
Offline Reinforcement Learning with Imbalanced Datasets

Li Jiang¹ Sijie Cheng^{2,3} Jieliu Qiu⁴ Haoran Xu³ WaiKin Chan¹ Ding Zhao⁴

Abstract

The prevalent use of benchmarks in current offline reinforcement learning (RL) research has led to a neglect of the imbalance of real-world dataset distributions in the development of models. The real-world offline RL dataset is often imbalanced over the state space due to the challenge of exploration or safety considerations. In this paper, we specify properties of imbalanced datasets in offline RL, where the state coverage follows a power law distribution characterized by skewed policies. Theoretically and empirically, we show that typically offline RL methods based on distributional constraints, such as conservative Q-learning (CQL), are ineffective in extracting policies under the imbalanced dataset. Inspired by natural intelligence, we propose a novel offline RL method that utilizes the augmentation of CQL with a retrieval process to recall past related experiences, effectively alleviating the challenges posed by imbalanced datasets. We evaluate our method on several tasks in the context of imbalanced datasets with varying levels of imbalance, utilizing the variant of D4RL. Empirical results demonstrate the superiority of our method over other baselines.

1. Introduction

Offline reinforcement learning (RL), known as batch RL, (Lange et al., 2012; Levine et al., 2020) holds great promise in learning high-quality policies from previously logged datasets, without further interaction with the environment and collect trajectories that may be dangerous or expensive (Tang & Wiens, 2021; Kalashnikov et al., 2021; Zhan et al., 2022). Such promise makes real-world RL more realistic

and enables better generalization abilities by incorporating diverse previous experiences (Levine et al., 2020).

To mitigate the fundamental challenge offline RL, i.e., distributional shift (Levine et al., 2020), most current studies generally enforce pessimism on policy updates (Fujimoto et al., 2019; Wu et al., 2019; Kumar et al., 2019a; Fujimoto & Gu, 2021), or value updates (Kumar et al., 2020; An et al., 2021; Bai et al., 2022). The pessimism explicitly or implicitly constrains the learned policy to the behavior policy and has been shown to be effective in standard benchmarks in (Fu et al., 2020; Gulcehre et al., 2020; Qin et al., 2021). However, it should be noted that while real-world distributions are rarely uniform, the datasets used in these benchmarks often feature near-uniform state coverage and near-balanced policies. While the imbalanced dataset has been widely studied in supervised learning (Buda et al., 2018; Liu et al., 2019), it remains under-explored in the offline RL community even though the static dataset is the only source to extract policies. Real-world datasets in offline RL are often imbalanced mainly due to the challenge of exploration or safety consideration, consisting of skewed policies.

Real-world datasets in offline RL exhibit imbalanced and follow Zipf’s law (Gabaix, 1999; Newman, 2005) across the entire state space, with skewed policies (see Section 3). The states in the dataset from sufficient to insufficient coverage are characterized by *varying policies from mixture to expert ones*. As an example, for safety considerations in industrial control systems, certain adjustments to key parameters may be rare but near-optimal, while other trivial adjustments may be more common but inferior.

Our study shows that most of the existing offline RL algorithms perform poorly when faced with imbalanced datasets. We then provide a provable explanation for this failure, which unfolds in two dimensions. Firstly, *state-agnostic pessimism* in most current approaches fail to guarantee policy improvement. Secondly, *uniform transition sampling* during the course of training may lead to large temporal-difference (TD) errors in off-policy evaluation for states with poor coverage due to inefficient sampling. Ironically, our empirical investigation exposes that re-sampling methods like Prioritized Experience Replay (PER, as outlined in (Schaul et al., 2015)) may lead to worse performance in

¹Shenzhen International Graduate School, Tsinghua University ²Department of Computer Science and Technology, Tsinghua University ³Institute for AI Industry Research, Tsinghua University ⁴Carnegie Mellon University. Correspondence to: Li Jiang <jiangli3859@gmail.com>.

imbalanced offline RL datasets, serving as an amplifier for the distributional shift problem.

Motivated by natural intelligence, which avoids forgetting rare phenomena by recalling related information from past experiences (McClelland et al., 1995; Leake, 1996), we introduce a new offline RL algorithm to overcome those limitations by augmenting standard offline RL algorithms, e.g., CQL (Kumar et al., 2020), with a *retrieval process*. Our proposed method, retrieval-based CQL (RB-CQL), enables agents to effectively utilize related experiences through nearest-neighbor matching from diverse and large-scale datasets, and to directly inform their actions, particularly for states with poor coverage. Our algorithm is evaluated on a series of tasks with imbalanced datasets in the variant of D4RL (Fu et al., 2020) and shows the competition compared with state-of-the-art offline RL methods. Note that RB-CQL represents a straightforward approach to alleviate offline RL in the context of real-world imbalanced datasets. We hope this work will inspire future research on real-world RL systems utilizing practical datasets, and serve as a foundation for large-scale offline RL.

2. Preliminaries

RL problem is typically characterized by a Markov Decision Process (MDP) (Sutton et al., 1998), which is specified by a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, r, \rho, \gamma \rangle$ consisting of a state space \mathcal{S} , an action space \mathcal{A} , a transition probability function $P(s'|s, a)$, a reward function $r(s, a) \in \mathbb{R}$, an initial state distribution ρ , and the discount factor $\gamma \in [0, 1)$. The object of RL is to extract a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that maximizes the expectation of the sum of discounted reward, known as the return $J(\pi) = \mathbb{E}_\pi [\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$.

Based on approximate dynamic programming, off-policy RL methods typically learn a state-action value function (Q-function), characterized as $Q^\pi(s, a) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ with policy π , the discounted return when the trajectory starts with (s, a) and all remaining actions are taken via π . For a given policy π , the Q-function can be obtained through the Bellman operator $\mathcal{T}^\pi : \mathbb{R}^{\mathcal{S} \times \mathcal{A}} \rightarrow \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$, defined as :

$$(\mathcal{T}^\pi Q)(s, a) := r(s, a) + \gamma \mathbb{E}_{\substack{s' \sim P(\cdot|s, a) \\ a' \sim \pi(\cdot|s')}} [Q(s', a')],$$

where $Q^\pi(s, a)$ is a unique fixed point with the contracted Bellman operator for $\gamma \in [0, 1)$ (Bertsekas & Tsitsiklis, 1995). The optimal Q-function is obtained via $Q^* = \max_\pi Q^\pi(s, a)$, which aligns with an optimal policy achieved through greedy action choices. Another important concept is the notion of discounted state occupancy, $d_\pi \in \Delta(\mathcal{S})$, defined as $d_\pi(s) := (1 - \gamma) \mathbb{E}_\pi [\sum_{t=0}^{\infty} \gamma^t \mathbb{1}[s_t = s]]$, which characterizes a distribution over the states visited by the policy.

We narrow our focus to offline RL settings, which aim to learn an optimal policy from a pre-collected dataset \mathcal{D} , generated by some unknown behavior policy β . The most critical problem in offline RL is that the training policy π may generate out-of-distribution action a' that causes erroneous estimation error of $Q(s', a')$ and propagate through the Bellman update, often leading to catastrophic failure to the policy learning. Most recent offline RL methods alleviate the above issue by introducing pessimism either on policy learning (Fujimoto et al., 2019; Wu et al., 2019; Kumar et al., 2019a) or value learning (Kostrikov et al., 2021; Kumar et al., 2020; Wu et al., 2021). Most of those methods can be formulated into a generic formulation of distribution-constrained offline RL algorithms (Kumar et al., 2020), either explicitly or implicitly. The goal of it is to maximize the return of the learned behavior policy π while minimizing the divergence $D(\pi, \beta)$ weighted by a hyperparameter α :

$$\max_{\pi} \mathbb{E}_{s \sim d^\pi} [J(\pi) - \alpha D(\pi, \beta)(s)]. \quad (1)$$

Conservative Q-learning. (Kumar et al., 2020) enforces pessimism on the value update, based on the actor-critic framework, which can be written as:

$$\min_{\theta} \alpha (\mathbb{E}_{s \sim \mathcal{D}, a \sim \rho} [Q_\theta(s, a)] - \mathbb{E}_{s \sim \mathcal{D}, a \sim \beta} [Q_\theta(s, a)]) + \frac{1}{2} \mathbb{E}_{(s, a, s') \sim \mathcal{D}} [(Q_\theta(s, a) - \mathcal{T}^\pi \bar{Q}_{\theta'}(s, a))^2].$$

The second term is the standard Bellman updates error (Mnih et al., 2015; Lillicrap et al., 2015; Fujimoto et al., 2019) and \bar{Q}_θ represents the target Q-value with parameters θ' , which is updated to the current network with parameters θ by averaging $\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$ for some small τ . Instead of giving pessimism on policy learning by choosing in-distribution actions, CQL enforces pessimism on value learning (the first term) by minimizing the Q-values under a specific policy distribution ρ , which is characterized by the high Q-values, and balancing by maximizing the Q-values of from the behavior policy β . Note that CQL implicitly constrains the learned policy π over the behavior policy β (see Eq. (13) in (Kumar et al., 2020)).

The hyperparameter α controls the degree of pessimism. A high value of α corresponds to a high level of pessimism, indicating that the optimal policy is highly similar to the behavior policy β , while a low value of α corresponds to a low level of pessimism. The pessimism level, i.e., α , is a constant for every state-action pair, and thus the value function learned by CQL is changed to the same extent at all possible state-action pairs. However, the consistent divergence between the learned policy and the behavior policy over all possible state-action pairs may ultimately lead to a decline in the performance of the learned policy and

even result in failure in imbalanced datasets. We illustrate these challenges by providing a motivating example and practical experiences in the next session.

3. Current Offline RL Methods fail in the Imbalanced Dataset

An imbalanced dataset over the state space in offline RL can be described as the state visitation of the behavior policy $d_\beta(s)$ in the dataset imbalanced. The state visitation of the behavior policy $d_\beta(s)$ follows a power law distribution (Gabaix, 1999; Newman, 2005) with exponent $\eta \in [0, \infty]$. The probability distribution for a random variable X is:

$$p(X = x) = \frac{1}{Z} \cdot \frac{1}{x^\eta}, \quad (2)$$

where Z is a normalizing constant term and η determines the degree of imbalance in the imbalanced offline RL dataset. A higher value of η exacerbates the imbalance in the state space and renders learning from rare experiences more challenging. Furthermore, rare experiences are of importance as they are likely, or in many cases, derived from expert policies. The significant decline in performance observed in current offline RL algorithms can mainly be attributed to two factors:

State-agnostic. Most current offline RL algorithms are state-agnostic and enforce consistent pessimism for all state-action pairs in the datasets. In the context of an imbalanced dataset, it is crucial for the learned policy to remain similar to the behavior policy, β , in states with insufficient coverage $d_\beta^-(s)$, as these states typically consist of expert demonstrations. Conversely, in states with sufficient coverage, it is necessary for the learned policy to deviate from β in order to stitch potential useful trajectories, as these states often consist of a mixture of demonstrations.

Uniformly sampling. In deep Q-learning offline RL methods, transitions are uniformly sampled from the dataset, contributing to Bellman updates errors with respect to the probability of transitions in the dataset:

$$\approx \mathbb{E}_{(s,a,s') \sim \mathcal{D}} \left[\left(Q_\theta(s,a) - \mathcal{T}^\pi \bar{Q}_{\theta'}(s,a) \right)^2 \right]. \quad (3)$$

During the course of training, those suboptimal transitions from $d_\beta^+(s)$ dominate over those near-optimal transitions from $d_\beta^-(s)$. It will cause inefficient sampling to near-optimal but rare transitions from $d_\beta^-(s)$, which may lead to inferior estimation over Q-functions, especially for tasks with sparse reward requiring enormous exploration.

In the following sections, we introduce a navigation task and a practical experience on Antmaze (Fu et al., 2020) with a varying imbalance to demonstrate how the current

state-of-the-art offline RL algorithm, i.e. CQL, fails in imbalanced datasets. We then theoretically reveal the failure of current offline RL algorithms under imbalanced datasets. Experiments details can be found in Appendix C.

3.1. State-Agnostic Pessimism

We first use a didactic `Four-room` navigation task in Fig. 1 to present the failure of CQL with state-agnostic pessimism. This task exemplifies the challenge of exploration with sparse reward and long horizon. The goal of this task is to find a path from the initial state (yellow grid in the first room) to the target state (red grid in the last room) in the grid space, where each room is only connected by one grid with its adjacent rooms. The state and action space are discrete, where the state presents the current location of the agent via the (x, y) coordinate and the action space is 4 corresponding to four orthogonal nearby grids. The agent will receive 10 rewards for reaching the goal state and 0 for all other actions at any feasible state.

The dataset is generated by a goal-reaching controller from the start to the goal state and the agent executes the correct action with increasing probabilities from 10 % probability to 100 %. The `Four-room` dataset is characterized by an imbalanced distribution of state coverage across the entire state space. Specifically, there is a high degree of coverage in the first room, which is accompanied by suboptimal actions, whereas there is insufficient coverage in the last room, where near-optimal policies are present. The color bar is limited to 30 to better highlight the differences among rare experiences in Fig. 1 (left). It should be noted that the samples of some states (e.g., those in the first room) reaching the limit are significantly larger than 30 with a high likelihood. We illustrate the action choices of two policies trained with CQL, one with low pessimism ($\alpha = 5$) and the other with high pessimism ($\alpha = 20$), over the entire feasible state space in Fig. 1 (center) and (right), respectively. One feasible state without actions represents that Q-functions are the same in the whole action space, which can be attributed to the missing data in the original dataset. The grid color (more saturated to yellow means a higher value) denotes the value function, which is obtained by taking the expectation of all possible actions.

Given a low level of pessimism (i.e., $\alpha = 5$), shown in Fig. 1 (center), the learned policy is entitled to deviate heavily from the behavior policy, encouraging the agent to stitch suboptimal trajectories and hence successfully get out of the first room. However, low pessimism leads to a poorer estimation of Q-functions in states with insufficient coverage and the failure of the last room. This can be attributed to the high distribution mismatch induced by low pessimism, where insufficient coverage serves as an amplifier to the inaccurate estimation of Q-functions. On the other hand, in

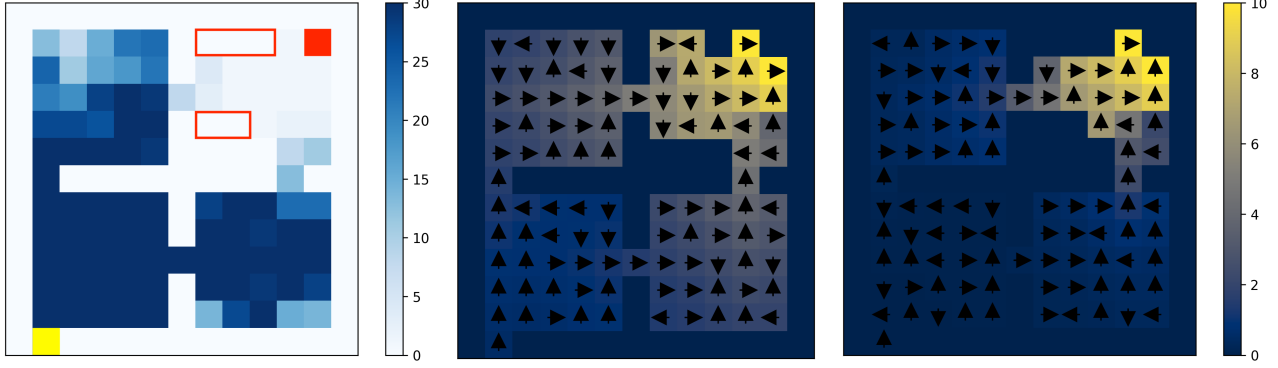


Figure 1. Left: The state coverage and task description of the imbalanced dataset in `Four-room`, where states coverage is featured with the heavy-tail property and states in red rectangle frame have no state coverage. It requires the agent to find a path from the start state (yellow grid) in the first room to the goal state (red grid) in the last room. **Center:** Policy performance with lower pessimism. The learned policy succeeds in the first room but fails in the last room. **Right:** Policy performance with higher pessimism. The learned policy succeeds in the last room but fails in the first room.

order to succeed in the last room, high pessimism is required to significantly constrain the learned policy to the behavior policy. However, the learned policy fails in the first room (suboptimal trajectories) because of the high similarity to the behavior policy, as shown in Fig. 1 (right).

In order to finish this task, the policy should be imposed state-specific pessimism. In detail, imposing low pessimism for states with sufficient coverage (e.g., states from the first room) and high pessimism for states with insufficient coverage (e.g., states from the last room). Note that lower pessimism is highly risky to out-of-distribution states problem at the test time, resulting in incorrect Q-functions over those states because of the missing in the training dataset.

3.2. Uniformly Sampling

To illustrate the failure of current offline RL algorithms, we then train CQL on the imbalanced dataset in AntMaze medium, which is a more sophisticated task with continuous space over states and actions. Similar to `Four-room` dataset, the expert trajectories reaching the target location in AntmMaze datasets are rare and the state coverage is imbalanced.

The imbalance in the dataset is increased with higher η and the performance significantly decreases (shown in Fig. 2 (left)). Except for the state-agnostic training process, we hypothesize that uniformly sampling also worsens the performance because of inefficient training on rare expert experiences from $d_{\beta}^{-}(s)$, failing to properly estimation of Q-functions. To validate our hypothesis, we track 500 state-action pairs near the target location (i.e., $d_{\beta}^{-}(s)$) and the start location (i.e., $d_{\beta}^{+}(s)$), respectively. In Fig. 2 (center), we graph the mean of TD errors over and find that TD errors of states from $d_{\beta}^{-}(s)$ are larger than states from $d_{\beta}^{+}(s)$ around

5. Can we draw techniques from common approaches to the long-tail distribution problem in supervised learning?

Prioritized Experience Replay (PER, as proposed in Schaul et al. (2015)) is a method that modifies the sampling probability of transitions based on their temporal difference (TD) errors. Specifically, it over-samples transitions with high TD errors, potentially adjusting the non-uniformity of the dataset. However, as demonstrated in Fig. 2 (right), re-weighting the loss in Eqn. (3) based on TD errors can result in a decrease in overall performance. We posit that the re-sampling process alters the distribution of $d_{\beta}(s)$, which dynamically shifts the behavior policy β with respect to TD errors, and results in an additional divergence between the true β and the modified β . As a result, the extra distributional shift problem is introduced to the whole training process and yields further incorrect estimations over Q-functions. On the other hand, those techniques are useful in online one-step Q-learning RL methods because the learning policy can collect new experiences and thus avoids the distributional shift problem.

3.3. Theoretical Analysis

So far we have empirically seen the continuous performance drop with the increasing imbalance of the imbalanced dataset, now we formally characterize why current pessimism without the consideration of dataset coverage fails to guarantee policy improvement in the imbalanced dataset. In contrast to previous common approaches analysis via concentrability coefficient (Liu et al., 2020; Rashidinejad et al., 2021), upper bounding the ratio of state-action visitation a specific policy $d_{\pi}(s, a)$ and the behavior policy from the dataset $d_{\beta}(s)$, i.e., $\max_{s,a} d_{\pi}(s, a)/d_{\beta}(s) \leq C^{\pi}$, we use a different metric from the variant of differential concentrability (Singh et al., 2022), aiming to measure the imbalance of

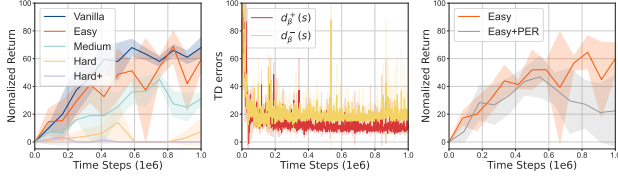


Figure 2. Left: The performance of CQL on Antmaze-medium task with imbalanced dataset. With the increasing imbalance of the given dataset from `Easy` to `Hard+` (increasing α), the performance continues to drop, and even fails in the hard+ dataset. **Center:** TD errors over 500 states from sufficient coverage $d_\beta^+(s)$ and insufficient coverage $d_\beta^-(s)$ on medium-level imbalance, respectively. TD errors from sufficient coverage $d_\beta^+(s)$ are smaller than from insufficient coverage $d_\beta^-(s)$ due to inefficient training. **Right:** The performance of CQL and CQL with PER on Antmaze-medium task on easy-level imbalance. PER worsens the final performance as the introduction of additional data distributional shift problem.

dataset coverage (i.e., the discrepancy of state visitation frequency between adequate and inadequate dataset coverage over the whole state space).

Definition 3.1 (Differential concentrability.). *Given a divergence D over the action space, the differential concentrability C_{diff}^π of a given policy π with respect to the behavior policy β is given by:*

$$\mathbb{E}_{\substack{s_1 \sim d_\beta(s)^+ \\ s_2 \sim d_\beta(s)^-}} \left[\left(\sqrt{\frac{D(\pi, \beta)(s_1)}{d_\beta(s_1)}} - \sqrt{\frac{D(\pi, \beta)(s_2)}{d_\beta(s_2)}} \right)^2 \right] \quad (4)$$

Eqn. (4) measures of imbalance in the dataset in the current state-agnostic methods, is characterized by the discrepancy between a given policy $\pi(a|s)$ and the behavior policy $\beta(a|s)$, weighted negatively by the density of states in the denominator, from states with sufficient and insufficient coverage (i.e., $d_\beta^+(s)$ and $d_\beta^-(s)$), respectively. Instead of considering a simpler scenario where $d_\beta(s) = \text{Unif}(s)$ (i.e., $d_\beta(s_1) = d_\beta(s_2)$) (Singh et al., 2022), we consider a more realistic scenario where $d_\beta(s_1) > d_\beta(s_2)$, and the imbalance of the dataset is exacerbated by increasing the value of η in Eqn. (2). This allows us to examine the effect of imbalanced datasets on the performance of our method.

Under the imbalanced dataset, if the policy divergence D with respect to any given policy π is state-agnostic, whether loose or tight constraints, C_{diff}^π would be large because the denominators are significantly disagreeable (i.e., $d_\beta(s_1)$ is large than $d_\beta(s_2)$). However, on the other hand, if we allow \mathcal{D} to be state-specific with respect to the state coverage (i.e., loose constraints to states from $d_\beta^+(s)$ and tight to states from $d_\beta^-(s)$), such that $D(\pi, \beta)(s_1)$ is large and $D(\pi, \beta)(s_2)$ is small. C_{diff}^π would be relatively low, as

the numerator, the policy divergence $D(\pi, \beta)(s)$, increases monotonically in relation to the denominator, the coverage of the dataset $d_\beta(s)$. Small C_{diff}^π indicates that the policy π stays close to the behavior policy β in $d_\beta^-(s)$ while deviates significantly in $d_\beta^+(s)$, consisting with our didactic example and conclusion in Section 3.1.

Given the definition of differential concentrability under the imbalanced state coverage, we follow Singh et al. (2022) and use it to bound the policy improvement over π w.r.t. β in the safe policy improvement framework (Laroche et al., 2019). We then show that large C_{diff}^π is hard to guarantee the policy improvement with a large margin, under the distributional constraints methods in Eqn. (1):

Theorem 3.2 (Limited policy improvement via distributional constraints.). *W.h.p. $\geq 1 - \delta$, for any prescribed level of safety ζ , the maximum possible policy improvement over choices of α , $J(\pi) - J(\beta) \leq \zeta^+$, where ζ^+ is given by:*

$$\zeta^+ := \max_{\alpha} h^*(\alpha) \cdot \frac{1}{(1 - \gamma)^2} \quad (5)$$

$$\text{s.t. } \frac{c_1}{(1 - \gamma)^2} \frac{\sqrt{C_{\text{diff}}^\pi}}{|\mathcal{D}|} - \frac{\alpha}{1 - \gamma} \mathbb{E}_{s \sim d_\pi(s)} [D(\pi, \beta)(s)] \leq \zeta$$

where h^* is a monotonically decreasing function of α , and $h(0) = \mathcal{O}(1)$.

Theorem 3.2 expresses the essential trade-off between distributional constraints and safe policy improvement. The LHS of the constraint in Eqn. (5) is a monotonically increasing function of C_{diff}^π . When the value of C_{diff}^π is low, a relatively small α can lead to a significant improvement in performance, as measured by $h^*(\alpha)$ in Eqn. (5), while also ensuring compliance with the ζ -safety condition. However, when the dataset is exposed to imbalance (i.e., C_{diff}^π is large) in the current state-agnostic constraints framework, satisfying the ζ -safety condition requires a larger α , which obtains smaller maximum possible improvement $h^*(\alpha)$. When the degree of imbalance η in Eqn. (2) approaches extremes, the improvement $h^*(\alpha)$ over the learned policy π is close to zero with high likelihood, i.e., the learned policy fails to improve over the behavior policy β . We remark that smaller C_{diff}^π is essential for the ζ -safety condition and hence a smaller α for the improvement $h^*(\alpha)$ with a large margin. The proofs of Theorem 3.2 can be found in (Singh et al., 2022).

4. Retrieval Augmented Offline RL

As discussed in Section 3 with empirical and theoretical analyses, current state-of-the-art offline RL algorithms can hardly solve imbalanced offline RL datasets. In Natural Language Processing (NLP) and computer vision (CV) communities, there is a line of works (Kandpal et al., 2022; Long

et al., 2022; Wu et al., 2008) that leverages external information or knowledge to alleviate the heavy-tail or long-tail problems. Actually, our designed imbalanced dataset in offline RL is similar to these studies, which are following Zipf’s law. As a preliminary study, we try to introduce non-parametric retrieval-augmented methods to enhance standard offline RL algorithms by combining retrieved relevant states from external resources.

Auxiliary dataset preparation. Firstly, we prepare external resources as our auxiliary dataset \mathcal{D}_{aux} for the retrieval process. The auxiliary dataset contains a large body of knowledge or experience, for example, trajectories from previous experiences of the current learned policy or other agents (Goyal et al., 2022; Humphreys et al., 2022). In our setups, we directly adopt the state information in the same and relevant environments as auxiliary dataset \mathcal{D}_{aux} .

Retrieval process. Secondly, we retrieve the nearest states with useful information from auxiliary dataset \mathcal{D}_{aux} . Formally, given the query state $s_{\text{ori}}^i \in \mathcal{D}$ and the external state $s_{\text{aux}}^i \in \mathcal{D}_{\text{aux}}$ from the auxiliary dataset, we encode all these states to representations as static indexes. For high dimensional states, the representations of both query state and external states can be denoted as $\text{Enc}(s_{\text{ori}}^i)$ and $\text{Enc}(s_{\text{aux}}^i)$. To mitigate additional computation burden and eliminate effects from other architecture, we evaluate our method on Mujoco (Todorov et al., 2012) tasks with low dimensional states. Then, after building the indexes, we compute the similarity score between these indexes based on Maximum Inner Product Search (MIPS) or other distance functions, such as Euclidean distance. Taking the dense inner product as an example, the detailed similarity score can be computed as follows:

$$\text{Sim}(s_{\text{ori}}^i, s_{\text{aux}}^i) = \frac{\exp(s_{\text{ori}}^i \cdot s_{\text{aux}}^i)}{\sum_{j=1}^{|\mathcal{D}_{\text{aux}}|} \exp(s_{\text{ori}}^i \cdot s_{\text{aux}}^j)} \quad (6)$$

According to the similarity scores $\text{Sim}(s_{\text{ori}}^i, s_{\text{aux}}^i)$ between the query state s_{ori}^i and external states s_{aux}^i , our retrieval process selects the top- k nearest retrieved states $\{s_{\text{ret}}^1, s_{\text{ret}}^2, \dots, s_{\text{ret}}^k\}$ to augment our agent, which will further introduce in the next part. Following Humphreys et al. (2022), we use $k = 10$ and adopt the SCaNN architecture (Guo et al., 2020) as the implementation to efficiently complete the approximate vector similarity search.

Retrieval-augmented agent. Finally, we hope to leverage these retrieved states $\{s_{\text{ret}}^1, s_{\text{ret}}^2, \dots, s_{\text{ret}}^k\}$ for our agent to alleviate the heavy-tail problem reflected in our exploration-challenging dataset. With query state s_{ori}^i , our agent further exploits the retrieved states by a straightforward and non-parametric approach. Considering that directly concatenating all retrieved states to the query state will lose the main information of the query state, we concatenate the original state s_{ori}^i with averaged retrieved states to produce the final

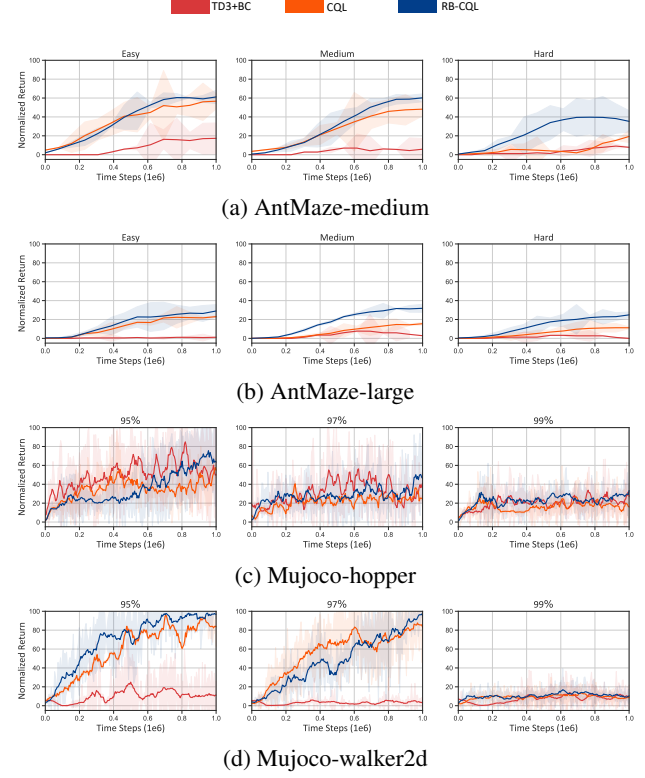


Figure 3. Average normalized scores of RB-CQL against other baselines over the whole training process. Regarding the final checkpoint, RB-CQL reaches the best performance in 10 out of 12 tasks, where an increasing margin with the increasing imbalance in AntMaze.

state s_{final} :

$$s_{\text{final}} = \left[s_{\text{ori}}^i \oplus \frac{\sum_{j=1}^k s_{\text{ret}}^j}{N} \right] \quad (7)$$

Where \oplus is the concatenate operator. We then feed the final state s_{final} to the policy network and value network, then further adopt the popular value-based offline RL method, e.g., CQL, to train and evaluate. It is worth noting that the only difference between the offline RL method and our retrieval-augmented offline RL method is that we introduce additional state information via s_{final} instead of the only state from the current transition s_{ori}^i . We summarize the pseudo-code of RB-CQL in Appendix A.

5. Experiments

Imbalanced datasets. In order to evaluate the effectiveness of Retrieval-based CQL (RB-CQL), we first establish four new tasks from two tasks with imbalanced datasets in the variant of the standard benchmark D4RL (Fu et al., 2020). Recall in Section 3, the distribution of imbalanced datasets follows Zipf’s law, and the magnitude of its coverage is negatively related to the degree of policies (i.e., sufficient to insufficient coverage is characterized by mixture and ex-

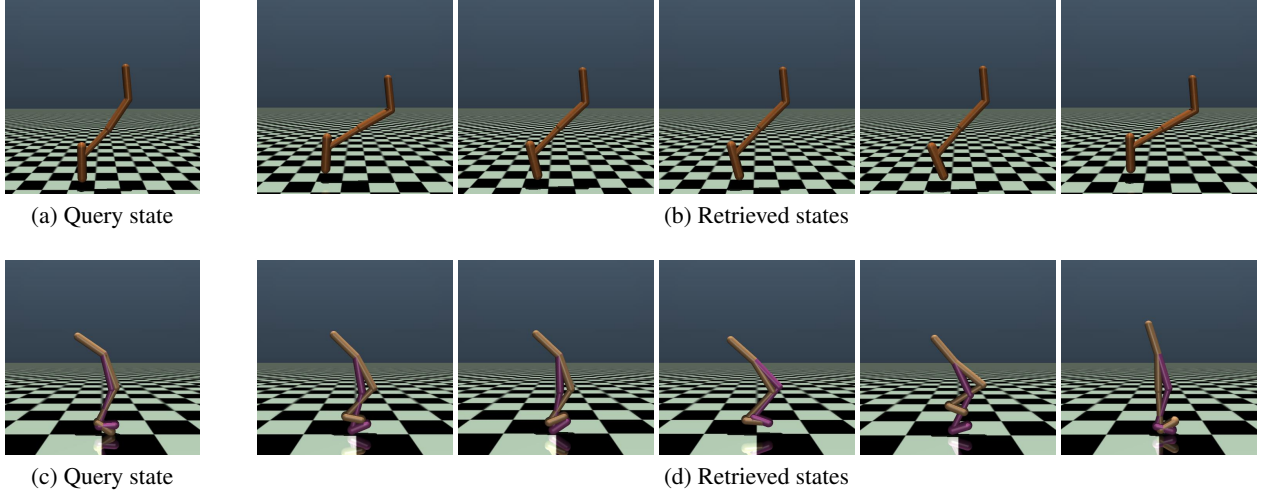


Figure 4. Visualisation of $N=5$ retrieved from RB-CQL for Mujoco locomotion tasks (first row for Hopper and second row for Walker2d.)

pert policies). For AntMaze tasks, we use a goal-reaching controller with a fixed start distribution (same as the evaluation process) and a goal distribution varying in the whole state space, where only rare goal distributions are identical to the evaluation process. It represents that the dataset is marked by imbalanced state coverage, where rare expert trajectories could reach the goal state in terms of evaluation. For Mujoco tasks, we simply combine the random and expert dataset within a few expert datasets. The increasing η in Eqn. (2) denotes the increasing imbalance over those datasets: for AntMaze navigation tasks in medium and large mazes, smaller near-optimal trajectories are corresponding from *Easy* to *Hard+* dataset; for Mujoco locomotion tasks, smaller near-optimal trajectories are corresponding to random dataset ratio from 95% to 99%. Note that AntMaze tasks are more challenging than Mujoco locomotion tasks due to the sparse reward setting and the heavy requirement for exploration.

We then compare RB-CQL with TD3+BC (Fujimoto & Gu, 2021) and CQL (Kumar et al., 2020), which are two state-of-the-art methods and enforce policy support constraints. For a fair comparison, we sweep important hyperparameters of each algorithm. The experimental results for these tasks are reported in Fig. 3. The shaded area in the plots represents the standard deviation of the results. Detail implementation and experimental details are provided in Appendix C and the performance profiles (Agarwal et al., 2021) based on score distribution are shown in Fig. 7.

For each AntMaze task with increasing imbalance (i.e., *Easy* to *Hard*), our approach, RB-CQL, is robust to the imbalance and maintains its performance to a certain degree, especially from *Easy* to *Medium* without any performance drop. However, the performance of other methods is vul-

nerable to the imbalance in the dataset, where they have a sharp decrease (e.g., CQL decreases from 60 to 20, and TD+BC decreases from 20 to 0 in AntMaze-medium). The performance gap between RB-CQL and CQL grows with the increasing imbalance (i.e., from *Easy* to *Hard*), which illustrates the effectiveness of the retrieval augmentation on CQL. All algorithms, including RB-CQL, fail in the extremely hard AntMaze task *Hard+*, shown in Appendix B. In Mujoco locomotion tasks, known for easier tasks compared to AntMaze, RB-CQL outperforms other methods with 95% and 97% random dataset ratio. Similarly, in the heavily imbalanced dataset of the *Hard+* AntMaze task, there was limited improvement observed with the use of the RB-CQL method compared to other methods.

We also visualized what we retrieved from our retrieval process, shown in Fig. 4. From the locomotion perspective, the posture is not exactly the same between the query state and retrieved states, but they are similar from the macroscopic point of view. It boosts the generalization ability by the augmentation from the auxiliary dataset.

6. Related Work

Offline RL (Ernst et al., 2005; Riedmiller, 2005; Lange et al., 2012; Levine et al., 2020) is interested in the problem of extracting a policy from a fixed dataset, without further interacting with the environment. It provides the promise to many real-world applications where data collection is expensive or dangerous such as healthcare (Tang & Wiens, 2021), autonomous driving (Kalashnikov et al., 2021), industrial control (Zhan et al., 2022), and task-oriented dialog systems (Verma et al., 2022). The task of offline RL, learning a policy to perform better than the behavior policy in the dataset, often leads to the distributional shift between the learned

policy and the behavior policy (Fujimoto et al., 2019; Levine et al., 2020), which can cause erroneous extrapolation error.

Offline RL without dataset consideration. To alleviate the above problem, previous methods enforce pessimism in actor updates (Wu et al., 2019; Fujimoto et al., 2019; Kumar et al., 2019a; Fujimoto & Gu, 2021; Xu et al., 2021; Zhou et al., 2021) or critic updates (Liu et al., 2020; Kumar et al., 2020; Bai et al., 2022), whose learned policy would query the Q-functions of out-of-distribution actions to outperform the behavior policy, implicitly or explicitly minimizing the divergence over those two policies to mitigate the distributional shift problem. Another branch of methods learns an optimal policy by using only in-sample transitions, also called in-sample learning, bypassing querying the values of unseen actions. This can be achieved by vanilla advantage-weighted regression (Peng et al., 2019; Nair et al., 2020), the extreme approximation of advantage-weighted regression (Chen et al., 2020; Kostrikov et al., 2021; Brandfonbrener et al., 2021), conditioning on some future information or target information (Kumar et al., 2019b; Chen et al., 2021; Emmons et al., 2021), or find the best next states in datasets (Xu et al., 2022). Those methods have achieved significant success in standard benchmarks (Gulcehre et al., 2020; Fu et al., 2020; Qin et al., 2021), however, they have not adequately considered the properties of the dataset, despite it being the primary source to extract policies. In this study, we shift our focus towards offline RL for real-world applications by taking into account the distribution of real-world datasets (i.e., imbalanced datasets), and examining the limitations of current algorithms in such scenarios.

Offline RL with dataset consideration. The prevalent use of benchmarks (Gulcehre et al., 2020; Fu et al., 2020; Qin et al., 2021) in current research has led to a neglect of the importance of the properties of the dataset in the development of models. Previous benchmarks have often assumed that datasets are not imbalanced, with rare but important trajectories, despite the fact that studies such as (Qin et al., 2021) provides dataset from real-world scenarios and recognizes that the exploration challenge is a fundamental challenge in offline RL. Several prior works have conducted extensive empirical analyses to establish various characteristics (Swazinna et al., 2021; Schweighofer et al., 2021) regarding the impact of dataset properties on performance. Despite these efforts, previous studies have primarily focused on standard benchmarks, without taking into account real-world datasets and have not adequately addressed the practical challenges posed by imbalanced datasets. While policies fail to adequately learn under such multi-task or lifelong learning settings with imbalanced datasets from different environments (Zhou et al., 2022; Chan et al., 2022), the single-task setting has been hardly investigated. Instead of allowing the agent to further actively interact with the environment to collect task-specific or task-agnostic datasets

to overcome the imbalance or exploration challenge (Riedmiller et al., 2022; Yarats et al., 2022; Lambert et al., 2022), we are interested in the setting that additional interaction is forbidden but other large-scale offline datasets are available.

Retrieval in RL. Retrieval-augmented methods have recently been widely used to leverage a large body of external information to support downstream tasks in natural language processing (Chen et al., 2017; Karpukhin et al., 2020; Izacard & Grave, 2021) or computer vision fields (Gur et al., 2021; Zhu et al., 2020). Recently, the RL community introduces the retrieval system to integrate experience with new information in a straightforward way to advise the current decision: mitigate the expensive computation and overcome the capacity limitation of models (Goyal et al., 2022; Humphreys et al., 2022). With the same purpose of utilizing related experiences to facilitate decisions, Paischer et al. (2022) use a frozen pre-trained language transformer to embed history information and improve sample efficiency in the partially observable Markov decision process.

The methods mostly similar to ours, RB-CQL, are Goyal et al. (2022); Humphreys et al. (2022), where they both augment a standard reinforcement learning agent with a retrieval process. However, the motivation is different. Our algorithm is designed to overcome the difficulties of the imbalanced datasets in the offline RL setting, while they are aiming to reduce the computation burden and integrate the previous experience into the model without additional gradient updates. Besides, the retrieval architecture of those two algorithms is parametric, which needs other computation sources, but ours is non-parametric.

7. Conclusions and Future Work

In this work, we investigate the challenges posed by imbalanced datasets in offline reinforcement learning, which mirror the distribution commonly found in real-world scenarios. We first characterize the imbalanced offline RL dataset with an imbalanced state coverage, varying policies ranging from a mixture of behaviors to expert ones from sufficient to poor state coverage. Empirically and theoretically, we show that the current offline RL method, CQL, is flawed in extracting high-quality policies under such datasets. To overcome the challenges in imbalanced datasets, We further propose a new method, retrieval-based CQL (RB-CQL), by the augmentation of CQL with a retrieval process, which directly incorporates related experiences from a larger source dataset into training. We evaluate our method on a range of tasks with varying levels of imbalance using the variant of D4RL and show that our method is robust to the imbalance and outperforms other baselines. Limitations of our work are that the retrieval process requires huge computation sources from the CPU and the lack of study on high dimensional inputs, which requires the embedding networks, to exclude

other influences. We hope this work will inspire further research in the field of offline RL, particularly utilizing practical and imbalanced datasets, and serve as a foundation for large-scale offline RL applications.

References

- Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A. C., and Bellemare, M. Deep reinforcement learning at the edge of the statistical precipice. *Proc. of NeurIPS*, 2021.
- An, G., Moon, S., Kim, J.-H., and Song, H. O. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Proc. of NeurIPS*, 2021.
- Bai, C., Wang, L., Yang, Z., Deng, Z., Garg, A., Liu, P., and Wang, Z. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. *arXiv preprint arXiv:2202.11566*, 2022.
- Bertsekas, D. P. and Tsitsiklis, J. N. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE conference on decision and control*, 1995.
- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs. 2018.
- Brandfonbrener, D., Whitney, W. F., Ranganath, R., and Bruna, J. Quantile filtered imitation learning. *arXiv preprint arXiv:2112.00950*, 2021.
- Buda, M., Maki, A., and Mazurowski, M. A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 2018.
- Chan, S. C., Lampinen, A. K., Richemond, P. H., and Hill, F. Zipfian environments for reinforcement learning. In *Conference on Lifelong Learning Agents*, 2022.
- Chen, D., Fisch, A., Weston, J., and Bordes, A. Reading Wikipedia to answer open-domain questions. In *Proc. of ACL*, 2017.
- Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Proc. of NeurIPS*, 2021.
- Chen, X., Zhou, Z., Wang, Z., Wang, C., Wu, Y., and Ross, K. Bail: Best-action imitation learning for batch deep reinforcement learning. *Proc. of NeurIPS*, 2020.
- Emmons, S., Eysenbach, B., Kostrikov, I., and Levine, S. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751*, 2021.
- Ernst, D., Geurts, P., and Wehenkel, L. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 2005.
- Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.
- Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. *Proc. of NeurIPS*, 2021.
- Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *Proc. of ICML*, 2019.
- Gabaix, X. Zipf’s law for cities: an explanation. *The Quarterly journal of economics*, 1999.
- Goyal, A., Friesen, A., Banino, A., Weber, T., Ke, N. R., Badia, A. P., Guez, A., Mirza, M., Humphreys, P. C., Konyushova, K., et al. Retrieval-augmented reinforcement learning. In *Proc. of ICML*, 2022.
- Gulcehre, C., Wang, Z., Novikov, A., Paine, T., Gómez, S., Zolna, K., Agarwal, R., Merel, J. S., Mankowitz, D. J., Paduraru, C., et al. RL unplugged: A suite of benchmarks for offline reinforcement learning. *Proc. of NeurIPS*, 2020.
- Guo, R., Sun, P., Lindgren, E., Geng, Q., Simcha, D., Chern, F., and Kumar, S. Accelerating large-scale inference with anisotropic vector quantization. In *Proc. of ICML*, 2020.
- Gur, S., Neverova, N., Stauffer, C., Lim, S.-N., Kiela, D., and Reiter, A. Cross-modal retrieval augmentation for multi-modal classification. *arXiv preprint arXiv:2104.08108*, 2021.
- Humphreys, P. C., Guez, A., Tieleman, O., Sifre, L., Weber, T., and Lillicrap, T. Large-scale retrieval for reinforcement learning. *arXiv preprint arXiv:2206.05314*, 2022.
- Izcard, G. and Grave, E. Leveraging passage retrieval with generative models for open domain question answering. In *Proc. of EACL*, 2021.
- Kalashnikov, D., Varley, J., Chebotar, Y., Swanson, B., Jonschkowski, R., Finn, C., Levine, S., and Hausman, K. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021.
- Kandpal, N., Deng, H., Roberts, A., Wallace, E., and Raffel, C. Large language models struggle to learn long-tail knowledge. *arXiv preprint arXiv:2211.08411*, 2022.
- Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. Dense passage retrieval for open-domain question answering. In *Proc. of EMNLP*, 2020.

- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. *ArXiv preprint*, 2021.
- Kumar, A., Fu, J., Tucker, G., and Levine, S. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. 2019a.
- Kumar, A., Peng, X. B., and Levine, S. Reward-conditioned policies. *arXiv preprint arXiv:1912.13465*, 2019b.
- Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Proc. of NeurIPS*, 2020.
- Lambert, N., Wulfmeier, M., Whitney, W., Byravan, A., Bloesch, M., Dasagi, V., Hertweck, T., and Riedmiller, M. The challenges of exploration for offline reinforcement learning. *arXiv preprint arXiv:2201.11861*, 2022.
- Lange, S., Gabel, T., and Riedmiller, M. Batch reinforcement learning. In *Reinforcement learning*. 2012.
- Laroche, R., Trichelair, P., and Des Combes, R. T. Safe policy improvement with baseline bootstrapping. In *Proc. of ICML*, 2019.
- Leake, D. B. Case-based reasoning: experiences, lessons, and future directions. 1996.
- Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.
- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. Provably good batch off-policy reinforcement learning without great exploration. *Proc. of NeurIPS*, 2020.
- Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., and Yu, S. X. Large-scale long-tailed recognition in an open world. In *Proc. of CVPR*, 2019.
- Long, A., Yin, W., Ajanthan, T., Nguyen, V., Purkait, P., Garg, R., Blair, A., Shen, C., and van den Hengel, A. Retrieval augmented classification for long-tail visual recognition. In *Proc. of CVPR*, 2022.
- McClelland, J. L., McNaughton, B. L., and O'Reilly, R. C. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 1995.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 2015.
- Nair, A., Gupta, A., Dalal, M., and Levine, S. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.
- Newman, M. E. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 2005.
- Paischer, F., Adler, T., Patil, V., Bitto-Nemling, A., Holzleitner, M., Lehner, S., Eghbal-zadeh, H., and Hochreiter, S. History compression via language models in reinforcement learning. *arXiv preprint arXiv:2205.12258*, 2022.
- Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.
- Qin, R., Gao, S., Zhang, X., Xu, Z., Huang, S., Li, Z., Zhang, W., and Yu, Y. Neorl: A near real-world benchmark for offline reinforcement learning. *arXiv preprint arXiv:2102.00714*, 2021.
- Rashidinejad, P., Zhu, B., Ma, C., Jiao, J., and Russell, S. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Proc. of NeurIPS*, 2021.
- Riedmiller, M. Neural fitted q iteration—first experiences with a data efficient neural reinforcement learning method. In *Proc. of ECML*, 2005.
- Riedmiller, M., Springenberg, J. T., Hafner, R., and Heess, N. Collect & infer—a fresh look at data-efficient reinforcement learning. In *Conference on Robot Learning*, 2022.
- Schaul, T., Quan, J., Antonoglou, I., and Silver, D. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.
- Schweighofer, K., Hofmarcher, M., Dinu, M.-C., Renz, P., Bitto-Nemling, A., Patil, V., and Hochreiter, S. Understanding the effects of dataset characteristics on offline reinforcement learning. *arXiv preprint arXiv:2111.04714*, 2021.
- Singh, A., Kumar, A., Vuong, Q., Chebotar, Y., and Levine, S. Offline rl with realistic datasets: Heteroskedasticity and support constraints. *arXiv preprint arXiv:2211.01052*, 2022.
- Sutton, R. S., Barto, A. G., et al. *Introduction to reinforcement learning*. MIT press Cambridge, 1998.

- Swazinna, P., Udluft, S., and Runkler, T. Measuring data quality for dataset selection in offline reinforcement learning. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2021.
- Tang, S. and Wiens, J. Model selection for offline reinforcement learning: Practical considerations for healthcare settings. In *Proc. of ML4H*, 2021.
- Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *Proc. of IROS*, 2012.
- Verma, S., Fu, J., Yang, M., and Levine, S. Chai: A chatbot ai for task-oriented dialogue with offline reinforcement learning. *arXiv preprint arXiv:2204.08426*, 2022.
- Wu, F., Hoffmann, R., and Weld, D. S. Information extraction from wikipedia: Moving down the long tail. In *Proc. of KDD*, 2008.
- Wu, Y., Tucker, G., and Nachum, O. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.
- Wu, Y., Zhai, S., Srivastava, N., Susskind, J., Zhang, J., Salakhutdinov, R., and Goh, H. Uncertainty weighted actor-critic for offline reinforcement learning. *arXiv preprint arXiv:2105.08140*, 2021.
- Xu, H., Zhan, X., Li, J., and Yin, H. Offline reinforcement learning with soft behavior regularization. *arXiv preprint arXiv:2110.07395*, 2021.
- Xu, H., Jiang, L., Li, J., and Zhan, X. A policy-guided imitation approach for offline reinforcement learning. *arXiv preprint arXiv:2210.08323*, 2022.
- Yarats, D., Brandfonbrener, D., Liu, H., Laskin, M., Abbeel, P., Lazaric, A., and Pinto, L. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. *arXiv preprint arXiv:2201.13425*, 2022.
- Zhan, X., Xu, H., Zhang, Y., Zhu, X., Yin, H., and Zheng, Y. Deepthermal: Combustion optimization for thermal power generating units using offline reinforcement learning. In *Proc. of AAAI*, 2022.
- Zhou, W., Bajracharya, S., and Held, D. Plas: Latent action space for offline reinforcement learning. In *Conference on Robot Learning*, 2021.
- Zhou, W., Bohez, S., Humplik, J., Heess, N., Abdolmaleki, A., Rao, D., Wulfmeier, M., and Harnoja, T. Forgetting and imbalance in robot lifelong learning with off-policy data. In *Conference on Lifelong Learning Agents*, 2022.
- Zhu, L., Song, J., Wei, X., Yu, H., and Long, J. Caesar: Concept augmentation based semantic representation for cross-modal retrieval. *Multimedia Tools and Applications*, 2020.

Offline Reinforcement Learning with Imbalanced Datasets:

Supplementary Material

A. Pseudo Code

Algorithm 1 Retrieval-Augmented Offline Reinforcement Learning

Require: Current training offline dataset \mathcal{D} , retrieval auxiliary dataset \mathcal{D}_{aux} , training timestep T .

```

1: Initialize  $R_\eta, \pi_\theta$ 
2: // Training
3: for  $i = 1, 2, \dots, T$  do
4:   Sample external states  $s_{\text{aux}} \sim \mathcal{D}_{\text{aux}}$ 
5:   Sample transitions  $(s, r, a, s') \sim \mathcal{D}$ 
6:   Select top- $k$  retrieved states  $s_{\text{ret}}$  and  $s'_{\text{ret}}$  via  $\text{Sim}(s, s_{\text{aux}}) = \frac{\exp(s \cdot s_{\text{aux}})}{\sum_{j=1}^{|\mathcal{D}_{\text{aux}}|} \exp(s \cdot s_{\text{aux}}^j)}$ 
7:   Get concatenated states  $\tilde{s} = [s \oplus s_{\text{ret}}], \tilde{s}' = [s \oplus s'_{\text{ret}}]$ 
8:   Use  $(\tilde{s}, r, a, \tilde{s}')$  for downstream offline RL training
9: end for
10: Return policy  $\pi_\theta$ 
11: // Evaluation
12: Get initial state  $s$ , set  $d$  as False
13: while not  $d$  do:
14:   Select top- $k$  retrieved states  $s_{\text{ret}}$  via the same  $\text{Sim}(s, s_{\text{aux}})$ 
15:   Get concatenated states  $\tilde{s} = [s \oplus s_{\text{ret}}]$ 
16:   Get action  $a = \pi_\theta(\tilde{s})$ 
17:   Roll out  $a$  and get  $(s', r, d)$ 
18:   Set  $s = s'$ 
19: end while

```

B. Additional Experiences

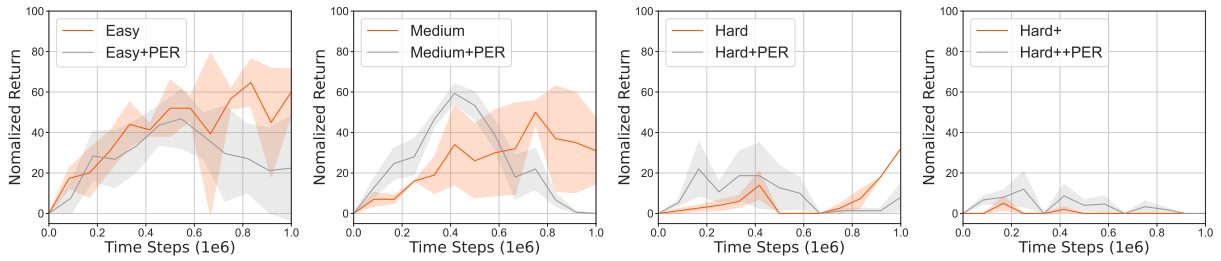


Figure 5. The performance of CQL and CQL with PER on Antmaze-medium task on easy to hard+ level imbalance. Except for the hard+ task, CQL with the augmentation of PER worsens the final performance, compared with CQL.)

C. Experiment Details

The variant of D4RL AntMaze datasets. The dataset we introduced in this work is all imbalanced with the heavy-tail property, featuring imbalanced coverage and rare but important trajectories. In AntMaze tasks, we introduce four new AntMaze datasets for the medium and large mazes from D4RL (Fu et al., 2020). For each type of maze, the imbalance

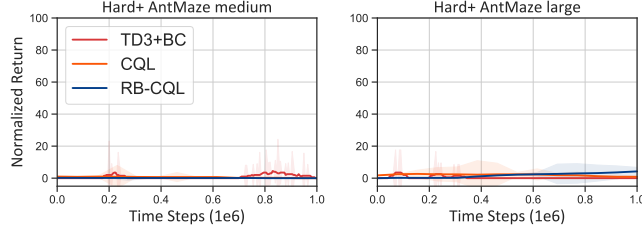


Figure 6. Average normalized scores of RB-CQL against other baselines over the whole training process in the extremely hard task, i.e., hard+.

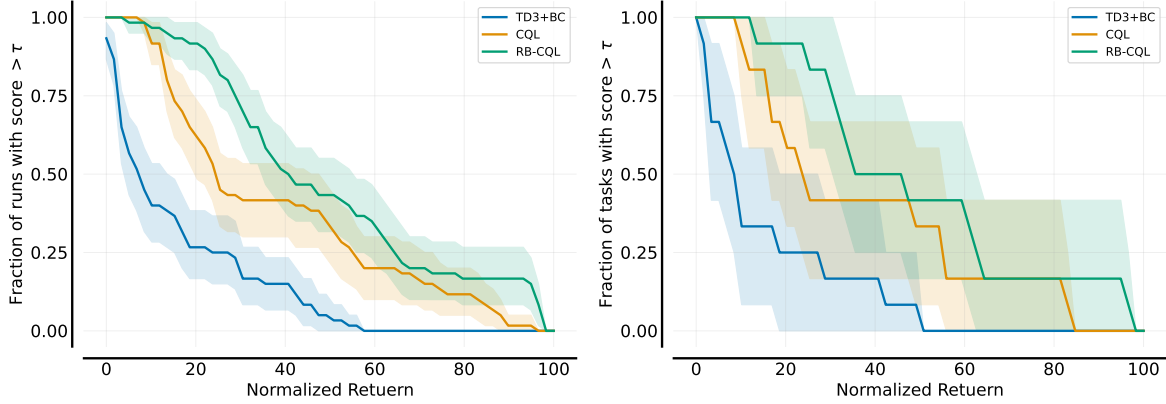


Figure 7. Performance profiles on the imbalanced dataset based on score distributions over five runs and an average of five runs (tasks) on every task. Shaded regions show pointwise 95% confidence bands based on percentile bootstrap with stratified sampling.

increases from *Easy* to *Hard+*, characterizing an optimal decreasing trajectory that can reach the target state distribution in evaluation. In detail, the start state distribution is fixed and aligned with the evaluation, whereas the goal state distribution can be varying and misaligned with the evaluation. In each episode, the goal state distribution can hardly be aligned with the goal state distribution in evaluation. Given an observation, a goal state from the environment, and optimal action from a pre-trained optimal goal-reaching policy, the agent executes the correct action given by the optimal policy. That simulates the imbalanced dataset we have introduced in our work, where the state coverage is heavy-tail and the rare trajectories near the target state are optimal in terms of evaluation. On the other hand, on Mujoco locomotion tasks, we simply combine the random and expert datasets with large ratios approaching 100% (i.e., 95%, 97%, and 99%). It also stimulates the imbalanced dataset we have described because only expert policies can reach certain states and thus gain significant performance, while random policies reach those states with low likelihood. The total transitions in every dataset are 1M. To give a more intuitive representation of the imbalance over the whole state space, we calculate the number of transitions that are in the start distribution grid, a medium grid at the center of the maze, and the target goal state distribution grid, shown in Table 1. We also present the total transitions and expert transitions in imbalanced Mujoco locomotion tasks, shown in Table 2.

For MuJoCo locomotion tasks, we average mean returns over 10 evaluations every 5000 training steps. For AntMaze tasks, we average over 100 evaluations every 0.1M training steps. They are both taken from 5 random seeds. For the source code of our experiences, we re-run TD+BC from author-provided source code https://github.com/sfujim/TD3_BC. Due to the introduction of new datasets and for a fair comparison, we sweep the hyper-parameter α in TD3+BC in $[0.1, 0.2, 0.5, 0.7, 1]$ and choose the best performance in our results. For CQL and our method, we use the open-source JAX (Bradbury et al., 2018) version code: <https://github.com/young-geng/JaxCQL>. Same as we do in TD3+BC, we sweep the hyper-parameter α in CQL in $[5, 10, 20, 50]$ and choose the best performance in our results. Note that RB-CQL and CQL use identical hyper-parameters in all imbalanced datasets. The details of important hyper-parameters α of TD3+BC and CQL (RB-CQL) are shown in Table 3, and other hyper-parameters of TD3+BC and CQL (RB-CQL) are shown in Table 4 and Table 5, where we note that the hyper-parameters in CQL (RB-CQL) with AntMaze tasks are suggested by the open-source code from <https://github.com/young-geng/CQL>.

Table 1. Details of the imbalanced dataset in AntMaze tasks.

Dataset (AntMaze)		Total tansitions	Start	Meidum	Target
medium	Vanilla	100,000	95, 000	51,000	6,000
	Easy	100,000	104, 000	45,000	800
	Medium	100,000	83, 000	45,000	300
	Hard	100,000	84, 000	63,000	90
	Hard+	100,000	90, 000	64,000	21
large	Vanilla	100,000	105, 000	45,000	8,000
	Easy	100,000	100, 000	41,000	4,000
	Medium	100,000	87, 000	54,000	800
	Hard	100,000	92, 000	62,000	350
	Hard+	100,000	86, 000	64,000	100

Table 2. Details of the imbalanced dataset in Mujoco locomotion tasks.

Env	Random ratio	Total transitions	Expert transitions	Random transitions
Hopper	95%	1,000,000	50,000	950,000
	97%	1,000,000	30,000	970,000
	99%	1,000,000	10,000	990,000
Walker2d	95%	1,000,000	50,000	950,000
	97%	1,000,000	30,000	970,000
	99%	1,000,000	10,000	990,000

Table 3. α used for TD3+BC, CQL, and RB-CQL.

Env		TD3+BC	CQL (RB-CQL)
AntMaze-Meidum	Easy	2	5
	Medium	2	5
	Hard	2	5
	Hard+	2	5
AntMaze-Large	Easy	2	5
	Medium	2	5
	Hard	2	5
	Hard+	2	5
Mujoco-hopper	95 random	5	20
	97 random	5	20
	99 random	5	20
Mujoco-walker2d	95 random	2	30
	97 random	2	30
	99 random	2	30

Table 4. The other hyperparameters of TD3+BC.

	Hyperparameter	Value
Architecture	Critic hidden dim	256
	Critic hidden layers	2
	Critic activation function	ReLU
	Actor hidden dim	256
	Actor hidden layers	2
	Actor activation function	ReLU
TD3+BC Hyperparameters	Optimizer	Adam (Kingma & Ba, 2014)
	Critic learning rate	3e-4
	Actor learning rate	3e-4
	Mini-batch size	256
	Discount factor	0.99
	Target update rate	5e-3
	Policy noise	0.2
	Policy noise clipping	(-0.5, 0.5)
	Policy update frequency	2

Table 5. The other hyperparameters of CQL (RB-CQL).

	Hyperparameter	AntMaze	Mujoco
Architecture	Critic hidden dim	256	256
	Critic hidden layers	3	2
	Critic activation function	ReLU	ReLU
	Actor hidden dim	256	256
	Actor hidden layers	3	3
	Actor activation function	ReLU	256
CQL Hyperparameters	Optimizer	Adam (Kingma & Ba, 2014)	Adam
	Critic learning rate	3e-4	3e-4
	Actor learning rate	1e-4	3e-4
	Mini-batch size	256	256
	Discount factor	0.99	0.99
	Target update rate	5e-3	5e-3
	policy_log_std_multiplier	0	1.0
	bc_epochs	40	0
	reward_scale	10	1
	reward_bias	-5	0
	Target entropy	-1 · Action Dim	-1 · Action Dim
	Entropy in Q target	True	True
	Lagrange	True	False
	orthogonal_init	True	False
	Num sampled actions (during eval)	10	10
	Num sampled actions (logsumexp)	10	10