# When to Trust Your Simulator: Dynamics-Aware Hybrid Offline-and-Online Reinforcement Learning

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Learning effective reinforcement learning (RL) policies to solve real-world complex tasks can be quite challenging without a high-fidelity simulation environment. In most cases, we are only given imperfect simulators with simplified dynamics, which inevitably lead to severe sim-to-real gaps in RL policy learning. The recently emerged field of offline RL provides another possibility to learn policies directly from pre-collected historical data. However, to achieve reasonable performance, existing offline RL algorithms need impractically large offline data with sufficient state-action space coverage for training. This brings up a new question: is it possible to combine learning from limited real data in offline RL and unrestricted exploration through imperfect simulators in online RL to address the drawbacks of both approaches? In this study, we propose the Dynamics-Aware **H**ybrid **O**ffline-and-**O**nline Reinforcement Learning (H2O) framework to provide an affirmative answer to this question. H2O introduces a dynamics-aware policy evaluation scheme, which adaptively penalizes the Q function learning on simulated state-action pairs with large dynamics gaps, while also simultaneously allowing learning from a fixed real-world dataset. Through extensive simulation and real-world tasks, as well as theoretical analysis, we demonstrate the superior performance of H2O against other cross-domain online and offline RL algorithms. H2O provides a brand new hybrid offline-and-online RL paradigm, which can potentially shed light on future RL algorithm design for solving practical real-world tasks.

## 1 Introduction

Over recent years, criticism against reinforcement learning (RL) continues to pour regarding its poor real-world applicability. Although RL has demonstrated superhuman performance in solving complex tasks such as playing games [Mnih *et al.*, 2013; Silver *et al.*, 2017], its success is heavily dependent on the availability of an unbiased interactive environment, either the real system or a high-fidelity simulator, as well as millions of unrestricted trials and errors. However, constructing high-fidelity simulators can be extremely expensive or even impossible due to the complex system dynamics and unobservable information in physical world. Less accurate simulators with simplifications on the other hand, are easy to build, however, also lead to severe visual and dynamics sim-to-real gaps when learning RL policies [Rao *et al.*, 2020; Peng *et al.*, 2018a]. Among different sources of sim-to-real gaps, visual gaps are relatively well-addressed in a number of methods, e.g. through domain randomization [Tobin *et al.*, 2017], proper model setups that are less impacted by visual gaps [Lee *et al.*, 2021], or using a robust and transferable state-encoder [Bewley *et al.*, 2019; Rao *et al.*, 2020]. Dynamics gap on the other hand causes a systematic bias that is not easy to address through a modeling process. These Dynamics gaps widely exist in almost every imperfect physical simulator with simplified dynamics, posing great challenges for RL when solving real-world tasks.

Recently, offline RL provides another possibility to directly learn policies from real-world data [Levine *et al.*, 2020; Fujimoto *et al.*, 2018; Kumar *et al.*, 2019], which has already been used in solving

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

a number of practical problems, such as industrial control [Zhan *et al.*, 2022a], robotics [Lee *et al.*, 2021] and interactive recommendation [Xiao and Wang, 2021]. However, the policies learned through existing offline RL algorithms are often over-conservative due to the use of data-related policy constraints [Fujimoto *et al.*, 2019; Kumar *et al.*, 2019; Fujimoto and Gu, 2021], value regularizations [Kumar *et al.*, 2020; Kostrikov *et al.*, 2021a; Xu *et al.*, 2022], or in-sample learning [Kostrikov *et al.*, 2021b] to combat the distributional shift [Kumar *et al.*, 2019]. This hurts the performance of offline RL policies and makes their performance strongly dependent on the size and state-action space coverage of the offline dataset. In real-world scenarios, data collection can be expensive or restrictive. The state-action space coverage of actual offline datasets can be quite narrow, which directly limit the potential performance of offline RL policies.

Despite all the drawbacks, online RL with an imperfect simulator and offline RL with the real-world dataset bear indispensable potential when jointly used to learn high performance policies and overcome the sim-to-real issues. Within a simulator, online RL agents can perform unrestricted exploration and have an access to a great quantity and diversity of state-action data, which offline datasets may have poor coverage. On the other hand, real-world datasets contain perfect dynamics information, which can be readily used to guide and correct the erroneous information in online training with simulation data. This intuition has motivated researchers to explore how to incorporating offline real-world data with simulation-based online training, however, existing studies only restricted to pure online [Eysenbach *et al.*, 2020] or offline [Liu *et al.*, 2021] learning settings. None of the prior works has successfully combined offline and online learning organically, while simultaneously handling the dynamics gaps between these two regimes.

In this paper, we present the first Hybrid Offline-and-Online Reinforcement Learning framework (H2O), that enables simultaneous policy learning with offline real-world datasets and simulation rollouts. The core of H2O is the introduction of dynamics-aware policy evaluation, which adaptively pushes down or pulls up Q values on simulated data according to the dynamics gap evaluated against real data. We provide analyses and insights on why such dynamics-aware learning can potentially bridge the sim-to-real gap. Finally, we analyze the impact of each design component in H2O and demonstrate its superior performance against other cross-domain RL algorithms through extensive simulation and real-world experimental validations.

## 2 Related Work

### 2.1 Reinforcement Learning Using Simulators with Dynamics Gap

Dealing with the dynamics gap in simulators has long been recognized as a challenging task. The most straightforward approach is to calibrate the simulated dynamics with the real-world counterpart using system identification methods [Ljung, 1998; Rajeswaran *et al.*, 2016; Chebotar *et al.*, 2019]. These methods typically need plenty of offline or even real-world interaction data to tune the parameters of the simulator [Yu *et al.*, 2017], sometimes can be costly or offer little improvement if the simulator is over-simplified. Domain Randomization (DR) [Peng *et al.*, 2018b; Andrychowicz *et al.*, 2020] provides another perspective that trains RL policies in a series of randomized simulated dynamics. DR has been shown to yield more adaptable policies in real-world, yet it often needs manually-specified randomized parameters and nuanced randomization distributions [Vuong *et al.*, 2019]. More recently, Dynamics Adaptation, e.g. DARC [Eysenbach *et al.*, 2020], augments the simulation reward $r(s, a)$ with a dynamics gap-related penalization term $\Delta r(s', s, a)$ in online simulation-based RL training, derived from minimizing the divergence of distributions over real and simulation trajectories. DARC uses real data and trains two discriminators to evaluate $\Delta r(s', s, a)$, rather than directly leverage the data for RL training. In a similar effort, DARA [Liu *et al.*, 2021] is the pure offline version of DARC, sacrificing the unlimited online exploration in the simulated domain. While existing RL studies have tackled the dynamics gap issue from different perspectives, none of them is able to develop a structured framework by combining offline data with online learning, simultaneously addressing the dynamics gap.

### 2.2 Offline Reinforcement Learning

The recent emerged offline RL methods provide a new alternative to learn policies directly from a fixed, pre-collected dataset. A key challenge of offline RL is the *distributional shift* problem [Kumar *et al.*, 2019; Levine *et al.*, 2020], which is caused by counterfactual queries of function approximators (e.g., value function, police network) on out-of-distribution (OOD) samples. Conventional off-policy RL methods will suffer severe value overestimation due to distributional shift. Existing offline RL

methods solve this problem by introducing various data-related regularizations to constrain policy learning from deviating too much from the offline dataset, such as using data distribution or support based policy constraints [Fujimoto *et al.*, 2019; Kumar *et al.*, 2019; Wu *et al.*, 2019; Fujimoto and Gu, 2021], regularizing the value function on OOD data [Kumar *et al.*, 2020; Kostrikov *et al.*, 2021a; Yu *et al.*, 2021; Li *et al.*, 2022; Xu *et al.*, 2022], in-sample learning [Kostrikov *et al.*, 2021b], and adding uncertainty-based penalties [Yu *et al.*, 2020; Kidambi *et al.*, 2020; Zhan *et al.*, 2022a,b]. Conservatism and pessimism are central principles [Buckman *et al.*, 2020] used in almost all existing offline RL algorithms, which cause the performances of these methods to heavily depend on the size, quality and state-action space coverage of the offline dataset. In most practical settings, getting a dataset with large coverage is impractical. Meanwhile, if online interaction with a simulator is involved, although may not be accurate, can substantially supplement the state-action space coverage of offline dataset in policy learning. This motivates a new paradigm of combining both offline and online RL, which is explored in our proposed H2O framework.

# 3 Background

## 3.1 Reinforcement Learning

We consider the standard Markov Decision Process (MDP) setting, which is specified by a tuple $\mathcal{M} := (\mathcal{S}, \mathcal{A}, r, P_{\mathcal{M}}, \rho, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $r(\mathbf{s}, \mathbf{a})$ is the reward function, $P_{\mathcal{M}}(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ refers to the transitional dynamics under $\mathcal{M}$, $\rho(\mathbf{s})$ depicts the initial state distribution, and $\gamma \in (0, 1)$ is the discount factor. The goal of RL is to learn a policy $\pi(\mathbf{a}|\mathbf{s})$ that maps a state $\mathbf{s}$ to an action $\mathbf{a}$ so as to maximize the expected cumulative discounted reward starting from $\rho$, i.e., $J(\mathcal{M}, \pi) = \mathbb{E}_{\mathbf{s}_0 \in \rho, \mathbf{a}_t \sim \pi(\cdot|\mathbf{s}_t), \mathbf{s}_{t+1} \sim P_{\mathcal{M}}(\cdot|\mathbf{s}, \mathbf{a})} \left[\sum_{t=0}^{\infty} \gamma^t r\left(\mathbf{s}_t, \mathbf{a}_t\right)\right]$. In conventional actor-critic formalism [Barto *et al.*, 1983; Sutton *et al.*, 1998], one learns an approximated Q function $\hat{Q}$ by minimizing the squared Bellman error (referred as *policy evaluation*), and optimizes the policy $\pi$ by maximizing the Q function (referred as *policy improvement*) as follows:

$$\hat{Q} \leftarrow \arg\min_{Q} \mathbb{E}_{\mathbf{s}, \mathbf{a}, \mathbf{s}' \sim \mathcal{U}} \left[\left(Q(\mathbf{s}, \mathbf{a}) - \hat{\mathcal{B}}^{\pi} \hat{Q}(\mathbf{s}, \mathbf{a})\right)^2\right] \qquad \text{(Policy evaluation)} \tag{1}$$

$$\hat{\pi} \leftarrow \arg\max_{\pi} \mathbb{E}_{\mathbf{s}, \mathbf{a} \sim \mathcal{U}} \left[\hat{Q}(\mathbf{s}, \mathbf{a})\right] \qquad \text{(Policy improvement)} \tag{2}$$

where $\mathcal{U}$ can be either the replay buffer generated by a previous version of policy $\hat{\pi}$ through online environment interactions, or can be a fixed dataset $\mathcal{D} = \left\{\left(\mathbf{s}_t^i, \mathbf{a}_t^i, \mathbf{s}_{t+1}^i, r_t^i\right)\right\}_{i=1}^{n}$ as in offline RL setting. $\hat{\mathcal{B}}^{\pi}$ is the Bellman operator, which is typically used as the Bellman evaluation operator $\hat{\mathcal{B}}^{\pi} \hat{Q}(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}, \mathbf{a}) + \gamma \mathbb{E}_{\mathbf{a}' \sim \hat{\pi}(\mathbf{a}'|\mathbf{s}')} \left[\hat{Q}\left(\mathbf{s}', \mathbf{a}'\right)\right]$.

## 3.2 Offline RL via Value Regularization

In the offline setting, performing the standard Bellman update in Eq. (1) can result in serious overestimation of Q values due to distributional shift. An effective approach to alleviate this problem is to regularize the value function on OOD actions, which is introduced in CQL [Kumar *et al.*, 2020]. CQL learns a policy $\hat{\pi}$ upon a lower-bounded Q function that additionally pushes down the overestimated Q on actions induced by $\mu(\mathbf{a}|\mathbf{s})$ and pulls up Q values on trustworthy offline data:

$$\min_{Q} \max_{\mu} \alpha \left(\mathbb{E}_{\mathbf{s} \sim \mathcal{D}, \mathbf{a} \sim \mu(\cdot|\mathbf{s})}[Q(\mathbf{s}, \mathbf{a})] - \mathbb{E}_{\mathbf{s}, \mathbf{a} \sim \mathcal{D}}[Q(\mathbf{s}, \mathbf{a})]\right) + \mathcal{E}\left(Q, \hat{\mathcal{B}}^{\pi} \hat{Q}\right) \tag{3}$$

where $\pi_{\beta}(\cdot|\mathbf{s})$ is the behavior policy in $\mathcal{D}$, and $\mu(\cdot|\mathbf{s})$ is some sampling distribution, which is often taken as $\hat{\pi}(\cdot|\mathbf{s})$ in prior works [Yu *et al.*, 2021; Li *et al.*, 2022]. $\mathcal{E}\left(Q, \hat{\mathcal{B}}^{\pi} \hat{Q}\right)$ denotes the Bellman error in Eq. (1). It is worth noting that above value regularization framework has the potential to handle data from two sources with some preference, i.e., the unreliable pushed-down data and reliable pulled-up data. This motivates us to devise the H2O framework to perform simultaneous offline and online policy learning using a similar value regularization recipe.

# 4 Hybrid Offline-and-Online Reinforcement Learning

Naïvely combining offline and online data from different distributions may cause the mismatch between data distributions, causing issues similar to the distributional shift problem as in offline RL settings [Kumar *et al.*, 2019]. Caution needs to be taken when dealing with the trustworthy

offline data from real world and the potentially problematic simulated online rollouts (described in Section 4.1). Secondly, the dynamics model underpinning the simulator is only roughly aligned with the real-world counterpart, and the dynamics sim-to-real gaps can be heterogeneous across different sate-action pairs. Hence uniform value regularization upon all simulated samples may not be appropriate to fully leverage the potential power of an imperfect simulator. A well-founded dynamics gap measurement for adaptive value regularization is needed (Section 4.2). Lastly, it is also worth noting that the biased simulated dynamics could induce erroneous next-state predictions which may lead to problematic Bellman updates. Such errors also need to be properly considered in the algorithm design(Section 4.3). We approach all the aforementioned issues with what we call Dynamics-Aware Hybrid Offline-and-Online (H2O) RL, which is described in following contents.

### 4.1 Incorporating Offline Data in Online Learning

To address the potential mismatching of training data distributions caused by different data sources, existing online RL frameworks are not suitable, however, offline RL approaches such as value regularization can provide a viable foundation. We modify upon the previous value regularization scheme in offline RL and propose the dynamics-aware policy evaluation, which has the following general form:

$$\min_{Q} \max_{d^\phi} \beta \left[\mathbb{E}_{\mathbf{s},\mathbf{a}\sim d^\phi(\mathbf{s},\mathbf{a})}[Q(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}[Q(\mathbf{s},\mathbf{a})] + \mathcal{R}(d^\phi)\right] + \widetilde{\mathcal{E}}\left(Q, \hat{\mathcal{B}}^\pi \hat{Q}\right) \tag{4}$$

where $\beta$ is a positive scaling parameter. $d^\phi(\mathbf{s},\mathbf{a})$ is a particular state-action sampling distribution that associates with high dynamics-gap samples, and $\mathcal{R}(d^\phi)$ is a regularization term for $d^\phi$ to enforce this designed behavior. Ideally, we would want to penalize Q values at the simulated samples with high dynamics gaps, rather than all of the samples. $\widetilde{\mathcal{E}}(Q, \hat{\mathcal{B}}^\pi \hat{Q})$ denotes the modified Bellman error of the mixed data from offline dataset $\mathcal{D}$ and the simulation rollout samples in online replay buffer $B$, which are generated by the real MDP $\mathcal{M}$ and the simulated MDP $\widehat{\mathcal{M}}$ respectively. Specifically, the Bellman error of the simulated data needs to be fixed. The terms marked in red and blue are the key design elements that account for the dynamics gaps in offline-and-online RL policy learning, which will be discussed in the following sub-sections.

### 4.2 Adaptive Value Regularization on High Dynamics-Gap Samples

To achieve adaptive value regularization, we can use the minimization term $\mathbb{E}_{\mathbf{s},\mathbf{a}\sim d^\phi(\mathbf{s},\mathbf{a})}[Q(\mathbf{s},\mathbf{a})]$ to penalize the high-dynamics gap samples, and use the maximization term $\mathbb{E}_{\mathbf{s}\sim\mathcal{D},\mathbf{a}\sim\pi_\beta(\mathbf{a}|\mathbf{s})}[Q(\mathbf{s},\mathbf{a})]$ to cancel the penalization on real offline data. The key question is how to properly design $d^\phi(\mathbf{s},\mathbf{a})$ to assign high probabilities to high-dynamics gap samples. In our study, we leverage the regularization term $\mathcal{R}(d^\phi)$ to control the behavior of $d^\phi(\mathbf{s},\mathbf{a})$. We choose $\mathcal{R}(d^\phi) = -D_{KL}(d^\phi(\mathbf{s},\mathbf{a})\|\omega(\mathbf{s},\mathbf{a}))$, where $D_{KL}(\cdot||\cdot)$ is the Kullback-Leibler (KL) divergence and $\omega(\mathbf{s},\mathbf{a})$ is a distribution that characterizes the dynamics gaps for samples in the state-action space. Hence minimizing over $\mathcal{R}(d^\phi)$ draws closure between $d^\phi$ and $\omega$, and achieves our desired behavior. Note that the inner maximization problem over $d^\phi$ in Eq. (4) under this design corresponds to the following optimization problem:

$$\max_{d^\phi} \mathbb{E}_{\mathbf{s},\mathbf{a}\sim d^\phi(\mathbf{s},\mathbf{a})}[Q(\mathbf{s},\mathbf{a})] - D_{KL}(d^\phi(\mathbf{s},\mathbf{a})\|\omega(\mathbf{s},\mathbf{a})) \quad \text{s.t.} \sum_{\mathbf{s},\mathbf{a}} d^\phi(\mathbf{s},\mathbf{a}) = 1, d^\phi(\mathbf{s},\mathbf{a}) \geq 0 \tag{5}$$

Above optimization problem admits a close-form solution $d^\phi(\mathbf{s},\mathbf{a}) \propto \omega(\mathbf{s},\mathbf{a}) \exp\left(Q(\mathbf{s},\mathbf{a})\right)$. Plugging this back to Eq. (4), the first term now corresponds to a weighted soft-maximum of Q-values at any state-action pair and the original problem transforms into the following form:

$$\min_{Q} \beta \left(\log \sum_{\mathbf{s},\mathbf{a}} \omega(\mathbf{s},\mathbf{a}) \exp\left(Q(\mathbf{s},\mathbf{a})\right) - \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}\left[Q(\mathbf{s},\mathbf{a})\right]\right) + \widetilde{\mathcal{E}}\left(Q, \hat{\mathcal{B}}^\pi \hat{Q}\right) \tag{6}$$

This result is intuitively reasonable, as we are penalizing more on Q-values with larger $\omega(\mathbf{s},\mathbf{a})$, corresponding to those high dynamics-gap samples. Then the next question is, how do we practically evaluate $\omega(\mathbf{s},\mathbf{a})$ given offline dataset and the online simulation rollouts. Specifically, we can measure the dynamics gap between real dynamics and the simulator on a state-action pair $(\mathbf{s},\mathbf{a})$ as $u(\mathbf{s},\mathbf{a}) := D_{KL}(P_{\widehat{\mathcal{M}}}(\mathbf{s}'|\mathbf{s},\mathbf{a})\|P_{\mathcal{M}}(\mathbf{s}'|\mathbf{s},\mathbf{a})) = \mathbb{E}_{s'\sim P_{\widehat{\mathcal{M}}}} \log(P_{\widehat{\mathcal{M}}}(\mathbf{s}'|\mathbf{s},\mathbf{a})/P_{\mathcal{M}}(\mathbf{s}'|\mathbf{s},\mathbf{a}))$. $\omega(s,a)$ can thus be represented as a normalized distribution of $u$, i.e., $\omega(\mathbf{s},\mathbf{a}) := u(\mathbf{s},\mathbf{a})/\sum_{\mathbf{s},\mathbf{a}} u(\mathbf{s},\mathbf{a})$. Now, the

challenge is how to evaluate the dynamics ratio $P_{\widehat{\mathcal{M}}}/P_{\mathcal{M}}$. Note that according to Bayes' rule:

$$\begin{aligned}\frac{P_{\widehat{\mathcal{M}}}\left(\mathbf{s}'|\mathbf{s},\mathbf{a}\right)}{P_{\mathcal{M}}\left(\mathbf{s}'|\mathbf{s},\mathbf{a}\right)} &= \frac{p\left(\mathbf{s}'|\mathbf{s},\mathbf{a},\text{sim}\right)}{p\left(\mathbf{s}'|\mathbf{s},\mathbf{a},\text{real}\right)} = \frac{p\left(\text{sim}|\mathbf{s},\mathbf{a},\mathbf{s}'\right)}{p\left(\text{sim}|\mathbf{s},\mathbf{a}\right)} \Big/ \frac{p\left(\text{real}|\mathbf{s},\mathbf{a},\mathbf{s}'\right)}{p\left(\text{real}|\mathbf{s},\mathbf{a}\right)} \\ &= \frac{p\left(\text{sim}|\mathbf{s},\mathbf{a},\mathbf{s}'\right)}{p\left(\text{real}|\mathbf{s},\mathbf{a},\mathbf{s}'\right)} \Big/ \frac{p\left(\text{sim}|\mathbf{s},\mathbf{a}\right)}{p\left(\text{real}|\mathbf{s},\mathbf{a}\right)} = \frac{1-p\left(\text{real}|\mathbf{s},\mathbf{a},\mathbf{s}'\right)}{p\left(\text{real}|\mathbf{s},\mathbf{a},\mathbf{s}'\right)} \Big/ \frac{1-p\left(\text{real}|\mathbf{s},\mathbf{a}\right)}{p\left(\text{real}|\mathbf{s},\mathbf{a}\right)}\end{aligned} \tag{7}$$

Hence, we can approximate $p\left(\text{real}|\mathbf{s},\mathbf{a},\mathbf{s}'\right)$ and $p\left(\text{real}|\mathbf{s},\mathbf{a}\right)$ with a pair of discriminators $D_{sas}\left(\mathbf{s},\mathbf{a},\mathbf{s}'\right)$ and $D_{sa}\left(\mathbf{s},\mathbf{a}\right)$ respectively that are optimized with standard cross-entropy loss between real offline data and the simulated samples as in DARC [Eysenbach *et al.*, 2020].

### 4.3 Fixing Bellman Error due to Dynamics Gap

Due to the existence of a dynamics gap in the simulator, directly computing the Bellman error for simulated samples in the online replay buffer $B$ (i.e., $\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim B}[(Q-\hat{\mathcal{B}}^{\pi}\hat{Q})(\mathbf{s},\mathbf{a})]^2$) is problematic. This is because the next state $\mathbf{s}'$ comes from a potentially biased simulated dynamics $P_{\widehat{M}}$, which can result in miscalculation of the target Q values in Bellman updates. Ideally, we wish the next state $\mathbf{s}'$ come from the real dynamics $P_{\mathcal{M}}$, however, such a $\mathbf{s}'$ given an arbitrary state-action pair $(\mathbf{s},\mathbf{a})\sim B$ is not obtainable during training. To fix this issue, we reuse the previously evaluated dynamics ratio in Eq. (7) as an importance sampling weight, and introduce the following modified Bellman error formulation for both offline dataset $\mathcal{D}$ and simulated data $B$:

$$\begin{aligned}\widetilde{\mathcal{E}}\left(Q,\hat{\mathcal{B}}^{\pi}\hat{Q}\right) &= \frac{1}{2}\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim\mathcal{D}}\left[\left(Q-\hat{\mathcal{B}}^{\pi}\hat{Q}\right)(\mathbf{s},\mathbf{a})\right]^2 + \frac{1}{2}\mathbb{E}_{\mathbf{s},\mathbf{a}\sim B}\mathbb{E}_{\mathbf{s}'\sim p_{\mathcal{M}}}\left[\left(Q-\hat{\mathcal{B}}^{\pi}\hat{Q}\right)(\mathbf{s},\mathbf{a})\right]^2 \\ &= \frac{1}{2}\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim\mathcal{D}}\left[\left(Q-\hat{\mathcal{B}}^{\pi}\hat{Q}\right)(\mathbf{s},\mathbf{a})\right]^2 + \frac{1}{2}\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim B}\left[\frac{P_{\mathcal{M}}(\mathbf{s}'|\mathbf{s},\mathbf{a})}{P_{\widehat{\mathcal{M}}}(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left(Q-\hat{\mathcal{B}}^{\pi}\hat{Q}\right)(\mathbf{s},\mathbf{a})\right]^2\end{aligned} \tag{8}$$

### 4.4 Practical Implementation

Combining Eq. (6) and (8), we obtain the final dynamics-aware policy evaluation procedure for H2O, which enables adaptive value regularization on high dynamics-gap samples as well as more reliable Bellman updates. The policy $\hat{\pi}$ can be learned by maximizing the value function following the conventional policy improvement procedure as in Eq. (4).

For practical considerations, we introduce two relaxations in our implementation. First, in Eq. (6), we need to sample from the whole state-action space to compute the weighted average of exponentiated Q-values, which can be impractical and unnecessary since not all state-action pairs are of interest for policy learning. Instead, we approximate this value using the state-action samples in the mini-batch of the simulated replay buffer $B$. Second, the evaluation of the KL divergence in $u(\mathbf{s},\mathbf{a}) = \mathbb{E}_{s'\sim P_{\widehat{\mathcal{M}}}}\log(P_{\widehat{\mathcal{M}}}(\mathbf{s}'|\mathbf{s},\mathbf{a})/P_{\mathcal{M}}(\mathbf{s}'|\mathbf{s},\mathbf{a}))$ involves sampling next states $\mathbf{s}'$ from $P_{\widehat{\mathcal{M}}}(\cdot|\mathbf{s},\mathbf{a})$, which can be infeasible given a black-box simulator. We take a simplified approach that approximates the expected value by averaging over $n$ random samples from a Gaussian distribution $\mathcal{N}(\mathbf{s}',\hat{\Sigma}_{\mathcal{D}})$, where $\hat{\Sigma}_{\mathcal{D}}$ is the covariance matrix of states computed from the real offline dataset. Although this treatment is simple, we find it highly effective and produces good performance in empirical experiments. The complete algorithm of H2O is presented in Appendix B.1.

## 5 Interpretation of Dynamics-Aware Policy Evaluation

In this section, we analyze upon H2O and provide intuition on how it works when combining both offline and online learning. Interestingly, we find that the final form of dynamics-aware policy evaluation as given in Eq. (6) and (8) actually leads to a equivalent adaptive reward adjustment term on the Q function, which depends on the dynamics gap distribution $\omega(\mathbf{s},\mathbf{a})$ and the state-action distribution of the dataset $d_M^{\pi_\beta}(\mathbf{s},\mathbf{a})$. To show this, note that the weighted log-sum-exp term $\log\sum_{\mathbf{s},\mathbf{a}}\omega(\mathbf{s},\mathbf{a})\exp\left(Q(\mathbf{s},\mathbf{a})\right) = \log\mathbb{E}_{\mathbf{s},\mathbf{a}\sim\omega(\mathbf{s},\mathbf{a})}\exp(Q(\mathbf{s},\mathbf{a}))$ in Eq. (6) can be bound as follows (see Appendix A.1 for detailed proof):

$$\mathbb{E}_{\mathbf{s},\mathbf{a}\sim\omega(\mathbf{s},\mathbf{a})}[Q(\mathbf{s},\mathbf{a})] \le \log\mathbb{E}_{\mathbf{s},\mathbf{a}\sim\omega(\mathbf{s},\mathbf{a})}\exp(Q(\mathbf{s},\mathbf{a})) \le \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\omega(\mathbf{s},\mathbf{a})}[Q(\mathbf{s},\mathbf{a})] + \frac{\text{Var}_{\omega}[\exp(Q(\mathbf{s},\mathbf{a}))]}{2\exp(2Q_{min})} \tag{9}$$

where we denote the range of learned Q values as $[Q_{min},Q_{max}]$, and $\text{Var}_{\omega}[\exp(Q(\mathbf{s},\mathbf{a}))]$ denotes the variance of $\exp(Q(\mathbf{s},\mathbf{a}))$ under samples drawn from distribution $\omega(\mathbf{s},\mathbf{a})$. The LHS inequality is a result of Jensen's inequality, and the RHS inequality is a direct results from Liao and Berg

[2018]. Above inequalities suggest that $\mathbb{E}_{\mathbf{s},\mathbf{a}\sim\omega(\mathbf{s},\mathbf{a})}[Q(\mathbf{s},\mathbf{a})]$ can be a reasonable approximation of our weighted log-sum-exp term if the Q function takes large values and the gap between $\exp(Q_{min})$ and $\sqrt{Var_{\omega}[\exp(Q(\mathbf{s},\mathbf{a}))]}$ is not too large. This can be practically satisfied if we let $\gamma \to 1$ and design the range of reward function and the above conditions properly.

With this approximation, we can consider the following approximated policy evaluation objective based on Eq. (6) and (8), which offers a much cleaner form for analysis:

$$\min_{Q} \beta \left( \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\omega(\mathbf{s},\mathbf{a})}[Q(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}\left[Q(\mathbf{s},\mathbf{a})\right] \right) + \widetilde{\mathcal{E}}\left(Q, \hat{\mathcal{B}}^{\pi}\hat{Q}^{k}\right) \tag{10}$$

Assuming that the Q function is tabular, we can find the Q function corresponding to the new objective by following an approximate dynamic programming approach and differentiating Eq. (10) with respect to $Q^k$ in iteration $k$ (see Appendix A.1 for details):

$$\hat{Q}^{k+1}(\mathbf{s},\mathbf{a}) = (\hat{\mathcal{B}}^{\pi}Q^{k})(\mathbf{s},\mathbf{a}) - \beta \left[ \frac{\omega(\mathbf{s},\mathbf{a}) - d_M^{\pi_\beta}(\mathbf{s},\mathbf{a})}{P_{\mathcal{M}}(\mathbf{s}'|\mathbf{s},\mathbf{a})\left(d_M^{\pi_\beta}(\mathbf{s},\mathbf{a}) + d_{\widehat{M}}^{\pi}(\mathbf{s},\mathbf{a})\right)} \right] \tag{11}$$

where $d_M^{\pi_\beta}(\mathbf{s},\mathbf{a})$ and $d_{\widehat{M}}^{\pi}(\mathbf{s},\mathbf{a})$ are state-action marginal distributions under behavioral policy $\pi_\beta$ and the learned policy $\pi$ respectively. If we represent the second term in the RHS of above equation as $\nu(\mathbf{s},\mathbf{a})$, we can see that $\nu(\mathbf{s},\mathbf{a})$ can be perceived as an adaptive reward adjustment term, which penalizes or boosts the reward at a state-action pair $(\mathbf{s},\mathbf{a})$ according to the relative difference between dynamics gap distribution $\omega(\mathbf{s},\mathbf{a})$ and marginal state-action distribution of the offline dataset $d_M^{\pi_\beta}(\mathbf{s},\mathbf{a})$. Specifically, we can make following interesting observations:

- If $\omega(\mathbf{s},\mathbf{a}) > d_M^{\pi_\beta}(\mathbf{s},\mathbf{a})$, then $\nu(\mathbf{s},\mathbf{a})$ serves as a reward penalty. This corresponds to the case that either the state-action pair $(\mathbf{s},\mathbf{a})$ has large dynamics gap, or it belongs to OOD or relatively low data density areas ($d_M^{\pi_\beta}(\mathbf{s},\mathbf{a}) = 0$ or $< \omega(\mathbf{s},\mathbf{a})$). Under such cases, we push down the Q values and adopt conservatism similar to the typical treatment in offline RL settings [Kumar *et al.*, 2020]. The higher the dynamics gap is (larger $\omega$), the more we penalize on the Q function.
- If $\omega(\mathbf{s},\mathbf{a}) \leq d_M^{\pi_\beta}(\mathbf{s},\mathbf{a})$, then we are evaluating at some good data areas with low dynamics gap or have more information from offline data to control the potential dynamics gap in the simulated online samples. Under such cases, $\nu(\mathbf{s},\mathbf{a})$ serves as a reward boost term and adopts optimism. Although it overestimates the Q values, we argue that this can be beneficial. As it can promote exploration in these "good" data regions and potentially help reduce variance during training.

With the adaptive reward adjustment term $\nu$ defined in Eq. (11), we can further show that the value function $V(\mathbf{s})$ is underestimated at high dynamics gap areas, which allows for safe policy learning in our offline-and-online setting. Detailed theoretical analysis can be found in Appendix A.2.

# 6 Experiments

In this section, we present empirical evaluations of H2O. We start by describing our experimental environment setups and the cross-domain RL baselines for comparison with our method. We then evaluate H2O against the baseline methods in simulation environments and on a real wheel-legged robot. Detailed ablation studies and empirical analyses of H2O are also reported. Detailed experiment settings and additional results can be found in Appendix B.2 and C.

## 6.1 Experimental Environment Setups

**Simulation-based experiments.** We conduct simulation-based experiments in the MuJoCo physics simulator [Todorov *et al.*, 2012]. In particular, we construct three new simulation task environments with intentionally introduced dynamics gaps upon the original MuJoCo-HalfCheetah task environment (serve as the real environment) by modifying the dynamics parameters: (1) **Gravity**: applying 2 times the gravitational acceleration in the simulation dynamics; (2) **Friction**: using 0.3 times the friction coefficient to make the agent harder to maintain balance; (3) **Joint Noise**: adding a random noise sampled from $\mathcal{N}(0,1)$ on every dimension of the action space, mimicking a system with large control noise. As for the offline dataset, we use the datasets of the corresponding task from standard offline RL benchmark D4RL [Fu *et al.*, 2020]. Specifically, we consider the Medium, Medium-Replay and Medium-Expert datasets, as in typical real-world scenarios, we do not use a random policy or or do not have an expert policy for system control. The online training is performed on the modified simulation environment, and we evaluate the performance of the average return of the learned policy in the original unchanged MuJoCo environment.

Table 1: Average returns for MuJoCo-HalfCheetah tasks. Results are averaged over 5 random seeds.

| Dataset | Unreal dynamics | SAC(sim) | CQL(real) | DARC | DARC+ | H2O |
|---|---|---|---|---|---|---|
| Medium | Gravity | 4513±513 | 6066±73 | 5011±456 | 5706±440 | **6777±490** |
| | Friction | 2684±2646 | 6066±73 | 6113±104 | 6047±112 | **6230±916** |
| | Joint Noise | 4137±805 | 6066±73 | 5484±171 | 5314±520 | **6895±434** |
| Medium Replay | Gravity | 4513±513 | 5947±59 | 5105±460 | 4958±540 | **6813±289** |
| | Friction | 2684±2646 | 5947±59 | 5503±263 | 5288±100 | **5928±896** |
| | Joint Noise | 4137±805 | 5947±59 | 5137±225 | 5230±209 | **6747±427** |
| Medium Expert | Gravity | 4513±513 | 3029±1134 | **4759±353** | 72±109 | **4707±779** |
| | Friction | 2684±2646 | 3029±1134 | **9038±1480** | 7989±3999 | 6745±562 |
| | Joint Noise | 4137±805 | 3029±1134 | 3822±811 | 733±767 | **5280±1329** |

**Real-world experiments.** We also evaluate the performance of H2O on a real wheel-legged robot that moves on a pair of wheels to keep balanced, as illustrated in Fig. 1(a). The state space of the robot is a quadratic-tuple $\mathcal{S} = (\theta, \dot{\theta}, x, \dot{x})$, where $\theta$ denotes the forward tilt angle of the body, $x$ is the displacement of the robot, $\dot{\theta}$ and $\dot{x}$ are the angular and linear velocity respectively. The control action is the torque $\tau$ of the motors at the two wheels. Here we construct two tasks for real-world validation: (1) **Standing still**: We want to make the robot stand still and not move or fall down; (2) **Moving straight**: We want to control the robot to move forward at a target velocity $v$ while keeping it's balance. We record 100,000 human controlled transitions with different reward functions for both the two tasks, which serve as the offline dataset used in RL training.

The control task is relatively challenging, as the robot has to use two wheels to keep balance and can easily fall to the ground. The simulation environment is constructed based on Isaac Gym [Makoviychuk *et al.*, 2021], depicted in Fig. 1(b). The simulation environment suffers from intricate dynamics gaps. Notably, the electric motors in the real robot bear dead-zones, which can lead to observational errors, whereas the simplified dynamics in simulation fails to capture this complicated situation. Also, the rolling friction applied on wheels is well-simulated, but the sliding friction cannot be modeled realistically in Isaac Gym. Furthermore, the friction coefficient between wheels and the ground as well as its wear-and-tear, are not possible to be configured exactly as in the real-world environment. All these dynamics gaps can cause serious sim-to-real transfer issues when deploying the RL policy learned in simulation to the real world scenarios.

### 6.2 Baselines

As H2O is the first RL framework under the hybrid offline-and-online setting, we can only compare it with some purely online or offline RL algorithms, as well as representative methods that partially incorporate offline data in online or offline policy learning in a less integrated manner, which are:

- **SAC (sim)** [Haarnoja *et al.*, 2018]: the SOTA off-policy online RL algorithm, which is trained within the imperfect simulator and evaluated in the original simulation or real-world environments.
- **CQL (real)** [Kumar *et al.*, 2020]: the representative offline RL algorithm using value regularization. We run CQL on the D4RL and real-world robotic control dataset, which does not impacted by the dynamics gap, but suffers from limited state-action space coverage in offline dataset.
- **DARC** [Eysenbach *et al.*, 2020]: DARC uses a reward correction term derived from a pair of binary discriminators to optimize the policy within an imperfect simulator. We train these discriminators with real and simulation samples, and learn policy within the simulation.
- **DARC+**: a variant of DARC, in which the real offline data are not only utilized for discriminator training, but also used in policy evaluation and improvement. DARC+ showcases the model behavior when we naïvely combine offline and online policy learning under the DARC framework.

### 6.3 Comparative Evaluation of H2O in Simulation and Real-World Experiments

In Table 1, we present the comparative results of H2O and other baselines on the simulation-based experiments. Notably, H2O achieves the best performance in almost all tasks compared with the baselines. Due to the relatively large dynamics gap, directly training SAC in simulation yields poorest performance in most of the tasks. Offline RL method CQL achieves reasonable performance as compared with the online RL counterpart, which achieves second highest scores in Medium and

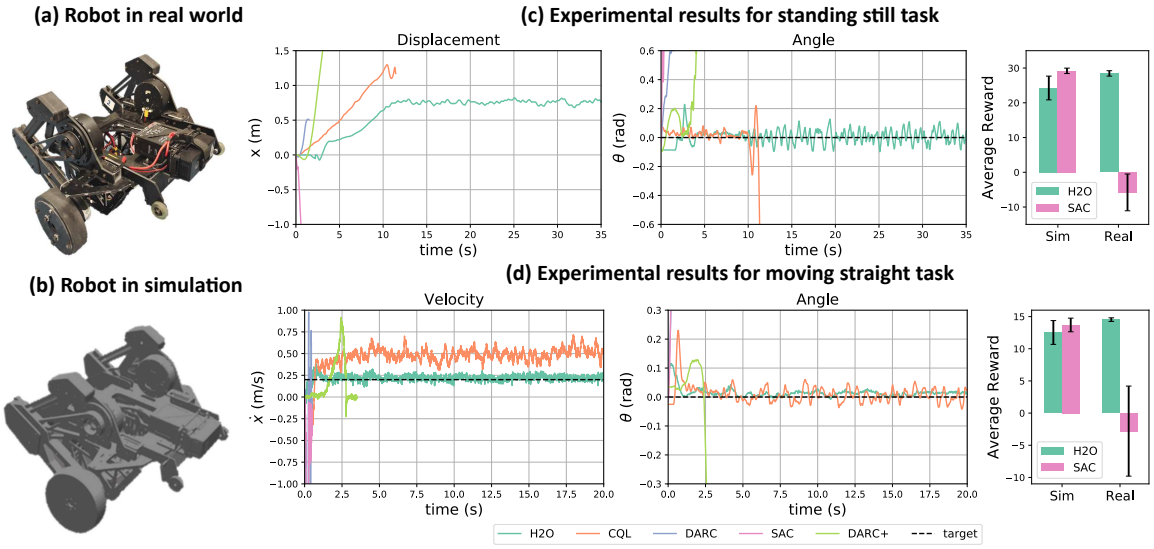


Figure 1: Real-world validation on a wheel-legged robot

Medium Replay tasks. DARC performs surprisingly very well in Medium Expert-Friction task, perhaps due to specific dataset and dynamics gap properties. DARC+ shows inferior performance compared with DARC in most tasks, indicating the insufficiency of naïvely combining offline and online policy learning. Nevertheless, H2O consistently provides superior performance under different offline dataset and dynamics gap settings, demonstrating the effectiveness of our proposed dynamics-aware policy evaluation scheme.

In the real-world experiments[1], H2O manifests overwhelmingly better transfer ability on both the standing still and moving straight tasks. In the standing still task, as demonstrated in the comparative curves in Fig. 1(c), the robot with H2O policy remains steady after 11 seconds (s), while CQL bumps into the ground and goes out of control at 12s. DARC, DARC+ and SAC are not able to keep the robot balanced and quickly fail after initialization In the moving straight task demonstrated in Fig. 1(d), H2O achieves great control performance to keep balanced and closely follow the target velocity $v = 0.2$m/s, while CQL exceeds $v$ by a fairly large margin, nearly doubling the desired target velocity. Additionally, the angle $\theta$ illustrates that the robot controlled by H2O runs more smoothly than CQL. Once again, DARC, DARC+ and SAC fail at the beginning. Interestingly, we plot the evaluated average returns in both simulation and the real-world robot for SAC and H2O, revealing that achieving high performance in the simulation environment provides no guarantee for real-world transferability, if the simulator has large sim-to-real gap. It is possible that policies with lower performance in the biased simulation could perform better in real-world scenarios. This suggests the potential need of reconsidering the widely adopted simulation-based evaluation or verification in mission-critical tasks, such as in autonomous driving and industrial control, that overly trust simulators with dynamics gaps can lead to serious consequences.

### 6.4 Ablation Study

In this section, we investigate the impact of each design component of H2O in the HalfCheetah-Medium Replay-Gravity task. In Table 2, we compare the performances of the variants of H2O by replacing adaptive value regularization weight $\omega$ into an uniform value, removing the importance weight in modified Bellman Error formulation $\widetilde{\mathcal{E}}$ or even removing the entire value regularization part from the framework. Without the adaptive value regularization, "H2O-a" demonstrates lower performance since the Q values at high dynamics-gap samples are not properly regularized during training. Without the dynamics ratio as importance weight to fix the Bellman error, "H2O-dr", "H2O-a-dr", and "H2O-reg-dr" shows severe performance deterioration against their calibrated counterparts, which signifies the importance to correctly handle the Bellman updates when involving multiple sources of data. Removing the entire value regularization part from H2O leads to substantial performance degradation. However, it is noteworthy that "H2O-reg-dr" outperforms "H2O-dr", again

[1]Please refer to the supplementary video for actual performance of H2O and baselines on the real robot.

Table 2: Ablations on H2O. “-a" denotes no adaptive weights $\omega$. “-dr" denotes no dynamics ratio to fix the Bellman error. “-reg" stands for no value regularization applied in the framework.

| | H2O | H2O-a | H2O-dr | H2O-a-dr | H2O-reg | H2O-reg-dr | CQL | SAC |
|---|---|---|---|---|---|---|---|---|
| Adaptive $\omega$ | ✓ | ✗ | ✓ | ✗ | - | - | - | - |
| Uniform $\omega$ | ✗ | ✓ | ✗ | ✓ | - | - | - | - |
| Modified $\widetilde{\mathcal{E}}\left(Q, \hat{\mathcal{B}}^{\pi}\hat{Q}\right)$ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | - | - |
| Average Return | **6642±324** | 6568±194 | 4866±303 | 5381±406 | 6396±175 | 5322±345 | 5947±59 | 5001±548 |

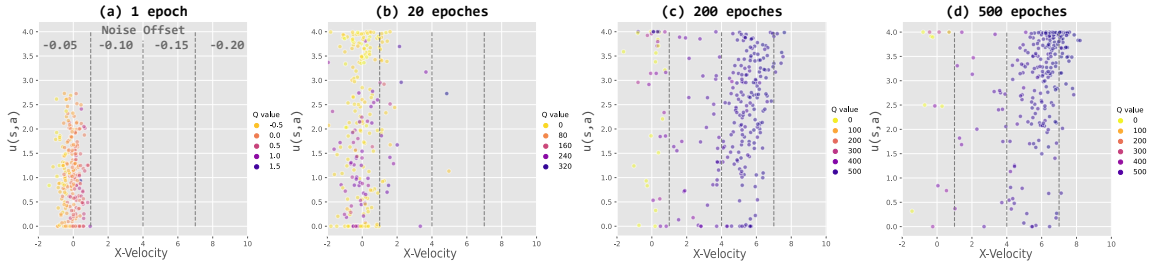


Figure 2: The dynamics gap measure $u(\mathbf{s}, \mathbf{a})$ evaluated during the training process

suggests that fixing the Bellman error due to dynamics gap is very important in hybrid offline-and-online policy learning.

To further investigate the quality of the dynamics gap measure $u(\mathbf{s}, \mathbf{a})$ in H2O as well as its impact on the adaptive value regularization during the learning process, we construct a new test environment based on HalfCheetah. We apply specific noise offsets to actions based on X-velocity (see Fig. 2) in the simulation, such that larger X-velocity bears larger noise offsets. We plot the dynamics gap measure $u(\mathbf{s}, \mathbf{a})$ and the corresponding Q values $Q(\mathbf{s}, \mathbf{a})$ for each simulated sample $(\mathbf{s}, \mathbf{a})$ from a batch in some stages of the learning process. Notably, in HalfCheetah task, higher X-velocity corresponds to higher rewards hence corresponds to higher Q-values, which at the same time may suffer from more severe dynamics gap. We observe that as the course of learning proceeds, the dynamics gap measures $u$ in H2O indeed captures the higher dynamics gap information at later part of the training, and works as expected to penalize the Q values on high dynamics gap samples.

## 7 Conclusion and Perspectives

In this paper, we propose the dynamics-aware Hybrid Offline-and-Online RL (H2O) framework to combine offline and online RL policy learning, while simultaneously addressing the sim-to-real dynamics gaps in an imperfect simulator. H2O introduces a dynamics-aware policy evaluation scheme, which adaptively penalizes the Q values as well as fixes the Bellman error on simulated samples with large dynamics gaps. This scheme can be shown equivalent to a special kind of reward adjustment, which places reward penalties on high dynamics-gap samples, but boost reward in regions with low dynmaics gaps and abundant offline data. Under H2O, both the offline dataset and the imperfect simulator are fully used and complement with each other to extract the maximum information for policy learning. In particular, the limited size and state-action space coverage of offline dataset are greatly supplemented by unrestricted simulated samples; the dynamics gaps in simulated samples can also be alleviated during policy learning given correct dynamics information from the real data. Through extensive simulation and real-world experiments, we demonstrate the superior performance of H2O against other cross-domain online and offline RL methods.

Online and offline RL have been two separately studied fields in the past, both face a set of specific challenges. Few attempts have been made to organically combining these two different RL paradigms. H2O provides a brand new hybrid offline-and-online RL paradigm, which shows promising results when leveraging offline data and simulation-based online learning in an integrated framework. Still, there are several future works can be done to further enhance the model performance, such as adopting a less conservative offline RL backbone algorithm as compared to CQL, and incorporating better dynamics gap quantification methods that do not need approximation. We hope the insights developed in this work can shed light on future hybrid RL algorithm design, which can potentially provide better RL solutions for practical deployment.

## References


Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. 2

Andrew G Barto, Richard S Sutton, and Charles W Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, (5):834–846, 1983. 3

Alex Bewley, Jessica Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, and Alex Kendall. Learning to drive from simulation without real world labels. In *2019 International conference on robotics and automation (ICRA)*, pages 4818–4824. IEEE, 2019. 1

Jacob Buckman, Carles Gelada, and Marc G Bellemare. The importance of pessimism in fixed-dataset policy optimization. In *International Conference on Learning Representations*, 2020. 3

Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019. 2

Benjamin Eysenbach, Shreyas Chaudhari, Swapnil Asawa, Sergey Levine, and Ruslan Salakhutdinov. Off-dynamics reinforcement learning: Training for transfer with domain classifiers. In *International Conference on Learning Representations*, 2020. 2, 5, 7

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020. 6

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021. 2, 3

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1587–1596, 2018. 1

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062. PMLR, 2019. 2, 3

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870, 2018. 7

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2020. 3

Ilya Kostrikov, Rob Fergus, Jonathan Tompson, and Ofir Nachum. Offline reinforcement learning with fisher divergence critic regularization. In *International Conference on Machine Learning*, pages 5774–5783. PMLR, 2021. 2, 3

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *Deep RL Workshop NeurIPS 2021*, 2021. 2, 3

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, pages 11761–11771, 2019. 1, 2, 3

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, 2020. 2, 3, 6, 7

Alex X Lee, Coline Manon Devin, Yuxiang Zhou, Thomas Lampe, Konstantinos Bousmalis, Jost Tobias Springenberg, Arunkumar Byravan, Abbas Abdolmaleki, Nimrod Gileadi, David Khosid, et al. Beyond pick-and-place: Tackling robotic stacking of diverse shapes. In *5th Annual Conference on Robot Learning*, 2021. 1, 2

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. 1, 2

Jinning Li, Chen Tang, Masayoshi Tomizuka, and Wei Zhan. Dealing with the unknown: Pessimistic offline reinforcement learning. In *Conference on Robot Learning*, pages 1455–1464. PMLR, 2022. 3

JG Liao and Arthur Berg. Sharpening jensen's inequality. *The American Statistician*, 2018. 5

Jinxin Liu, Zhang Hongyin, and Donglin Wang. Dara: Dynamics-aware reward augmentation in offline reinforcement learning. In *International Conference on Learning Representations*, 2021. 2

Lennart Ljung. System identification. In *Signal analysis and prediction*, pages 163–173. Springer, 1998. 2

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021. 7

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 1

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018. 1

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3803–3810, 2018. 2

Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. Epopt: Learning robust neural network policies using model ensembles. *arXiv preprint arXiv:1610.01283*, 2016. 2

Kanishka Rao, Chris Harris, Alex Irpan, Sergey Levine, Julian Ibarz, and Mohi Khansari. Rl-cyclegan: Reinforcement learning aware simulation-to-real. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11157–11166, 2020. 1

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017. 1

Richard S Sutton, Andrew G Barto, et al. Introduction to reinforcement learning. 1998. 3

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)*, pages 23–30. IEEE, 2017. 1

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. 6

Quan Vuong, Sharad Vikram, Hao Su, Sicun Gao, and Henrik I Christensen. How to pick the domain randomization parameters for sim-to-real transfer of reinforcement learning policies? *arXiv preprint arXiv:1903.11774*, 2019. 2

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019. 3

Teng Xiao and Donglin Wang. A general offline reinforcement learning framework for interactive recommendation. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, volume 2021, 2021. 2

Haoran Xu, Xianyuan Zhan, and Xiangyu Zhu. Constraints penalized q-learning for safe offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 2, 3

Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. *arXiv preprint arXiv:1702.02453*, 2017. 2

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. In *Neural Information Processing Systems (NeurIPS)*, 2020. 3

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in Neural Information Processing Systems*, 34, 2021. 3

Xianyuan Zhan, Haoran Xu, Yue Zhang, Xiangyu Zhu, Honglei Yin, and Yu Zheng. Deepthermal: Combustion optimization for thermal power generating units using offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 2, 3

Xianyuan Zhan, Xiangyu Zhu, and Haoran Xu. Model-based offline planning with trajectory pruning. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence, IJCAI-ECAI 2022*, 2022. 3

# Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 7, Conclusions and Perspectives
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section **??**5 and Appendix A
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section 5 and Appendix A
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix B.2
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 6
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix B.2
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]