Analysis and method comparsion of online and offline reinforcement learning

Changhang Zheng*

Faculty of Information Science and Engineering, Ocean University of China, Qingdao, Shandong, China

Abstract. In this paper, an exploration of the online and offline precepts of reinforcement learning and the associated algorithms of paradigms is carried out in a systematic manner. Online reinforcement learning is a balance between exploration and exploitation in which there is real-time interaction with the environment, but which incurs significant costs of interaction as well as hazards. In expensive environments, offline reinforcement learning applies trial sets by earlier-collected data to create safety and efficiency procedures. The examples of representative algorithms are Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC) in the online reinforcement learning context, or Conservative Q-Learning (CQL) and Implicit Q-Learning (IQL) in the offline reinforcement learning context. Through comparative analysis, it can be established that online reinforcement learning lacks the ability to use samples efficiently and deploy flexibly, whereas offline reinforcement learning doesn't have issues with utilizing data and ensuring the safety of data when dealing with distribution change. This paper gives recommendations on where the field of reinforcement learning should go in the present age, based on an overview of four algorithms and analyses of case studies.

1 Introduction

Reinforcement learning, as a fundamental subdivision of machine learning, is dedicated to ensuring that an agent is able to learn how to make optimal decisions in response to exposure to the environment to maximize the total rewards it receives. This learning paradigm is connected with the trial-and-error adaptation of organisms to their surroundings. Problems of reinforcement learning typically consist of Markov decision processes, with the states, actions, reward signals, and value functions. Technology divided reinforcement learning into online and offline. Online reinforcement learning employs the concept of learning by doing to gather information and modify strategies by operating in real-time. Nevertheless, offline reinforcement learning is another recent learning-by-observation paradigm that learns strategies using a fixed set of data with no interaction with any other environment. Due to the active evolution of artificial intelligence, reinforcement learning is being applied to natural language processing, computer vision, autonomous driving, and medical diagnostics. Complex, high-dimensional, and high-risk tasks continue to be a challenge for traditional

* Corresponding author's email: Henry2024@stu.ouc.edu.cn

reinforcement learning systems. In autonomous driving, the optimal decision-making techniques need to be learned by the agents during a small number of interactions, but online reinforcement learning is expensive in terms of interactions, as well as having risks associated with exploration. Conversely, offline reinforcement learning is a safe and efficient reinforcement learner for high-risk situations, which utilizes pre-collected historical static data to learn policies. However, offline learning also faces core challenges such as data distribution bias and insufficient model generalization capabilities. Therefore, how to improve the generalization ability and robustness of models while ensuring data security has become an important research direction.

This paper aims to provide theoretical support and practical guidance for research and applications in related fields by systematically analyzing the theoretical foundations, representative algorithms, and application cases of online and offline reinforcement learning. This paper will first introduce the definitions, characteristics, and challenges of online and offline reinforcement learning, then analyze two representative algorithms in each field, and finally, this paper will conduct a systematic and objective comparative analysis of online and offline reinforcement learning from multiple dimensions, in order to reveal the trade-offs and choices in their practical applications.

2 Online Reinforcement Learning

Online reinforcement learning is the most classic form of reinforcement learning, with its core feature being the intelligent agent's real-time and continuous interaction with the environment to collect experience and use these new experiences to update its policy immediately or later [1]. In this loop, every time the agent performs an action, the environment provides a new state and reward, and the agent adjusts its subsequent behavior based on this information. In this mode, data collection and strategy learning are tightly coupled and interleaved. A key intrinsic requirement of online learning is active information acquisition [2], which leads to the famous Exploration-Exploitation Dilemma. The agent has to strike a balance between the exploitation of strategies that are known to give a high yield and the exploration of unknown actions in the hope of trying to find better strategies. Experience replay is used by many online algorithms (mostly off-policy algorithms) to improve the efficiency of using this data in the future, storing past experience to use during policy adaptation [3]. The weakness of online reinforcement learning is its reliance on ambient interaction. Several real-life scenarios can be expensive to interact with the environment: autonomous driving, the manufacturing industry, or robotics. Each trial can be time-consuming, energy-consuming, or physically demanding [1]; highly risky: in healthcare, financial transactions, or critical system control, mature exploration can cause disastrous consequences and make it inapplicable to complex problems [4]; and inefficient: achieving good performance can require millions of such interactions, with excruciatingly complicated samples [5]. The approach to the use of data in RL algorithms can be classified into two categories: on-policy and off-policy.

This paper presents two general algorithms of each type: Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC). PPO is a policy gradient algorithm that maximizes the policy network to maximize the expected rewards [6]. PPO should help in overcoming challenges regarding the choice of update step sizes and training instability of conventional policy gradient techniques. According to TRPO, it has a more convenient and efficient implementation tool. The PPO cabs policy changes so that they do not go too far, in contrast to the previous one. PPO applies a series of minibatch updates to optimize the use of data, without any change in training stability. The best-known application of the PPO limits policy uses a clipped surrogate objective function to update its policy. The core part of this objective

function is the probability ratio of the new and old policies taking a certain action in a certain state

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{1}$$

The objective function of PPO is as follows:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[min\left(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right] \tag{2}$$

This clipping method is also useful in avoiding an overly rapid policy update, rendering the algorithm much more stable [7]. The majority of scholars have opined that PPO is an on-policy algorithm. It has to refresh with recent policy information. When the update occurs, old information is deleted, requiring the new agent to re-interact with the environment using the new policy to obtain new information. PPO has a significant disadvantage, that of low sampling efficiency [8]. PPO is more effective than on-policy techniques because it uses importance sampling to perform multiple updates from a single data collection; however, it cannot utilize past data. PPO is fairly sensible in its simplicity, stability, robustness, and complexity of the sample. Its implementation is far easier than TRPO and gives excellent results both on continuous and discrete control problems (including MuJoCo robot motion simulators and Atari games) [6]. Nevertheless, there are also some limitations to it, mainly related to the inefficiency of the sample that is caused by its policy properties, as well as its comparatively high sensitivity to the choice of hyperparameters. Proximal Policy Optimization (PPO) constrains the magnitude of updates in which the policy is updated so as to reduce aggressive updates, thus improving training stability and sample efficiency. PPO has done remarkably well in some benchmark tasks, including the ones involving Atari games and robot control tasks, where it is more sample efficient and stable compared to traditional policy gradient methods [6]. Maximum entropy reinforcement learning is applied in off-policy deep learning at SAC. Accumulative reward and policy entropy are maximized in SAC, unlike classical reinforcement learning. Growing policy entropy is an exploratory, less certain behavior. The SAC objective function can be expressed as:

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[r(s_t,a_t) + \alpha\mathcal{H}\left(\pi(\cdot\,|s_t)\right)\right] \tag{3}$$

This entropy is added in significant amounts to drive agent exploration to avoid early convergence to suboptimal deterministic strategies, and it enhances the robustness of the algorithm and its exploration performance. Technical terms are off-policy, experience replay, stochastic policy, and double Q-networks. To start with, off-policy and experience replay: in the off-policy algorithm, SAC has a large experience replay buffer where the historical interaction data are stored [5]. SAC randomly samples small batches of data in the buffer during training, and the innovations of earlier experience are reused, enabling it to attain extremely high sample efficiency. Second, stochastic policy: SAC also learns a stochastic policy during training (typically a Gaussian distribution), which inherently fits into the paradigm of maximum entropy [9]. Double Q-networks: similar to the TD3 method, SAC employs two Q-networks to minimize the overestimation of Q-values by taking the smaller of the two to estimate the goal Q-value and enhance more stable learning. The strengths of SAC are sample efficiency and consistency. It may, in many continuous control benchmark experiments (including MuJoCo), outperform other highly developed algorithms, such as PPO, DDPG, or TD3. Due to its off-policy nature, it is particularly suitable for online learning scenarios with high data collection costs that require efficient utilization of each experience.

SAC is an offline reinforcement learning algorithm based on maximum entropy reinforcement learning. It improves the policy's exploration and robustness by maximizing the balance between entropy and reward. SAC has demonstrated excellent performance in continuous control tasks, such as achieving expert-level performance in robotic control tasks within the MuJoCo simulator [5].

3 Offline Reinforcement Learning

Offline reinforcement learning aims to learn a decision-making policy entirely from a fixed, pre-collected dataset, without any new interactions with the environment during the learning process. This static dataset can come from human expert demonstrations, previously run policies, or various log data related to the task. Offline RL emerged primarily to overcome the deployment barriers of online learning in the real world. In many areas, people have vast amounts of historical data but cannot afford the cost or risk of exploring online. The vision of offline reinforcement learning is to use this data to learn a strategy that is better than the data collection strategy, which can open up new possibilities in areas such as medical diagnosis, recommendation systems, robotics skill learning, and autonomous driving. The main advantages of offline reinforcement learning are security and data utilization capabilities. The complete decoupling of data collection and strategy learning makes it possible to apply reinforcement learning in scenarios where online interaction is not possible. At the same time, offline reinforcement learning also faces unique and severe challenges, the core of which is Distributional Shift [10]. Since agents can only access state-action pairs within the dataset, when evaluating an "out-of-distribution" (OOD) action that never or rarely appeared during policy learning, the value function estimation may incur substantial, uncontrollable errors. Such errors propagate through Bellman updates, ultimately causing policy collapse. Therefore, effectively constraining the policy to remain within the bounds supported by the dataset is central to the design of offline reinforcement learning algorithms. The core of offline reinforcement learning algorithms is addressing the distribution shift problem. Conservative Q-Learning (CQL) and Implicit Q-Learning (IQL) are representative methods with different approaches. Principle and motivation of CQL: counteracting out-of-distribution actions. The central idea of CQL is to modify the learning objective to explicitly penalize high Q-values for actions not seen in the dataset while boosting Q-values for actions that do appear in the dataset. It aims to learn a conservative Q-function such that the expected value under the true policy is a lower bound of its true value [11]. Concretely, CQL adds a regularization term to the standard Bellman error loss. This regularization term minimizes the Q-values of all possible actions under the current policy, while maximizing the Q-values of actions recorded in the dataset. In this way, CQL can effectively suppress the overestimation of Q-values for OOD actions, thereby preventing the policy learner from selecting these uncertain actions. CQL is relatively simple to implement, requiring only the addition of a regularization term to the objective function of existing Q-learning or actor-critic algorithms (such as SAC), with minimal code changes and easy integration. CQL has various variants, such as CQL(H), which differ in the specific form of the regularization term to adapt to different scenarios and needs. At the same time, CQL demonstrates strong performance and robustness in offline tasks dealing with complex data distributions (e.g., multimodal, suboptimal data), achieving excellent results on standard offline RL benchmarks (such as D4RL). Its main challenges lie in the need for further research to theoretically guarantee its effectiveness with deep neural networks, and the significant impact of the regularization term's weight on performance, requiring careful tuning. [11] CQL is an offline reinforcement learning method that addresses the distribution shift problem in offline learning by explicitly penalizing the Q-values of out-of-distribution actions, thereby preventing the policy from selecting actions that do not exist in the dataset. CQL is used as a

baseline method in the D4RL benchmark to evaluate the performance of offline reinforcement learning algorithms. [12] IQL Algorithm Principle: Avoiding Explicit Q-Value Queries. The approach that IQL has to issues of distribution is distinct from that of CQL. To avoid punishment in cases of out-of-distribution (OOD) operations, it does not query them during policy updates at all. IQL states that incompetent estimations deceive the policy only if uncertain Q-values are interrogated. This job is done by the independent study of the state value function V(s) and the advantage function A(s, a). Weighted behavioral cloning is used to extract the policy and make the the functions of the dataset activities more advantageous. One of the methods that is important in IQL is the expectile regression, which is used to learn the state value V(s) [13]. Uniqueness Sometimes expectile regression can learn a unique data distribution quantile, which is not possible with mean squared error regression. Adjustment of quantile parameters. IQL does not require any action to compute an upper bound of state value, which can be used to approximate the process of maximization of Q(s, a), though it does not explicitly maximize this function. IQL is also easy to apply, with only some superficial adjustments in the SARSA-like temporal difference algorithm. The simplicity of the concept, ease of implementation, and computing efficiency are known to make IQL so appealing. Despite its simple idea, it has achieved performance comparable to or even better than CQL on standard offline RL benchmarks like D4RL, proving the effectiveness of this "avoiding problems" rather than "solving problems" approach. IQL is an offline reinforcement learning method that enhances policy performance by avoiding direct queries of Q-values for actions not present in the dataset, instead implicitly estimating the state value function. IQL demonstrates performance comparable to CQL on the D4RL benchmark and achieves state-of-the-art results across multiple offline tasks [14].

4 Comparative Analysis and Discussion

This section will conduct an in-depth comparison between online and offline reinforcement learning across key dimensions such as sample efficiency, deployment challenges, convergence, and hyperparameter sensitivity. The summary is shown in Table 1. First, regarding data interaction and sample efficiency, online deep learning is generally considered to have low sample efficiency. For instance, advanced online algorithms like SAC may require millions of interactions with the environment to learn an effective policy [15]. Because every minor policy improvement needs to be validated and data collected through new online interactions. This continuous interaction requirement is also known as low "deployment efficiency" because the learning process needs to deploy many different policies to collect data [15]. Offline reinforcement learning is extremely sample-efficient in the "interaction" sense because it generates no new interactions at all. The goal is to extract all useful information from a fixed dataset. However, the performance ceiling is constrained by the quality and coverage of the dataset. If the dataset lacks information about optimal behavior, offline algorithms will struggle to learn optimal strategies. Therefore, while offline reinforcement learning saves interaction costs, it demands higher data quality (As shown in Table 1). The second challenge is deployment difficulties in real-world applications. The primary deployment challenges of online reinforcement learning are high costs and safety risks. In domains like autonomous driving, healthcare, or finance, online exploration is almost infeasible [15]. Therefore, online RL is currently more commonly applied in simulators, games, or low-cost environments with controllable risks. Offline reinforcement learning was initially developed to take into consideration deployment issues in the real world. Offline RL applies secure data to generalize reinforcement learning in key areas. Generalization and distributional shift are, however, drawbacks in the deployment of offline RL. The policies trained on the offline datasets can be poor in practice when applied to real-world settings because of minor dissimilarity between the state distribution of the real setting

and the dataset (as presented in Table 1). Lastly, the policy's convergence and sensitivity to the hyperparameter are factors to consider. Theoretical analysis of convergence shows that the convergence process of online algorithms involves exploration, which may result in fluctuations. Offline algorithm convergence is a phenomenon that is independent of external noise and depends only on a fixed set of data and algorithm design. However, if the algorithm is poorly designed (e.g., failing to handle distribution shift effectively), it might not converge to a meaningful policy at all. Some research suggests that using offline data can accelerate the convergence of online learning. Hyperparameter sensitivity is a common issue in deep reinforcement learning. Studies indicate that offline reinforcement learning algorithms are generally more sensitive to hyperparameters than online reinforcement learning algorithms. The reason is that in offline learning, there's no online interaction to provide "correction" signals. A poor hyperparameter choice can lead to an infinite amplification of cumulative errors in value estimation, whereas online learning can partially mitigate this problem through real-time feedback from the environment. How to reliably select hyperparameters for offline algorithms is itself an active area of research, as traditional cross-validation methods are difficult to apply directly in an offline setting.

Table 1. Key Dimensions Comparison Between Online and Offline Reinforcement Learning

Feature Dimension	Online reinforcement learning	Offline reinforcement learning
Sample efficiency and data interaction	Continuous interaction with the environment is required, resulting in relatively low sample efficiency. Policy fine-tuning relies on newly added trajectories [15].	No new interactions; highly sample-efficient in terms of interaction cost, but constrained by data quality and coverage limits.
Deployment challenges and applications	High costs and safety/ethical risks limit exploration in genuinely high-risk environments.[15]	Leveraging existing security data reduces deployment risks but is constrained by distribution shifts and generalization challenges.
Convergence	Accompanied by exploration, the curve is prone to fluctuation; deviations can be corrected through environmental feedback.	Determined by data and algorithms; improper handling may lead to difficulty in convergence; offline pre-training can accelerate convergence in the online phase.
Hyperparameter sensitivity	Sensitive Yet Buffered by Online Feedback.	More sensitive; lacking online error correction, incorrect hyperparameters amplify value estimation errors.

Note: Deployment efficiency refers to how many different policies need to be deployed to collect data during the learning process; distributional shift refers to the performance degradation caused by the mismatch between the state-action-reward distribution of training data and the real deployment distribution

5 Conclusion

Online and offline reinforcement learning represent two distinct paradigms of data-driven decision-making, each with its own advantages and disadvantages, and applicable to different scenarios. Online learning, with its flexibility and continuous optimization capabilities, shows great potential in simulated environments and low-risk scenarios. Offline learning, on the other hand, with its safety and ability to leverage existing data, has successfully paved the way for reinforcement learning in high-risk, high-cost real-world applications. Examination of some sample algorithms indicates that online reinforcement learning

techniques, such as PPO and SAC, take different tradeoffs when it comes to stability, exploration efficiency, and sample efficiency. The offline learning approaches to reinforcement, such as CQL and IQL, provide new concepts that address the fundamental issue of distribution shifting classically in the adversarial and avoidance viewpoints. In comparison, the availability of data, the cost of interaction, the safety factor, and the characteristics of the problem dictate the selection of either of the two. The way forward in future research could be to focus more on hybrid research approaches that combine the best of the two. Offline-to-Online Fine-tuning How offline data can be pretrained and fine-tuned in an online setting: How to use offline data to pretrain strategies and optimize them in a safe online setting or How to design algorithms that use offline data with high efficiency and limited and safe online exploration How to design algorithms that do restricted and safe online exploration How to use offline data to pretrain strategies and optimize them in a safe online setting. The study will also result in the further evolution of reinforcement learning theory and its faster introduction to a wider field of more complicated real-life issues.

References

1. Q. Wen, W. Chen, L. Sun, Z. Zhang, L. Wang, R. Jin, T. Tan, Onenet: Enhancing time series forecasting models under concept drift by online ensembling. Adv. Neural Inf. Process. Syst. 36, 69949–69980 (2023)
2. B. Kim, M.H. Oh, Model-based offline reinforcement learning with count-based conservatism, in Proceedings of the International Conference on Machine Learning, 16728–16746 (2023)
3. T. Xie, D.J. Foster, Y. Bai, N. Jiang, S.M. Kakade, The role of coverage in online reinforcement learning. arXiv:2210.04157 (2022)
4. C. Lu, P.J. Ball, T.G. Rudner, J. Parker-Holder, M.A. Osborne, Y.W. Teh, Challenges and opportunities in offline reinforcement learning from visual observations. arXiv:2206.04779 (2022)
5. T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in Proceedings of the International Conference on Machine Learning, 1861–1870 (2018)
6. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms. arXiv:1707.06347 (2017)
7. L. Wang, X. Hu, Y. Wang, S. Xu, S. Ma, K. Yang, W. Wang, Dynamic job-shop scheduling in smart manufacturing using deep reinforcement learning. Comput. Netw. 190, 107969 (2021)
8. C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, Y. Wu, The surprising effectiveness of PPO in cooperative multi-agent games. Adv. Neural Inf. Process. Syst. 35, 24611–24624 (2022)
9. R. Reiter, A. Ghezzi, K. Baumgärtner, J. Hoffmann, R.D. McAllister, M. Diehl, AC4MPC: Actor-critic reinforcement learning for nonlinear model predictive control. arXiv:2406.03995 (2024)
10. Z. Jia, A. Rakhlin, A. Sekhari, C.Y. Wei, Offline reinforcement learning: Role of state aggregation and trajectory data, in Proceedings of the Conference on Learning Theory, 2644–2719 (2024)
11. A. Kumar, A. Zhou, G. Tucker, S. Levine, Conservative Q-learning for offline reinforcement learning. Adv. Neural Inf. Process. Syst. 33, 1179–1191 (2020)

12. J. Lyu, X. Ma, X. Li, Z. Lu, Mildly conservative Q-learning for offline reinforcement learning. Adv. Neural Inf. Process. Syst. 35, 1711–1724 (2022)
13. J. Luo, P. Dong, J. Wu, A. Kumar, X. Geng, S. Levine, Action-quantized offline reinforcement learning for robotic skill learning, in Proceedings of the Conference on Robot Learning, 1348–1361 (2023)
14. I. Kostrikov, A. Nair, S. Levine, Offline reinforcement learning with implicit Q-learning. arXiv:2110.06169 (2021)
15. T. Matsushima, H. Furuta, Y. Matsuo, O. Nachum, S. Gu, Deployment-efficient reinforcement learning via model-based offline optimization. arXiv:2006.03647 (2020)