Playback $\mathscr{B}$.

3.2. Algorithm 2.TD-PBCQ

1) Calculate the importance sampling weight: $w_j = \left(\frac{1}{\mathscr{M}} \cdot \frac{1}{P(j)} \cdot \frac{1}{\max_i w_i}\right)$

2) Update priority: Calculate the δ_j error , Update data priority $p_j \leftarrow |\delta_j| + \sigma$

3) cumulative Q value network parameter variation Δ_θ Variation of parameters in the disturbance network Δ_ϕ :

cumulative Q value network parameter variation $\Delta_\theta \leftarrow \Delta_\theta + \nabla_\theta \left(y_{\mathscr{B}} - Q_\theta\left(s_j, a_j\right)\right)^2$ cumulative perturbation network parameter variation $\Delta_\phi \leftarrow \Delta_\phi - \nabla_\phi Q_{\theta_1}\left(s_j, a_j + \xi_\phi\left(s_j, a_j, \Phi\right)\right)$ $a_j \sim G_\omega\left(s\right)$

3.3. Algorithm 3.M-PBCQ

1)Calculate importance sampling weights:M-PBCQ does not calculate importance sampling weights.

2) Update priority: Calculate priority u_j ,Update experience data priority.

3) cumulative Q value network parameter variation Δ_θ and the variation of the network parameters Δ_ϕ :

cumulative Q value network parameter variation $\Delta_\theta \leftarrow \Delta_\theta + \nabla_\theta \left(y_{\mathscr{B}} - Q_\theta\left(s_j, a_j\right)\right)^2$ cumulative perturbation network parameter variation $\Delta_\phi \leftarrow \Delta_\phi - \nabla_\phi Q_{\theta_1}\left(s_j, a_j + \xi_\phi\left(s_j, a_j, \Phi\right)\right)$ $a_j \sim G_\omega\left(s\right)$

4. Results and analysis

To begin,the public offline datasets curated by D4RL—encompassing tasks like Ant,HalfCheetah,Hopper,and Walker2d on both medium and expert data partitions—underwent empirical investigation utilizing TD-PBCQ,M-PBCQ,and BCQ algorithms respectively. Emergent from this phase of experimentation are performance metrics derived for each method within these operational contexts. Following this,tested were the same algorithmic variants—TD-PBCQ,M-PBCQ,and BCQ—within the Torcs environment,for which an offline experience memory buffer served as the foundation upon which procedural assessments were executed. Demonstrated through such deployments appears an explorative breadth in algorithm applicability across distinct task families.The above content is shown in Figures 3 and 4.

DOI: 10.54254/2755-2721/2026.31513

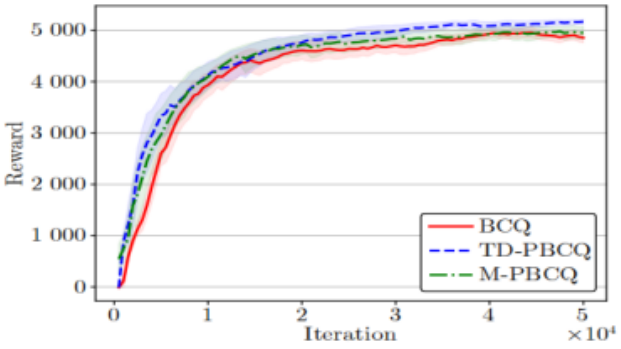

Figure 3. Ant

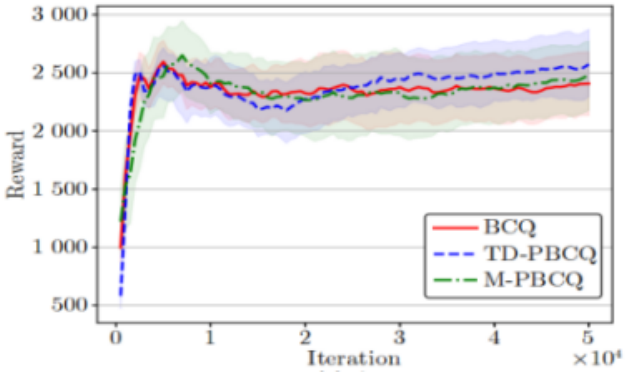

Figure 4. HalfCheetah

5. Conclusion

Manifestations of reinforcement learning, wherein agents by virtue of continual engagement with their operational environments are enabled to discern strategies considered optimal, have in recent discourses come to be regarded as a foundational paradigm for addressing perception-mediated decision-making quandaries manifesting within intricate contextualities. Instances from the prevailing literature indicate that the conventionally espoused methodology of online data accrual frequently encounters obstacles—security risks being present among them; constraints imposed temporally and considerations of economic expenditure likewise belong to this array—whose presence induces considerable circumscription upon real-world implementational scope.

Offline reinforcement learning, whose pertinence emerges from its sole reliance on pre-collected experiential datasets obviating direct environmental interaction, presents an alternative characterized by its foundation in historical informational substrates. New vistas it thereby opens for pursuits toward generalized artificial intelligence can be recognized in these traits. Yet, observed has been the phenomenon whereby qualitative attributes inherent to offline datasets exert significant influences, often detrimental, upon algorithmic efficacy in policy derivation processes. The development of robust strategical constructs directly from such imperfect archives thus manifests complexity.

Centered upon amelioration of this predicament is the present inquiry, which investigates methodologies for distillation of samples exhibiting higher utility from repositories comprising exclusively offline-captured experiences. Utilized herein are quantities defined through temporal difference errors in conjunction with stochastic properties encapsulated by martingale constructs, through which hierarchies of sample prioritization may then be established. Constructed and elaborated across two distinct lines—namely, a model grounded within temporal difference error frameworks and alternatively another predicated upon martingale dynamics—are the proposed sampling architectures.

Within epochs of agentic instruction, deployment of these models facilitates purposive curation of experience subsets so as to more precisely inform estimative procedures attending value function calculation and subsequent optimization of policy representations. Further synthesis is achieved through amalgamation of said sampling formulations into the BCQ structure, yielding dual prioritized BCQ variants: one informed by temporal difference errors, and the other by martingale criteria. What distinguishes the sampling schemes delineated herein, observable through integration practices, is their adaptability—such that interoperability with heterogeneous clusters of existing offline reinforcement learning techniques results unimpeded.

DOI: 10.54254/2755-2721/2026.31513

References

[1] Sun Chang-Yin, Mu Chao-Xu. Important scientific problems of multi-agent deep reinforcement learning. Acta Automatica Sinica, 2020, 46(7): 1301−1312.
[2] Wu Xiao-Guang, Liu Shao-Wei, Yang Lei, Deng Wen-Qiang, Jia Zhe-Heng. A gait control method for biped robot on slope based on deep reinforcement learning. Acta Automatica Sinica, 2021, 47(8): 1976−1987.
[3] Yin Lin-Fei, Chen Lv-Peng, Yu Tao, Zhang Xiao-Shun. Lazy reinforcement learning through parallel systems and social system for real-time economic generation dispatch and control. Acta Automatica Sinica, 2019, 45(4): 706−719.
[4] Chen Jin-Yin, Zhang Yan, Wang Xue-Ke, Cai Hong-Bin, Wang Jue, Ji Shou-Ling. A survey of attack, defense and related security analysis for deep reinforcement learning. Acta Automatica Sinica, 2022, 48(1): 21−39.
[5] Hu Y L, Skyrms B, Tarrès P. Reinforcement learning in signaling game. arXiv preprint arXiv: 1103.5818, 2011.