Bayesian Design Principles for Offline-to-Online Reinforcement Learning

Hao Hu [*1] **Yiqin Yang** [*2] **Jianing Ye** [1] **Chengjie Wu** [1] **Ziqing Mai** [1] **Yujing Hu** [3] **Tangjie Lv** [3] **Changjie Fan** [3]
Qianchuan Zhao [4] **Chongjie Zhang** [5]

Abstract

Offline reinforcement learning (RL) is crucial for real-world applications where exploration can be costly or unsafe. However, offline learned policies are often suboptimal, and further online fine-tuning is required. In this paper, we tackle the fundamental dilemma of offline-to-online fine-tuning: if the agent remains pessimistic, it may fail to learn a better policy, while if it becomes optimistic directly, performance may suffer from a sudden drop. We show that Bayesian design principles are crucial in solving such a dilemma. Instead of adopting optimistic or pessimistic policies, the agent should act in a way that matches its belief in optimal policies. Such a probability-matching agent can avoid a sudden performance drop while still being guaranteed to find the optimal policy. Based on our theoretical findings, we introduce a novel algorithm that outperforms existing methods on various benchmarks, demonstrating the efficacy of our approach. Overall, the proposed approach provides a new perspective on offline-to-online RL that has the potential to enable more effective learning from offline data. Our code is public online at https://github.com/YiqinYang/BOORL.

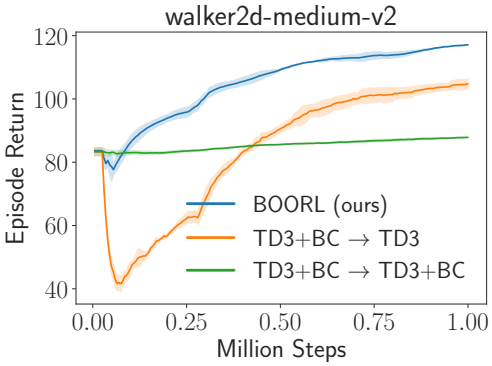


Figure 1: Fine-tuning dilemma in offline-to-online setting. if the algorithm remains pessimistic as it does in offline algorithms, the agent learns slowly due to a lack of exploration (green). Conversely, when the algorithm is optimistic, the agent's performance may suffer from a sudden drop due to inefficient use of offline knowledge and radical exploration (orange). We adopt a probability-matching approach to attain a fast and robust performance improvement (blue).

1. Introduction

Reinforcement learning (RL) has shown impressive success in solving complex decision-making problems such as board games (Silver et al., 2016) and video games (Mnih et al., 2013), and has been applied to many real-world problems like plasma control (Degrave et al., 2022), and human preference alignment (Ouyang et al., 2022). However, online RL algorithms rely on a significant amount of exploration, which can be time-consuming and expensive. Offline RL (Levine et al., 2020) tackles such a problem by utilizing previously collected data and has gained increasing attention in recent years, with the potential to leverage large-scale and diverse datasets (Kumar et al., 2022). However, offline learned policies can be sub-optimal and generalize poorly due to insufficient data and overoptimization (Gao et al., 2023), necessitating further online fine-tuning.

To address this challenge, a hybrid approach (Nair et al., 2020; Lee et al., 2022; Song et al., 2022) has been proposed, enabling sample-efficient learning utilizing both previously collected data and online environments. Pure online or offline strategies are known to fail on the offline-to-online (off-to-on) transition, and various solutions have been proposed, including a balanced replay buffer (Lee et al., 2022; Ball et al., 2023), ensembling (Lee et al., 2022; Ball et al., 2023), and value calibration (Nakamoto et al., 2023). However, there is a lack of a theoretical understanding of the optimistic-pessimistic dilemma, as depicted in Figure 1, and

[*]Equal contribution [1]Institute for Interdisciplinary Sciences, Tsinghua University, China [2]Institute of Automation, Chinese Academy of Sciences, China [3]Fuxi AI Lab, Netease, Inc., Hangzhou, China [4]Department of Automation, Tsinghua University, China [5]Washington University in St. Louis, USA. Correspondence to: Hao Hu <huh22@mails.tsinghua.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

previous methods do not fully address such a dilemma in a principled way. This naturally leads to the question:

What are the design principles to balance offline data reuse and online exploration in offline-to-online reinforcement learning?

To answer this question, we integrate information-theoretic concepts into the design and analysis of RL algorithms. Our results show that Bayesian principles are crucial in solving the dilemma and lead to methods superior to pure online and offline methods. Intuitively, by taking a probability matching approach, which samples from the posterior rather than the most optimistic or the most pessimistic policy, it balances between reusing known experiences and exploring the unknowns. We derive a concrete bound in linear MDPs and conduct experiments in didactic bandits to further demonstrate the superiority of such a Bayesian approach in off-to-on settings.

Based on the theoretical results, we design a simple yet efficient offline-to-online algorithm using approximated posterior sampling with bootstrapped datasets (Osband et al., 2016). Experiments show that our algorithm effectively resolves the dilemma, which effectively explores while avoiding a sudden drop in performance. Also, our algorithm is generally compatible with off-the-shelf offline RL methods for offline-to-online transition.

Our contribution is threefold: (1) we provide an information-theoretic characterization of RL algorithms' performance that links online and offline performance with the agent's gained information about the environment, (2) we demonstrate the superiority of the Bayesian approach in offline-to-online RL theoretically, and (3) we develop a practical approach with bootstrapping for offline-to-online RL and achieve superior performance on various tasks. Overall, our proposed approach provides a new perspective on offline-to-online fine-tuning that has the potential to enable more effective learning from offline data.

1.1. Related Works

Offline RL. Offline RL methods mainly address the extrapolation error issue in value and policy estimation, and can be roughly divided into policy constraint (Yang et al., 2021; Ma et al., 2021; Yang et al., 2023), pessimistic value estimation (Kumar et al., 2020), and model-based methods (Yu et al., 2021; Hu et al., 2022). Policy constraint methods keep the policy close to the behavior under a probabilistic distance (Yang et al., 2021; Ma et al., 2021; Yang et al., 2023) while pessimistic value estimation methods enforce a regularization constraint on the critic loss to penalize over-generalization (Kumar et al., 2020). Model-based methods attempt to learn a model from offline data with minimal modification to the policy learning (Yu et al., 2020).

Offline-to-Online RL. On the empirical side, Nair et al. (2020) is among the first to propose a direct solution to offline-to-online RL. Lee et al. (2022) proposes a balanced replay buffer and pessimistic Q-ensembles. Zhang et al. (2023) proposes a policy expansion scheme to ensure a smooth offline-to-online transition. Ball et al. (2023) leverages ensemble methods and a high update-to-data ratio to enable faster online learning with offline data. Nakamoto et al. (2023) observes an optimistic-pessimistic dilemma similar to ours and proposes calibrating the offline and online learned value function. However, they do not formally point out such a dilemma nor analyze it in a principled way. Wagenmaker & Pacchiano (2023) proposes a novel alignment step in actor-critic RL to bridge the gap in online and offline value learning, which also help mitigate the performance drop during the finetuning stage.

Theoretically, Xie et al. (2021) shows the importance of online exploration when the offline dataset only has partial coverage. Song et al. (2022) demonstrates cases where a purely offline dataset can fail while a hybrid approach succeeds, and Xie et al. (2022) shows an interesting connection between offline concentration coefficient and online learning efficiency. Yu & Zhang (2023) analyzes online learning with offline data in linear MDPs. Their algorithm allows for the incorporation of offline data without degraded performance compared to pure online learning.

Bayesian RL and Information-Theoretic Analysis. Osband & Van Roy (2017); Russo & Van Roy (2014) theoretically justify the effectiveness of Bayesian methods like Thompson sampling. Russo & Van Roy (2016); Lu & Van Roy (2019) discuss the performance of TS in linear bandits. Lu & Van Roy (2019); Ouyang et al. (2017) provide the performance guarantee of TS on tabular MDPs. Dann et al. (2021); Zhong et al. (2022) analyze the frequentist regret of optimistic TS in general function approximation. Uehara & Sun (2021) analyzes the performance of Bayesian methods in the offline setting. Lu & Van Roy (2019) derives an information-theoretical formulation to analyze the regret bound of online learning algorithms like UCB and TS. Xu & Zeevi (2023) proposes an adaptive approach to backup Bayesian algorithms with a frequentist regret. Our work extends these works to offline and off-to-on settings. On the empirical side, (Osband et al., 2016) first adopts a Bayesian view on the exploration in deep RL. (Chua et al., 2018) proposes a model-based approach for Bayesian exploration. Ghosh et al. (2022) adopts the Bayesian principle in the offline setting. Our work extends these works to the off-to-on setting.

2. Preliminaries

2.1. Episodic Reinforcement Learning

We consider finite-horizon episodic Markov Decision Processes (MDPs), defined by the tuple $(\mathcal{S}, \mathcal{A}, H, \mathcal{P}, r)$, where $\mathcal{S}$ is a state space, $\mathcal{A}$ is an action space, H is the horizon and $\mathcal{P} = \{\mathcal{P}_h\}_{h=1}^H, r = \{r_h\}_{h=1}^H$ are the transition function and reward function, respectively.

A policy $\pi = \{\pi_h\}_{h=1}^H$ specifies a decision-making strategy in which the agent chooses its actions based on the current state, i.e., $a_h \sim \pi_h(\cdot \mid s_h)$. The value function $V_h^\pi : \mathcal{S} \to \mathbb{R}$ is defined as the sum of future rewards starting at state s and step $h \in [H]$, and similarly, the Q-value function, i.e.

$$V_h^\pi(s) = \mathbb{E}_\pi\Big[\sum_{t=h}^H r_t(s_t, a_t) \,\Big|\, s_h = s\Big],$$
$$Q_h^\pi(s,a) = \mathbb{E}_\pi\Big[\sum_{t=h}^H r_h(s_t, a_t) \,\Big|\, s_h = s, a_h = a\Big]. \quad (1)$$

where the expectation is w.r.t. the trajectory τ induced by π. We define the Bellman operator as

$$(\mathbb{B}_h f)(s,a) = \mathbb{E}\big[r_h(s,a) + f(s')\big], \quad (2)$$

for any $f : \mathcal{S} \to \mathbb{R}$ and $h \in [H]$. The optimal Q-function Q^*, optimal value function V^* and are related by the Bellman optimality equation

$$V_h^*(s) = \max_{a \in \mathcal{A}} Q_h^*(s,a),$$
$$Q_h^*(s,a) = (\mathbb{B}_h V_{h+1}^*)(s,a), \quad (3)$$

while the optimal policy is defined as

$$\pi_h^*(\cdot \mid s) = \operatorname*{argmax}_\pi \mathbb{E}_{a\sim\pi} Q_h^*(s,a).$$

We define the suboptimality, or the per-episode regret as the performance difference of the optimal policy π^* and the current policy π_k given the initial state $s_1 = s$. That is

$$\Delta_k = \text{SubOpt}(\pi_k; s) = V_1^{\pi^*}(s) - V_1^{\pi_k}(s).$$

2.2. Linear Function Approximation

To derive a concrete bound for Bayesian offline-to-online learning, we consider the *linear MDP* (Jin et al., 2020; 2021) as follows, where the transition kernel and expected reward function are linear with respect to a feature map, which indicate that the value function is also linear.

Definition 1 (Linear MDP)**.** $\mathrm{MDP}(\mathcal{S}, \mathcal{A}, \mathrm{H}, \mathbb{P}, \mathrm{r})$ *is a* linear MDP *with a feature map* $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$*, if for any* $h \in [H]$*, there exist* d unknown *(signed) measures* $\mu_h = (\mu_h^{(1)}, \ldots, \mu_h^{(d)})$ *over* $\mathcal{S}$ *and an* unknown *vector* $\theta_h \in \mathbb{R}^d$*, such that for any* $(s,a) \in \mathcal{S} \times \mathcal{A}$*, we have*

$$\mathbb{P}_h(\cdot \mid s,a) = \langle \phi(s,a), \mu_h(\cdot)\rangle, \quad r_h(s,a) = \langle \phi(s,a), \theta_h\rangle. \quad (4)$$

Without loss of generality, we assume $||\phi(s,a)|| \le 1$ *for all* $(s,a) \in \mathcal{S} \times \mathcal{A}$*, and* $\max\{||\mu_h(\mathcal{S})||, ||\theta_h||\} \le \sqrt{d}$ *for all* $h \in [H]$.

2.3. Information Gain and Bayesian Learning

Let $\mathcal{H}_{k,h} = (s_{1,1}, a_{1,1}, r_{1,1}, \ldots, s_{k,h-1}, a_{k,h-1}, r_{k,h-1}, s_{k,h})$ be all the history up to step h of episode k. We use subscript k,h to indicate quantities conditioned on $\mathcal{H}_{k,h}$, i.e. $\mathbb{P}_{k,h} = \mathbb{P}(\cdot|\mathcal{H}_{k,h}), \mathbb{E}_{k,h}[\cdot] = \mathbb{E}[\cdot|\mathcal{H}_{k,h}]$. The filtered mutual information is defined as

$$I_{k,h}(X;Y) = D_{\mathrm{KL}}(\mathbb{P}_{k,h}(X,Y)||\mathbb{P}_{k,h}(X)\mathbb{P}_{k,h}(Y)),$$

which is a random variable of $\mathcal{H}_{k,h}$. For a horizon dependent quantity $f_{k,h}$, we define $\mathbb{E}_k[f_k] = \sum_{h=1}^H \mathbb{E}_{k,h}[f_{k,h}]$ and similarly for $\mathbb{P}_k$. We use t instead of k,h for simplicity when it does not lead to confusion, e.g., $I_t \triangleq I_{k,h}$.

We assume that the MDP can be described by an unknown model parameter $w = \{w_h\}_{h=1}^H$, which governs the outcome distribution. This is true in linear MDP since Q-function is always linear, i.e., $Q_{w_h}(s,a) = \langle w_h, \phi(s,a)\rangle$. The agent's belief over the environment at the k-th episode is represented as a distribution β_k over w.

We also define the information ratio (Russo & Van Roy, 2016) as the ratio between the expected single step regret and the expected reduction in entropy of the unknown parameter as follows

Definition 2 (Information Ratio)**.** *The information ratio* $\Gamma_{k,h}$ *given history* $\mathcal{H}_{k,h}$ *is the minimum value* Γ *such that the following event*

$$|Q_{w_h}(s,a) - \mathbb{E}Q_{w_h}(s,a)| \le \frac{\Gamma}{2}\sqrt{I_{k,h}(w_h; r_h, s_{h+1} \mid s,a)}, \quad (5)$$

holds for all $h \in [H], s \in \mathcal{S}, a \in \mathcal{A}$ *with probability* $1 - \delta/2$.

We consider the Bayesian regret of an algorithm π over T periods

$$\text{BayesRegret}(T,\pi) = \mathbb{E}[\text{Regret}(T,\pi)] = \mathbb{E}[\sum_{k=1}^K \mathbb{E}_{\beta_k}[\Delta_k]],$$

where $T = HK$ and the expectation is taken over the randomness in outcomes, algorithm π, as well as the posterior distribution β_k over w. We also use $\text{BayesRegret}(N,T,\pi)$

to denote the offline-to-online regret of an algorithm π that uses an offline dataset of size N and interacts online for T steps.

Similar to the definition of the coverage coefficient in offline RL literature (Jin et al., 2021; Uehara & Sun, 2021), we can generalize such a concept by taking the expectation over the belief β. Specifically, we have the following definition from Uehara & Sun (2021).

Definition 3. *The Bayesian coverage coefficient with respect to the feature map $\phi(s,a)$ and posterior β is defined as*

$$C_\beta^\dagger = \max_{h\in[H]} \mathbb{E}_{w\sim\beta} \sup_{\|x\|=1} \frac{x^\top \Sigma_{\pi_w^*,h} x}{x^\top \Sigma_{\rho_h} x}, \tag{6}$$

where $\Sigma_{\pi_w^*,h} = \mathbb{E}_{(s,a)\sim d_{\pi_w^*,h}(s,a)}[\phi(s,a)\phi(s,a)^\top], \Sigma_{\rho_h} = \mathbb{E}_{\rho_h}[\phi(s,a)\phi(s,a)^\top]$.

Bayesian coverage coefficient is a natural generalization of normal coverage coefficient (Uehara & Sun, 2021; Jin et al., 2021; Rashidinejad et al., 2021) in Baysian settings.

3. Theoretical Analysis

It is known that we should adopt optimistic algorithms (e.g., UCB (Auer, 2002)) in online settings to avoid missing optimal strategies, and we should adopt pessimistic algorithms (e.g., LCB (Rashidinejad et al., 2021)) to avoid overconfidence in unknown regions. However, it is unclear what the principled way for offline-to-online settings is where both an offline dataset and an online environment are available. As Figure 1 demonstrates, optimistic online algorithms (e.g., TD3 (Fujimoto et al., 2018)) can mismanage prior knowledge in the dataset, leading to a sudden drop in performance. On the other hand, pessimistic offline algorithms (e.g., TD3+BC (Fujimoto & Gu, 2021)) can be too conservative in exploration, which leads to slow learning.

Such a dilemma is fundamentally intractable under the frequentist (i.e., worst-case) scenario. For instance, consider a multi-arm bandit problem where one arm has limited historical pulls in the dataset. Opting for exploration, we risk significant suboptimality in scenarios where this arm proves less favorable. Conversely, avoiding exploration might incur substantial regret in cases where the arm is optimal. This predicament highlights the limitations of a frequentist analysis for offline-to-online RL. Therefore, we resort to the Bayesian point of view and conduct an information-theoretic analysis in Section 3.1 to understand how we can use both the dataset and the online environment properly under the average regret criteria. Interestingly, we show that a probability matching agent (e.g., Thompson Sampling; TS) can make a good balance and outperform pure online and offline agents. Intuitively, uniformly sampling from the posterior rather than acting according to the most optimistic or pessimistic estimation strikes a proper balance between efficient exploration and safe exploitation.

In Section 3.2, we derive a concrete performance bound for such agents in linear MDPs. Such theoretical performance bound matches well with empirical observations on the didactic bandit problem, as shown in Figure 2. The probability-matching approach contrasts drastically with pure optimism, pure pessimism, and a naive approach that gradually switches from pessimism to optimism.

3.1. Information-Theoretic Analysis

A good exploration strategy should avoid careless trials and incur regret only when it can learn enough from such action. Therefore, the suboptimality should be proportional to the possible information gain from the current policy. Similarly, after learning from the offline dataset, a good exploitation strategy should incur a suboptimality only due to its uncertainty about the environment. The following theorem formalizes such an intuition under the lens of information theory.

Theorem 4. *Then the per-episode regret of Thompson Sampling and UCB agents satisfies*

$$\mathbb{E}_k[\Delta_k] \le \sum_{h=1}^{H} \Gamma_{k,h} \sqrt{I_{k,h}(w_h; a_{k,h}, r_{k,h}, s_{k,h+1})} + 2\delta H^2, \tag{7}$$

where $a_{k,h} \sim \pi_{k,h}$. Similarly, the per-episode regret of Thompson Sampling and LCB agents satisfies

$$\mathbb{E}_k[\Delta_k] \le \sum_{h=1}^{H} \Gamma_{k,h} \sqrt{I_{k,h}(w_h; a_h^*, r_{k,h}, s_{k,h+1})} + 2\delta H^2, \tag{8}$$

where $a_h^* \sim \pi_h^*$.

Proof. Please refer to Appendix C.1 for detailed proof. □

Here UCB, LCB and Thompson Sampling strategies are defined in an abstract and information-theoretic manner, with details shown in Appendix A.1.

Theorem 4 leads to an information-theoretic performance bound for both online exploration and offline exploitation. Equation (7) indicates an online $\widetilde{\mathcal{O}}(\sqrt{T})$-regret bound using the chain rule of mutual information, as depicted in Proposition 8. With additional assumption on the coverage of the dataset, Equation (8) implies an $\widetilde{\mathcal{O}}(\sqrt{C/N})$ offline performance bound where C is the coverage coefficient. Note that we have no assumptions on the structures of the MDPs (e.g., linear MDP, etc.). Instead, we only assume that the uncertainty in the Q-value function can be reduced at a certain rate as we gain more information about the environment and

the total information available does not grow too fast. This assumption generally holds in various settings, including linear MDPs (Jin et al., 2020), factored MDPs (Lu & Van Roy, 2019), and kernel MDPs (Yang & Wang, 2020). Please refer to Lu & Van Roy (2019) for a detailed derivation.

Note that Thompson sampling enjoys both regret bounds in Equation (7) and Equation (8), which indicates that Thompson sampling is suitable for both offline and online settings. Moreover, it indicates that a Bayesian approach enjoys better guarantees in offline-to-online settings since it can avoid sudden performance drop (due to Equation (8)) and explore efficiently (due to Equation (7)). This is summarized in Table 1, where we classify existing settings and corresponding doctrines. Table 1 suggests that the Bayesian approach is consistent across different settings and recommends a realist approach in offline-to-online settings instead of optimism or pessimism.

Setting	Doctrine	Algorithm
Online Learning	Optimism	TS, UCB
Offline Learning	Pessimism	TS, LCB
Offline-to-online	Realism	TS

Table 1: A taxonomy of the doctrines in different settings of reinforcement learning. a Bayesian approach like TS is generally suitable for online, offline and offline-to-online settings, and is the only one that works in the offline-to-online setting.

3.2. Specification in Linear MDPs

To provide more insights about Theorem 4, we consider a concrete regret bound for Bayesian methods in linear MDPs. Applying Theorem 4 to linear MDPs as defined in Definition 1, we have the following theorem.

Theorem 5 (Regret of Bayesian Agents in Linear MDPs, informal)**.** *Given an offline dataset $\mathcal{D}$ of size N, the regret of Thompson sampling during online interaction satisfies the following bound:*

$$\text{BayesRegret}(N,T,\pi) \le c\sqrt{d^3H^3\iota}\left(\sqrt{\frac{N}{C_\beta^\dagger}+T}-\sqrt{\frac{N}{C_\beta^\dagger}}\right), \tag{9}$$

where ι is a logarithmic factor and c is an absolute constant.

Proof. Please refer to Appendix C.2 for detailed proof. □

Theorem 5 demonstrates that the Bayesian approach provides a robust regret guarantee. From simple algebraic observations that $\sqrt{a+b}-\sqrt{a}\le\sqrt{b}$ and $\sqrt{a+b}-\sqrt{a}\le b/(2\sqrt{a})$, Theorem 5 indicates that Bayesian agent can converge to the optimal policy at an $\widetilde{\mathcal{O}}(\sqrt{T})$ rate while having a bounded single-step suboptimality (i.e., $\widetilde{\mathcal{O}}(\sqrt{C_\beta^\dagger/N})$), a feat neither naive online nor offline approaches can accomplish alone. This is further formalized in Propositions 6 and 7.

Proposition 6. *Under the same assumption of Theorem 5, the expected first-step suboptimality of UCB can be unbounded (i.e. $\widetilde{O}(1)$), while the expected suboptimality of Thompson sampling satisfies*

$$\text{SubOpt}(N,T,\pi) \le c\sqrt{\frac{C_\beta^\dagger d^3H^3\iota}{N}} = \widetilde{\mathcal{O}}\left(\sqrt{\frac{C_\beta^\dagger d^3H^3}{N}}\right),$$

where ι is a logarithmic factor and c is an absolute constant.

Proof. Please refer to Appendix C.3 for detailed proof. □

Proposition 7. *Under the same assumption of Theorem 5, the regret of LCB can be unbounded (i.e. $\widetilde{O}(T)$), while the regret of Thompson sampling satisfies*

$$\text{BayesRegret}(N,T,\pi) \le 2c\sqrt{d^3H^3T\iota} = \widetilde{\mathcal{O}}(\sqrt{d^3H^3T}),$$

where ι is a logarithmic factor and c is an absolute constant.

Proof. Please refer to Appendix C.4 for detailed proof. □

The performance bound in Theorem 5 incorporates the number of online interactions T and the offline dataset size N, demonstrating that both elements play a key role in minimizing the average regret. This contrasts significantly with previous theoretical findings. Xie et al. (2021) analyzes the benefit of online exploration when offline data only has partial coverage. At the same time, our result shows that we can benefit from online exploration in Bayesian settings, even if offline data has full coverage. Song et al. (2022) proposes Hy-Q, which allows a weaker notion of coverage by combining a certain proportion of online and offline data, while our result shows that Bayesian strategies achieve the best of both worlds regardless of data proportions.

To further verify our theoretical findings, we conducted experiments on didactic Bernoulli bandit, and the result is shown in Figure 2. The performance of the TS agent aligns well with our predictions in Equation (9) outperforms both pure online and offline agents. We also compare TS with a naive approach where we gradually switch from pessimism to optimism by interpolating the weight on the bonus term. This naive agent performs badly regardless of the choice of the interpolation scheme. This result highlights the importance of adopting a realistic probability match strategy rather than committing to optimism or pessimism. Please refer to Appendix E for more results and details on the didactic bandits.

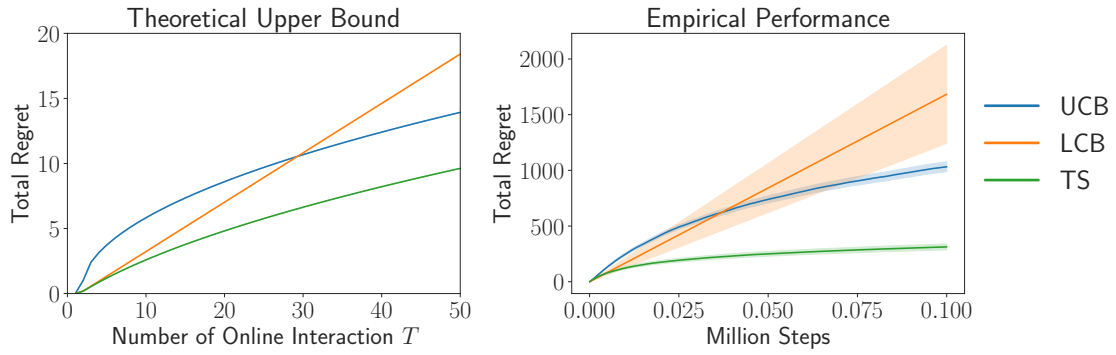


Figure 2: Theoretical upper bound in Theorem 5 and experiment results on Bernoulli bandits. The performance of a Bayesian approach matches the performance of LCB at an early stage by using prior knowledge in the dataset properly and matches the performance of UCB in the run by allowing efficient exploration. Therefore, a realistic Bayesian agent performs better than both optimistic UCB and pessimistic LCB agents.

Algorithm 1 BOORL, Offline Phase

Require: Ensemble size N, offline dataset $\mathcal{D}^{\text{off}}$, masking distribution M
Initialize parameters of N independent TD3+BC agents $\{Q_{\theta_i}, \pi_{\phi_i}\}_{i=1}^N$
for $i = 1, \cdots, N$ **do**
  Sample bootstrap mask $m \sim M$ (e.g., Bernoulli distribution)
  Add m to $\mathcal{D}^{\text{off}}$ as $\mathcal{D}_i^{\text{off}}$
  for each training iteration **do**
    Sample a random minibatch $\{\tau_j\}_{j=1}^B \sim \mathcal{D}_i^{\text{off}}$
    Calculate $L_{\text{critic}}^{\text{offline}}(\theta_i)$ and update θ_i
    Calculate $L_{\text{actor}}^{\text{offline}}(\phi_i)$ and update ϕ_i
  end for
end for
Return $\{Q_{\theta_i}, \pi_{\phi_i}\}_{i=1}^N$

Algorithm 2 BOORL, Online Phase

Require: $\{Q_{\theta_i}, \pi_{\phi_i}\}_{i=1}^N$, offline dataset $\mathcal{D}^{\text{off}}$
Initialize empty online replay buffer $\mathcal{D}^{\text{on}}$
for each iteration **do**
  Obtain initial state from environment s_0
  for step $t = 1, \cdots, T$ **do**
    Construct a categorical distribution based on

$$p_i = \frac{\exp(Q_{\theta_i}(s_t, \pi_{\phi_i}(s_t)))}{\sum_j \exp(Q_{\theta_j}(s_t, \pi_{\phi_j}(s_t)))}$$

    Pick an policy to act $a_t \sim \pi_{\phi_n}(\cdot \mid s_t)$ by sampling index n based on $p = (p_1, p_2, ..., p_N)$
    Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}^{\text{on}}$
    Sample minibatch b from $\mathcal{D}^{\text{off}}$ and $\mathcal{D}^{\text{on}}$
    for $i = 1, \cdots, N$ **do**
      With b, calculate $L_{\text{critic}}^{\text{online}}(\theta_i)$ and update θ_i
    end for
    for $i = 1, \cdots, N$ **do**
      With b, calculate $L_{\text{actor}}^{\text{online}}(\phi_i)$ and update ϕ_i
    end for
  end for
end for

4. Method

Designing a posterior sampling-based offline-to-online agent faces two main challenges. The first is to ensure pessimism while obtaining a posterior distribution during offline pretraining. Naively applying posterior sampling during the offline phase loses frequentist (i.e., worst-case) guarantees (Anonymous, 2024) and can perform poorly in practice. The second challenge is to update the posterior smoothly despite the distributional shift between offline and online data. Otherwise, it may lead to unstable performance improvements.

To address the above challenges, we propose a simple yet effective Bayesian Offline-to-Online Reinforcement Learning (BOORL) method to balance offline policy reuse and online exploration.

Our method consists of two phases. We combine the bootstrapping mechanism and pessimistic offline algorithms during the offline phase to obtain a posterior belief over optimal policies while ensuring offline performance. During the online phase, we generate the posterior belief over policies based on softmax Q-values, and we act by sampling from such posterior. The observation from the environment is then added on the fly to each bootstrapped dataset from the offline phase to update the posterior. We reweight the newly added online data to guarantee a smooth posterior update.

Task	Type	ODT	PEX	Cal-QL	BOORL	RLPD
Hopper	random	10.1→30.8	7.6→10.1	9.3→11.9	8.8→75.7	**84.1**
	medium	66.9→97.5	63.8→78.6	75.8→100.6	61.9→**109.8**	107.3
	medium-replay	86.6→88.8	89.8→103.3	95.4→106.1	75.5→**111.1**	58.9
	medium-expert	107.6→**111.1**	91.5→107.8	85.0→**111.6**	89.0→103.4	95.2
	expert	108.1→**110.7**	102.4→96.6	94.8→**110.3**	111.5→**109.2**	100.4
Walker2d	random	4.6→8.8	2.9→40.7	14.8→17.3	4.8→**93.6**	76.4
	medium	72.1→76.7	79.8→94.8	80.8→89.6	83.6→**107.7**	**108.6**
	medium-replay	68.9→76.8	73.6→89.3	83.8→94.5	69.1→**114.4**	**115.0**
	medium-expert	108.1→108.7	109.6→**117.9**	106.8→111.0	110.8→**116.2**	**115.1**
	expert	108.2→107.6	108.6→111.9	108.8→109.2	110.0→118.6	**119.8**
Halfcheetah	random	1.1→2.2	9.6→61.2	22.0→45.1	10.7→**97.7**	63.0
	medium	42.7→42.1	47.3→67.8	48.0→72.3	47.9→**98.7**	90.5
	medium-replay	39.9→40.4	44.1→55.2	46.5→59.5	44.5→**91.5**	87.6
	medium-expert	86.8→94.1	86.7→91.0	48.0→90.2	77.7→**97.9**	94.3
	expert	87.3→94.3	90.5→95.5	64.5→92.1	97.5→**98.4**	93.2
Antmaze	umaze	56.6→83.5	81.6→**100.0**	78.5→**100.0**	81.7→**100.0**	95.6
	umaze-diverse	51.4→78.9	74.1→78.7	75.1→**96.0**	78.3→**99.4**	93.3
	medium-play	0.0→0.0	68.6→90.8	59.4→91.9	50.6→**100.0**	96.3
	medium-diverse	0.0→0.0	61.3→86.4	69.5→84.1	61.7→86.6	**94.6**
	large-play	0.0→0.0	49.9→68.2	24.2→55.9	61.0→75.8	**81.6**
	large-diverse	0.0→0.0	45.7→70.0	35.2→69.1	50.9→79.9	**86.8**
δ_{sum} (0.2M)		146.0	326.8	392.0	**698.1**	-

Table 2: Normalized score before and after the online fine-tuning with five random seeds. Each method is pre-trained with 1M steps and then fine-tuned with 0.2M online steps. Since offline algorithms' performance differs, we focus on performance improvement within a limited time, δ_{sum} (0.2M), which denotes the sum of performance improvement on all tasks within 0.2M steps. RLPD starts from random initialization, so we only compare the final performance with RLPD rather than the performance improvement.

Offline Phase. In the offline phase, we resample the dataset by generating a set of masks $\{M_\ell\}_{\ell=1}^L$. Each mask $M_\ell \in \{0,1\}^N$ is a binary vector representing if each data point in $\mathcal{D}^{\text{off}}$ is selected in the dataset $\mathcal{D}_\ell^{\text{off}}$. We adopt a standard bootstrapping method, and each M_ℓ is sampled independently from the Bernoulli distribution $\text{Ber}(p)$. These masks are stored in the memory replay buffer to represent $\mathcal{D}_\ell^{\text{off}}$ and we train L policy networks and corresponding Q-value networks $\{\pi_{\phi_\ell}, Q_{\theta_\ell}\}_{\ell=1}^L$ concerning each resampled dataset. Next, each one of these offline policies is trained against its own pessimistic Q-value network and bootstrapped dataset with the offline RL loss (e.g., TD3+BC (Fujimoto & Gu, 2021)):

$$\mathcal{L}_{\text{critic}}(\theta_\ell) = \mathbb{E}_{(s,a,r,s',m_\ell)\sim\mathcal{D}_\ell^{\text{off}}}\Big[(r+\gamma Q_{\theta'_\ell}(s',\widetilde{a}) - Q_{\theta_\ell}(s,a))^2 * M_\ell\Big], \quad (10)$$

$$\mathcal{L}_{\text{actor}}(\phi_\ell) = -\mathbb{E}_{(s,a,m_\ell)\sim\mathcal{D}_\ell^{\text{off}}}\left[(\lambda Q_{\theta_\ell}(s,\pi_{\phi_\ell}(s)) - (\pi_{\phi_\ell}(s)-a)^2) * M_\ell\right], \quad (11)$$

where $\widetilde{a} = \pi_{\phi'_\ell}(s') + \epsilon$ is the target policy smoothing regularization and λ is the hyper-parameter for behavior cloning. θ'_ℓ and ϕ'_ℓ are the parameters for the Q-value and policy target network, respectively.

Online Phase. When acting in the online environment, we first compute posterior beliefs over candidate policies $\{\phi_\ell\}_{\ell=1}^L$ according to

$$p_\ell = \frac{\exp(Q_{\theta_\ell}(s,a_\ell))}{\sum_j \exp(Q_{\theta_j}(s,a_j))}, \quad (12)$$

where $a_\ell \sim \pi_{\phi_\ell}(s)$. Then, we sample index $\ell \in \{1,\cdots,L\}$ based on $p = (p_1, p_2, ..., p_L)$ at each time step and following π_n to collect online data. Each loaded policy and Q-value network is continued to be trained with the online RL loss (e.g., TD3 (Fujimoto et al., 2018)). Additionally, to incorporate prior data better, we reweight the online data so that for the training batch, 50% of the sampled data are from the online phase, and the remaining 50% from the offline replay buffer $\mathcal{D}^{\text{off}}$ (Ross & Bagnell, 2012).

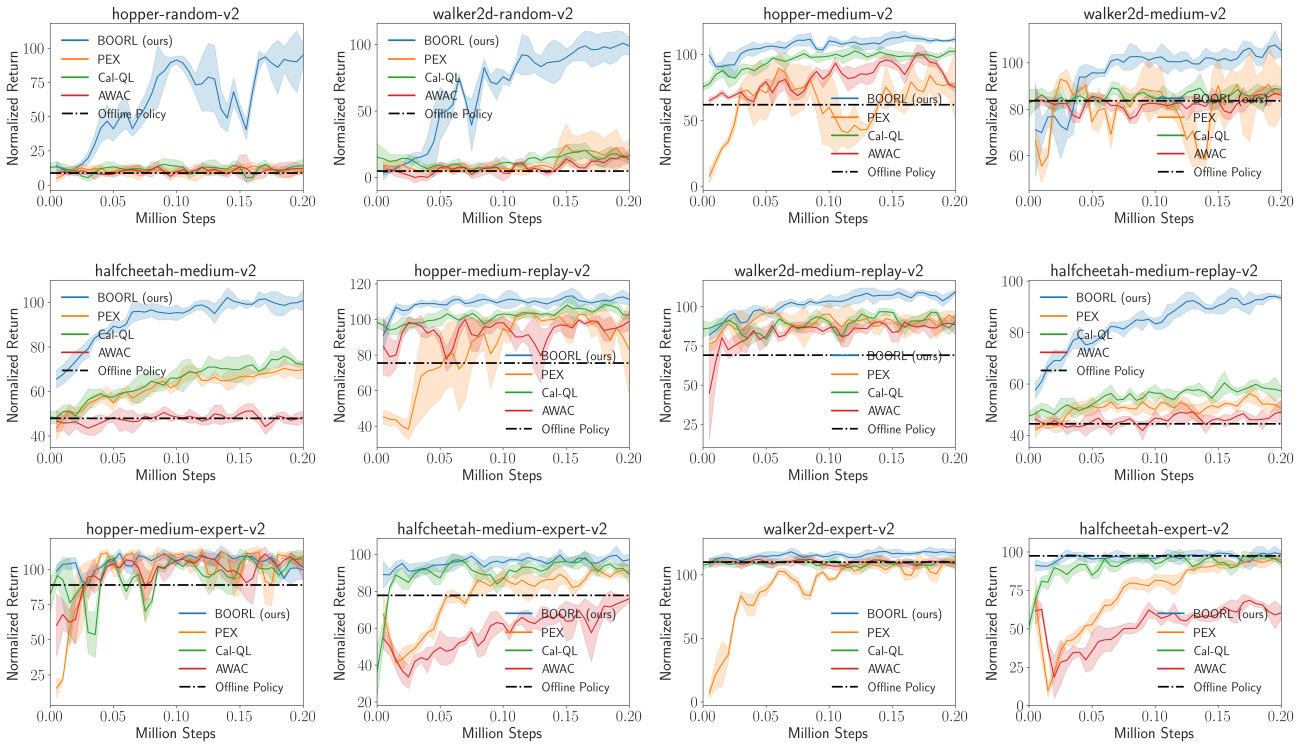


Figure 3: Experiments between several baselines and BOORL within 0.2M time steps. The reference line is the performance of TD3+BC. The experimental results are averaged with five random seeds. Please refer to Appendix G for more results.

Intuitively, the Bayesian-based agent allows a "robust" policy improvement since it adopts a mild exploration strategy by sampling from a soft distribution, which reduces "catastrophic" attempts from optimistic exploration. The overall algorithm is summarized in Algorithm 1 and Algorithm 2. We highlight elements important to our approach in Purple. Our approach only requires minor adjustments to existing methods, making its effectiveness easily implementable and following.

5. Experiments

We design our experiments to answer the following questions: (1) Whether BOORL can effectively solve the dilemma in offline-to-online RL? (2) How does BOORL compare with other state-of-the-art approaches for finetuning pre-trained policies? (3) Is BOORL general, and can it be effectively combined with other off-the-shelf offline RL algorithms?

To answer the questions above, we conduct experiments to test our proposed approach on the D4RL benchmark (Fu et al., 2020), which encompasses a variety of dataset qualities and domains. We adopt the normalized score metric proposed by the D4RL benchmark (Fu et al., 2020), averaging over five random seeds with standard deviation.

Answer of Question 1: We compare BOORL with the online version of TD3+BC (Fujimoto & Gu, 2021), named `TD3+BC (online)`, as well as directly using TD3 for finetuning, named `TD3 (finetune)`. For the fair comparison, these three methods are all pre-trained based on TD3+BC for 1 million time steps and adopt the TD3 algorithm for online learning. The results in Figure 6 in Appendix F show TD3+BC exhibits safe but slow performance improvement, resulting in worse asymptotic performance. On the other hand, TD3 suffers from initial performance degradation, especially in narrow distribution datasets (e.g., expert datasets), while BOORL attains a fast performance improvement with a smaller regret.

Answer of Question 2: We compare BOORL with several strong offline-to-online algorithms, inclduing ODT (Zheng et al., 2022), AWAC (Nair et al., 2020), PEX (Zhang et al., 2023), Cal-QL (Nakamoto et al., 2023) and RLPD (Ball et al., 2023). We re-run the official implementation to offline pre-train for 1 million steps. Then, we report the fine-tuned performance for 200k online steps. As for BOORL, we use TD3+BC and TD3 as the backbone of offline and online algorithms. Table 2 shows that our algorithm achieves superior fine-tuning performance and notable performance improvement δ_{sum} compared with other fine-tuning ap-

Task	Type	BOORL	Bayesian	δ	Hybrid RL	δ
Hopper	random	75.7±1.3	85.4±3.3	-9.7	75.2±3.9	0.5
	medium	109.8±1.6	109.6±1.5	0.2	91.4±1.2	18.4
	medium-replay	111.1±0.3	110.6±0.6	0.5	103.5±2.7	7.6
Walker2d	random	93.6±4.4	92.4±4.7	1.2	15.4±0.8	78.2
	medium	107.7±0.5	96.5±3.5	11.2	86.4±0.4	21.3
	medium-replay	114.4±0.9	103.7±2.1	10.7	99.7±2.4	14.7
Halfcheetah	random	97.7±1.1	94.5±4.2	3.2	85.2±0.5	12.5
	medium	98.7±0.3	97.7±0.5	1.0	80.3±0.2	18.4
	medium-replay	91.5±0.9	90.5±0.5	1.0	84.8±1.0	6.7

Table 3: Ablation results on Mujoco tasks with the normalized score metric.

proaches. The results in Figure 3 show that our method achieves better learning efficiency and stability than these baselines. AWAC has limited efficiency due to a lack of online adaptability. To further test the performance of our method across various timesteps, we compare our method with PEX over 1M steps. Results in Appendix G show that our method outperforms PEX across different timesteps. Cal-QL achieves comparable stability due to calibration, but our method generally demonstrates better sample efficiency.

Answer of Question 3: We incorporate BOORL with another offline RL algorithm, IQL, and evaluate it on the sparse reward task in the D4RL benchmark, Antmaze. Consistent with the previous experimental setup, we first offline train IQL for 1 million time steps and then load the same pre-trained weight for BOORL. The experimental results in Table 2 show that BOORL achieves superior performance and higher sample efficiency than other baselines. This demonstrates that BOORL can be easily extended to various offline RL algorithms.

Ablation Study. To delve deeper into the performance of Bayesian methods, we enforced a strict offline $\rightarrow$ online transition. Specifically, we exclusively loaded the offline-trained policy and Q-network module, omitting the offline data during the online phase. We refer to this setup as `Bayesian`. Furthermore, we examined the naive offline-to-online (TD3+BC $\rightarrow$ TD3) with the Hybrid RL framework to examine the effects of integrating offline data, termed `Hybrid RL`. Results in Table 3 reveal that Thompson Sampling perform well in most tasks. Please refer to Appendix H for additional ablation studies.

6. Conclusion

Our work presents a novel perspective on offline-to-online RL under Bayesian principles. By utilizing a probability-matching strategy, we can effectively tackle the inherent challenge of balancing exploration efficiency and utilizing offline data. Such intuition is formalized with an information-theoretic analysis, and we provide a concrete regret bound on linear MDPs. Previous methods like a balanced replay buffer and ensembling can be rationalized as implementing an implicit probability matching strategy. Based on the above theoretical insights, we design a two-phase algorithm that implements approximate posterior sampling with bootstrapping, which provides a smooth and robust transition from offline to online. Our algorithm outperforms previous methods and demonstrates superior outcomes across various tasks. Adopting more advanced Bayesian methods to enable a more efficient and robust offline-to-online transition is an interesting future direction.

Impact Statement

This paper focuses on the offline reinforcement learning area. This paper offers a reliable method for effectively improving online reinforcement learning methods by offline pre-training, substantiated by sufficient proof, which provides considerable societal importance. In contexts where substantial data is available, our method can save significant time and cost for online reinforcement learning and avoid the performance drop. This has the potential to expand the current boundaries of application in the field of offline reinforcement learning, making it more accessible and applicable in a broader range of societal contexts.

As for ethical aspects, to the best of our knowledge, the research presented in this paper does not directly engage with them. However, we acknowledge the importance of ethical considerations in machine learning research and strive to ensure our work aligns with general ethical standards.

References

Anonymous. Posterior sampling via langevin monte carlo for offline reinforcement learning, 2024. URL https://openreview.net/forum?id=WwCirclMvl.

Auer, P. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Ball, P. J., Smith, L., Kostrikov, I., and Levine, S. Efficient online reinforcement learning with offline data. *arXiv preprint arXiv:2302.02948*, 2023.

Cai, Q., Yang, Z., Jin, C., and Wang, Z. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pp. 1283–1294. PMLR, 2020.

Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

Dann, C., Mohri, M., Zhang, T., and Zimmert, J. A provably efficient model-free posterior sampling method for episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 34:12040–12051, 2021.

Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de Las Casas, D., et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

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. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Ghosh, D., Ajay, A., Agrawal, P., and Levine, S. Offline rl policies should be trained to be adaptive. In *International Conference on Machine Learning*, pp. 7513–7530. PMLR, 2022.

Gittens, A. and Tropp, J. A. Tail bounds for all eigenvalues of a sum of random matrices. *arXiv preprint arXiv:1104.4513*, 2011.

Hu, H., Yang, Y., Zhao, Q., and Zhang, C. On the role of discount factor in offline reinforcement learning. In *International Conference on Machine Learning*, pp. 9072–9098. PMLR, 2022.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pp. 2137–2143. PMLR, 2020.

Jin, Y., Yang, Z., and Wang, Z. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pp. 5084–5096. PMLR, 2021.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191, 2020.

Kumar, A., Agarwal, R., Geng, X., Tucker, G., and Levine, S. Offline q-learning on diverse multi-task data both scales and generalizes. *arXiv preprint arXiv:2211.15144*, 2022.

Lee, S., Seo, Y., Lee, K., Abbeel, P., and Shin, J. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning*, pp. 1702–1712. PMLR, 2022.

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.

Lu, X. and Van Roy, B. Information-theoretic confidence bounds for reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Ma, X., Yang, Y., Hu, H., Liu, Q., Yang, J., Zhang, C., Zhao, Q., and Liang, B. Offline reinforcement learning with value-based episodic memory. *arXiv preprint arXiv:2110.09796*, 2021.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Nair, A., Gupta, A., Dalal, M., and Levine, S. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Nakamoto, M., Zhai, Y., Singh, A., Mark, M. S., Ma, Y., Finn, C., Kumar, A., and Levine, S. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. *arXiv preprint arXiv:2303.05479*, 2023.

Osband, I. and Van Roy, B. Why is posterior sampling better than optimism for reinforcement learning? In *International conference on machine learning*, pp. 2701–2710. PMLR, 2017.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Ouyang, Y., Gagrani, M., Nayyar, A., and Jain, R. Learning unknown markov decision processes: A thompson sampling approach. *Advances in neural information processing systems*, 30, 2017.

Rashidinejad, P., Zhu, B., Ma, C., Jiao, J., and Russell, S. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021.

Ross, S. and Bagnell, J. A. Agnostic system identification for model-based reinforcement learning. *arXiv preprint arXiv:1203.1007*, 2012.

Russo, D. and Van Roy, B. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39 (4):1221–1243, 2014.

Russo, D. and Van Roy, B. An information-theoretic analysis of thompson sampling. *The Journal of Machine Learning Research*, 17(1):2442–2471, 2016.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Song, Y., Zhou, Y., Sekhari, A., Bagnell, J. A., Krishnamurthy, A., and Sun, W. Hybrid rl: Using both offline and online data can make rl efficient. *arXiv preprint arXiv:2210.06718*, 2022.

Uehara, M. and Sun, W. Pessimistic model-based offline reinforcement learning under partial coverage. *arXiv preprint arXiv:2107.06226*, 2021.

Wagenmaker, A. and Pacchiano, A. Leveraging offline data in online reinforcement learning. In *International Conference on Machine Learning*, pp. 35300–35338. PMLR, 2023.

Xie, T., Jiang, N., Wang, H., Xiong, C., and Bai, Y. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. *Advances in neural information processing systems*, 34:27395–27407, 2021.

Xie, T., Foster, D. J., Bai, Y., Jiang, N., and Kakade, S. M. The role of coverage in online reinforcement learning. *arXiv preprint arXiv:2210.04157*, 2022.

Xu, Y. and Zeevi, A. Bayesian design principles for frequentist sequential learning. In *International Conference on Machine Learning*, pp. 38768–38800. PMLR, 2023.

Yang, L. and Wang, M. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pp. 10746–10756. PMLR, 2020.

Yang, Y., Ma, X., Li, C., Zheng, Z., Zhang, Q., Huang, G., Yang, J., and Zhao, Q. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:10299–10312, 2021.

Yang, Y., Hu, H., Li, W., Li, S., Yang, J., Zhao, Q., and Zhang, C. Flow to control: Offline reinforcement learning with lossless primitive discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10843–10851, 2023.

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., and Ma, T. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., and Finn, C. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

Yu, Z. and Zhang, X. Actor-critic alignment for offline-to-online reinforcement learning. In *International Conference on Machine Learning*, pp. 40452–40474. PMLR, 2023.

Zhang, H., Xu, W., and Yu, H. Policy expansion for bridging offline-to-online reinforcement learning. *arXiv preprint arXiv:2302.00935*, 2023.

Zheng, Q., Zhang, A., and Grover, A. Online decision transformer. In *International Conference on Machine Learning*, pp. 27042–27059. PMLR, 2022.

Zhong, H., Xiong, W., Zheng, S., Wang, L., Wang, Z., Yang, Z., and Zhang, T. Gec: A unified framework for interactive decision making in mdp, pomdp, and beyond. *arXiv preprint arXiv:2211.01962*, 2022.

A. Algorithm Details

A.1. Details of abstract algorithms

In this section, we provide an information-theoretic abstraction of the UCB, LCB and Thompson Sampling algorithm when both offline dataset and online environment are present. Here we use t to represent (k, h) for simplicity.

Algorithm 3 Upper Confidence Bound

Require: offline dataset $\mathcal{D}^{\text{off}}$, online interaction episodes K, exploration coefficient Γ_t,
Initialize prior β_0 with the offline dataset $\mathcal{D}^{\text{off}}$.
for $k = 1, \cdots, K$ **do**
  for $h = 1, \cdots, H$ **do**
    Calculate posterior mean $\bar{Q}_{w,t}(\cdot,\cdot) = \mathbb{E}_{w\sim\beta_t}[Q_{t,w}(\cdot,\cdot)]$.
    Calculate optimistic value function $\widehat{Q}(\cdot,\cdot) = \bar{Q}_{w,t}(\cdot,\cdot) + \frac{\Gamma_t}{2}\sqrt{I_t(w;\cdot,\cdot)}$.
    Execute $a_t = \text{argmax}_{a_t} \widehat{Q}(s_t, a)$ and receive feedback s_{t+1}, r_t.
    Update posterior β_t with evidence (a_t, r_t, s_{t+1}).
  end for
end for

Algorithm 4 Lower Confidence Bound

Require: offline dataset $\mathcal{D}^{\text{off}}$, online interaction episodes K, exploration coefficient Γ_t,
Initialize prior β_0 with the offline dataset $\mathcal{D}^{\text{off}}$.
for $k = 1, \cdots, K$ **do**
  for $h = 1, \cdots, H$ **do**
    Calculate posterior mean $\bar{Q}_{w,t}(\cdot,\cdot) = \mathbb{E}_{w\sim\beta_t}[Q_{t,w}(\cdot,\cdot)]$.
    Calculate pessimistic value function $\widehat{Q}(\cdot,\cdot) = \bar{Q}_{w,t}(\cdot,\cdot) - \frac{\Gamma_t}{2}\sqrt{I_t(w;\cdot,\cdot)}$.
    Execute $a_t = \text{argmax}_{a_t} \widehat{Q}(s_t, a)$ and receive feedback s_{t+1}, r_t.
    Update posterior β_t with evidence (a_t, r_t, s_{t+1}).
  end for
end for

Algorithm 5 Thompson Sampling

Require: offline dataset $\mathcal{D}^{\text{off}}$, online interaction episodes K, exploration coefficient Γ_t,
Initialize prior β_0 with the offline dataset $\mathcal{D}^{\text{off}}$.
for $k = 1, \cdots, K$ **do**
  for $h = 1, \cdots, H$ **do**
    Sample parameter w_t from posterior β_t.
    Calculate corresponding value function $\widehat{Q}(\cdot,\cdot) = Q_{w_t,t}(\cdot,\cdot)$.
    Execute $a_t = \text{argmax}_{a_t} \widehat{Q}(s_t, a)$ and receive feedback s_{t+1}, r_t.
    Update posterior β_t with evidence (a_t, r_t, s_{t+1}).
  end for
end for

B. Additional Propositions

Proposition 8. *Suppose the following equation for each episode holds with $\Gamma_t \leq \Gamma$ for all $k \in [K], h \in [H]$,*

$$\mathbb{E}_k[\Delta_k] \leq \sum_{h=1}^{H} \Gamma_t \sqrt{I_t(w_h; a_t, r_t, s_{t+1})} + \epsilon_t. \tag{13}$$

Then

$$\mathbb{E}[\textit{Regret}(T, \pi)] \leq \Gamma \sqrt{T I(w; \mathcal{H}_T)} + \mathbb{E} \sum_{k=1}^{K} \sum_{h=1}^{H} \epsilon_t.$$

Proof. By leveraging the chain rule of mutual information, i.e.

$$I(X; Y_1, \ldots, Y_N) = \sum_{i=1}^{N} I(X; Y_i | Y_1, \ldots, Y_{i-1}),$$

and the Cauchy-Schwartz inequality, we have the result immediately. □

C. Missing Proofs

C.1. Proof of Theorem 4

Proof. By regret decomposition in Lemma 12, we have

$$\begin{aligned}
\mathbb{E}_k[\Delta_k] &= \sum_{h=1}^{H} \mathbb{E}_{\pi^*} \left[\langle Q_t(s_h, \cdot), \pi_h^*(\cdot \mid s_h) - \pi_{h,k}(\cdot \mid s_h) \rangle \right] + \sum_{h=1}^{H} \left(\mathbb{E}_{\pi^*}[\iota_t(s,a)] - \mathbb{E}_k[\iota_t(s,a)] \right) \\
&\leq \sum_{h=1}^{H} \left(\mathbb{E}_{\pi^*}[\iota_t(s,a)] - \mathbb{E}_k[\iota_t(s,a)] \right),
\end{aligned}$$

where $\iota_t(s,a) = r_t(s,a) + (\mathbb{B}_h V_{t+1})(s,a) - Q_{t+1}(s,a)$. The inequality is due to the fact that π_t is the greedy policy with respect to Q_t.

Let $\mathcal{W}_k$ be the confidence set of w such that Equation (5) holds, we have

$$\mathbb{P}_k(w \in \mathcal{W}_k) \geq 1 - \frac{\delta}{2}.$$

For a UCB algorithm with upper confidence estimation $Q_t(s,a) = \bar{Q}_{t,w}(s,a) + \frac{\Gamma_t}{2} \sqrt{I_t(w_h; r_{t,a}, s_{t+1,a})}$, we have

$$-\Gamma_t \sqrt{I_t(w_h; r_{t,a}, s_{t+1,a})} \leq \iota_t(s,a) \leq 0$$

when $w \in \mathcal{W}_k$. Here $r_{t,a}$ and $s_{t+1,a}$ are the random variables for the reward and next state given action a at time t.

Then we have

$$
\begin{aligned}
&\sum_{h=1}^{H} \left(\mathbb{E}_{\pi^*}[\iota_t(s,a)] - \mathbb{E}_k[\iota_t(s,a)]\right) \\
\leq &\sum_{h=1}^{H} \left\{ \mathbb{1}_{w \in \mathcal{W}_k} \left\{\mathbb{E}_{\pi^*}[\iota_t(s,a)] - \mathbb{E}_k[\iota_t(s,a)]\right\} + \frac{1}{2}\delta \cdot 2H \right\} \\
\leq &\sum_{h=1}^{H} \mathbb{E}_k \left[\sum_{a \in \mathcal{A}} \mathbb{P}(a_t = a) \Gamma_t \sqrt{I_t(w_h; r_{t,a}, s_{t+1,a})} \right] + \delta H^2 \\
\leq &\sum_{h=1}^{H} \mathbb{E}_k \left[\Gamma_t \sqrt{\sum_{a \in \mathcal{A}} \mathbb{P}(a_t = a) I_t(w_h; r_{t,a}, s_{t+1,a})} \right] + \delta H^2 \\
= &\sum_{h=1}^{H} \mathbb{E}_k \left[\Gamma_t \sqrt{\sum_{a \in \mathcal{A}} \mathbb{P}(a_t = a) I_t(w_h; r_{t,a_t}, s_{t+1,a_t} \mid a_t = a)} \right] + \delta H^2 \\
= &\sum_{h=1}^{H} \mathbb{E}_k \left[\Gamma_t \sqrt{I_t(w_h; a_t, r_t, s_{t+1})} \right] + \delta H^2.
\end{aligned}
$$

Similarly, for a LCB algorithm with lower confidence functions $Q_t(s,a) = \bar{Q}_{t,w}(s,a) - \frac{\Gamma_t}{2}\sqrt{I_t(w_h; r_{t,a}, s_{t+1,a})}$, we have

$$
0 \leq \iota_t(s,a) \leq \Gamma_t \sqrt{I_t(w_h; r_{t,a}, s_{t+1,a})}
$$

when $w \in \mathcal{W}_k$.Then we have

$$
\begin{aligned}
&\sum_{h=1}^{H} \left(\mathbb{E}_{\pi^*}[\iota_t(s,a)] - \mathbb{E}_k[\iota_t(s,a)]\right) \\
\leq &\sum_{h=1}^{H} \left\{ \mathbb{1}_{w \in \mathcal{W}_k} \left\{\mathbb{E}_{\pi^*}[\iota_t(s,a)] - \mathbb{E}_k[\iota_t(s,a)]\right\} + \frac{1}{2}\delta \cdot 2H \right\} \\
\leq &\sum_{h=1}^{H} \mathbb{E}_{\pi^*} \left[\sum_{a \in \mathcal{A}} \mathbb{P}(a_t^* = a) \Gamma_t \sqrt{I_t(w_h; r_{t,a}, s_{t+1,a})} \right] + \delta H^2 \\
\leq &\sum_{h=1}^{H} \mathbb{E}_{\pi^*} \left[\Gamma_t \sqrt{\sum_{a \in \mathcal{A}} \mathbb{P}(a_t^* = a) I_t(w_h; r_{t,a}, s_{t+1,a})} \right] + \delta H^2 \\
= &\sum_{h=1}^{H} \mathbb{E}_{\pi^*} \left[\Gamma_t \sqrt{\sum_{a \in \mathcal{A}} \mathbb{P}(a_t^* = a) I_t(w_h; r_{t,a_t^*}, s_{t+1,a_t^*} \mid a_t^* = a)} \right] + \delta H^2 \\
= &\sum_{h=1}^{H} \mathbb{E}_{\pi^*} \left[\Gamma_t \sqrt{I_t(w_h; a_t^*, r_t, s_{t+1})} \right] + \delta H^2.
\end{aligned}
$$

For Thompson Sampling, note that the probability matching property implies that $\mathbb{P}_k(\widehat{w}_k \in \mathcal{W}_k) = \mathbb{P}_k(w \in \mathcal{W}_k) \geq 1 - \frac{\delta}{2}$, we have

$$
\begin{aligned}
&\sum_{h=1}^{H} \left(\mathbb{E}_{\pi^*}[\iota_t(s,a)] - \mathbb{E}_k[\iota_t(s,a)]\right) \\
\leq &\sum_{h=1}^{H} \left\{\mathbb{1}_{w,\widehat{w}_k \in \mathcal{W}_k} \left\{\mathbb{E}_{\pi^*}[\iota_t(s,a)] - \mathbb{E}_k[\iota_t(s,a)]\right\} + \delta \cdot 2H\right\} \\
\leq &\sum_{h=1}^{H} \mathbb{E}_{\pi^*}\left[\sum_{a\in\mathcal{A}} \mathbb{P}(a_t^* = a)\Gamma_t \sqrt{I_t(w_h; r_{t,a}, s_{t+1,a})}\right] + 2\delta H^2 \\
= &\sum_{h=1}^{H} \mathbb{E}_{k}\left[\sum_{a\in\mathcal{A}} \mathbb{P}(a_t = a)\Gamma_t \sqrt{I_t(w_h; r_{t,a}, s_{t+1,a})}\right] + 2\delta H^2.
\end{aligned}
$$

The last equality and the last equality is due to the fact that $\pi_{k,h}$ is the optimal policy under $\widehat{w}_k$, and w and $\widehat{w}_k$ have the same distribution. The rest of the proof is similar to the case of UCB and LCB and is omitted for simplicity. □

C.2. Proof of Theorem 5

Theorem 9 (Regret of Bayesian Agents in Linear MDPs, restatment)**.** *Given an offline dataset $\mathcal{D}$ of size N, and a fixed posterior β during the online interaction phase, the regret of Thompson sampling during online interaction satisfies the following bound:*

$$
\text{BayesRegret}(N, T, \pi) \leq 4c\sqrt{d^3H^3}\iota \left(\sqrt{\frac{N}{C_\beta^\dagger} + T} - \sqrt{\frac{N}{C_\beta^\dagger}}\right), \tag{14}
$$

for sufficiently large N or T. Here ι is a logarithmic factor and c is an absolute constant.

Proof. At each online episode k, we have

$$
\begin{aligned}
&\sum_{h=1}^{H} \Gamma_t \mathbb{E}_{\pi^*}\left[\sqrt{I_t(w_h; a_t^*, r_t, s_{t+1})}\right] \\
=&\sum_{h=1}^{H} \Gamma_t \mathbb{E}_{\pi^*}\left[\log\left(1 + \phi(s_h,a_h)^\top \Lambda_h^{-1} \phi(s_h,a_h)\right)^{1/2}\right] \\
\leq&\Gamma_t \mathbb{E}_{\pi^*}\Big[\sum_{h=1}^{H} \left(\phi(s_t,a_t)^\top \Lambda_k^{-1} \phi(s_t,a_t)\right)^{1/2}\Big] \\
=&\Gamma_t \mathbb{E}_{\pi^*}\Big[\sum_{h=1}^{H} \sqrt{\text{Tr}\left(\phi(s,a)\phi(s,a)^\top \Lambda_k^{-1}\right)}\Big] \\
\leq&\Gamma_t \sum_{h=1}^{H} \sqrt{\text{Tr}\left(\mathbb{E}_{d_h^{\pi^*}}\left[\phi(s,a)\phi(s,a)^\top\right]\Lambda_t^{-1}\right)} \\
=&\Gamma_t \sum_{h=1}^{H} \sqrt{\text{Tr}\left(\Sigma_{\pi^*,h}^\top \Lambda_t^{-1}\right)},
\end{aligned} \tag{15}
$$

where $\Sigma_{\pi^*,h} \triangleq \mathbb{E}_{d_h^{\pi^*}}\left[\phi(s,a)\phi(s,a)^\top\right]$. The first equality uses Lemma 14, The first inequality uses the fact that $\log(1+x) \leq x \forall x \geq 0$. The second equality uses the trace trick and the last inequality due to Jensen inequality and the linearity of the trace function.

By the definition of Bayesian coverage coefficient, we have

$$\mathbb{E}\left[\sum_{\ell=1}^{L}\phi(s_{\ell,h},a_{\ell,h})\phi(s_{\ell,h},a_{\ell,h})^\top\right]\succeq\frac{L}{C_\beta^\dagger}\Sigma_{\pi_\beta^*,h},$$

where $\Sigma_{\pi_\beta^*,h}\triangleq\mathbb{E}_{w\sim\beta}\mathbb{E}_{d_h^{\pi_w^*}}\left[\phi(s,a)\phi(s,a)^\top\right]$.

From the probability matching property of Thompson sampling method, we have

$$\mathbb{E}\left[\sum_{k=1}^{K}\phi(s_t,a_t)\phi(s_t,a_t)^\top\right]=K\Sigma_{\pi_\beta^*,h}.$$

From matrix concentration inequalities (Gittens & Tropp, 2011), with a probability $1-\xi$ where $\xi=\frac{d}{H}e^{-\frac{4(L+KC_\beta^\dagger)}{C_\beta^\dagger\kappa_\beta^2}}$, we have

$$\sum_{\ell=1}^{L}\phi(s_{\ell,h},a_{\ell,h})\phi(s_{\ell,h},a_{\ell,h})^\top+\sum_{k=1}^{K}\phi(s_t,a_t)\phi(s_t,a_t)^\top\succeq\frac{1}{2}(\frac{L}{C_\beta^\dagger}+K)\Sigma_{\pi_\beta^*}. \tag{16}$$

Here $\kappa_\beta=\max_{h\in[H]}\frac{\lambda_{\max}(\Sigma_{\pi_\beta^*,h})}{\lambda_{\min}(\Sigma_{\pi_\beta^*,h})}$ is the condition number for the feature matrix under expert policy π^*, $\lambda_{\max}$ is the largest eigenvalue and $\lambda_{\min}$ is the smallest *non-zero* eigenvalue. Let $\mathcal{E}$ be the event such that Equation (16) holds, then we have

$$\begin{aligned}
&\mathbb{E}_\beta[\Delta_k]\\
\leq&\mathbb{E}_\beta\left[\sum_{h=1}^{H}\Gamma_t\mathbb{E}_{\pi^*}\left[\sqrt{I_t(w_h;a_t^*,r_t,s_{t+1})}\right]\right]+2\delta H^2\\
\leq&\Gamma\sum_{h=1}^{H}\sqrt{\mathbb{E}_\beta\operatorname{Tr}\left(\Sigma_{\pi^*,h}^\top\Lambda_t^{-1}\right)}+2\delta H^2\\
\leq&\mathbb{1}_{\mathcal{E}}\left\{\Gamma\sum_{h=1}^{H}\sqrt{\mathbb{E}_\beta\operatorname{Tr}\left(\Sigma_{\pi^*,h}^\top\Lambda_t^{-1}\right)}\right\}+\xi H^2+2\delta H^2\\
\leq&\Gamma\sum_{h=1}^{H}\sqrt{\mathbb{E}_\beta\operatorname{Tr}\left(\Sigma_{\pi^*,h}^\top\big(\lambda\cdot I+\frac{1}{2}(\frac{L}{C_\beta^\dagger}+K)\Sigma_{\pi^*,h}\big)^{-1}\right)}+\xi H^2+2\delta H^2\\
\leq&\Gamma\sum_{h=1}^{H}\sqrt{\sum_{j=1}^{d}\frac{\lambda_j(h)}{\lambda+\frac{1}{2}(\frac{L}{C_\beta^\dagger}+K)\cdot\lambda_j(h)}}+(2\delta+\xi)H^2.
\end{aligned}$$

Here $\{\lambda_j(h)\}_{j=1}^d$ are the eigenvalues of $\Sigma_{\pi^*,h}$ for all $h\in[H]$. The first inequality follows from Lemma 15, and the second to last inequality follows from the Jensen inequality and the definition of event $\mathcal{E}$.

Meanwhile, by definition, we have $\|\phi(s,a)\|\leq 1$ for all $(s,a)\in\mathcal{S}\times\mathcal{A}$. By Jensen's inequality, we have

$$\|\Sigma_{\pi^*,h}\|_{\text{op}}\leq\mathbb{E}_{\pi^*}\left[\|\phi(s,a)\phi(s,a)^\top\|_{\text{op}}\right]\leq 1 \tag{17}$$

for all $h\in[H]$. As $\Sigma_{\pi^*,h}$ is positive semidefinite, we have $\lambda_j(h)\in[0,1]$ for all $h\in[H]$ and all $j\in[d]$.

Then we have

$$\begin{aligned}\mathbb{E}_\beta[\Delta_k] \leq & \Gamma \sum_{h=1}^{H} \sqrt{\sum_{j=1}^{d} \frac{1}{\lambda + \frac{1}{2}(\frac{L}{C_\beta^\dagger} + K)}} + (2\delta + \xi)H^2 \\ \leq & H\Gamma \sqrt{\frac{2d}{\frac{L}{C_\beta^\dagger} + K}} + (2\delta + \xi)H^2 \\ \leq & 2c \sqrt{\frac{2d^3 H^3 \iota}{\frac{L}{C_\beta^\dagger} + K}} + (2\delta + \xi)H^2,\end{aligned}$$

where $\iota = \log \frac{4dT}{\delta}$. For sufficiently large L and K such that $\xi = (\frac{L}{C_\beta^\dagger} + K)^{-1/2}K^{-1}$and let $\delta = (\frac{L}{C_\beta^\dagger} + K)^{-1/2}K^{-1}$. Using the fact that

$$\sum_{k=1}^{K} \sqrt{\frac{1}{a+bk}} \leq \int_0^K \sqrt{\frac{1}{a+bx}} dx \leq \frac{2}{b}(\sqrt{a+bK} - \sqrt{a}),$$

we have the desired result. □

C.3. Proof of Proposition 6

Proof. Let $T = 1$ in Theorem 5 and note that $\sqrt{x+1} - \sqrt{x} \leq 2/\sqrt{x}$, we have the result for Thompson sampling immediately. For an counterexample for UCB, Please refer to Lemma 10. □

C.4. Proof of Proposition 7

Proof. Let $N = 0$ in Theorem 5, and we have the result for Thompson sampling immediately. For an counterexample for LCB, Please refer to Lemma 11.

□

C.5. Failure of UCB and LCB

Lemma 10 (Failure of UCB)**.** *For any $\epsilon < 0.05$, $N \geq 500$, there exists a bandit problem with two arms such that for $\widehat{a}^{UCB} = \text{argmax}_a \widehat{r}(a) + k\sqrt{\frac{\log N}{N_a}}$, with $k > 0$, one has*

$$\mathbb{E}_{\mathcal{D}}[r(a^\star) - r(\widehat{a}^{UCB})] \geq \epsilon \tag{18}$$

Proof. The following proof is mainly adapted from Proposition 1 in Rashidinejad et al. (2021). Consider a two-arm bandit $\mathcal{A} = \{1, 2\}$, where $r(a_1) = 2\epsilon$, and

$$r(a_2) = \begin{cases} 2.1\epsilon & p = 0.5 \\ 0 & p = 0.5 \end{cases} \tag{19}$$

and the behavior policy for data collection satisfies $\mu(a_1) = (N-1)/N, \mu(a_2) = 1/N$. Then consider the following event $\mathcal{E} = \{N(a_2) = 1\}$, We have $P(\mathcal{E}) = (1 - 1/N)^{N-1}$. As long as $N \geq 500$, we have $P(\mathcal{E}) \geq 0.36$. Then with probability $p \geq 0.18$, we have $\widehat{r}(1) = 2\epsilon, \widehat{r}(2) = 2.1\epsilon$. Note that the bonus term for a_2 is larger than for a_1 sinee a_1 is pulled more than a_2, then we have $P(\widehat{a}_{\text{UCB}} = a_2) \geq p = 0.18$.

Finally, we have

$$\mathbb{E}_{\mathcal{D}}[r(a^\star) - r(\widehat{a}_{\text{UCB}})] \geq 0.95\epsilon \cdot p \geq 0.1\epsilon. \tag{20}$$

□

Lemma 11 (Failure of LCB). *For any $\epsilon < 0.05$, $N \geq 500$, there exists a bandit problem with two arms such that for $\widehat{a}_{LCB} = \text{argmax}_a \widehat{r}(a) - k\sqrt{\frac{\log N}{N_a}}$, with $k > 0$, one has*

$$\mathbb{E}_{\mathcal{D}}\left[\sum_{t=1}^{T}(r(a^\star) - r(\widehat{a}_t^{LCB}))\right] \geq 0.1\epsilon \cdot T \tag{21}$$

Proof. Similar to Lemma 10, consider a two-arm bandit $\mathcal{A} = \{1, 2\}$, where $r(a_1) = \epsilon$, and

$$r(a_2) = \begin{cases} 4\epsilon & p = 0.5 \\ 0 & p = 0.5 \end{cases} \tag{22}$$

and the behavior policy for data collection satisfies $\mu(a_1) = (N-1)/N, \mu(a_2) = 1/N$. Then consider the following event $\mathcal{E} = \{N(a_2) = 1\}$, We have $P(\mathcal{E}) = (1 - 1/N)^{N-1}$. As long as $N \geq 500$, we have $P(\mathcal{E}) \geq 0.36$. Then with probability $p \geq 0.18$, we have $\widehat{r}(1) = \epsilon, \widehat{r}(2) = 0$. Note that the bonus term for a_2 is larger than for a_1 sinee a_1 is pulled more than a_2, then for any t, we have $P(\widehat{a}^{\text{LCB}} = a_1) \geq p = 0.18$. Note that pulling a_1 does not obtain any information, we can conclude that $P(\widehat{a}_t^{\text{LCB}} = a_1) \geq p = 0.18$ for any t.

Finally, we have

$$\mathbb{E}_{\mathcal{D}}\left[\sum_{t=1}^{T}(r(a^\star) - r(\widehat{a}_t^{\text{LCB}}))\right] \geq \epsilon \cdot p \cdot T \geq 0.1\epsilon \cdot T \tag{23}$$

□

D. Auxiliary Lemmas

Lemma 12 (Regret Decomposition (Cai et al., 2020)). *We define the model prediction error as*

$$\iota_{k,h}(s,a) = r_{k,h}(s,a) + (\mathbb{B}_h V_{k,h+1})(s,a) - Q_{k,h+1}(s,a), \tag{24}$$

which arises from estimating $\mathbb{P}_h V_{h+1}^k$ in the Bellman equation based on only finite historical data. Also, we define the following filtration generated by the state-action sequence and reward functions.

Definition 13 (Filtration). *For any $(t) \in [K] \times [H]$, we define $\mathcal{F}_{t,1}$ as the σ-algebra generated by the following state-action sequence and reward functions,*

$$\{(s_{\tau,i}, a_{\tau,i})\}_{(\tau,i)\in[k-1]\times[H]} \cup \{r^\tau\}_{\tau\in[k]} \cup \{(s_{k,h}, a_{k,h})\}_{i\in[h]},$$

and $\mathcal{F}_{t,2}$ as the σ-algebra generated by

$$\{(s_{\tau,i}, a_{\tau,i})\}_{(\tau,i)\in[k-1]\times[H]} \cup \{r^\tau\}_{\tau\in[k]} \cup \{(s_{k,h}, a_{k,h})\}_{i\in[h]} \cup \{s_{h+1}^k\},$$

where, for the simplicity of discussion, we define s_{H+1}^k as a null state for any $k \in [K]$.

It holds that

$$\begin{aligned} \textit{Regret}(T) &= \sum_{k=1}^{K}\big(V_1^{\pi^*,k}(s_1^k) - V_1^{\pi^k,k}(s_1^k)\big) \\ &= \sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}_{\pi^*}\left[\langle Q_h^k(s_h,\cdot), \pi_h^*(\cdot \mid s_h) - \pi_{k,h}(\cdot \mid s_h)\rangle\right] + \mathcal{M}_{K,H,2} \\ &\quad + \sum_{k=1}^{K}\sum_{h=1}^{H}\big(\mathbb{E}_{\pi^*}[\iota_h^k(s_h,a_h)] - \iota_h^k(s_{k,h},a_{k,h})\big). \end{aligned} \tag{25}$$

Here $\{\mathcal{M}_{t,m}\}_{(t,m)\in[K]\times[H]\times[2]}$ is a martingale adapted to the filtration $\{\mathcal{F}_{t,m}\}_{(t,m)\in[K]\times[H]\times[2]}$.

Proof. See Lemma 4.2 in Cai et al. (2020) for a detailed proof. □

Lemma 14 (Mutual Information in Linear MDP)**.** *It hold that*

$$I_t(w_h; a_t, r_t, s_{t+1}|\mathcal{D}) = \frac{1}{2}\log\left(1 + \phi(s_t, a_t)^\top \Lambda_t^{-1}\phi(s_t, a_t)\right).$$

Proof. Let the prior be $w_h \sim \mathcal{N}(0, \lambda \cdot I)$, then we have the following closed form posterior

$$w_h|\mathcal{D} \sim \mathcal{N}(\widehat{w}_h, \Lambda_t^{-1}),$$

where

$$\widehat{w}_h = \Lambda_h^{-1}(\sum_{k=1}^{K} \phi(s_t, a_t) \cdot (r_t + \widehat{V}_{h+1}(s_{t+1}))),$$
$$\Lambda_h = \sum_{k=1}^{K} \phi(s_t, a_t)\phi(s_t, a_t)^\top + \lambda \cdot I.$$

Note that this is equivalent to the regularized least-square solution for linear MDPs (Jin et al., 2021). Then we have

$$\begin{aligned}
I_t(w_h; a_t, r_t, s_{t+1}|\mathcal{D}) &= H(w_h|\mathcal{D}) - H(w_h|\mathcal{D} \cup \{(r_t, a_t, s_{t+1})\}) \\
&= \frac{1}{2}\log\frac{\det(\Lambda_t^\dagger)}{\det(\Lambda_t)} \\
&= \frac{1}{2}\log\det(I + \Lambda_t^{-1/2}\phi(s_t, a_t)\phi(s_t, a_t)^\top \Lambda_t^{-1/2}) \\
&= \frac{1}{2}\log\left(1 + \phi(s_t, a_t)^\top \Lambda_t^{-1}\phi(s_t, a_t)\right).
\end{aligned}$$

where $\Lambda_t^\dagger = \Lambda_t + \phi(s_h, a_h)\phi(s_h, a_h)^\top$. □

Lemma 15. *Under linear MDP, we have*

$$\mathbb{P}_k\left(\left|Q_{t,w}(s,a) - \bar{Q}_{t,w}(s,a)\right| \le \frac{\Gamma_t}{2}\sqrt{I_t(w_h; r_{t,a}, s_{t+1,a})}, \forall h \in [H], s \in \mathcal{S}, a \in \mathcal{A}\right) \ge 1 - \frac{\delta}{2}$$

With $\Gamma_t \equiv \Gamma = 2cHd\sqrt{\log\frac{4dT}{\delta}}$*, where* c *is an absolute constant and* $\bar{Q}_{t,w}(s,a) = r_{h,w}(s,a) + \mathbb{P}_h V_{h+1,w}(s,a)$*.*

Proof. Following a similar argument in Lemma 5.2 in Jin et al. (2021), We have with probability $1 - \delta$,

$$\begin{aligned}
&\left|Q_{t,w}(s,a) - \bar{Q}_{t,w}(s,a)\right| \\
\le&\beta\sqrt{\phi(s,a)\Lambda_h^{-1}\phi(s,a)} \\
\le&\beta\sqrt{\log(1 + \phi(s,a)\Lambda_h^{-1}\phi(s,a)) \cdot \frac{\phi(s,a)\Lambda_h^{-1}\phi(s,a)}{\log(1 + \phi(s,a)\Lambda_h^{-1}\phi(s,a))}} \\
\le&\beta\sqrt{2\log(1 + \phi(s,a)\Lambda_h^{-1}\phi(s,a))} \\
\le&2\beta\sqrt{I_t(w_h; r_{t,a}, s_{t+1,a})},
\end{aligned}$$

where $\beta = cHd\sqrt{\log\frac{4dT}{\delta}}$ and Λ_h is defined as in Lemma 14. The last inequality use the fact that $\phi(s,a)\Lambda_h^{-1}\phi(s,a) \le 1$ and $2\log(1+x) \ge x$ for $x \in [0,1]$. The last step follows from Lemma 14.

□

E. Experiments on Bernoulli Bandits

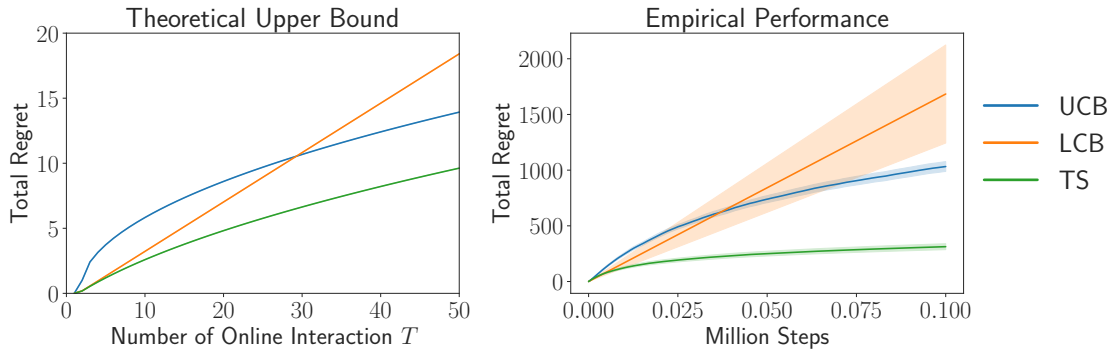
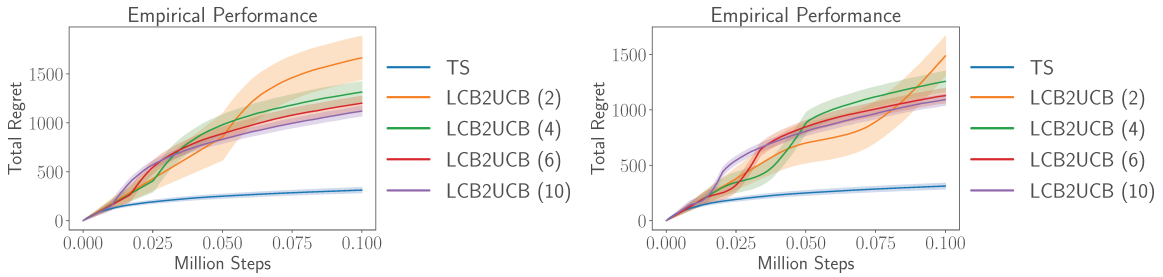


Figure 4: Theoretical upper bounds in Theorem 5 and experiment result on the Bernoulli bandit.



(a) Hard switch of LCB to UCB

(b) Soft switch of LCB to UCB

Figure 5: Performance of different switch schemes from LCB to UCB on the Bernoulli bandit. It incurs a large regret to switch from pessimism to optimism regardless of interpolation schemes. x in `LCB2UCB` (x) represents the switch parameter.

Experiment Setup. We consider a didactic Bernoulli bandit, where each arm pull yielded a binary reward based on probability p_i. the parameters p_n are i.i.d. drawn from a Beta distribution for all arms a_i. The objective was to identify the arm associated with the maximum probability, $p^* = \max\{p_i\}_{i=1}^n$. We use a uniform policy to collect the offline data. We use 10 arms, 1000 offline data points and the online phase lasts for 100000 steps.

Baseline Setup. We use both offline and online data to calculate the statistics (i.e., mean of the reward $\widehat{\mu}$ and the corresponding confidence bound $\widehat{\sigma}$). Then Upper Confidence Bound (UCB) policy, π_{UCB} select arms by $\text{argmax}_i\, \widehat{\mu}(a_i) + \widehat{\sigma}(a_i)$, while the Lower Confidence Bound (LCB) policy, π_{LCB} select arms by $\text{argmax}_i\, \widehat{\mu}(a_i) - \widehat{\sigma}(a_i)$.

We also consider two switching scheme that switches from LCB to UCB. The soft switch interpolates the confidence weight by $k_t = \min\{At/T - 1, 1\}$ where A is a parameter. Then it select the arm by $\pi^t_{\text{soft}} = \text{argmax}_i\, \widehat{\mu}(a_i) + k_t \cdot \widehat{\sigma}(a_i)$. The hard switch interpolates the confidence weight by $k'_t = 2 * \mathbb{1}\{t \geq T/B\} - 1$ where B is the parameter. Then it select the arm by $\pi^t_{\text{hard}} = \text{argmax}_i\, \widehat{\mu}(a_i) + k'_t \cdot \widehat{\sigma}(a_i)$.

Results. The results are shown in Figure 4 and Figure 5, respectively. We can see in Figure 4 that UCB performs badly during the initial stage (i.e., the regret increases quickly), while LCB suffers from a linear regret in the long run. Bayesian agents, on the contrary, enjoys guarantees from the both world, which performs well at initial stage while achieving a sublinear regret. From Figure 5, we can see that naively switches from LCB to UCB leads to suboptimal performance regardless of the interpolation scheme. It suffers from a sudden increase in regret the first time pessimism is switched to optimism. On the contrary, Bayes-based methods enjoys a smooth regret curve.

F. Complete Experimental Results

The results in Figure 6 show TD3+BC exhibits safe but slow performance improvement, resulting in worse asymptotic performance. On the other hand, TD3 suffers from initial performance degradation, especially in narrow distribution datasets (e.g., expert datasets). Differently, BOORL attains a fast performance improvement with a smaller regret. Due to the offline bootstrap, the initial performance in the online phase between BOORL and baselines exits a small difference, while it does not change the conclusion.

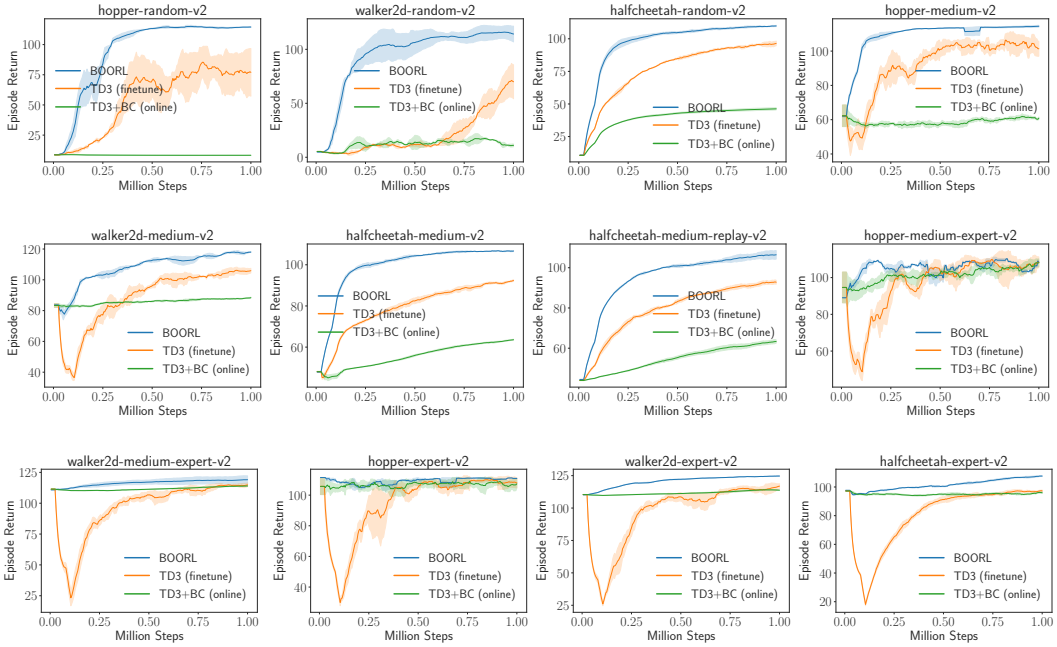


Figure 6: Comparison between BOORL and baselines in the finetune phase. We adopt datasets of various quality for offline training and then load same pre-trained weight for online learning. We adopt normalized score metric averaged with five random seeds.

G. Comparison with PEX

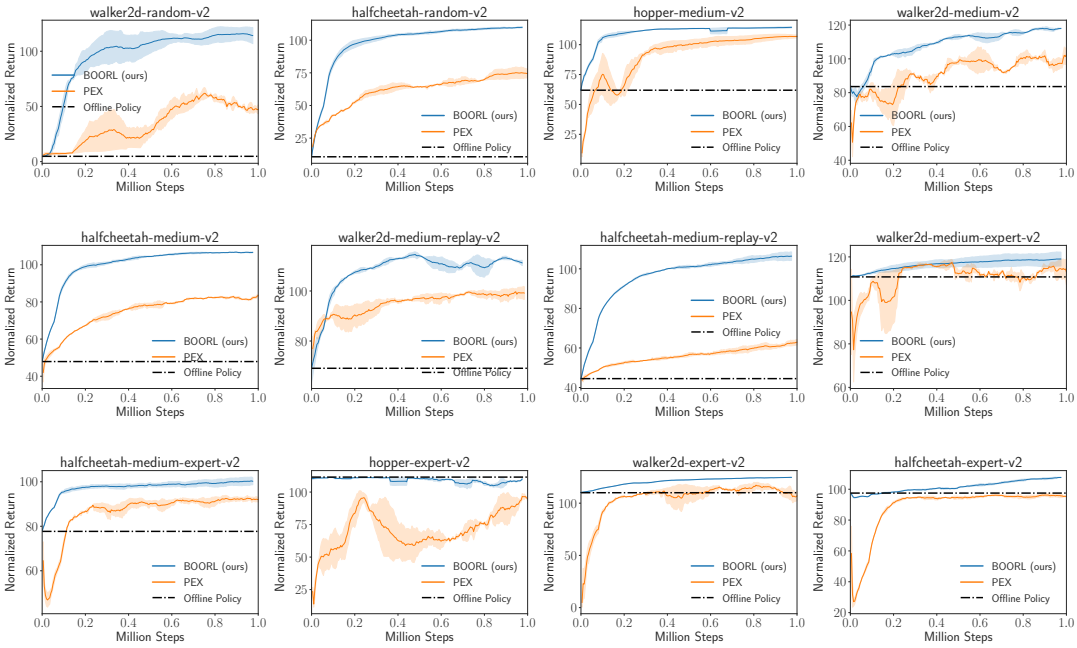


Figure 7: Comparison between BOORL and PEX in the finetune phase. We adopt datasets of various quality for offline training and then load same pre-trained weight for online learning. We adopt normalized score metric averaged with five random seeds.

H. Additional Experiments

H.1. Ablation for BOORL

We aim to understand the behavior of BOORL by performing ablation studies. (1) We store the offline and online data into the same buffer for uniform sampling, named BOORL (Uniform Buffer). (2) We set the Ensemble Number to 1 to investigate the effect of the Thompson Sampling, named BOORL (Ensemble Num=1). The experimental results in Figure 8 show that each module is essential to the superior performance of our algorithm. In the ablation studies, we use TD3+BC and TD3 as the backbone of offline and online algorithms.

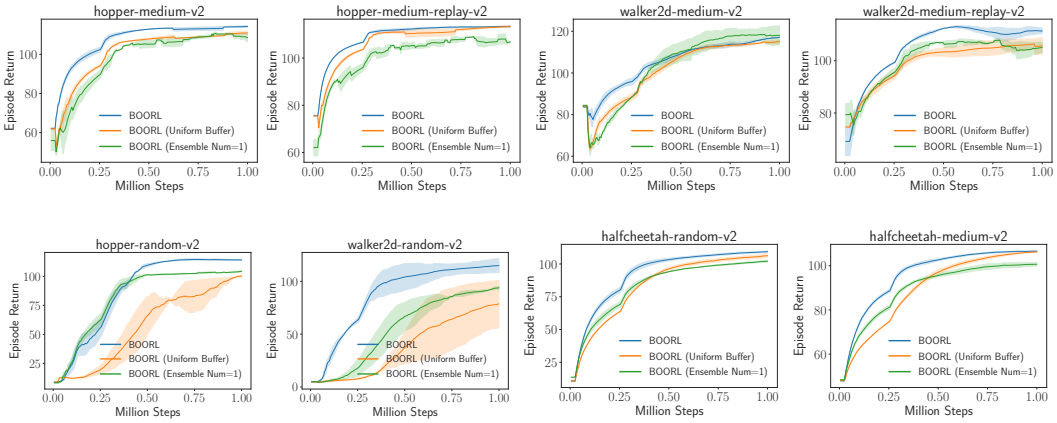


Figure 8: Module ablation study of BOORL.

H.2. Ablation for mask ratio

In addition, we conduct ablation studies for mask ratio p. The experimental results in Figure 9 show that the performance of our algorithm is robust to the changes of p. Similar results are also found in (Osband et al., 2016). Therefore, we select the uniform parameter $p = 0.9$ in all experiments.

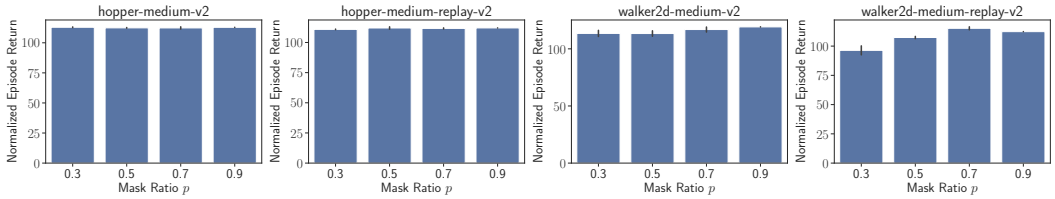


Figure 9: The performance comparison between various mask ratios p.

H.3. Ablation for computational overhead

We aim to provide a comparison between the computational overhead of using an ensemble versus not using an ensemble in our approach. Specifically, we train our method with various ensemble sizes N on the same computational device (GeForce RTX 3090 GPU). The time required to complete 1M training is shown in Table 4. Since we adopt the multi-head structure and share part of the network, the computational overhead does not increase significantly as the number of N increases.

Ensemble Size	1	5	10	20
Computational Overhead	2.5 h	2.7 h	2.9 h	3.2 h

Table 4: Ablation results for computational overhead.

H.4. Additional substantiation to the drop in performance

Prior knowledge is encoded in the offline learned value function and the learned policy function. Mismanagement of prior knowledge is mainly influenced by two factors: a sudden change in the replay data distribution and a sudden change in the loss function.

we conducted additional experiments about the performance drop. We first gradually increase the proportion of the offline data in the training dataset from 15% to 50% to validate the effect of the sudden change in the replay data distribution. A smaller proportion indicates a more sudden change in the replay data distribution. We report the difference between the initial and minimum performance of the performance drop. Larger values represent a more severe performance drop. The experimental results in Table 5 indicate that a smaller proportion of the offline data leads to a more severe performance drop. This suggests that sudden changes in replay data distribution exacerbate performance drop.

In addition, we use various update intervals of the value function network to show the effect of the sudden change of the value function. Specifically, we vary the update interval of the value function network from 1 to 16. A larger update interval indicates a slower change in the value function. The experimental results in Table 6 demonstrate that the slow update of the value function alleviates the performance drop. This suggests that sudden changes in the value function also exacerbate the performance drop.

Proportion	15%	30%	35%	45%	50%
Hopper-medium	20.8±3.3	14.1±2.7	13.8±2.2	7.7±1.6	4.9±1.3
Walker2d-medium	40.1±2.9	32.7±3.1	28.7±3.8	26.9±3.5	21.2±3.7
Halfcheetah-medium	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
Hopper-medium-replay	23.7±1.6	15.5±2.2	14.8±2.5	13.9±1.3	13.2±1.1
Walker2d-medium-replay	25.5±3.7	20.3±2.3	19.0±3.5	18.2±2.7	16.7±3.6
Halfcheetah-medium-replay	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0

Table 5: The effect of the replay data distribution on the performance drop. We adopt the normalized score metric, averaging numbers across five seeds.

Update Interval	1	2	4	8	16
Hopper-medium	4.9±1.3	4.5±1.0	3.8±0.6	3.7±0.7	3.5±0.4
Walker2d-medium	21.2±3.7	18.2±2.5	17.9±3.6	16.2±2.7	15.4±3.2
Halfcheetah-medium	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
Hopper-medium-replay	13.2±1.1	12.6±2.4	8.7±1.6	0.0±0.0	0.0±0.0
Walker2d-medium-replay	16.7±3.6	16.2±2.5	12.8±3.3	10.1±2.0	6.9±1.6
Halfcheetah-medium-replay	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0

Table 6: The effect of the value function on the performance drop. We adopt the normalized score metric, averaging numbers across five seeds.

I. Experimental Details

Experimental Setting. For TD3+BC (online) and TD3 (finetune), we first load the offline dataset into the online replay buffer and add the online collected data into the buffer. Then, we uniformly sample data to train from the online buffer.

Hyper-parameters. We adopt the TD3+BC and TD3 as the backbone of offline and online algorithms. Therefore, we build BOORL based on the code of the TD3+BC. We outline the hyper-parameters used by BOORL in Table 7.

Hyperparameter	Value
Optimizer	Adam
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)
TD3+BC parameter α	2.5
IQL parameter	0.9
Architecture	**Value**
Critic hidden dim	256
Critic hidden layers	2
Critic activation function	ReLU
Actor hidden dim	256
Actor hidden layers	2
Actor activation function	ReLU
BOORL Parameters	**Value**
Mask ratio p	0.9
Ensemble Number	5
UTD ratio G	5

Table 7: Hyper-parameters sheet of BOORL.

Baselines Implementation. We adopt the author-provided implementations from GitHub for TD3 [1], TD3+BC [2], CQL [3], IQL [4], Off2On [5], ODT [6], PEX [7] and Cal-QL [8]. We use the official implementation in the author-provided code for TD3+BC (online) and IQL (online). All experiments are conducted on the same experimental setup, a single GeForce RTX 3090 GPU and an Intel Core i7-6700k CPU at 4.00GHz.

[1] https://github.com/sfujim/TD3
[2] https://github.com/sfujim/TD3_BC
[3] https://github.com/aviralkumar2907/CQL
[4] https://github.com/ikostrikov/implicit_q_learning
[5] https://github.com/shlee94/Off2OnRL
[6] https://github.com/facebookresearch/online-dt
[7] https://github.com/Haichao-Zhang/PEX
[8] https://github.com/nakamotoo/Cal-QL