MOORL: A Framework for Integrating Offline-Online Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

Sample efficiency and exploration remain critical challenges in Deep Reinforcement Learning (DRL), particularly in complex domains. Offline RL, which enables agents to learn optimal policies from static, pre-collected datasets, has emerged as a promising alternative. However, offline RL is constrained by issues such as out-of-distribution (OOD) actions that limit policy performance and generalization. To overcome these limitations, we propose Meta Offline-Online Reinforcement Learning (MOORL), a hybrid framework that unifies offline and online RL for efficient and scalable learning. While previous hybrid methods rely on extensive design choices and added complexity to utilize offline data effectively, MOORL introduces a meta-policy that seamlessly adapts across offline and online trajectories. This enables the agent to leverage offline data for robust initialization while utilizing online interactions to drive efficient exploration. Importantly, MOORL addresses the key challenges of hybrid RL in terms of being design-free. Our theoretical analysis demonstrates that the hybrid approach enhances exploration by effectively combining the complementary strengths of offline and online data. Furthermore, we demonstrate that MOORL learns a stable Q-function without relying on extensive design choices. Extensive experiments on 28 tasks from the D4RL and V-D4RL benchmarks validate its effectiveness, showing consistent improvements over state-of-the-art offline and hybrid RL baselines. With minimal computational overhead, MOORL achieves strong performance, underscoring its potential for practical applications in real-world scenarios.

1 Introduction

Deep reinforcement learning (DRL) has been tremendously successful in solving a variety of complex problems, including robotics (Tang et al., 2024), autonomous driving (Kiran et al., 2021), healthcare (Yu et al., 2021), game-playing (Silver et al., 2017; Vinyals et al., 2019), intelligent perception system (Chaudhary et al., 2023), and finance (Charpentier et al., 2021). However, one of the primary drawbacks of DRL algorithms is their sample inefficiency, i.e., the number of state-action-state transition samples they require to train a policy. Typically, these algorithms require millions of such interactions, making them impractical in real-world scenarios, particularly in safety-critical domains like robotics and autonomous driving (Kiran et al., 2021). Learning policies in controlled simulation environments can offer a partial remedy, as these policies often fail to generalize to real-world situations due to the well-known simulation-to-reality (sim2real) gap (Tobin et al., 2017).

An effective strategy to mitigate these problems is Offline Reinforcement Learning (RL) (Levine et al., 2020), which enables policy learning from historical datasets without requiring online exploration. Offline RL can train policies safely and cost-effectively using pre-collected human demonstrations or previously logged interactions. Although it alleviates some concerns related to sample complexity, the reliance on static offline data introduces new challenges, such as extrapolation errors and out-of-distribution (OOD) actions (Bai et al., 2022), which can result in sub-optimal behaviors when policies are tested in real environments or unfamiliar states (Kim et al., 2024). Offline-to-Online (O2O) RL (Lee et al., 2022; Wagenmaker & Pacchiano, 2023) dilute the limitations of purely offline RL to some extent. O2O RL setups typically pre-train the agent using offline data, followed by fine-tuning via limited online interactions. However, this approach often experiences performance drops due to compounded Bellman errors (Sun et al., 2023)due to changes in reward distributions and distributional shifts between offline data and online interaction (Farahmand et al., 2010; Munos, 2005).

In this article, we aim to tackle inherent challenges faced by offline RL and online RL algorithms. We believe directly integrating offline data into online RL training can lead to more stable learning and mitigate issues such as out-of-distribution (OOD) actions and inefficient exploration. While recent efforts, such as RLPD (Ball et al., 2023) and Hy-Q (Song et al., 2022), have attempted to address offline-online integration, each comes with its own challenges. RLPD, which combines offline and online data, requires tailored design parameters and careful task-specific tuning. RLPD employs a large Q-ensemble and a high Update-to-Data (UTD) ratio to stabilize learning and optimize performance. These dependencies add complexity to the algorithm, making it more challenging and compute-intensive, limiting its scalability.

On the other hand, Hy-Q offers a more streamlined approach by integrating offline data into online training without necessitating extensive task-specific design choices. However, Hy-Q has its own limitations; it requires maintaining separate Q-value functions for each timestep within a fixed horizon, significantly increasing computational and memory overhead, especially in environments with longer or variable horizons. Additionally, Hy-Q's reliance on a predefined horizon makes it less adaptable to tasks with dynamic episode lengths, limiting its scalability and flexibility across diverse RL scenarios. While Hy-Q reduces the need for task-specific tuning compared to RLPD, it still struggles with computational efficiency and adaptability in more complex or heterogeneous environments.

Further, as highlighted by (Furuta et al., 2021; Engstrom et al., 2019; Henderson et al., 2018), the RL algorithms are difficult to optimize and tune where minor hyperparameter changes can have a non-trivial impact on performance we believe it is important to limit RL algorithm design choices. These limitations underscore the necessity for a hybrid approach that effectively integrates offline and online data and maintains computational efficiency and adaptability across various tasks. With this motivation in this work, we propose a framework called Meta Offline-Online RL (MOORL) that addresses the stated issues by utilizing a unified set of design choices without the need for a large Q-ensemble, high UTD, and separate a Q-function per horizon step, offering a more robust and generalizable solution for hybrid reinforcement learning. In particular, our contributions can be summarized as follows.

- We provide theoretical insights showing that mixing offline and online data acts as a regularizer, improving stability and policy learning efficiency.
- We leverage the off-policy RL framework, Soft-Actor-Critic (SAC) (Haarnoja et al., 2018), to seamlessly integrate offline and online data via meta-learning for efficient design-free policy learning without introducing any new hyperparameters.
- Our proposed framework, MOORL, uses meta-learning principles (Finn et al., 2017; Nichol & Schulman, 2018) to train policies under a single meta-objective, enabling the dynamic balancing of offline and online data. The learned meta-policy adapts across varying distributions, minimizing the impact of distributional shifts and extrapolation errors.
- We validate our methodology through 28 comprehensive experiments on benchmark D4RL (Fu et al., 2020) and V-D4RL (Lu et al., 2022) environments, demonstrating that MOORL outperforms state-of-the-art methods in reward accumulation while being stable across diverse tasks, including dense and sparse reward scenario.

2 Preliminaries

2.1 Markov Decision Process

We consider a Markov Decision Process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, T, H, R, \gamma, \rho \rangle$, where $\mathcal{S}$ indicates the state space, $\mathcal{A}$ denotes the action space, $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ represents the state transition dynamics (where $\Delta(\mathcal{S})$ defines the collection of all probability distributions over $\mathcal{S}$), $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, $\gamma \in (0, 1)$ is the discount factor, H is the length of the horizon of the episodes, and $\rho \in \Delta(\mathcal{S})$ is the initial state distribution.

At each time instant t, the agent observes the state s_t, executes an action a_t, and as a result, transitions to the next state $s_{t+1} \sim T(\cdot|s_t, a_t)$, and receives a reward $r_t = R(s_t, a_t, s_{t+1})$. The goal in reinforcement learning is to learn a (stochastic stationary) policy $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ that maximizes the expected cumulative reward $J_\pi = \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t r_t \mid \pi, \rho\right]$ where the expectation is obtained over π-induced trajectories of length H that start from the initial state distribution, ρ. The state-value and state-action value functions are defined, respectively as $V_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t r_t \mid s_0 = s, \pi\right]$ and $Q_\pi(s, a) = \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t r_t \mid s_0 = s, a_0 = a, \pi\right]$, where the expectations are obtained over π-induced trajectories of length H. Note that $J_\pi = \mathbb{E}_{s\sim\rho}\left[V_\pi(s)\right]$. In this paper, we obtain the optimal policy that maximizes J_π by combining offline and online data using ideas from meta-reinforcement learning, which is described below.

2.2 Meta-Reinforcement Learning

Meta-RL aims to solve a distribution of tasks given by $\mathcal{P}_{\mathcal{T}}(\cdot)$, rather than a single fixed task, where each task is characterized by an MDP $\mathcal{M}_i = \langle \mathcal{S}_i, \mathcal{A}_i, T_i, H_i, R_i, \gamma_i, \rho_i \rangle$. In our setting, the meta-learning process is used to combine offline and online data. Following the meta-RL paradigm, we maintain separate replay buffers for each type of data, denoted as $\mathcal{D}_{\text{offline}}$ and $\mathcal{D}_{\text{online}}$ respectively, which store transition tuples in the form (s_i, a_i, r_i, s'_i). The training process alternates between sampling from these replay buffers to update the policy while adapting to the changing data as the agent gathers more online experiences. A meta-episode consists of sampling data from either of these replay buffers and forming the associated trajectories to update the meta policy. A significant challenge arises from the distributional shift between offline and online data, which can lead to instability in the training process. To mitigate this issue, our meta-RL approach employs gradient-based meta-learning to effectively balance updates between the offline and online data sources, enhancing the robustness of the policy against distribution mismatches.

3 Why Meta Learning?

Meta-learning, or "learning to learn," is a powerful paradigm that performs well across diverse task distributions (Finn et al., 2017; Nichol & Schulman, 2018). By leveraging meta-learning to integrate offline and online data, Meta Offline Reinforcement Learning (MOORL) can achieve the following benefits:

- **Reduced Extrapolation Error and Improved Distribution Generalization:** MOORL adapts the policy to changing data distributions by training a meta-policy over online and offline data. This approach minimizes extrapolation errors by optimizing the meta-policy to generalize across distributions rather than being confined to a single dataset.
- **Balanced Integration of Expert and Agent Behaviors:** Rather than forcing the agent to mimic the expert purely, MOORL learns a meta-policy that harmonizes both expert (offline) and agent (online) behaviors. This flexibility reduces over-reliance on expert data, enabling effective exploration while leveraging expert guidance.
- **Improved Credit Assignment:** MOORL utilizes a meta-objective that optimizes task performance across various data sources, helping isolate beneficial behaviors. By employing gradient-based meta-learning techniques, the meta-policy can assign credit more accurately, focusing on the trajectory aspects that generalize well between online and offline data.
- **Simplified Algorithmic Complexity:** MOORL streamlines the learning process by employing a single meta-objective that integrates agent and expert data without extensive design choices. This meta-objective is robust across varying distributions, reducing the need for task-specific hyperparameter tuning and automating many design decisions that would require manual adjustments.

4 Integrating Online RL with Offline Data

This work aims to bridge the gap between online and offline reinforcement learning by presenting a unified approach that integrates both paradigms without requiring additional hyperparameter tuning or introducing new design elements. Our method utilizes an off-policy reinforcement learning algorithm (Haarnoja et al., 2018) to leverage data from various distributions effectively. The proposed framework, MOORL, is designed to handle variations in offline data quality, making it adaptable to a wide range of scenarios. Additionally, MOORL demonstrates consistent performance across different problem settings, including environments with state-based or pixel-based observations, as well as those with dense, sparse, or even binary rewards. To support this, we first offer theoretical insights into why combining offline expert data with online agent data may be a more effective strategy than relying solely on online learning. We then introduce the MOORL framework and highlight its independence from specific design constraints (refer to Appendix C).

4.1 Impact of Mixing Offline and Online Data

Consider an RL setup where the agent is exposed to online and offline data. Below, we analyze a performance bound of the expected reward when the replay buffer is composed of a mixture of offline and online data.

4.1.1 Problem Setup

Let π be the current (online) policy of the agent, μ be an offline policy that generated the offline data, and $\mathcal{D}$ be a dataset of trajectories, λ fraction of those being generated by the offline policy, μ, and the rest being generated via π. Let $d^\pi(\cdot,\cdot), d^\mu(\cdot,\cdot)$ denote the state-action distributions generated by π, μ respectively. Also, let $d^{\mathcal{D}}(\cdot,\cdot)$ be the state-action distribution generated by a randomly chosen trajectory taken from $\mathcal{D}$. Clearly,

$$d^{\mathcal{D}}(s,a) = \lambda \cdot d^\mu(s,a) + (1-\lambda) \cdot d^\pi(s,a). \tag{1}$$

4.1.2 Expected Reward and its Performance Bound

The expected reward over trajectories τ sampled from the mixed dataset $\mathcal{D}$ is defined as:

$$\mathbb{E}_{\tau\sim\mathcal{D}}[R(\tau)] = \sum_{(s,a)} d^{\mathcal{D}}(s,a) \cdot r(s,a). \tag{2}$$

where $R(\tau) = \sum_{t=0}^{H-1} r(s_t,a_t)$ is the cumulative reward over a trajectory $\tau = (s_0, a_0, \ldots, s_{H-1}, a_{H-1})$, and $r(s,a)$ is the immediate reward at state-action pair (s,a). We can express the expected reward using linearity of expectation and definition of $d^{\mathcal{D}}$:

$$\sum_{(s,a)} d^{\mathcal{D}}(s,a) Q^\pi(s,a) = \lambda \sum_{(s,a)} d^\mu(s,a) Q^\pi(s,a) + (1-\lambda) \sum_{(s,a)} d^\pi(s,a) Q^\pi(s,a). \tag{3}$$

where $Q^\pi(s,a)$ is the action-value function under the online policy π.

The KL divergence (Kullback & Leibler, 1951) between online policy π and offline policy μ state-action visitation distribution can be defined as:

$$\mathbb{D}_{\mathrm{KL}}(d^\pi \parallel d^\mu) = \sum_{(s,a)} d^\pi(s,a) \log \frac{d^\pi(s,a)}{d^\mu(s,a)}. \tag{4}$$

Rearranging Equation 4, we use the property of logarithms, $\log \frac{d^\pi(s,a)}{d^\mu(s,a)} = \log d^\pi(s,a) - \log d^\mu(s,a)$, to write:

$$\mathbb{D}_{\mathrm{KL}}(d^\pi \parallel d^\mu) = \sum_{(s,a)} d^\pi(s,a) \log d^\pi(s,a) - \sum_{(s,a)} d^\pi(s,a) \log d^\mu(s,a). \tag{5}$$

Rearranging to isolate $\sum_{(s,a)} d^\pi(s,a) \log d^\pi(s,a)$ gives:

$$\sum_{(s,a)} d^\pi(s,a) \log d^\pi(s,a) = \sum_{(s,a)} d^\pi(s,a) \log d^\mu(s,a) + \mathbb{D}_{\mathrm{KL}}(d^\pi \parallel d^\mu). \tag{6}$$

Since $\mathbb{D}_{\mathrm{KL}}(d^\pi \parallel d^\mu) \geq 0$ (non-negativity of KL divergence). Dropping this non-negative term leads to:

$$\sum_{(s,a)} d^\pi(s,a) \log d^\pi(s,a) \geq \sum_{(s,a)} d^\pi(s,a) \log d^\mu(s,a) - \mathbb{D}_{\mathrm{KL}}(d^\pi \parallel d^\mu). \tag{7}$$

To relate $d^\mu(s,a)$ and $d^\pi(s,a)$, note that $d^\mu(s,a) = d^\pi(s,a) \cdot \exp(-\log \frac{d^\pi(s,a)}{d^\mu(s,a)})$. Using this relationship for $d^\mu(s,a)$:

$$\sum_{(s,a)} d^\mu(s,a) Q^\pi(s,a) = \sum_{(s,a)} \left[d^\pi(s,a) \cdot \exp\left(-\log \frac{d^\pi(s,a)}{d^\mu(s,a)} \right) \right] Q^\pi(s,a). \tag{8}$$

The term $\exp\left(-\log \frac{d^\pi(s,a)}{d^\mu(s,a)}\right)$ penalizes the contribution of $Q^\pi(s,a)$ for state-action pairs where $d^\pi(s,a)$ diverges significantly from $d^\mu(s,a)$.

Since the distribution will consist of two subsets of state-action space:

$S_1 = \{(s,a) | d^\pi(s,a) \geq d^\mu(s,a)\}$,

$S_2 = \{(s,a) | d^\pi(s,a) < d^\mu(s,a)\}$,

These subsets partition the state-action space, i.e., $S = S_1 \cup S_2$, and are disjoint, $S_1 \cap S_2 = \emptyset$ with both subsets $S_1 \neq \emptyset$ and $S_2 \neq \emptyset$.

In S_1, where $d^\pi(s,a) \geq d^\mu(s,a)$:

$$\log \frac{d^\pi(s,a)}{d^\mu(s,a)} \geq 0 \quad \text{and} \quad \exp\left(-\log \frac{d^\pi(s,a)}{d^\mu(s,a)} \right) \leq 1. \tag{9}$$

Thus:

$$\sum_{(s,a) \in S_1} d^\mu(s,a) Q^\pi(s,a) \leq \sum_{(s,a) \in S_1} d^\pi(s,a) Q^\pi(s,a). \tag{10}$$

For an entropy-regularized policy, from equation 7 and the above relationship, a tighter bound can be defined as:

$$\sum_{(s,a) \in S_1} d^\mu(s,a) Q^\pi(s,a) \leq \sum_{(s,a) \in S_1} d^\pi(s,a) Q^\pi(s,a) - \mathbb{D}_{\mathrm{KL}}^{S_1}(d^\pi \parallel d^\mu). \tag{11}$$

Here, the KL-divergence is computed after normalization, and the given KL-divergence term implicitly accounts for the subset's normalization process. From equation 3 and equation 11:

$$\sum_{(s,a) \in S_1} d^{\mathcal{D}} Q^\pi(s,a) \leq \sum_{(s,a) \in S_1} d^\pi(s,a) Q^\pi(s,a) - \lambda \cdot \mathbb{D}_{\mathrm{KL}}^{S_1}(d^\pi || d^\mu). \tag{12}$$

In S_2, where $d^\pi(s,a) < d^\mu(s,a)$:

$$\log \frac{d^\pi(s,a)}{d^\mu(s,a)} < 0 \quad \text{and} \quad \exp\left(-\log \frac{d^\pi(s,a)}{d^\mu(s,a)} \right) > 1. \tag{13}$$

Thus:

$$\sum_{(s,a) \in S_2} d^\mu(s,a) Q^\pi(s,a) > \sum_{(s,a) \in S_2} d^\pi(s,a) Q^\pi(s,a). \tag{14}$$

For an entropy-regularized policy, from Equation equation 7 and the above relationship, a tighter bound can be defined as:

$$\sum_{(s,a) \in S_2} d^\mu(s,a) Q^\pi(s,a) > \sum_{(s,a) \in S_2} d^\pi(s,a) Q^\pi(s,a) + \mathbb{D}_{\mathrm{KL}}^{S_2}(d^\pi \parallel d^\mu). \tag{15}$$

Here, the KL-divergence is computed after normalization and the given KL-divergence term implicitly accounts for the subset's normalization process. From equation 3 and equation 15:

$$\sum_{(s,a)\in S_2} d^{\mathcal{D}} Q^{\pi}(s,a) > \sum_{(s,a)\in S_2} d^{\pi}(s,a) Q^{\pi}(s,a) + \lambda \cdot \mathbb{D}_{\mathrm{KL}}^{S_2}(d^{\pi}||d^{\mu}). \tag{16}$$

Now, to understand the effect of mixing of the data, let ΔR denote the performance gain when utilizing mixed trajectories from offline data and the agent's self-generated experiences:

$$\Delta R = \mathbb{E}_{\tau\sim\mathcal{D}}[R(\tau)] - \mathbb{E}_{\tau\sim\pi}[R(\tau)], \tag{17}$$

To analyze the performance gain, we decompose the reward difference using insights from equation 12 and equation 16 in equation 3. These insights reveal that the performance gain ΔR is influenced by the KL divergence term, which balances exploration and exploitation while acting as a regularizer for the policy.

Based on the above insights, we can summarize that state action pairs for which the online policy's visitation frequency dominates the offline data, the KL divergence term acts as a penalty, encouraging the policy to focus on high-reward regions within its known state-action space. This improves exploitation and ensures the policy prioritizes reliable, high-reward trajectories. otherwise, the KL term acts as a bonus, encouraging exploration of these underrepresented areas. This helps mitigate the risk of converging to sub-optimal solutions by leveraging the diversity of offline trajectories.

Thus, the mixed distribution strategy inherently balances exploration and exploitation. By leveraging offline trajectories alongside self-generated experiences, the agent improves sample efficiency and robustness, achieving better exploration and reducing the likelihood of converging to local optima. However, care must be taken to address potential distributional mismatch issues (Ball et al., 2023), which can destabilize training.

4.2 MOORL: Meta Offline-Online RL

This section introduces the Meta Offline-Online Reinforcement Learning (MOORL) framework, which addresses the challenges of integrating offline and online data in off-policy reinforcement learning. MOORL combines the strengths of off-policy learning (Haarnoja et al., 2018) and meta-learning (Finn et al., 2017), leveraging offline data for efficient learning while enabling online exploration, all while ensuring stable Q-learning updates. By employing a meta-learning strategy to dynamically adapt Q-function updates, MOORL mitigates issues such as distributional mismatch, overestimation bias, and instability in Q-learning.

4.2.1 Problem Definition

The task is to learn a policy using two distinct data distributions:

- **Offline Data** ($\mathcal{D}_{\text{offline}}$): This data consists of trajectories previously collected from one or more policies. While offline data may include high-reward sequences, it is often derived from sub-optimal or outdated policies, leading to potential biases. Direct incorporation of this data into training may cause overestimation bias in learned Q-values due to limited diversity and representativeness of experiences.
- **Online Data** ($\mathcal{D}_{\text{online}}$): This data is collected through interactions with the environment based on the current policy. Initially, online data may yield low rewards due to early-stage exploration but typically improves as the policy refines.

The primary challenge comes from the distributional mismatch between offline and online data. Directly mixing these two distributions in off-policy RL algorithms can lead to *instability in Q-learning*, as the value estimates can become biased towards the high-reward offline data, resulting in overestimated Q-values. The proposed MOORL framework addresses this by learning a meta Q-function $Q_{\text{meta}}(s,a;\theta_{\text{meta}})$, parameterized by θ_{meta}, that aims to generalize across both offline and online data distributions. The meta Q-function is optimized using a meta-Q objective using Reptile (Nichol & Schulman, 2018) meta-learning algorithm, balancing contributions from offline and online data.

Algorithm 1 MOORL: Meta Offline-Online Reinforcement Learning

```
Initialize: Meta-policy parameters actor $\phi_{\text{meta}}$, critic $\theta_{\text{meta}}$, and temperature $\alpha$.
Offline dataset $\mathcal{D}_{\text{offline}}$ and empty online buffer $\mathcal{D}_{\text{online}}$.
Meta-learning rate $\eta_{\text{meta}}$, inner-loop learning rate $\eta$, number of iterations $N$.
for $n = 1$ to $N$ do
    Select Buffer: Choose $\mathcal{D}_{\text{offline}}$ or $\mathcal{D}_{\text{online}}$ as the data buffer.
    Inner-loop Adaptation:
    Collect trajectories in online environment and store in $\mathcal{D}_{\text{online}}$.
    Sample mini-batch from the selected data buffer.
    Perform $K$ inner actor $\tilde{\phi}$ and critic $\tilde{\theta}$ updates using data from $\mathcal{D}_i$.
    Meta-update:
    Update meta-policy parameters of both actor and critic using $\phi_{\text{meta}} \leftarrow \phi_{\text{meta}} - \eta_{\text{meta}} \nabla_{\phi_{\text{meta}}}[\mathcal{L}(\tilde{\phi})]$ and
    $\theta_{\text{meta}} \leftarrow \theta_{\text{meta}} - \eta_{\text{meta}} \nabla_{\theta_{\text{meta}}}[\mathcal{L}(\tilde{\theta})]$, respectively.
end for
```

4.2.2 Meta Q-Function Learning

MOORL learns robust meta Q-values that generalize across offline and online data distributions to mitigate overestimation bias. The MOORL framework aims to learn a meta Q-function $Q(s, a; \theta_{\text{meta}})$, parameterized by θ_{meta}, that generalizes across both offline and online distributions. The parameter θ_{meta} can be interpreted as a solution to the Bellman error minimization problem stated below.

$$\min_{\theta_{\text{meta}}} \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}} \left[\left(Q(s, a; \theta_{\text{meta}}) - \left(r + \gamma \mathbb{E}_{s'\sim T(s'|s,a)} \left[\max_{a'} Q(s', a'; \theta_{\text{meta}}) \right] \right) \right)^2 \right], \tag{18}$$

where $\mathcal{D}$ is a dataset containing both offline and online data.

The combined loss ensures that $Q(\cdot, \cdot; \theta_{\text{meta}})$ provides consistent value estimates across both offline and online data. However, this objective is similar to mixing offline and online data distribution as done by RLPD (Ball et al., 2023), which simply mixes data from different distributions and learns a Q-function that can generalize across data distribution which requires different design choices to stabilize learning. In this work, we apply a meta-learning perspective. Specifically, our algorithm progresses in multiple epochs, performing one meta-policy update at each epoch. At the start of an epoch, we randomly choose either the offline or the online replay buffer and perform K inner updates for distribution adaptation followed by meta-update.

Specifically, for a given data distribution, $i \in \{\text{offline}, \text{online}\}$, we take a mini-batch $\mathcal{B}_i \subset \mathcal{D}_i$ of length B, and define the following loss functions for inner loop adaptation of the sampled distribution.

$$\mathcal{L}_i^{\text{critic}}(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{B}_i} \left[\left(Q(s, a; \theta) - (r + \gamma Q(s', a'; \theta')) \right)^2 \right] \tag{19}$$

where θ' is the parameter of a target function that is used for stabilizing the learning. Following Q-learning paradigm, θ and θ' are synced periodically. The above loss is used to update the critic parameter, θ, via a gradient-descent approach. The weights of the critic in the inner update loop are initialized with meta-critic weights followed by K inner loop gradient steps for distribution adaptation. where η denotes a learning parameter. Mathematically, each gradient descent step is defined as follows.

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_i^{\text{critic}}(\theta) \tag{20}$$

Let the final critic parameter (after K inner loop steps) be $\tilde{\theta}$. We similarly define an actor loss function as follows, where ϕ denotes the parameter of the actor (policy approximator), and α is a tunable parameter.

$$\mathcal{L}_i^{\text{actor}}(\phi, \theta) = \mathbb{E}_{s\sim\mathcal{B}_i, a\sim\pi(\cdot|s,\phi)} \left[\alpha \log \pi(a|s; \phi) - Q(s, a; \theta) \right] \tag{21}$$

Starting from meta-policy weights, the parameter ϕ is also updated K number of times as follows.

$$\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}_i^{\text{actor}}(\phi, \theta) \tag{22}$$

Here, we use the principle of Soft Actor Critic (Haarnoja et al., 2018) that maximizes the expected reward (Q-value) while ensuring sufficient exploration through entropy regularization controlled by temperature parameter α. The entropy term encourages the policy to remain stochastic, promoting diverse actions and balancing exploration and exploitation. Let the final value of the actor parameter be $\tilde{\phi}$. After inner loop distribution adaptation, in the outer loop, we update the meta actor and critic parameters θ_{meta}, ϕ_{meta} using the updated inner loop parameters as follows, where η_{meta} is a tunable learning parameter.

$$\theta_{\text{meta}} \leftarrow \theta_{\text{meta}} - \eta_{\text{meta}} \nabla_{\theta_{\text{meta}}} [\mathcal{L}(\tilde{\theta})] \tag{23}$$

$$\phi_{\text{meta}} \leftarrow \phi_{\text{meta}} - \eta_{\text{meta}} \nabla_{\phi_{\text{meta}}} [\mathcal{L}(\tilde{\phi}, \tilde{\theta})] \tag{24}$$

For computing meta-updates ϕ_{meta} and θ_{meta} in the last step in the direction of $\phi_{\text{meta}} - \tilde{\phi}_{\text{meta}}$ and $\theta_{\text{meta}} - \tilde{\theta}_{\text{meta}}$, we treat $\phi_{\text{meta}} - \tilde{\phi}_{\text{meta}}$ and $\theta_{\text{meta}} - \tilde{\theta}_{\text{meta}}$ as a gradient similar to (Nichol & Schulman, 2018) and plunge it into an adaptive algorithm such as Adam (Kinga et al., 2015). In summary, the learning process consists of two steps: first, adapting the distribution through K inner updates, followed by a meta-update aimed at generalizing across distributions

Hence, by leveraging meta-learning capabilities of task adaptation (Finn et al., 2017), we enable adaptation across diverse data distributions generated by different policies. This distribution adaptation strategy allows the learned Q-function to approximate a combined Bellman Q-function. Further, Figure 1 illustrates that the Q-values learned via MOORL and RLPD exhibit similar convergence trends. This demonstrates that learning a combined Bellman function, as done by RLPD or exploring a meta Q-function, yields comparable Q-values. However, MOORL offers the advantage of not relying on specific design choices for stability, highlighting its robustness and simplicity.

5 Experiments

We evaluate the proposed MOORL approach on the D4RL benchmark (Fu et al., 2020) and V-D4RL (Lu et al., 2022), comparing its performance against state-of-the-art methods, including hybrid RL approach RLPD (Ball et al., 2023), Hy-Q (Song et al., 2022) and offline RL method ReBRAC (Tarasov et al., 2024), for completeness. Each baseline has distinct design choices and operational paradigms that provide valuable context for evaluating MOORL's strengths in hybrid offline-online RL settings. Through our chosen baselines and benchmark tasks, we aim to address the following questions:

- Can MOORL effectively integrate offline data into an online RL setting without extensive design-specific configurations?
- Does MOORL ensure stable learning across diverse data distributions, minimizing the need for task-specific tuning?
- Can MOORL be extended to high-dimensional image-based observation data?
- Does MOORL maintain consistent performance across varying offline data qualities by learning a stable Q-function?

5.1 Evaluation on Offline D4RL Tasks

To evaluate MOORL's performance, we select a range of D4RL tasks to assess robustness across diverse data distributions:

- **D4RL Locomotion**(Fu et al., 2020): This set includes 15 dense-reward locomotion tasks, with offline data covering varying levels of optimality, from expert to random trajectories.
- **D4RL Maze-Navigation**(Fu et al., 2020): We utilize 6 AntMaze navigation tasks with sparse binary reward structures, each with different complexities.
- **D4RL Adroit** (Fu et al., 2020): The tasks in this set (Pen, Door, Hammer) involve complex manipulation and sparse rewards, with offline data consisting of expert-level trajectories.

Table 1: Performance comparison across Locomotion tasks. The score represents the average normalized score across 10 different unseen seeds. The symbol ± represents the standard deviation across the seeds.

Task Name	**TD3+BC**	**ReBRAC**	**Hy-Q**	**RLPD (UTD=20)**	**MOORL, our**
half-cheetah-expert	93.4 ± 0.4	105.9 ± 1.7	84.0	**111.1 ± 0.1**	105.9 ± 0.8
half-cheetah-medium-expert	89.1 ± 5.6	101.1 ± 5.2	86.0	**105.8 ± 0.2**	103.4 ± 2.9
half-cheetah-medium-replay	45.0 ± 1.1	51.0 ± 0.8	89.0	72.2 ± 0.4	**96.9 ± 2.0**
half-cheetah-medium	54.7 ± 0.9	65.6 ± 1.0	88.0	85.4 ± 0.1	**99.2 ± 1.9**
half-cheetah-random	30.9 ± 0.4	29.5 ± 1.5	80.0	85.1 ± 9.9	**99.0 ± 4.8**
hopper-expert	109.6 ± 3.7	100.1 ± 8.3	54.0	101.7 ± 16.4	**111.6 ± 4.1**
hopper-medium-expert	87.8 ± 10.5	**107.0 ± 6.4**	100.0	97.1 ± 12.7	101.5 ± 5.9
hopper-medium-replay	55.1 ± 31.7	98.1 ± 5.3	77.0	81.3 ± 24.8	**99.6 ± 4.3**
hopper-medium	60.9 ± 7.6	102.0 ± 1.0	106.0	90.8 ± 20.1	**107.9 ± 2.4**
hopper-random	8.5 ± 0.7	8.1 ± 2.4	80.0	92.9 ± 25.8	**101.9 ± 3.3**
walker2d-expert	110.0 ± 0.6	112.3 ± 0.2	112.0	**127.5 ± 5.6**	123.6 ± 3.7
walker2d-medium-expert	110.4 ± 0.6	111.6 ± 0.3	95.0	**128.0 ± 5.4**	117.2 ± 7.5
walker2d-medium-replay	68.0 ± 19.2	77.3 ± 7.9	103.0	105.7 ± 7.1	**111.2 ± 3.5**
walker2d-medium	77.7 ± 2.9	82.5 ± 3.6	86.0	**115.1 ± 6.4**	114.1 ± 1.7
walker2d-random	2.0 ± 3.6	18.4 ± 4.5	90.0	73.7 ± 31.7	**93.8 ± 1.1**
Average	66.9 ± 8.8	78.0 ± 2.8	88.7	98.2 ± 11.1	**105.8 ± 3.3**

Table 2: Performance comparison across AntMaze tasks. The score represents the average normalized score across 10 different unseen seeds. The symbol ± represents the standard deviation across the seeds.

Task Name	**TD3+BC**	**ReBRAC**	**Hy-Q**	**RLPD (UTD=20)**	**RLPD (UTD=1)**	**MOORL, our**
antmaze-umaze	66.3 ± 6.2	97.8 ± 1.0	-	99.0 ± 0.5	88.2 ± 28.0	**99.2 ± 0.7**
antmaze-umaze-diverse	53.8 ± 8.5	88.3 ± 13.0	-	97.8 ± 1.0	93.4 ± 3.5	**99.0 ± 1.0**
antmaze-medium-play	26.5 ± 18.4	84.0 ± 4.2	25.0	**98.5 ± 0.5**	94.4 ± 3.6	98.2 ± 0.9
antmaze-medium-diverse	25.9 ± 15.3	76.3 ± 13.5	02.0	98.0 ± 1.0	93.6 ± 4.3	**98.5± 1.2**
antmaze-large-play	0.0 ± 0.0	60.4 ± 26.1	00.0	**88.0 ± 2.5**	57.8 ± 16.3	82.3 ± 10.1
antmaze-large-diverse	0.0 ± 0.0	54.4 ± 25.1	00.0	**87.5 ± 3.7**	50.2 ± 20.7	85.6± 6.4
Average	28.7±8.0	76.8±13.8	6.8	**94.8 ±1.3**	79.6 ± 10.5	93.8±3.4

5.1.1 Choice of Baselines

Hybrid RL: We evaluate the performance of MOORL against current state-of-the-art hybrid RL approaches including RLPD (Ball et al., 2023) and Hy-Q (Song et al., 2022). These approaches aim to integrate offline and online learning. To ensure stable learning, they introduce many design elements, resulting in computational overhead and limiting their broader impact on real-world scenarios.

Offline RL: For the offline RL method, we use ReBRAC (Tarasov et al., 2024), which builds on offline RL and investigates the effect of many design elements on the performance highlighting how different design elements can enhance the performance of offline RL agents. ReBRAC is primarily designed for offline RL, while less relevant to MOORL's hybrid offline-online framework, is included for completeness.

RL+BC: These approaches use a minimalistic approach for learning from offline data. To demonstrate the effectiveness of MOORL against the Behavior Cloning (BC) regularized (Bain & Sammut, 1995) RL method, we use TD3+BC (Fujimoto & Gu, 2021) and DrQ+BC (Yarats et al., 2021) for state and pixel-based tasks.

5.1.2 Empirical Results

The baselines TD3+BC and ReBRAC results are taken from (Tarasov et al., 2024) while Hy-Q and DrQ+BC results are taken from (Nakamoto et al., 2024). Our results demonstrate that MOORL achieves competitive

Table 3: Performance comparison across Adroit tasks. The score represents the average normalized score across 10 different unseen seeds. The symbol ± represents the standard deviation across the seeds.

Task Name	BC	TD3+BC	ReBRAC	RLPD (UTD=20)	MOORL, our
pen	85.1	146.3 ± 7.3	**154.1 ± 5.4**	137.8 ±7.2	150.0 ± 2.4
door	34.9	84.6 ± 44.5	104.6 ± 2.4	105.5±6.2	**107.1 ± 2.0**
hammer	125.6	117.0 ± 30.9	133.8 ± 0.7	**140.3 ±8.5**	137.2 ± 3.3
Average	81.9	115.97 ± 27.6	130.8 ± 2.8	127.9±7.3	**131.4±2.6**

performance with RLPD, Hy-Q, and ReBRAC across the D4RL and V-D4RL tasks while requiring minimal hyperparameter adjustments. MOORL exhibits stable learning in a hybrid RL setting, i.e., offline-online learning, and achieves robust cumulative rewards with fewer design-specific configurations.

On D4RL locomotion benchmarks, MOORL achieves the highest average performance. For the tasks with suboptimal data, hybrid approaches such as MOORL, Hy-Q, and RLPD highlight the advantage of using a hybrid learning approach. It is evident from Table 1 that MOORL performs most optimally across tasks, specifically where offline data is suboptimal. MOORL achieves $8 - 10\%$ performance improvement over RLPD without using large critic ensembles, high UTD, and layer normalization.

The performance of RLPD is competitive to MOORL on the Antmaze navigation task, but as highlighted in Table 2, the performance of RLPD drops significantly with a lower UTD ratio, whereas MOORL performs significantly superior with a similar number of gradient steps performed on meta policy. For the D4RL AntMaze navigation tasks, MOORL avoids incorporating design choices such as Clipped Double Q-learning (CDQ) (Fujimoto et al., 2018) and Entropy Backup similar to RLPD, which are used in other tasks. These design choices tend to perform poorly in the sparse reward structure of AntMaze tasks, leading to overly conservative learning and suboptimal policy performance. While RLPD demonstrates the best performance under standard configurations, achieving a modest $3\%-4\%$ improvement over MOORL, its reliance on a high Update-to-Data (UTD) ratio makes it sensitive to this hyperparameter. When evaluated with a low UTD ratio, RLPD experiences a significant performance drop. MOORL achieves a $13\% - 17\%$ improvement over RLPD with UTD=1. This result highlights MOORL's ability to maintain effective learning without reliance on aggressive update schedules, making it particularly advantageous in scenarios with limited computational budgets or where low UTD ratios are preferred.

Adroit tasks pose significant challenges due to their high-dimensional action spaces and sparse reward structures. To evaluate the performance of various approaches, we conducted experiments on Adroit tasks using high-quality offline expert data. As shown in Table 3, the performance of all methods, including MOORL, remains consistent across tasks, with no single approach demonstrating a definitive advantage. Depending on the specific task, RLPD, ReBRAC, and MOORL occasionally outperform one another. However, the differences in performance are not statistically significant. This lack of clear improvement can likely be attributed to the inherent complexity of the Adroit tasks, which makes it challenging for any single method to achieve a distinct edge over others in this domain.

5.2 MOORL's Adaptability to Pixel-Based Observation Spaces

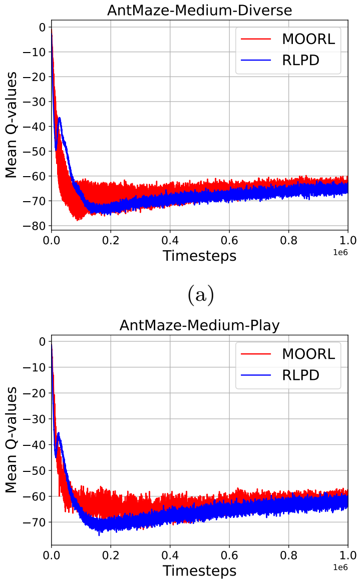
This section investigates MOORL's ability to operate effectively with high-dimensional image-based observations. Unlike methods that require extensive hyperparameter tuning or specialized architectural adjustments, MOORL seamlessly extends to this challenging domain by utilizing a shared feature extraction encoder within its actor-critic framework. This encoder processes raw pixel input into meaningful feature representations, enabling efficient learning even in visually complex environments. We evaluate MOORL's performance on high-dimensional DeepMind Control Suite (DMC) (Tassa et al., 2018) tasks, leveraging datasets with pixel-based observations that test robustness and adaptability. As shown in Table 4, MOORL outperforms RLPD on 3 out of 4 DMC tasks, while RLPD achieves slightly better average cumulative performance. Though RLPD remains competitive, its reliance on large Q-ensembles makes it computationally expensive, particu-

Table 4: Performance comparison across V-D4RL Locomotion tasks. The score represents the normalized score across 10 different unseen seeds. The symbol ± represents the standard deviation across the seeds.

Task Name	**BC**	**DrQ+BC**	**ReBRAC**	**RLPD**	**MOORL, our**
walker-walk-expert	91.5± 3.9	68.4 ± 7.5	81.4 ± 10.0	91.5 ±3.3	**94.6** ± **0.6**
walker-walk-medium	40.9 ± 3.9	46.8 ± 2.3	52.5 ± 3.2	84.9±2.1	**87.9** ± **5.5**
cheetah-run-expert	67.4 ± 6.8	34.5 ± 8.3	35.6 ± 5.3	**68.3** ± **2.9**	53.2 ± 7.5
cheetah-run-medium	51.6 ± 1.4	53.0 ± 3.0	**59.0** ± **0.7**	49.0 ± 4.0	49.9 ± 4.8
Average	62.9 ± 4.0	50.7 ± 5.3	57.1 ± 4.8	**73.4**±**3.0**	71.4±2.6

larly for image-based tasks. MOORL, on the other hand, achieves comparable results without using large critic-ensembles, emphasizing its practicality for resource-constrained scenarios.

These findings highlight MOORL's capacity for generalization and computational efficiency in high-dimensional reinforcement learning tasks. By maintaining strong performance without complex design alterations or fine-tuning, MOORL demonstrates its suitability for visually demanding environments, reaffirming its flexibility and reliability across diverse settings.



(a)

(b)

Figure 1: Learning curves showing the mean Q-values for the AntMaze-Medium task on Diverse and Play datasets. Figures 1a and 1b depict the performance of the MOORL and RLPD frameworks, demonstrating their learning stability and effectiveness across both datasets.

5.3 Does MOORL learn Stable Q-Values

To this end, we demonstrate that MOORL enables stable learning without relying on extensive design choices. Specifically, we present the mean Q-values in Figure 1 to showcase the stability of MOORL's learning architecture. Despite the data quality, MOORL consistently learns a stable meta Q-function, translating into a stable meta-policy.

In Figure 1, we evaluate MOORL stability in the challenging and sparsely rewarded AntMaze navigation task. The diverse dataset is created by assigning random goal locations in the maze and directing the agent to navigate to them. In contrast, the play dataset consists of trajectories from specific, hand-picked initial positions to hand-picked goal locations. Remarkably, MOORL achieves similar Q-value trajectories across these datasets, emphasizing its robustness to variations in data quality.

For comparison, we conducted a similar analysis on RLPD to further substantiate the stability of Q-value learning across different frameworks. This analysis highlights MOORL's capability to maintain stable learning dynamics even under diverse and challenging data conditions without requiring specific design choices.

6 Related Work

6.1 Offline Reinforcement Learning

Offline reinforcement learning (RL) has gained significant attention for its ability to learn policies from pre-collected datasets without additional environmental interaction. However, distributional shift remains a fundamental challenge, as highlighted by Levine et al. (2020), making it crucial to design algorithms that generalize effectively from limited data. Several approaches address this challenge by incorporating regularization techniques. TD3+BC (Fujimoto & Gu, 2021) integrates behavioral cloning into the actor loss to constrain policy learning, while CQL (Kumar et al., 2020) enforces conservatism by penalizing the critic for assigning high values to out-of-distribution (OOD) actions. IQL (Kostrikov et al., 2021) takes a different approach by leveraging advantage-weighted regression to avoid sampling OOD actions altogether. More recent methods further improve policy learning by incorporating representation learning, such as pre-training action encoders (Akimov et al., 2022; Chen et al., 2022) or estimating dataset uncertainty using VAEs (Wu et al., 2022) and RND (Nikulin et al., 2023). Another line of work improves policy robustness by leveraging ensemble-based uncertainty estimation. SAC-N (An et al., 2021) achieves strong results using large Q-function ensembles, though some tasks require ensembles of up to 500 members, making it computationally expensive. MSG (Ghasemipour et al., 2022) mitigates this by introducing independent target updates, reducing the ensemble size to four in MuJoCo tasks but still requiring 64 members for AntMaze.

Further, Fujimoto & Gu (2021) shows that performance is significantly influenced by non-algorithmic factors, and ReBRAC (Tarasov et al., 2024) performs a detailed analysis to understand the effect of design choice on the performance of offline RL. Offline RL also struggles when datasets lack full coverage, making pessimism computationally challenging (Jin et al., 2021; Zhang et al., 2022) and often requiring strong representation conditions (Xie et al., 2021). To overcome these limitations, hybrid approaches incorporating limited online interaction have been proposed as a promising alternative.

6.2 Bridging Online and Offline RL

Integrating online and offline reinforcement learning (RL) is a critical research area. Empirical studies have examined how online learners can leverage logged data to improve performance (Rajeswaran et al., 2017; Nair et al., 2017; Hester et al., 2017; Ball et al., 2023; Nakamoto et al., 2024; Zheng et al., 2023). While practical benefits are evident, formal guarantees in this setting remain limited. (Ross & Bagnell, 2012) proposed a framework where a learner can choose between executing a logging policy μ or an alternative online policy, effectively bridging the gap between online and offline data exploitation. (Xie et al., 2021) demonstrated that no approach could achieve strictly better sample complexity than purely online or offline methods when using data collected from a logging policy. This result highlights the challenges of balancing online exploration with offline data utilization. In contrast, our research assumes access to a pre-collected offline dataset and the ability to interact online, enabling the refinement of a near-optimal policy while minimizing online interactions. (Song et al., 2022) proposed "Hybrid RL" using the Hybrid Q-learning algorithm (Hy-Q) for low bilinear rank Markov Decision Processes (MDPs) (Du et al., 2021). Under certain conditions, Hy-Q can achieve optimal policies efficiently. However, the method's performance may degrade when offline data coverage of the optimal policy is insufficient, illustrating the importance of comprehensive offline data. Further, a study by (Xie et al., 2022) delves into purely online contexts or frameworks involving offline datasets, offering insights into the concealability coefficient—a parameter critical in establishing guarantees for offline RL. This approach bridges the analytical methods used in online and offline settings. Recent work by (O'Donoghue et al., 2018; Wagenmaker & Pacchiano, 2022), and others explore synergies between these methodologies, revealing strategies for effectively integrating offline data with online exploration.

Our approach builds on these foundations by addressing sample complexity when merging offline datasets with online learning. We aim to enhance the unified integration of offline data into online reinforcement learning without extensive design choices and added computational complexity (Ball et al., 2023; Song et al., 2022), which limits the border applicability of such hybrid-RL approaches.

7 Conclusion

In this work, we demonstrated that off-policy RL approaches can effectively combine offline data with online data distributions. We also showed that learning a meta-policy over these two distributions enables a design-free learning paradigm capable of learning stable Q-values independent of the quality of the data. Our method's efficacy was validated through extensive experiments on 28 diverse tasks spanning state and pixel observations. Concretely, we showed that learning a hybrid RL policy enhances sample efficiency and ensures robust performance across varying data qualities. By avoiding task-specific design parameters, our approach generalizes well to different environments, making it a scalable and practical solution for real-world applications. These findings underscore the potential of integrating offline and online learning to address long-standing challenges in reinforcement learning.

References

Dmitriy Akimov, Vladislav Kurenkov, Alexander Nikulin, Denis Tarasov, and Sergey Kolesnikov. Let offline rl flow: Training conservative agents in the latent space of normalizing flows. *arXiv preprint arXiv:2211.11096*, 2022.

Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.

Chenjia Bai, Lingxiao Wang, Zhuoran Yang, Zhihong Deng, Animesh Garg, Peng Liu, and Zhaoran Wang. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. *arXiv preprint arXiv:2202.11566*, 2022.

Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, pp. 103–129, 1995.

Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pp. 1577–1594. PMLR, 2023.

Arthur Charpentier, Romuald Elie, and Carl Remlinger. Reinforcement learning in economics and finance. *Computational Economics*, pp. 1–38, 2021.

Gaurav Chaudhary, Laxmidhar Behera, and Tushar Sandhan. Active perception system for enhanced visual signal recovery using deep reinforcement learning. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2023. doi: 10.1109/ICASSP49357.2023.10097084.

Xi Chen, Ali Ghadirzadeh, Tianhe Yu, Jianhao Wang, Alex Yuan Gao, Wenzhe Li, Liang Bin, Chelsea Finn, and Chongjie Zhang. Lapo: Latent-variable advantage-weighted policy optimization for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:36902–36913, 2022.

Simon Shaolei Du, Sham M. Kakade, Jason D. Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in rl. *ArXiv*, abs/2103.10897, 2021. URL `https://api.semanticscholar.org/CorpusID:232290507`.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep rl: A case study on ppo and trpo. In *International conference on learning representations*, 2019.

Amir-massoud Farahmand, Csaba Szepesvári, and Rémi Munos. Error propagation for approximate policy and value iteration. *Advances in neural information processing systems*, 23, 2010.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020. URL `https://arxiv.org/abs/2004.07219`.

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

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

Scott Fujimoto, Edoardo Conti, Mohammad Ghavamzadeh, and Joelle Pineau. Benchmarking batch deep reinforcement learning algorithms. *arXiv preprint arXiv:1910.01708*, 2019.

Hiroki Furuta, Tadashi Kozuno, Tatsuya Matsushima, Yutaka Matsuo, and Shixiang Shane Gu. Identifying co-adaptation of algorithmic and implementational innovations in deep reinforcement learning: A taxonomy and case study of inference-based algorithms. *arXiv preprint arXiv:2103.17258*, 2021.

Kamyar Ghasemipour, Shixiang Shane Gu, and Ofir Nachum. Why so pessimistic? estimating uncertainties for offline rl through ensembles, and why their independence matters. *Advances in Neural Information Processing Systems*, 35:18267–18281, 2022.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Todd Hester, Matej Vecerík, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, Gabriel Dulac-Arnold, John P. Agapiou, Joel Z. Leibo, and Audrunas Gruslys. Deep q-learning from demonstrations. In *AAAI Conference on Artificial Intelligence*, 2017. URL `https://api.semanticscholar.org/CorpusID:10208474`.

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

Taewoo Kim, Ho Suk, and Shiho Kim. Chapter nine - offline reinforcement learning methods for real-world problems. In Shiho Kim and Ganesh Chandra Deka (eds.), *Artificial Intelligence and Machine Learning for Open-world Novelty*, volume 134 of *Advances in Computers*, pp. 285–315. Elsevier, 2024. doi: https://doi.org/10.1016/bs.adcom.2023.03.001. URL `https://www.sciencedirect.com/science/article/pii/S0065245823000372`.

D Kinga, Jimmy Ba Adam, et al. A method for stochastic optimization. In *International conference on learning representations (ICLR)*, volume 5, pp. 6. San Diego, California;, 2015.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, 2021.

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

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning*, pp. 1702–1712. PMLR, 2022.

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

Cong Lu, Philip J Ball, Tim GJ Rudner, Jack Parker-Holder, Michael A Osborne, and Yee Whye Teh. Challenges and opportunities in offline reinforcement learning from visual observations. *arXiv preprint arXiv:2206.04779*, 2022.

Rémi Munos. Error bounds for approximate value iteration. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, pp. 1006. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and P. Abbeel. Overcoming exploration in reinforcement learning with demonstrations. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6292–6299, 2017. URL `https://api.semanticscholar.org/CorpusID:3543784`.

Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

Alexander Nikulin, Vladislav Kurenkov, Denis Tarasov, and Sergey Kolesnikov. Anti-exploration by random network distillation. In *International Conference on Machine Learning*, pp. 26228–26244. PMLR, 2023.

Brendan O'Donoghue, Ian Osband, Remi Munos, and Volodymyr Mnih. The uncertainty bellman equation and exploration. In *International conference on machine learning*, pp. 3836–3845, 2018.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *ArXiv*, abs/1709.10087, 2017. URL `https://api.semanticscholar.org/CorpusID:4780901`.

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

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

Yuda Song, Yifei Zhou, Ayush Sekhari, J Andrew Bagnell, Akshay Krishnamurthy, and Wen Sun. Hybrid rl: Using both offline and online data can make rl efficient. *arXiv preprint arXiv:2210.06718*, 2022.

Yihao Sun, Jiaji Zhang, Chengxing Jia, Haoxin Lin, Junyin Ye, and Yang Yu. Model-bellman inconsistency for model-based offline reinforcement learning. In *International Conference on Machine Learning*, pp. 33177–33194. PMLR, 2023.

Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes, 2024. URL `https://arxiv.org/abs/2408.03539`.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy P. Lillicrap, and Martin A. Riedmiller. Deepmind control suite. *CoRR*, abs/1801.00690, 2018. URL `http://arxiv.org/abs/1801.00690`.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017. doi: 10.1109/IROS.2017.8202133.

Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojciech M Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, et al. Alphastar: Mastering the real-time strategy game starcraft ii. *DeepMind blog*, 2:20, 2019.

Andrew Wagenmaker and Aldo Pacchiano. Leveraging offline data in online reinforcement learning. *ArXiv*, abs/2211.04974, 2022. URL `https://api.semanticscholar.org/CorpusID:253420756`.

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

Jialong Wu, Haixu Wu, Zihan Qiu, Jianmin Wang, and Mingsheng Long. Supported policy optimization for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:31278–31291, 2022.

Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. *Advances in neural information processing systems*, 34: 27395–27407, 2021.

Tengyang Xie, Dylan J Foster, Yu Bai, Nan Jiang, and Sham M Kakade. The role of coverage in online reinforcement learning. *arXiv preprint arXiv:2210.04157*, 2022.

Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International conference on learning representations*, 2021.

Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021.

Xuezhou Zhang, Yiding Chen, Xiaojin Zhu, and Wen Sun. Corruption-robust offline reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 5757–5773. PMLR, 2022.

Han Zheng, Xufang Luo, Pengfei Wei, Xuan Song, Dongsheng Li, and Jing Jiang. Adaptive policy learning for offline-to-online reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11372–11380, 2023.

A Hyperparameters

Table 5: RLPD Hyperparameters

Parameter	Value
Batch size	256
Discount (γ)	0.99
Optimizer	Adam
Learning rate	3×10^{-4}
Critic EMA Weight (ρ)	0.005
Inner Gradient Steps (K)	4
Network Width	256 Units
Number of Layers	2
Initial Entropy Temperature (α)	1.0
Target Entropy	$-\frac{\dim(A)}{2}$

B Detailed Task Definition

B.1 D4RL: Locomotion

In these tasks, the reward is dense and based on the agent's forward velocity, penalizing large control inputs to encourage stable movement. The goal is to maximize the cumulative reward over the episode by learning an efficient and stable gait. The standard evaluation metric is the normalized score, which is computed by normalizing the agent's return relative to expert and random policies, as defined by (Fujimoto et al., 2019). The datasets are generated from policies of varying expertise, including *random*, *medium*, *medium-replay*, *medium-expert*, and *expert* trajectories. Episodes typically last for 1,000 timesteps without early termination.

B.2 D4RL: AntMaze

In these tasks, the reward is sparse and binary, indicating whether the agent has reached the goal. Upon reaching the goal, the episode terminates. The normalized return is measured as the proportion of successful trials out of evaluation trials following prior work. The dataset consists of *play-based* and *diverse* demonstrations, where the former includes goal-directed trajectories, and the latter contains broader movement data. The challenge in this domain arises from long-horizon credit assignment and the need for effective exploration in a sparse reward setting.

B.3 D4RL: Adroit

The Adroit suite consists of dexterous hand manipulation tasks that require controlling a 24-DoF simulated Shadow Hand robot to perform complex actions such as hammering a nail, opening a door, or twirling a pen. This domain is specifically designed to assess the impact of narrowly distributed expert demonstrations on learning in a high-dimensional robotic manipulation setting with sparse rewards. Unlike standard Gym MuJoCo tasks, Adroit exhibits several distinct characteristics. First, the dataset is sourced from human demonstrations. Second, due to the sparse reward structure and inherent exploration difficulties, solving these tasks with online RL alone proves challenging. Lastly, the high-dimensional nature of these tasks introduces additional complexity in representation learning.

B.4 V-D4RL: DeepMind Control Suite (DMC)

The DMC tasks involve controlling physics-based agents with dense rewards that encourage smooth, efficient movement. The standard evaluation metric is the normalized score, computed using the return of the agent normalized against the performance of a well-trained SAC policy. The datasets include *expert* and *medium* policies, allowing evaluation of an agent's ability to learn from varying data quality. Episodes typically run for 1,000 timesteps without early termination.

C MOORL: Embracing Design Independence

Recent works have suggested that significant performance improvement can be achieved through offline data in reinforcement learning. Current state-of-the-art in hybrid RL (RLPD (Ball et al., 2023)) and offline RL (ReBRAC (Tarasov et al., 2024)) have demonstrated that challenges associated with offline data can be reduced through the incorporation of different design elements. MOORL framework aims to answer the key questions: Can a similar performance be achieved without introducing new design elements?

ReBRAC, a state-of-the-art method for offline reinforcement learning, focuses on fine-tuning various design parameters—such as deeper networks, larger batch sizes, and layer normalization—to optimize policy performance. While effective, this dependence on specific configurations can reduce adaptability across diverse learning environments. On the other hand, RLPD integrates offline data into online learning and aligns closely with the goals of MOORL. However, RLPD introduces many design elements and depends heavily on large Q-ensembles and a high UTD ratio for its performance, as shown in Table 2.

In contrast, MOORL's design independence enables it to adapt seamlessly to various tasks without introducing design components. This freedom from rigid design constraints allows MOORL to effectively address distribution shifts and limited exploration challenges. By removing these fixed design choices, MOORL enhances its adaptability and robustness in dynamic environments, providing a more versatile approach to integrating offline and online data.

The tables 6 and 7 illustrate how MOORL's approach differs from ReBRAC and RLPD in terms of the need for specific design components. Unlike ReBRAC and RLPD, which require fine-tuning various parameters for optimal performance, MOORL operates effectively without these additional design dependencies.

Table 6: Comparison of Design Choices with ReBRAC (Tarasov et al., 2024)

Component	**ReBRAC (Design Required)**	**MOORL (Design-Free)**
Deeper Networks	✓	✗
Larger Batches	✓	✗
Layer Normalization	✓	✗
Decoupled Penalization	✓	✗
Adjusted Discount Factor	✓	✗

Table 7: Comparison of Design Choices with RLPD (Ball et al., 2023)

Component	**RLPD (Design Required)**	**MOORL (Design-Free)**
Sampling Strategy	✓	✗
Layer Normalization	✓	✗
Random Ensemble Distillation	✓	✗
Clipped Double Q-Learning	✓	✓
Network Architectures	✓	✗
Update to Data Ratio	✓	✗
Entropy Backup	✓	✓

In summary, MOORL's lack of reliance on specific design components simplifies the learning process and allows for a more flexible and adaptable approach to reinforcement learning. This contrasts ReBRAC and RLPD, which require detailed configuration to achieve optimal results. MOORL provides a more generalizable and robust solution across various tasks by embracing design independence.