Efficient and Stable Offline-to-online Reinforcement Learning via Continual Policy Revitalization

Rui Kong, Chenyang Wu, Chen-Xiao Gao, Zongzhang Zhang*, Ming Li

National Key Laboratory for Novel Software Technology, Nanjing University, China
School of Artificial Intelligence, Nanjing University, China
{kongr, wucy, gaocx}@lamda.nju.edu.cn, {zzzhang, lim}@nju.edu.cn

Abstract

In offline Reinforcement Learning (RL), the pre-trained policies are utilized for initialization and subsequent online fine-tuning. However, existing methods suffer from instability and low sample efficiency compared to pure online learning. This paper identifies these limitations stemming from direct policy initialization using offline-trained policy models. We propose Continual Policy Revitalization (CPR) as a novel efficient, stable fine-tuning method. CPR incorporates a periodic policy revitalization technique, restoring the overtrained policy network to full learning capacity while ensuring stable initial performance. This approach enables fine-tuning without being adversely affected by low-quality pre-trained policies. In contrast to previous research, CPR initializes the new policy with an adaptive policy constraint in policy optimization. Such optimization keeps the new policy close to behavior policy constructed from historical policies. This contributes to stable policy improvement and optimal converged performance. Practically, CPR can seamlessly integrate into existing offline RL algorithms with minimal modification. We empirically validate the effectiveness of our method through extensive experiments, demonstrating substantial improvements in learning stability and efficiency compared to previous approaches. Our code is available at https://github.com/LAMDA-RL/CPR.

1 Introduction

Deep Reinforcement Learning (RL) involves agents learning through interactions with the environment and encoding decision-making knowledge with neural networks. It has demonstrated significant potential in mastering complex tasks and developing robust policies. However, attaining high performance often demands millions or even billions of samples [Ye *et al.*, 2021; Wu and Zhang, 2023]. The notorious sample inefficiency of deep RL obstructs its application to many problems where only limited interactions with the environment are acceptable [Yu, 2018; Zhou *et al.*, 2024].

Offline RL algorithms offer a novel approach, enabling the learning of policies from offline data generated by a behavior policy. This avenue has garnered significant interest from both academia and industry [Gao *et al.*, 2024; Ran *et al.*, 2023]. Though the offline methods exempt RL from costly online interactions, they often learn policies inadequate to meet the practical needs due to the insufficient data coverage [Kumar *et al.*, 2019].

To further fine-tune these agents in the online environment, Offline-to-Online (O2O) RL [Lee *et al.*, 2021] is proposed. Fine-tuning is expected to increase the agent's performance monotonically in RL just as the supervised fine-tuning. Still, it is observed empirically that directly tuning the pre-trained policy model gets it ruined [Uchendu *et al.*, 2022; Yu and Zhang, 2023]. We refer to this paradigm of updating policy function with offline trained checkpoints as **direct policy initialization**. We argue such a paradigm would cause instability and low sample efficiency due to two issues:

- Distribution shift between the offline state-action distributions and the online environment would cause unreliable value prediction. On the one hand, extrapolation error propagation [Fujimoto *et al.*, 2019] in Out-Of-Distribution (OOD) data devastates the pre-trained policy quickly with a few online updates. On the other hand, over-conservatism in offline learning [Nakamoto *et al.*, 2023] hinders online value function updating.
- Primacy bias of the overtrained policy function impedes subsequent learning [Nikishin *et al.*, 2022]. The loss of plasticity in pre-trained policy [Abbas *et al.*, 2023] hinders continual learning and degenerates the agent's asymptotic performance especially when the offline dataset is of low quality.

Existing work focuses on alleviating the negative impact caused by direct policy initialization. Several O2O algorithms have been proposed to mitigate this issue of distribution shift. Unfortunately, these methods have their drawbacks. In addition, the primacy bias issue has not been identified in the context of O2O RL previously.

In this work, we delve into the negative effects of direct policy initialization in O2O RL. Our contributions unfold in two main facets: firstly, the identification and analysis of is-

*Zongzhang Zhang is the corresponding author.

sues associated with direct policy initialization, substantiated by empirical evidence; and secondly, the introduction of a stable and efficient O2O RL methodology named **Continual Policy Revitalization (CPR)**. CPR is designed to replace an overtrained offline policy with a newly revitalized one during the policy revitalization stage. This revitalized policy begins as a blank slate, maintaining receptiveness to new knowledge, whereas the historical policies are preserved in the policy set to prevent forgetting. To balance learning and forgetting, we compose all existing policies to decide which action to choose. The new policy is initialized by efficient offline training with an adaptive policy constraint. CPR enjoys benefits absent from the previous methods. It is free of assumptions about offline pre-trained algorithms and does not assume access to offline datasets. The combination of policy revitalization and adaptive policy constraint provides stability and efficiency in the learning process. We empirically confirm the effectiveness of our method by a comprehensive comparison with existing algorithms, highlighting the efficacy of independent components within CPR.

2 Background

2.1 Markov Decision Process

We consider a standard Markov Decision Process (MDP) defined by $M = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \rho, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $R : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$ is the normalized reward function, $P : \mathcal{S} \times \mathcal{A} \mapsto \Delta_{\mathcal{S}}$ is the transition function, Δ_X is the set of distribution over the set X, ρ is the initial state distribution, and $\gamma \in [0, 1)$ is the discount factor.

A deterministic policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ is a function that maps a state to an action. For each policy, the corresponding value function $V^\pi : \mathcal{S} \mapsto \mathbb{R}$ and Q-function $Q^\pi : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ are the expected discounted cumulative reward (a.k.a. return):

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s \right],$$

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, a_0 = a \right].$$

We measure the performance of a policy by its expected return $J(\pi) = \mathbb{E}_{s\sim\rho}[V^\pi(s)]$. The goal in an MDP is to find a policy that maximizes $J(\pi)$, i.e., $\pi^\star = \arg\max_\pi J(\pi)$.

2.2 Offline RL

Online interaction with real environments could be costly or unsafe, in which cases offline RL algorithms [Levine *et al.*, 2020] are preferred. These algorithms train a policy π_β on an offline logged dataset $\mathcal{D}$ collected by a behavior policy from the environment. The dataset is composed of transition tuples (s, a, r, s'), where $r = R(s, a)$ is the reward obtained when taking action a at state s, and $s' \sim P(\cdot \mid s, a)$ is the next state sampled by the environment. The trained policy π_β is expected to maximize the expected return as much as possible given the information presented by the dataset.

2.3 Offline-to-Online (O2O) RL

O2O RL learns the optimal policy by interacting with the environment starting with a pre-trained policy π_β obtained by offline RL training. Other forms of knowledge can also be accessible under certain assumptions, such as offline datasets and value functions. An O2O method tries to transfer the knowledge acquired in the offline stage to enable sample-efficient online learning and poses unique challenges.

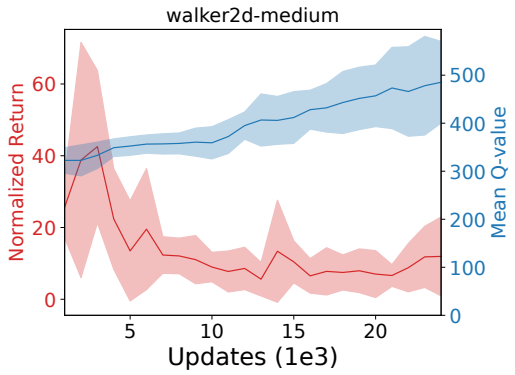

Figure 1: D4RL normalized return and mean Q-value of SAC on Walker. Normalized return of 0 corresponds to a random policy and 100 corresponds to an expert policy. The network is initialized from an offline pre-trained AWAC model on walker2d-medium-v2. The results are averaged on experiments of 5 random seeds.

3 Issues of Direct Policy Initialization

This section exposes the main issues of direct policy initialization in O2O RL. We empirically demonstrate the damage of these issues, which motivates the proposal of our method.

3.1 Distribution Shift

It is observed in previous research that O2O RL via off-policy RL algorithms suffers from a performance drop in the initial stage, which is attributed to distribution shift [Lee *et al.*, 2021; Yu and Zhang, 2023]. As demonstrated in Figure 1, we conduct a tuning experiment with direct policy initialization. We train a SAC policy [Haarnoja *et al.*, 2018] initialized from an AWAC [Nair *et al.*, 2020] model pre-trained on the same task. We observe the performance drop is concomitant with severe value overestimation of the Q-value function. Therefore, we hypothesize that the distribution shift in the tuning stage leads to the well-known extrapolation error problem [Fujimoto *et al.*, 2019] and the extrapolation error further leads to performance drop.

Though extrapolation error is thoroughly discussed in offline RL, we would like to take one step further in the context of O2O RL. The OOD actions would also be encountered in off-policy learning, however, direct policy initialization from a checkpoint and updating the model with naive off-policy learning is a common practice in online RL. In online off-policy learning, things do not get too bad because the overestimated value gets corrected when the policy executes these actions in the stage of training. Nevertheless, this problem is exacerbated in offline RL, where the policy does not get a chance to interact with the environment to fix its value estimation. In this case, the extrapolation error of value estimation at OOD actions gets continuously amplified by bootstrapping and makes the Q-value estimation unreliable. Ini-

Step	Average Return $\hat{R}'$	Drop Ratio ρ
0	3135.55	34.15%
1000	2745.57	42.34%
3000	2516.06	47.16%
5000	2377.49	50.07%

Table 1: The performance drop of SAC after updating the policy on Hopper. The average return, denoted as $\hat{R}'$, is computed based on the evaluation of 100 episodes using 5 different random seeds. In each run, the policy is initialized with the offline model pre-trained by AWAC with the same seed. The drop ratio is denoted as $\rho = \frac{\hat{R}-\hat{R}'}{\hat{R}}$, where $\hat{R}$ is the evaluated return of the pre-trained models.

tializing from offline Q-value function makes finetuning extremely unstable compared to common off-policy learning.

This distinction is the fundamental difference between the tuning processes of offline models and online learning. At the initial stage of O2O RL, the online buffer contains only a few samples, which makes numerous actions out-of-distribution, and the extrapolated value estimate for some of them could be fairly high. Therefore, the policy has an incentive for deviating from the pre-trained policy and selecting these OOD actions, which immediately causes a performance drop.

Since Figure 1 only presents the co-occurrence of performance drop and over-estimation, more evidence is required to confirm it. We designed a controlled experiment to investigate whether the policy update is biased because of the unreasonable Q-value prediction. We fine-tune a pre-trained policy with the SAC algorithm in the online MuJoCo [Todorov *et al.*, 2012] Hopper environment. The policy is pre-trained on the D4RL [Fu *et al.*, 2020] hopper-medium-expert-v2 dataset with the AWAC algorithm. During O2O RL, we first collect 10000 transition tuples from the online environment with pre-trained policy and update the policy with the SAC algorithm. Then, we keep the policy unchanged and update the Q-value function with additional gradient steps using online data. After that, the Q-value network is frozen and we conduct policy improvement for 1000 gradient steps. As shown in Table 1, additional updates hurt the performance due to the amplification of value extrapolation error and policy deviation.

Our analysis and empirical results show that visiting OOD actions should be avoided in online fine-tuning updates to achieve a stable fine-tuning process, especially in the beginning stage. This requires adding constraints on the policy update to make the visiting distribution not too far from the offline dataset. When it comes to the case that the offline dataset is not accessible, we should let the policy stay close to the offline trained policy.

3.2 Primacy Bias in Offline RL

Another issue of direct policy initialization in O2O RL is the primacy bias in neural networks. The primacy bias in deep RL is first observed as a tendency to overfit early experiences in online learning, which damages the rest of the learning process [Nikishin *et al.*, 2022]. Although such observation is intuitive in supervised learning and transfer learning, primacy bias in single-task offline RL is more difficult to identify. Excessive training on the early collected data could trap the neural network into unrecoverable plasticity loss. In common offline RL experiments, networks are trained for hundreds or thousands of epochs on the offline dataset, inevitably leading to overfitting.

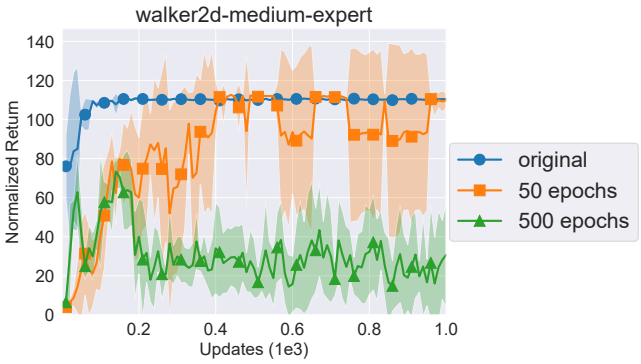

Figure 2: The performance of TD3+BC on the Walker2d-medium-expert-v2 dataset when loading pre-trained model after different epochs (1000 updates) of training on the Walker2d-random-v2 dataset. Scores are averaged over five random seeds, and the shaded areas in the plot represent the standard deviation. A score of 100 represents the average performance of a domain expert, and a score of 0 represents the performance of a uniform random policy.

Unfortunately, even if the policy suffers from plasticity loss, popular evaluation metrics may fail to exhibit any evidence of primacy bias. A policy may perform well in terms of evaluation scores but can also be extremely inefficient in the further learning process. So, simple online evaluation cannot reflect the actual continual learning ability of a policy.

We design a learning ability test experiment to demonstrate how primacy bias could suppress the learning ability of a pre-trained model. We first train the offline model by the TD3+BC algorithm [Fujimoto and Gu, 2021] in the walker2d-random-v2 dataset of D4RL. We then use the saved checkpoint of different training epochs to initialize the policy that is further trained on the walker2d-medium-expert-v2 dataset. We use this dataset as an unseen data distribution to test the continual learning capability. This fixed dataset helps to ablate the effects of data collection in the online stage. As depicted in Figure 2, the policy initialized with a model trained for 500 epochs falls short of attaining the performance level it would have reached if learned from scratch. On the contrary, the policy initialized with an insufficiently fitted model recovers from the distribution shift at the beginning stage. Considering that offline RL algorithms normally train over 1000 epochs on the D4RL MuJoCo datasets, 500 epochs should be considered as a common setting in practice.

The damage of primacy bias in offline RL suggests direct policy initialization from the offline trained policy could cause low learning efficiency. This issue limits the performance of direct policy initialization.

4 Methodology

This section presents our method to resolve the issues of direct policy initialization. In Section 4.1, we first introduce the continual policy revitalization technique. Then we discuss the

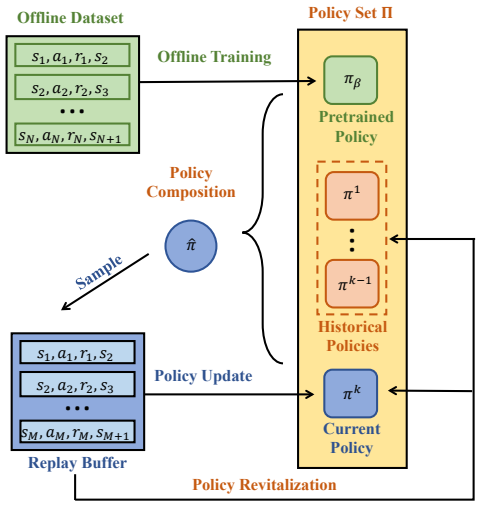

Figure 3: The overall framework of CPR. It maintains a policy set Π to preserve all frozen historical policies $\{\pi_\beta, \pi^1, \ldots, \pi^{k-1}\}$ and the current policy π^k. These policies are composed to be $\hat{\pi}$ as the online behavior policy. Periodically, CPR adds a new policy with full learning capacity to Π. This new policy is initialized with an adaptive policy constraint in the policy revitalization procedure.

proper design of policy constraint after revitalization in Section 4.2. The practical implementation is given in Section 4.3. The overall framework is shown in Figure 3.

4.1 Continual Policy Revitalization

As discussed before, the existing issues of direct policy initialization pose a challenge to designing appropriate O2O RL algorithms. On the one hand, the policy should be as close to the offline trained policy to alleviate the distribution shift. On the other hand, direct policy initialization from the over-trained offline policy could cause primacy bias, hindering online learning efficiency. Our intuitive idea is to find a surrogate of the given offline model with full learning capacity and comparable performance.

Inspired by this intuition, we propose Continual Policy Revitalization (CPR) to reinvigorate the continual learning capability of the policy function. Specifically, we first define the procedure of **policy revitalization** which is periodically performed every T_r epoch. At the k-th round of policy revitalization, we initialize the parameter of the policy network to generate a new policy π^{k+1}. While maintaining the value network Q_μ of parameter μ unchanged, we exploit an arbitrary offline training algorithm $\mathrm{A}_{\text{offline}}$ to initialize π^{k+1} on the online replay buffer $\mathcal{D}$ with batch RL training:

$$\pi^{k+1} \leftarrow \mathrm{A}_{\text{offline}}(\mathcal{D}, Q_\mu). \tag{1}$$

This designed procedure ensures that the revitalized policy π^{k+1} is not severely affected by previous training and is well-conditioned for continual learning. Since the size of the online replay buffer $\mathcal{D}$ is limited, policy revitalization uses the offline RL method to achieve sample efficiency and stable policy initialization with a high replay ratio in batch training.

Of course, the offline training in policy revitalization could not clone the previous behavior policy perfectly. Inspired by PEX [Zhang *et al.*, 2023], we maintain a policy set Π and add the **policy composition** mechanism in the action selection stage. At the k-th round of policy revitalization, we add the current policy π^k to the policy set $\Pi \leftarrow \Pi \cup \{\pi^k\}$ to avoid catastrophic forgetting. When interacting with the environment at state s, each policy π^i in the policy set Π proposes a candidate action a_i and the agent would select the action with the Boltzmann distribution of the predicted Q-value $Q^{\pi^k}(s, a_i)$. Here π^k is the current policy. The probability of selecting action proposed by policy π^i, $P^\star(i)$, is defined as:

$$P^\star(i) = \frac{\exp\left(Q^{\pi^k}(s, a_i)/\eta\right)}{\sum_j \exp\left(Q^{\pi^k}(s, a_j)/\eta\right)}, i \in \{1, \cdots, k\}. \tag{2}$$

Here η is the temperature hyper-parameter to balance exploitation and exploration. In this way, we perform a one-step policy improvement with maximal entropy at every action selection step in the following form:

$$P^\star = \arg\max_P \left[\sum_{i=1}^{k} P(i) Q^{\pi^k}(s, a_i) - \eta P(i) \log P(i)\right]. \tag{3}$$

We use the notation of $\hat{\pi}^k$ to represent the composed policy distribution at the k-th round of policy revitalization. By sampling from the composed policy $\hat{\pi}^k$, we combine the decision knowledge of all policies. With a higher η, the mixed policy tends to explore and randomly choose from the proposed actions. On the contrary, with a small η, the mixed policy makes greedy decisions based on the evaluated action quality of the proposed actions. Since all policies are archived in the policy set, historical knowledge could be preserved even if old data is no longer stored. This mechanism protects the policy from radical forgetting and policy improvement.

4.2 Adaptive Policy Constraint

Since we set the composed policy $\hat{\pi}$ as the behavior policy in the online replay buffer, CPR does not adopt a fixed policy constraint in pure offline RL or an iterative policy regularization in online off-policy learning. Instead, CPR uses the policy set to propose an adaptive policy constraint. We use subscript to denote the policy iteration step and superscript to represent the number of policy revitalization rounds. Thus, the optimization of the current policy at the k-th round of policy revitalization can be formalized as:

$$\begin{aligned} \pi^k = &\arg\max_\pi \mathbb{E}_{s\sim\rho_{\hat{\pi}^k}} \mathbb{E}_{a\sim\pi(\cdot|s)} Q^\pi(s, a) \\ &\text{s.t. } \mathbb{E}_{s\sim\rho_{\hat{\pi}^k}} D_{\mathrm{KL}}\left(\hat{\pi}^k(\cdot \mid s) \| \pi(\cdot \mid s)\right) \le \epsilon, \end{aligned} \tag{4}$$

where $\rho_{\hat{\pi}^k}$ is the state visitation distribution of $\hat{\pi}^k$, ϵ is a small positive number, and D_{KL} is the Kullback–Leibler (KL) divergence. Eq. 4 implements a trust region policy update adapting to the change of behavior policy.

Compared to iterative policy constraint in PPO [Schulman *et al.*, 2017] which constrains the updated policy to close with

the current policy, our adaptive policy constraint can alleviate harmful forgetting in recent policy updates with no significant drop of useful knowledge from the offline dataset. Thus CPR can gain a more stable performance improvement. Different from fixed policy constraints in offline RL [Fujimoto *et al.*, 2019], CPR can achieve higher performance even if the offline pre-trained policy is of low quality.

All historical policies within the policy set engage in competition, each vying for their actions to be manifested in the behavioral policy. Over time, only proficient actions are preserved in the replay buffer. The quality of the behavior policy of CPR is improved together with the update of the Q-value function. This, in turn, elevates the initial performance of the freshly revitalized policy. Thus our policy composition mechanism leads to an **adaptive policy constraint** where the target policy distribution is improved in learning.

4.3 Practical Implementation

Though the analytical distribution of $\hat{\pi}^k$ is complex, we show that the implementation can be simple in practice. The KL divergence inequality constraint of Eq. 4 can be written as the Lagrangian $\mathcal{L}(\pi, \alpha) = \mathbb{E}_{s\sim\rho_{\hat{\pi}^k}}\mathbb{E}_{a\sim\pi(\cdot|s)}Q^{\pi}(s,a) + \alpha\big[\epsilon - \mathbb{E}_{s\sim\rho_{\hat{\pi}^k}} D_{\mathrm{KL}}\big(\hat{\pi}^k(\cdot \mid s)\|\pi(\cdot \mid s)\big)\big]$, where α is a coefficient. To maximize $\mathcal{L}(\pi, \alpha)$, we can simplify the optimization of the second term into the following forms:

$$\begin{aligned} &\arg\min_{\pi} \mathbb{E}_{s\sim\rho_{\hat{\pi}^k}} D_{\mathrm{KL}}\big(\hat{\pi}^k(\cdot \mid s)\|\pi(\cdot \mid s)\big) \\ =&\arg\min_{\pi} \mathbb{E}_{s\sim\rho_{\hat{\pi}^k}, a\sim\hat{\pi}^k}\left[\log \hat{\pi}^k(a \mid s) - \log \pi(a \mid s)\right] \\ =&\arg\max_{\pi} \mathbb{E}_{s\sim\rho_{\hat{\pi}^k}, a\sim\hat{\pi}^k} \log \pi(a \mid s). \end{aligned} \tag{5}$$

Note this is the form of the Behavioral-Cloning (BC) objective [Pomerleau, 1991], which tries to mimic the likelihood under the state-action distribution of $\hat{\pi}^k$. We use the BC loss to minimize the proposed adaptive policy constraint.

Policy update through Eq. 4 is of on-policy fashion, which uses only the data collected by the last behavior policy $\hat{\pi}^k$ and can be inefficient. Therefore, in practice, we instead update the Q-function and enforce the constraint on all states in the replay buffer, leading to a TD3+BC[Fujimoto and Gu, 2021] update. The BC-regularized policy loss is given as:

$$\mathcal{L}_{\pi}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[-\lambda Q_{\mu}\big(s, \pi_{\theta}(s)\big) + \big(\pi_{\theta}(s) - a\big)^2\right], \tag{6}$$

where θ is the policy function parameter, $\lambda = \xi/\bar{Q}$, $\bar{Q} = \mathbb{E}_{(s,a)\sim\mathcal{D}}[|Q_{\mu}(s,a)|]$, and ξ is a hyper-parameter to balance the Q loss and the BC loss term. The loss function of the Q-value function Q_{μ} is given as:

$$\mathcal{L}_{Q}(\mu) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\big(y(r,s') - Q_{\mu}(s,a)\big)^2\right], \tag{7}$$

where $y(r,s') = r + Q_{\mu}\big(s', \pi_{\theta}(s)\big)$.

We present the practical implementation of CPR in Algorithm 1. The policy set Π starts with the pre-trained policy π_{β} and the mixed policy specified by Eq. 2 is equivalent to π_{β}. After collecting trajectories with π_{β}, we add a random policy π^1 to Π. Then CPR initializes π^1 via offline batch updates on the replay buffer to minimize Eq. 6. As Π contains more policies, CPR collects new samples using the mixed policy. At each step, we make a gradient update to the Q-value function Q_{μ}, and the current policy π^k. All other policies in Π are kept unchanged. Periodically, we add a new policy π^{k+1} to Π to maintain the continual learning capability.

Algorithm 1 CPR

Input: Pre-trained policy π_{β}, parameter μ of the pre-trained Q-value function, revitalization interval T_r

Initialize k with 0, parameters μ_1, μ_2 of twin Q-value functions with μ, the online replay buffer $\mathcal{D}$ with $\emptyset$, and the policy set Π with $\{\pi_{\beta}\}$
for each epoch **do**
  for each sampling step t **do**
    Select action with probabilities specified by Eq. 2
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r(s_t, a_t), s_{t+1})\}$
    if $k > 0$ **then**
      Update the Q-value function by minimizing Eq. 7 for one step of stochastic gradient descent
      Update the policy by minimizing Eq. 6 for one step of stochastic gradient descent
    end if
    if $t \,\%\, T_r == 0$ **then**
      Initialize the new policy π^{k+1} by minimizing Eq. 6 and add it to the policy set Π
      $k \leftarrow k + 1$
    end if
  end for
end for

5 Related Work

Existing O2O RL Methods. Existing O2O RL methods can be divided into policy constraint, pessimistic value estimation, and offline data replay. Explicit policy constraint slows online learning by constraining the policy towards the initial pre-trained policy [Nair *et al.*, 2020; Beeson and Montana, 2022; Zhao *et al.*, 2022]. It is especially detrimental when the pre-trained policy is poor. Conservative value function may hinder exploration and deteriorate learning efficiency [Kostrikov *et al.*, 2022; Nakamoto *et al.*, 2023; Zhang *et al.*, 2024]. The offline data replay requires access to the offline dataset which might not be accessible due to privacy and safety concerns [Lee *et al.*, 2021].

Policy Set. The idea of maintaining more than one policy function has been proposed for stabilizing the policy [Mnih *et al.*, 2016], exploration [Uchendu *et al.*, 2022], knowledge transfer [Lai *et al.*, 2020], and safety concern [Xu *et al.*, 2021]. In O2O RL, it was first proposed in PEX to construct a set containing two policies: the frozen policy π_{β} and the current learnable policy π_{θ} [Zhang *et al.*, 2023]. CPR inherits the idea of PEX and extends it. By periodically adding a new policy to the policy set, CPR restores the learning capability of the policy network while preserving historical knowledge.

Primacy Bias. Resetting sub-networks and the advantage of forgetting have been discussed in supervised learning

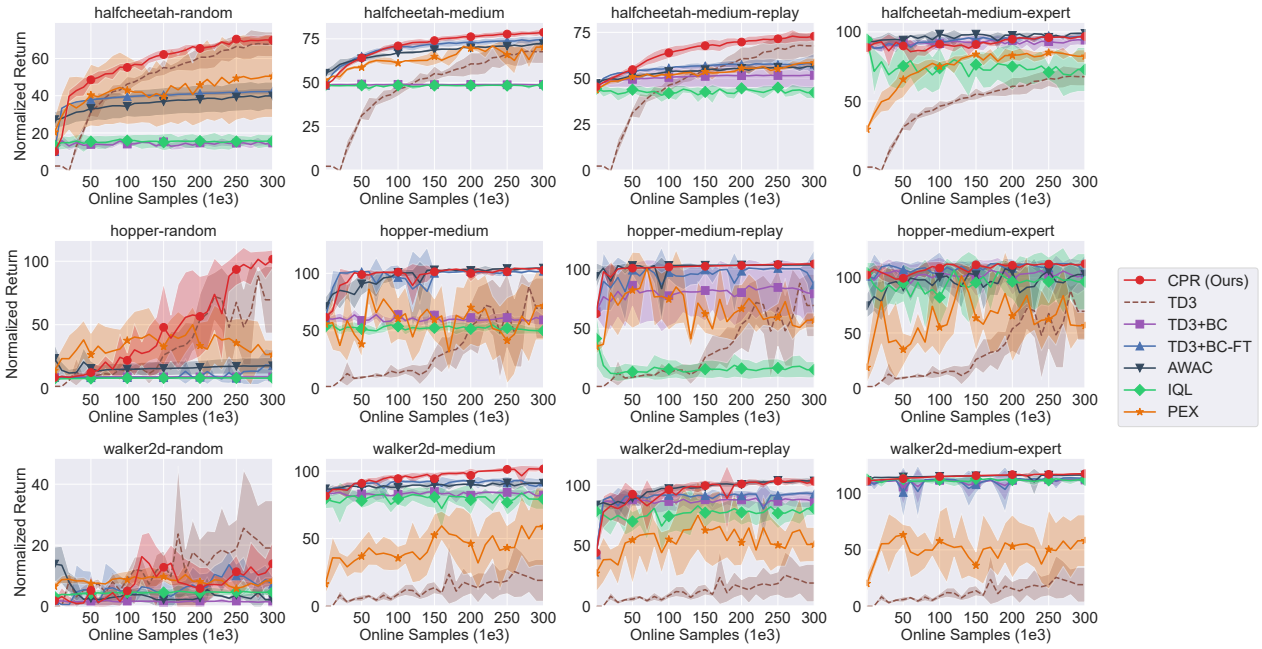

Figure 4: Normalized return learning curves on D4RL MuJoCo Locomotion benchmark.

[Zhou *et al.*, 2022]. Early experiments on the effect of example ordering report that early examples have greater influence in supervised learning [Erhan *et al.*, 2010]. The effect of primacy bias in deep RL is currently considered in the area of online RL [Nikishin *et al.*, 2022] and multi-task RL [Abbas *et al.*, 2023]. To our knowledge, CPR is the first algorithm to consider the primacy bias issue in O2O RL.

Reincarnating RL. Reincarnating RL (RRL) [Agarwal *et al.*, 2022] is proposed for reusing previous computation results without re-training from scratch in online learning. Different from RRL, CPR is designed for solving issues of direct policy initialization in O2O RL.

6 Evaluation

In this section, we conduct extensive experiments on MuJoCo to answer the following questions:

(i) How does CPR compare to other state-of-the-art methods for O2O RL? (See Section 6.2).

(ii) Can our continual policy revitalization method alleviate the primacy bias issue and prevent forgetting? (See Figure 4 and Section 6.3).

We present the ablation study of our method and extra experiments in Appendix C[1].

6.1 Experiment Settings

Our experiments are conducted with the following settings:

- **Task and offline datasets.** We select the popular MuJoCo locomotion tasks from D4RL [Fu *et al.*, 2020] as our benchmark for performance comparison. We select the random and three medium levels for each task to simulate offline datasets with different quality.
- **Offline training protocol.** All offline algorithms are trained for 1000 epochs of 1000 random mini-batches each. For each algorithm, we run 5 random seeds. The last checkpoint file is used as the pre-trained model.
- **Online training protocol.** We run all methods for 300 episodes with 5 random seeds. In each episode, there are 1000 online interaction steps.

We compare the following baselines:

- **TD3** [Fujimoto *et al.*, 2018], which represents the performance of learning from scratch with online RL.
- **TD3+BC** [Fujimoto and Gu, 2021], which adds a BC regularization term upon the policy update loss in TD3. As a pure offline RL algorithm, we initialize an offline buffer by the offline dataset.
- **TD3+BC-FT** [Beeson and Montana, 2022], which anneals the weight of BC term in TD3+BC for fine-tuning.
- **AWAC** [Nair *et al.*, 2020], which performs a KL divergence constraint in the policy improvement step.
- **IQL** [Kostrikov *et al.*, 2022], which uses expectile regression to learn the policy. IQL can be directly transferred to online fine-tuning without any modification.
- **PEX** [Zhang *et al.*, 2023], which freezes the pre-trained policy and adds a learnable policy for online learning.

The offline performance of all baselines is listed in Appendix C.1 for a fair comparison. In the experiments, CPR uses the same set of hyper-parameters on most tasks. We set revitalization interval $T_r = 10$ and revitalization fitting epochs $N_r = 32$. All hyper-parameter value selections are discussed in Appendix B in detail.

[1]https://www.lamda.nju.edu.cn/kongr/CPR/appendix.pdf

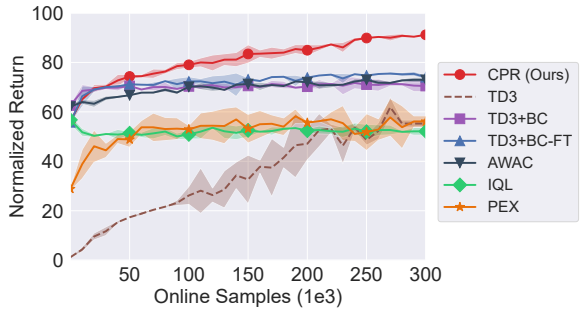

Figure 5: Aggregated learning curves of normalized return on all MuJoCo Locomotion tasks. See Figure 4 for more details.

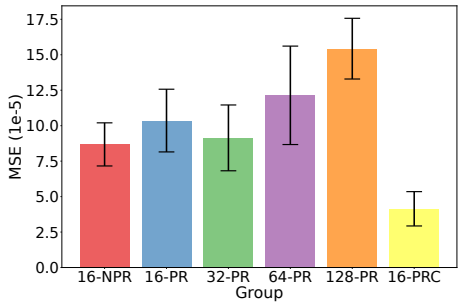

Figure 6: MSE loss of different methods on the testing set. The error bar in the plot represents the standard deviation over 50 seeds.

6.2 Main Results

We demonstrate the normalized score curves of all the tasks in Figure 4, and the aggregated normalized return across all tasks is illustrated in Figure 5. From the reported results, the dilemma of direct policy initialization is shown clearly. Because of the distribution shift issue, conservatism should be considered in the algorithm design to overcome performance drop in the early stage. However, over-conservatism severely hinders policy improvement. TD3+BC shows little improvement due to the fixed policy constraint to the offline dataset. With an unchanged hyper-parameter during the whole tuning process, IQL suffers from low sample efficiency compared to other algorithms. AWAC and TD3+BC-FT share the advantage of applying an implicit policy constraint over the online replay buffer and outperforming on medium tasks. Compared to AWAC, TD3+BC-FT updates the hyper-parameters with annealing. Though this could support more improvement on the top of the pre-trained policy, it also suffers from distribution shift when updating the actor too fast and causes policy improvement unstable. The adaptive policy constraint and policy design help CPR to achieve a stable policy improvement without a severe performance drop.

For the initial model pre-trained on the random level dataset, the primacy bias is observed and all the baseline algorithms fail to beat online algorithm learning from scratch. PEX uses a fresh learnable policy to learn from the online environment, it transfers knowledge from the offline model to the new policy by directly accessing the offline dataset. This could suppress the learning ability of the new policy and lead to suboptimal performance and high variance. In contrast, the policy revitalization procedure makes CPR achieve significant improvement even on tasks with low-quality datasets.

Our proposed method CPR outperforms all the baseline methods on most of the tasks and achieves the best overall performance, demonstrating its effectiveness.

6.3 Memory Test

We conduct an experiment to show the agent's memory capacity after revitalization. We split the set of samples $(s, \pi_\beta(s))$ collected by the pre-trained policy π_β to be a training set and a testing set. We keep the testing set not exposed to the agent and only utilize the training set to train a new policy π_0. We then sample states s from the testing set and let the agent predict actions $\pi_0(s)$ on those states. By assessing the L2 loss $\mathbb{E}_{s\sim\mathcal{D}_v}\left(\pi_\beta(s) - \pi_0(s)\right)^2$ on the validation set $\mathcal{D}_v$, we show that expanding a new policy could help in remembering. The Mean Squared Error (MSE) results are shown in Figure 6. The first number of each group means the number of training epochs. We use NPR to denote the no-policy-revitalization group and PR for the policy-revitalization group. PRC is the group where actions are selected from composition policy $\hat{\pi}$. As more rounds of fitting on the training set are performed, the loss of the testing set decreases at first and then increases which suggests over-fitting could happen when a high replay ratio is applied. As the control group, we also set an NPR group which simply fits the training set in the same way as other revitalized policies.

Compared with the control group, simply increasing fitting rounds only shows little improvement on memory and would cause overfitting if the replay ratio is too high. However, by introducing the policy composition mechanism CPR shows good performance on memorizing actions from behavior policy and outperforms even with limited fitting rounds.

7 Limitations and Discussion

The limitations of CPR include the extra running time and restricted policy model size. We list the analysis of running time results in Appendix C.4. The extra cost of computation is acceptable for most tasks. Contrary to O2O RL algorithms, our method does not introduce any specific constraints to the offline training portion to adapt different offline RL algorithms. This leads to a passive solution of primacy bias by resetting a part of the parameters to obtain a policy with full learning capacity. Appealing future directions are extending CPR to large-scale policy models and developing efficient utilization methods of pre-trained models.

8 Conclusion

This paper identifies the distribution shift and primacy bias issues and attributes these issues to direct policy initialization. Leveraging this analysis, we propose CPR to replace direct policy initialization in O2O RL. Our method periodically performs policy revitalization to reactivate the learning capacity and preserves good initial performance by adaptive policy constraint. Our experiment results show CPR's ability to improve learning efficiency and stability.

Acknowledgments

This work is supported by the National Science Foundation of China (62276126, 62076121, 62250069), and the Tencent AI Lab (RBFR2023011).

References

[Abbas *et al.*, 2023] Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C. Machado. Loss of plasticity in continual deep reinforcement learning. *arXiv preprint arXiv:2303.07507*, 2023.

[Agarwal *et al.*, 2022] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Reincarnating reinforcement learning: Reusing prior computation to accelerate progress. In *Advances in Neural Information Processing Systems*, pages 28955–28971, 2022.

[Beeson and Montana, 2022] Alex Beeson and Giovanni Montana. Improving TD3-BC: Relaxed policy constraint for offline learning and stable online fine-tuning. In *Offline RL Workshop: Offline RL as a "Launchpad"*, 2022.

[Brockman *et al.*, 2016] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym, 2016.

[Erhan *et al.*, 2010] Dumitru Erhan, Aaron C. Courville, Yoshua Bengio, and Pascal Vincent. Why does unsupervised pre-training help deep learning? In *International Conference on Artificial Intelligence and Statistics*, pages 201–208, 2010.

[Fu *et al.*, 2020] 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.

[Fujimoto and Gu, 2021] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 20132–20145, 2021.

[Fujimoto *et al.*, 2018] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1582–1591, 2018.

[Fujimoto *et al.*, 2019] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062, 2019.

[Gao and Rui, 2023] Chen-Xiao Gao and Kong Rui. OfflineRL-Lib: Benchmarked Implementations of Offline RL Algorithms, February 2023.

[Gao *et al.*, 2024] Chen-Xiao Gao, Chenyang Wu, Mingjun Cao, Rui Kong, Zongzhang Zhang, and Yang Yu. ACT: Empowering decision transformer with dynamic programming via advantage conditioning. In *AAAI Conference on Artificial Intelligence*, pages 12127–12135, 2024.

[Haarnoja *et al.*, 2018] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1856–1865, 2018.

[Harris *et al.*, 2020] Charles R. Harris, K. Jarrod Millman, Stéfan van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 2020.

[Kostrikov *et al.*, 2022] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022.

[Kumar *et al.*, 2019] 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*, pages 11761–11771, 2019.

[Lai *et al.*, 2020] Kwei-Herng Lai, Daochen Zha, Yuening Li, and Xia Hu. Dual policy distillation. In *International Joint Conference on Artificial Intelligence*, pages 3146–3152, 2020.

[Lee *et al.*, 2021] Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning*, pages 1702–1712, 2021.

[Levine *et al.*, 2020] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[Li *et al.*, 2023] Jianxiong Li, Xiao Hu, Haoran Xu, Jingjing Liu, Xianyuan Zhan, and Ya-Qin Zhang. PROTO: Iterative policy regularized offline-to-online reinforcement learning. *arXiv preprint arXiv:2305.15669*, 2023.

[Mnih *et al.*, 2016] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

[Nair *et al.*, 2020] Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. Accelerating online reinforcement learning with offline datasets. *arXiv preprint arxiv:2006.09359*, 2020.

[Nakamoto *et al.*, 2023] Mitsuhiko Nakamoto, Yuexiang Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-QL: Calibrated offline RL pre-training for efficient online fine-tuning. *arXiv preprint arXiv:2303.05479*, 2023.

[Nikishin *et al.*, 2022] Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron C. Courville. The primacy bias in deep reinforcement learning. In *International Conference on Machine Learning*, pages 16828–16847, 2022.

[Pomerleau, 1991] Dean A Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991.

[Ran *et al.*, 2023] Yuhang Ran, Yi-Chen Li, Fuxiang Zhang, Zongzhang Zhang, and Yang Yu. Policy regularization with dataset constraint for offline reinforcement learning. In *International Conference on Machine Learning*, pages 28701–28717, 2023.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Tarasov *et al.*, 2022] Denis Tarasov, Alexander Nikulin, Dmitry Akimov, Vladislav Kurenkov, and Sergey Kolesnikov. CORL: Research-oriented deep offline reinforcement learning library. In *Offline RL Workshop: Offline RL as a "Launchpad"*, 2022.

[Todorov *et al.*, 2012] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[Uchendu *et al.*, 2022] Ikechukwu Uchendu, Ted Xiao, Yao Lu, Banghua Zhu, Mengyuan Yan, Joséphine Simon, Matthew Bennice, Chuyuan Fu, Cong Ma, Jiantao Jiao, Sergey Levine, and Karol Hausman. Jump-start reinforcement learning. *arXiv preprint arXiv:2204.02372*, 2022.

[Wu and Zhang, 2023] Chenyang Wu and Zongzhang Zhang. Surfing information: The challenge of intelligent decision-making. *Intelligent Computing*, 2:0041, 2023.

[Xu *et al.*, 2021] Tengyu Xu, Yingbin Liang, and Guanghui Lan. CRPO: A new approach for safe reinforcement learning with convergence guarantee. In *International Conference on Machine Learning*, pages 11480–11491, 2021.

[Ye *et al.*, 2021] Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering Atari games with limited data. In *Advances in Neural Information Processing Systems*, pages 25476–25488, 2021.

[Yu and Zhang, 2023] Zishun Yu and Xinhua Zhang. Actor-critic alignment for offline-to-online reinforcement learning. In *International Conference on Machine Learning*, pages 40452–40474, 2023.

[Yu, 2018] Yang Yu. Towards sample efficient reinforcement learning. In *International Joint Conference on Artificial Intelligence*, pages 5739–5743, 2018.

[Zhang *et al.*, 2023] Haichao Zhang, Wei Xu, and Haonan Yu. Policy expansion for bridging offline-to-online reinforcement learning. In *International Conference on Learning Representations*, 2023.

[Zhang *et al.*, 2024] Yinmin Zhang, Jie Liu, Chuming Li, Yazhe Niu, Yaodong Yang, Yu Liu, and Wanli Ouyang. A perspective of Q-value estimation on offline-to-online reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pages 16908–16916, 2024.

[Zhao *et al.*, 2022] Yi Zhao, Rinu Boney, Alexander Ilin, Juho Kannala, and Joni Pajarinen. Adaptive behavior cloning regularization for stable offline-to-online reinforcement learning. In *European Symposium on Artificial Neural Networks*, 2022.

[Zhou *et al.*, 2022] Hattie Zhou, Ankit Vani, Hugo Larochelle, and Aaron C. Courville. Fortuitous forgetting in connectionist networks. In *International Conference on Learning Representations*, 2022.

[Zhou *et al.*, 2024] Renzhe Zhou, Chen-Xiao Gao, Zongzhang Zhang, and Yang Yu. Generalizable task representation learning for offline meta-reinforcement learning with data limitations. In *AAAI Conference on Artificial Intelligence*, pages 17132–17140, 2024.