CPE: A New Paradigm for Policy Extraction in Offline Reinforcement Learning

Extended Abstract

Zhaohui Yang
Institute of Automation, Chinese Academy of Sciences
Beijing, China
yangzhaohui2023@ia.ac.cn

Xiaoxuan Wang
Institute of Automation, Chinese Academy of Sciences
Beijing, China
xiaoxuan.wang@ia.ac.cn

Linjing Li
Institute of Automation, Chinese Academy of Sciences
Beijing, China
linjing.li@ia.ac.cn

ABSTRACT

Offline reinforcement learning (RL) aims to extract the optimal policy from static offline datasets but always encounters the notorious distribution shift problem. In order to address this problem, many previous offline RL algorithms primarily rely on modifications at policy evaluation stage. However, the performance gap between different policy extraction methods is significant even under the same value function. Thus, to address this issue, we focuses on the policy extraction stage and introduces a novel policy extraction method called Contrastive Policy Extraction (CPE), which samples action pairs at each state and leverages their relative values to improve the policy. By reformulating the optimal policy parameterization problem as a root-finding problem, CPE enhances the policy extraction capability and surpasses current prominent extraction methods in offline RL, such as AWAC and TD3BC. The proposed CPE is implemented within the iterative actor-critc framework and it substantially outperforms current state-of-the-art (SOTA) offline RL algorithms on D4RL benchmarks.

KEYWORDS

Offline Reinforcement Learning; Policy Extraction Method; Deep Learning

ACM Reference Format:
Zhaohui Yang, Xiaoxuan Wang, and Linjing Li. 2025. CPE: A New Paradigm for Policy Extraction in Offline Reinforcement Learning: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

1 INTRODUCTION

Offline reinforcement learning (RL) algorithms mainly can be divided into two categories: the onestep algorithm and the iterative algorithm. Both algorithms consist of two stages: policy evaluation and policy extraction. Recent studies [4, 8] have found that the choice of a policy extraction method often has a larger impact on performance than the policy evaluation algorithm. Therefore, we propose a novel policy extraction method, named Contrastive Policy Extraction (CPE). It overcomes the drawbacks of the weighted behavior cloning method [6, 7, 9]: the constrained action sampling space and the non-negativity of the gradient, while maintaining a concise implementation. On the D4RL benchmark, we achieve SOTA performance acorss multiple datasets.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

2 METHOD

2.1 Formulation of the Optimal Policy

Firstly, we theoretically derive the mathematical formulation of the optimal policy. We define the expected improvement of current policy π over a sampling policy $\mu(a|s)$ as

$$\eta(\pi) = J(\pi) - J(\mu) = \mathbb{E}_{\tau\sim p_\pi(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t A^\mu(s_t, a_t)\right]. \tag{1}$$

Nevertheless, in offline RL, we cannot interact with the environment to obtain trajectories under π. Consequently, we restrict the distance between the current policy π and the behavior policy μ, then we approximate $\eta(\pi)$ under state distribution of μ,

$$\hat{\eta}(\pi) = \mathbb{E}_{s\sim d_\mu, a\sim\pi}\left[A^\mu(s,a)\right]. \tag{2}$$

Using this objective, we can formulate the following constrained policy search problem:

$$\begin{aligned}\arg\max_\pi \quad & \int_s d_\mu(s)\int_a \pi(a|s)A^\mu(s,a)dads \\ \text{s.t.} \quad & \mathrm{D}_{\mathrm{KL}}\left(\pi(\cdot|s)||\mu(\cdot|s)\right) \le \epsilon, \quad \forall s\end{aligned} \tag{3}$$

According to KKT conditions, it is straightforward to derive that the optimal policy π^* has the following form[9],

$$\pi^*(a|s) = \frac{1}{Z(s)}\mu(a|s)\exp\left(\beta A^\mu(s,a)\right), \tag{4}$$

where

$$Z(s) = \int_a \mu(a|s)\exp\left(\beta A^\mu(s,a)\right)da, \tag{5}$$

is the partition function.

2.2 Extraction of the Optimal Policy

Since calculating $Z(s)$ is forbidden, we cannot directly use equation (4) to calculate π^*. The AWAC[7] parameterizes the optimal policy by minimizing the Kullback-Leibler (KL) divergence, but researches [4, 8] have shown that this approach tends to be overly conservative. The issue with this conservativeness is that AWAC does not provide a clear signal to reduce the probability of poor actions. While it assigns more weight to good actions, it may not effectively suppress bad ones. This phenomenon has also been observed in the field of

RLHF, where negative gradients are crucial for policy learning in an offline setting [10].

CPE overcomes the above shortcoming by contrasting action pairs. For each state s, we sample action pairs from π. By comparing the $Q^\pi(s,a)$ of the two actions, we explicitly increase the probability of the good action while decreasing the probability of the bad one. Comparing action pairs on each state provides explicit signals that facilitate the learning process of the model. At the same time, we use behavior cloning (BC) loss to control the overall policy shift, which ensures a series of steady improvements until convergence.

2.2.1 Objective of CPE. For the sake of clarity in derivation, we first define

$$g_{\pi,(s,a^1,a^2)} := \frac{\pi(a^1|s)}{\pi(a^2|s)}, \tag{6}$$

$$\Delta Q_{\mu,(s,a^1,a^2)} := Q^\mu_{s,a^1} - Q^\mu_{s,a^2}. \tag{7}$$

According to equation (4), π^* must satisfy to the following relationship:

$$\frac{\pi^*(a^1|s)}{\pi^*(a^2|s)} = \frac{\mu(a^1|s)}{\mu(a^2|s)} \exp\left(\beta\left(Q^\mu_{s,a^1} - Q^\mu_{s,a^2}\right)\right). \tag{8}$$

Taking the logarithm of both sides of equation (8) yields the following equivalent expression:

$$\log \frac{g_{\pi^*,(s,a^1,a^2)}}{g_{\mu,(s,a^1,a^2)}} = \beta \Delta Q_{\mu,s,a^1,a^2}. \tag{9}$$

Now the key idea is to consider a policy π to solve the equation (9). We can further reformulate the above root-finding problem into an optimization problem:

$$L_1(\pi) = \mathbb{E}_{a^1,a^2\sim\mu}\left[\left(\log \frac{g_{\pi,(s,a^1,a^2)}}{g_{\mu,(s,a^1,a^2)}} - \beta \Delta Q_{\mu,s,a^1,a^2}\right)^2\right]. \tag{10}$$

Then we use a neural network π_θ to parameterize π and solve the optimization problem (10) by gradient descent methods.

In addition, we incorporate a BC loss $L_2(\theta) = -\log\pi(a|s;\theta)$ to control the overall degree of distribution shift, which keeps the proposed CPE simple and efficient. Then we get the final objective of CPE,

$$L(\theta) = L_1 + \lambda L_2. \tag{11}$$

Algorithm 1 instantiates a version of iterative actor-critic framework with CPE (hereinafter referred to as IAC-CPE). The complete expression of $L(\theta)$ is,

$$L(\theta) = \mathbb{E}_{s\sim D,(a^1,a^2)\sim\pi'_k}\left[\left(\log \frac{g_{\pi_k,(s,a^1,a^2)}}{g_{\pi'_k,(s,a^1,a^2)}} - \beta \Delta Q_{\pi'_k,s,a^1,a^2}\right)^2\right] - \lambda \mathbb{E}_{(s,a)\sim D}\left[\log(a|s)\right]. \tag{12}$$

Algorithm 1 Iterative actor-critic framework with CPE (IAC-CPE)

for $k = 1, \ldots, K$ **do**
  Sample minibatch $B = \{(s,a,r,s')\}$ form D.
  Compute target Q-values

$$y = r + \gamma(\min_{i=1,2} Q_{\phi'_i}(s', a')), a' \sim \pi'_k(\cdot \mid s') \tag{13}$$

  Update each Q function Q_{ϕ_i} with gradient descent using

$$\nabla_{\phi_i} \frac{1}{|B|} \sum_{(s,a,r,s')\in B} \left(\left(Q_{\phi_i}(s,a) - y(r,s')\right)\right)^2 \tag{14}$$

  Sample action pairs $\{(s,a^1,a^2,r)\}$, $a^1, a^2 \sim \pi'_k(\cdot \mid s)$.
  Update actor network π_k: $\theta \leftarrow \theta - \alpha\nabla_\theta L(\theta)$.
  $\pi'_k \leftarrow (1-\tau)\pi'_k + \tau\pi_k$.
  For $(Q_\phi, Q_{\phi'}) \in \{(Q_{\phi_i}, Q_{\phi'_i})\}$, update target

$$Q_{\phi'_i} = (1-\tau)Q_{\phi'_i} + \tau Q_{\phi_i}. \tag{15}$$

end for

3 EXPERIMENTS

D4RL [3] is a widely used evaluation environment for offline RL, encompassing a wide range of tasks and datasets. To evaluate the performance of the proposed CPE, we implement IAC-CPE and conduct experiments on both single quality and mixed quality datasets in D4RL. The results for 10%BC, and DT [1] are based on performance summarized in the work by Emmons et al [2]. For other SOTA algorithms, including AWAC, TD3BC [5] and IQL [6], we rerun experiments based on the public repositories [11] and report results in Table 1. To evaluate the performance fairly, we run IAC-CPE on 10 evaluation trajectories and 5 random seeds. The results summarized in Table 1 demonstrate that IAC-CPE significantly outperforms other algorithms, achieving a lead of over 10%.

Table 1: Evaluation results of IAC-CPE and baselines on the D4RL dataset. The performance is measured by the normalized scores at the last training iteration. Bold indicates the best performance in each task. The abbreviations h, hp, and w correspond to HalfCheetah, Hopper, and Walker2D respectively. The suffixes r, m, mr and me stand for random, medium, medium-replay, and medium-expert respectively.

Task	10%BC	AWAC	DT	TD3BC	IQL	IAC-CPE
h-r	2.0	11.3	2.2	11.1	9.5	**26.67**
hp-r	4.1	**15.7**	7.5	8.6	7.4	8.12
w-r	1.7	3.3	2.0	0.4	4.0	**5.62**
h-m	42.5	49.9	42.6	48.3	48.3	**57.63**
hp-m	56.9	64.5	67.6	63.8	59.5	**100.61**
w-m	75.0	76.1	74.0	**80.5**	80.9	79.85
h-mr	40.6	45.7	44.7	36.6	43.7	**51.18**
hp-mr	75.9	97.6	82.7	55.6	89.4	**99.95**
w-mr	62.5	74.6	66.6	76.4	80.6	**91.35**
h-me	92.9	**95.8**	86.8	88.1	91.2	94.74
hp-me	**110.9**	107.3	107.6	95.6	105.9	103.02
w-me	109.0	103.6	108.1	110.5	**112.1**	108.91

4 ACKNOWLEDGEMENTS

This work was supported in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA27030100 and the National Natural Science Foundation of China under Grant 72293575.

REFERENCES

[1] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems* 34 (2021), 15084–15097.

[2] Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. 2021. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751* (2021).

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

[4] Yuwei Fu, Di Wu, and Benoit Boulet. 2022. A closer look at offline rl agents. *Advances in Neural Information Processing Systems* 35 (2022), 8591–8604.

[5] Scott Fujimoto and Shixiang Shane Gu. 2021. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems* 34 (2021), 20132–20145.

[6] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2021. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169* (2021).

[7] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. 2020. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359* (2020).

[8] Seohong Park, Kevin Frans, Sergey Levine, and Aviral Kumar. 2024. Is Value Learning Really the Main Bottleneck in Offline RL? *arXiv preprint arXiv:2406.09329* (2024).

[9] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. 2019. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177* (2019).

[10] Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. 2024. Preference fine-tuning of llms should leverage suboptimal, on-policy data. *arXiv preprint arXiv:2404.14367* (2024).

[11] Denis Tarasov, Alexander Nikulin, Dmitry Akimov, Vladislav Kurenkov, and Sergey Kolesnikov. 2024. CORL: Research-oriented deep offline reinforcement learning library. *Advances in Neural Information Processing Systems* 36 (2024).