Appendix

A Theoretical Derivation and Analysis of H2O

In this section, we provide the derivation of the closed-form solution to Eq. (5) and the detailed theoretical analysis of H2O as discussed in Section 5. In particular, we first provide an approximated dynamics-aware policy evaluation objective of H2O based on Eq. (6), after we claim the derivation of the original one, and (8) in the main text, which offers a much cleaner form for theoretical analysis. The approximation can be tight under some reasonable problem setups. Based on the approximated objective, we can show that the dynamics-aware policy evaluation is equivalent to adding an adaptive reward adjustment term to the original MDP. We can further show that this leads to an underestimated value function $V(\mathbf{s})$ in high dynamics-gap areas, which achieves desirable learning behavior under our offline-and-online policy learning setting involving imperfect simulators.

A.1 Derivation of the closed-form solution for $d^\phi(\mathbf{s}, \mathbf{a})$

The Lagrangian of the primal optimization problem in Eq. (5) is given by:

$$\mathcal{L}\left(d^\phi; \mu, \boldsymbol{\lambda}\right) = \mathbb{E}_{\mathbf{s},\mathbf{a}\sim d^\phi} Q(\mathbf{s}, \mathbf{a}) - D_{KL}\left(d^\phi(\mathbf{s}, \mathbf{a}) \| \omega(\mathbf{s}, \mathbf{a})\right) + \mu\left(\sum_{\mathbf{s},\mathbf{a}} d^\phi(\mathbf{s}, \mathbf{a}) - 1\right) + \boldsymbol{\lambda}(\mathbf{s}, \mathbf{a}) d^\phi(\mathbf{s}, \mathbf{a}) \tag{12}$$

where μ is the Lagrangian dual variable for the normalization constraint $\sum_{\mathbf{s},\mathbf{a}} d^\phi(\mathbf{s}, \mathbf{a}) = 1$, and $\boldsymbol{\lambda}(\mathbf{s}, \mathbf{a})$ is the Lagrangian dual variable for positivity constraints on d^ϕ. Setting the gradient of the Lagrangian w.r.t. d^ϕ to 0 yields:

$$d^{\phi *}(\mathbf{s}, \mathbf{a}) = \omega(\mathbf{s}, \mathbf{a}) \exp[Q(\mathbf{s}, \mathbf{a}) + \mu \cdot \mathbf{1} + \boldsymbol{\lambda}(\mathbf{s}, \mathbf{a}) - \mathbf{1}], \forall (\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A} \tag{13}$$

If we assume the dynamics gap distribution $\omega(s, a) > 0$ holds for all state-action pairs, hence $d^{\phi *}(\mathbf{s}, \mathbf{a}) > 0$ trivially holds, which implies $\boldsymbol{\lambda}(\mathbf{s}, \mathbf{a}) = 0$ for each state-action pair according to the complementary slackness condition. Utilizing the normalization constraint $\sum_{\mathbf{s},\mathbf{a}} d^\phi(\mathbf{s}, \mathbf{a}) = 1$, we have:

$$\sum_{\mathbf{s},\mathbf{a}} d^{\phi *}(\mathbf{s}, \mathbf{a}) = \sum_{\mathbf{s},\mathbf{a}} \omega(\mathbf{s}, \mathbf{a}) \exp[Q(\mathbf{s}, \mathbf{a})] e^{(\mu - 1)\cdot \mathbf{1}} = 1 \tag{14}$$

Solving $e^{(\mu-1)\cdot\mathbf{1}}$ using Eq. (14) and plugging it into Eq. (13), we then obtain the final closed-form solution for $d^{\phi *}$:

$$d^{\phi *}(\mathbf{s}, \mathbf{a}) = \frac{\omega(\mathbf{s}, \mathbf{a}) \exp[Q(\mathbf{s}, \mathbf{a})]}{\sum_{\mathbf{s},\mathbf{a}} \omega(\mathbf{s}, \mathbf{a}) \exp[Q(\mathbf{s}, \mathbf{a})]} \propto \omega(\mathbf{s}, \mathbf{a}) \exp[Q(\mathbf{s}, \mathbf{a})] \tag{15}$$

Note that if we plug the exact solution $d^{\phi *}$ and the regularization term $\mathcal{R}(d^\phi) = -D_{KL}(d^\phi || \omega)$ in Eq. (4), we have:

$$\begin{aligned}
&\beta\left[\mathbb{E}_{\mathbf{s},\mathbf{a}\sim d^\phi(\mathbf{s},\mathbf{a})}[Q(\mathbf{s}, \mathbf{a}) - \log(d^\phi(\mathbf{s}, \mathbf{a})/\omega(\mathbf{s}, \mathbf{a}))] - \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}[Q(\mathbf{s}, \mathbf{a})]\right] + \widetilde{\mathcal{E}}\left(Q, \hat{\mathcal{B}}^\pi \hat{Q}\right) \\
&= \beta\left[\mathbb{E}_{\mathbf{s},\mathbf{a}\sim d^\phi(\mathbf{s},\mathbf{a})}\left[Q(\mathbf{s}, \mathbf{a}) - \log\left(\frac{\exp[Q(\mathbf{s}, \mathbf{a})]}{\sum_{\mathbf{s},\mathbf{a}} \omega(\mathbf{s}, \mathbf{a}) \exp[Q(\mathbf{s}, \mathbf{a})]}\right)\right] - \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}[Q(\mathbf{s}, \mathbf{a})]\right] + \widetilde{\mathcal{E}}\left(Q, \hat{\mathcal{B}}^\pi \hat{Q}\right) \\
&= \beta\left[\mathbb{E}_{\mathbf{s},\mathbf{a}\sim d^\phi(\mathbf{s},\mathbf{a})}\left[\log\left(\sum_{\mathbf{s},\mathbf{a}} \omega(\mathbf{s}, \mathbf{a}) \exp[Q(\mathbf{s}, \mathbf{a})]\right)\right] - \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}[Q(\mathbf{s}, \mathbf{a})]\right] + \widetilde{\mathcal{E}}\left(Q, \hat{\mathcal{B}}^\pi \hat{Q}\right) \\
&= \beta\left[\log \sum_{\mathbf{s},\mathbf{a}} \omega(\mathbf{s}, \mathbf{a}) \exp[Q(\mathbf{s}, \mathbf{a})] - \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}[Q(\mathbf{s}, \mathbf{a})]\right] + \widetilde{\mathcal{E}}\left(Q, \hat{\mathcal{B}}^\pi \hat{Q}\right)
\end{aligned} \tag{16}$$

In the last step, we can remove $\mathbb{E}_{\mathbf{s},\mathbf{a}\sim d^\phi(\mathbf{s},\mathbf{a})}$ as $\log \sum_{\mathbf{s},\mathbf{a}} \omega(\mathbf{s}, \mathbf{a}) \exp[Q(\mathbf{s}, \mathbf{a})]$ is a value that does not depend on $(\mathbf{s}, \mathbf{a})$ any more. The objective in the last equation is exactly the objective we have used in Eq. (6). Note that the derivation is based on the exact solution of $d^{\phi *}$ rather than the proportional one.

A.2 Adaptive Reward Adjustment Under Dynamics-Aware Policy Evaluation

The weighted log-sum-exp term in Eq. (6) is quite cumbersome to work with. To draw more insights from H2O as well as to make the analysis simpler, we will derive a reasonably approximate version of the original dynamics-aware policy evaluation objective. Before presenting the final form, we first introduce the following lemma from [Liao and Berg, 2018]:

Lemma 1. *(A sharpened version of Jensen's inequality [Liao and Berg, 2018]). Let X be a one-dimensional random variable with $P(X \in (a,b)) = 1$, where $-\infty \leq a < b \leq \infty$. Let $\varphi(x)$ be a twice differentiable function on (a,b), we have:*

$$\inf_{x\in(a,b)} \frac{\varphi''(x)}{2} var(X) \leq E[\varphi(X)] - \varphi(E(X)) \leq \sup_{x\in(a,b)} \frac{\varphi''(x)}{2} var(X) \tag{17}$$

Define $[Q_{min}, Q_{max}]$ as the range of learned Q-values (we assume $r > 0$, hence $Q_{min} > 0$), and $\mathrm{Var}_\omega[\exp(Q(\mathbf{s},\mathbf{a}))]$ is the variance of $\exp(Q(\mathbf{s},\mathbf{a}))$ under $(\mathbf{s},\mathbf{a})$ samples drawn from the distribution $\omega(\mathbf{s},\mathbf{a})$. With Lemma 1, we can have the following result on the weighted log-sum-exp term:

Corollary 1. *The weighted log-sum-exp term $\log \sum_{\mathbf{s},\mathbf{a}} \omega(\mathbf{s},\mathbf{a}) \exp\left(Q(\mathbf{s},\mathbf{a})\right)$ can be reasonably approximated by $\mathbb{E}_{\mathbf{s},\mathbf{a}\sim\omega(\mathbf{s},\mathbf{a})}[Q(\mathbf{s},\mathbf{a})]$ if $\sqrt{Var_\omega[\exp(Q(\mathbf{s},\mathbf{a}))]}/\exp(Q_{min})$ is small. In particular, following bounds on the weighted log-sum-exp term holds:*

$$\mathbb{E}_{\mathbf{s},\mathbf{a}\sim\omega(\mathbf{s},\mathbf{a})}[Q(\mathbf{s},\mathbf{a})] \leq \log \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\omega(\mathbf{s},\mathbf{a})} \exp(Q(\mathbf{s},\mathbf{a})) \leq \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\omega(\mathbf{s},\mathbf{a})}[Q(\mathbf{s},\mathbf{a})] + \frac{Var_\omega[\exp(Q(\mathbf{s},\mathbf{a}))]}{2\exp(2Q_{min})} \tag{18}$$

Proof. The LHS inequality is a straightforward result of Jensen's inequality:

$$\log \sum_{\mathbf{s},\mathbf{a}} \omega(\mathbf{s},\mathbf{a}) \exp\left(Q(\mathbf{s},\mathbf{a})\right) = \log \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\omega(\mathbf{s},\mathbf{a})} \exp(Q(\mathbf{s},\mathbf{a})) \geq \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\omega(\mathbf{s},\mathbf{a})}[Q(\mathbf{s},\mathbf{a})]$$

The RHS inequality directly follows from Lemma 1 by setting $\varphi(\cdot) = \log(\cdot)$, $x = \exp(Q(\mathbf{s},\mathbf{a}))$ with $(\mathbf{s},\mathbf{a})$ sampled from the dynamics gap distribution $\omega(\mathbf{s},\mathbf{a})$, $a = Q_{min}$ and $b = Q_{max}$:

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

where we use the relationship $\inf_{x\in(a,b)} \log(x)''/2 = \inf_{x\in(a,b)} -1/2x^2 = \inf_{\exp(Q)\in(\exp(Q_{min}),\exp(Q_{max}))} -1/2\exp(2Q) = -1/2\exp(2Q_{min})$. □

Corollary 1 suggests that $\mathbb{E}_{\mathbf{s},\mathbf{a}\sim\omega(\mathbf{s},\mathbf{a})}[Q(\mathbf{s},\mathbf{a})]$ can be a reasonable approximation of the weighted log-sum-exp term in Eq. (6) if $\sqrt{\mathrm{Var}_\omega[\exp(Q(\mathbf{s},\mathbf{a}))]}/\exp(Q_{min})$ is relatively small compared to $\mathbb{E}_{\mathbf{s},\mathbf{a}\sim\omega(\mathbf{s},\mathbf{a})}[Q(\mathbf{s},\mathbf{a})]$. This can be practically satisfied if we properly design the value range of reward function $r \in [R_{min}, R_{max}]$ and the episode done condition to control the gap between Q_{max} and Q_{min}, as well as let $\gamma \to 1$ to encourage the learned Q function in taking large values.

With the above approximation, we instead consider the following approximated policy evaluation objective of H2O based on Eq. (6) and (8) in the main text, which is much easier for our analysis:

$$\begin{aligned}\min_Q \beta \left(\mathbb{E}_{\mathbf{s},\mathbf{a}\sim\omega(\mathbf{s},\mathbf{a})}[Q(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\mathbf{s},\mathbf{a}\sim\mathcal{D}}\left[Q(\mathbf{s},\mathbf{a})\right]\right) + \frac{1}{2}\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim\mathcal{D}}\left[\left(Q - \hat{\mathcal{B}}^\pi\hat{Q}\right)(\mathbf{s},\mathbf{a})\right]^2 \\ + \frac{1}{2}\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim B}\left[\frac{P_{\mathcal{M}}(\mathbf{s}'|\mathbf{s},\mathbf{a})}{P_{\widehat{\mathcal{M}}}(\mathbf{s}'|\mathbf{s},\mathbf{a})}\left(Q - \hat{\mathcal{B}}^\pi\hat{Q}\right)(\mathbf{s},\mathbf{a})\right]^2\end{aligned} \tag{19}$$

Note that in the original objective of H2O (Eq. (6)), we perform minimization on the weighted soft-maximum of Q-values, whereas in the approximated objective, we are minimizing on the weighted mean of Q-values. Both objectives penalize Q-values with high dynamics gap density ω, but the original objective can be seen as minimizing the worst-case objective as in robust optimization [Bertsimas *et al.*, 2011], which often leads to better results under uncertainty. Additional empirical comparisons between the original H2O policy evaluation objective in Eq. (6) and the approximated version in Eq. (19) are provided in Appendix C.1. Despite the differences, the approximated objective provides a much cleaner form of our analysis, from which we can gain some insights into how H2O works by combining both offline and online learning.

Consider the tabular and approximate dynamic programming setting, by setting the derivative of the approximated objective Eq. (19) with respect to Q^k to zero in iteration k, we have

$$\begin{aligned}
&\beta\left(\omega(\mathbf{s},\mathbf{a}) - d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a})\right) + \mathbb{E}_{\mathbf{s}'} d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a}) P_\mathcal{M}(\mathbf{s}'|\mathbf{s},\mathbf{a}) \left(Q - \hat{\mathcal{B}}^\pi \hat{Q}\right)(\mathbf{s},\mathbf{a}) \\
&\qquad + \mathbb{E}_{\mathbf{s}'} d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s},\mathbf{a}) P_{\widehat{\mathcal{M}}}(\mathbf{s}'|\mathbf{s},\mathbf{a}) \cdot \frac{P_\mathcal{M}(\mathbf{s}'|\mathbf{s},\mathbf{a})}{P_{\widehat{\mathcal{M}}}(\mathbf{s}'|\mathbf{s},\mathbf{a})} \cdot \left(Q - \hat{\mathcal{B}}^\pi \hat{Q}\right)(\mathbf{s},\mathbf{a}) = 0 \\
\Rightarrow\ &\beta\left(\omega(\mathbf{s},\mathbf{a}) - d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a})\right) + \mathbb{E}_{\mathbf{s}'} P_\mathcal{M}(\mathbf{s}'|\mathbf{s},\mathbf{a}) \left(d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a}) + d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s},\mathbf{a})\right) \left(Q - \hat{\mathcal{B}}^\pi \hat{Q}\right)(\mathbf{s},\mathbf{a}) = 0 \\
\Rightarrow\ &\hat{Q}^{k+1}(\mathbf{s},\mathbf{a}) = (\hat{\mathcal{B}}^\pi \hat{Q}^k)(\mathbf{s},\mathbf{a}) - \beta \left[\frac{\omega(\mathbf{s},\mathbf{a}) - d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a})}{d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a}) + d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s},\mathbf{a})}\right] = (\hat{\mathcal{B}}^\pi \hat{Q}^k)(\mathbf{s},\mathbf{a}) - \beta\nu(\mathbf{s},\mathbf{a}) \quad (20)
\end{aligned}$$

where $d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a})$ and $d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s},\mathbf{a})$ are state-action marginal distributions under behavioral policy $\pi_\mathcal{D}$ and the learned policy π respectively. Note that due to the introduction of the dynamics ratio $P_\mathcal{M}/P_{\widehat{\mathcal{M}}}$ as importance sampling weight, both the Bellman operators for Bellman errors of the offline dataset $\mathcal{D}$ and simulated data B are now defined on the true dynamics $\mathcal{M}$ (i.e., $\hat{\mathcal{B}}^\pi \hat{Q} = \hat{\mathcal{B}}^\pi_\mathcal{M} \hat{Q}$), hence can be combined. This is not possible without $P_\mathcal{M}/P_{\widehat{\mathcal{M}}}$, as we face $\hat{\mathcal{B}}^\pi_\mathcal{M}\hat{Q}$ and $\hat{\mathcal{B}}^\pi_{\widehat{\mathcal{M}}}\hat{Q}$ for the Bellman error of offline data $\mathcal{D}$ and simulated data B respectively.

We can see that $\nu(\mathbf{s},\mathbf{a}) = \frac{\omega(\mathbf{s},\mathbf{a}) - d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a})}{d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a}) + d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s},\mathbf{a})}$ in Eq.(20) corresponds to an adaptive reward adjustment term, which penalizes or boosts the reward at a state-action pair $(\mathbf{s},\mathbf{a})$ depending on the relative difference between $\omega(\mathbf{s},\mathbf{a})$ and $d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a})$. If $\omega(\mathbf{s},\mathbf{a}) > d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a})$ (high dynamics gap or OOD areas), ν acts as a reward penalty on state-action pair $(\mathbf{s},\mathbf{a})$; otherwise, it serves as a reward boost term to encourage exploration in low dynamics-gap areas. For more discussion on ν, please refer to Section 5 in the main text.

A.3 Lower Bounded Value Estimates on High Dynamics-Gap Samples

In this section, we show that the approximated dynamics-aware policy evaluation of H2O in Eq. (19) learns an underestimated value function on high dynamics-gap areas. We first discuss in Theorem 1, the case under the absence of sampling error, and further incorporate sampling error in Theorem 2 under some mild assumptions. All theoretical analyses are given under tabular settings. In continuous control problems, the continuous state-action space can be approximately discretized into a tabular form, but the tabular form may be large.

Theorem 1. *Assuming no sampling error in the empirical Bellman updates ($\hat{\mathcal{B}}^\pi = \mathcal{B}^\pi$), the value function learned via Eq. (19) lower bounds the actual value function (i.e., $\hat{V}^\pi(\mathbf{s}) \leq V^\pi(\mathbf{s})$) in high dynamics-gap data regions, which satisfy $\sum_\mathbf{a} \omega(\mathbf{s},\mathbf{a}) > \sum_\mathbf{a} d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a})\zeta^\pi(\mathbf{s},\mathbf{a})$, with $\zeta^\pi(\mathbf{s},\mathbf{a})$ given as:*

$$\zeta^\pi(\mathbf{s},\mathbf{a}) = \frac{d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s}) \max_\mathbf{a}\{\frac{\pi_\mathcal{D}(\mathbf{a}|\mathbf{s})}{\pi(\mathbf{a}|\mathbf{s})}\} + d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s})}{d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s})\frac{\pi_\mathcal{D}(\mathbf{a}|\mathbf{s})}{\pi(\mathbf{a}|\mathbf{s})} + d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s})} \geq 1, \quad \forall \mathbf{s},\mathbf{a} \tag{21}$$

Proof. Note that in Eq. (20), for state-action pairs with $\omega(\mathbf{s},\mathbf{a}) > d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a})$, we have potential underestimation on the Q-function. But this condition can be over-restrictive. We derive a more relaxed lower bounded condition for the state-value function $\hat{V}^\pi(\mathbf{s})$.

Taking expectation of Eq. (20) over the distribution $\pi(\mathbf{a}|\mathbf{s})$, we have

$$\begin{aligned}
\hat{V}^{k+1}(\mathbf{s}) &= (\hat{\mathcal{B}}^\pi \hat{V}^k)(\mathbf{s}) - \mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}\left[\beta\left(\frac{\omega(\mathbf{s},\mathbf{a}) - d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a})}{d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a}) + d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s},\mathbf{a})}\right)\right] \\
&= (\hat{\mathcal{B}}^\pi \hat{V}^k)(\mathbf{s}) - \beta \cdot \underbrace{\sum_\mathbf{a} \frac{\omega(\mathbf{s},\mathbf{a}) - d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s},\mathbf{a})}{d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s})\frac{\pi_\mathcal{D}(\mathbf{a}|\mathbf{s})}{\pi(\mathbf{a}|\mathbf{s})} + d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s})}}_{\Delta(\mathbf{s})}
\end{aligned} \tag{22}$$

where $d^{\pi_\mathcal{D}}_\mathcal{M}(\mathbf{s})$ and $d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s})$ are state marginal distributions of behavior policy $\pi_\mathcal{D}$ and the learned policy π. Above condition implies that the value iterates on states with $\Delta(\mathbf{s}) > 0$ will incur some

underestimation, i.e., $\hat{V}^{k+1}(\mathbf{s}) < (\hat{\mathcal{B}}^{\pi}\hat{V}^{k})(\mathbf{s})$. We are interested to find how does this underestimation correspond to the extent of dynamics gaps in these states. Note that

$$\sum_{\mathbf{a}} \frac{\omega(\mathbf{s},\mathbf{a})}{d^{\pi_{\mathcal{D}}}_{\mathcal{M}}(\mathbf{s})\frac{\pi_{\mathcal{D}}(\mathbf{a}|\mathbf{s})}{\pi(\mathbf{a}|\mathbf{s})} + d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s})} \geq \frac{\sum_{\mathbf{a}}\omega(\mathbf{s},\mathbf{a})}{d^{\pi_{\mathcal{D}}}_{\mathcal{M}}(\mathbf{s})\max_{\mathbf{a}}\{\frac{\pi_{\mathcal{D}}(\mathbf{a}|\mathbf{s})}{\pi(\mathbf{a}|\mathbf{s})}\} + d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s})} \tag{23}$$

We can consider a relaxed condition to make $\Delta(\mathbf{s}) > 0$ by enforcing the following relationship:

$$\frac{\sum_{\mathbf{a}}\omega(\mathbf{s},\mathbf{a})}{d^{\pi_{\mathcal{D}}}_{\mathcal{M}}(\mathbf{s})\max_{\mathbf{a}}\{\frac{\pi_{\mathcal{D}}(\mathbf{a}|\mathbf{s})}{\pi(\mathbf{a}|\mathbf{s})}\} + d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s})} > \sum_{\mathbf{a}} \frac{d^{\pi_{\mathcal{D}}}_{\mathcal{M}}(\mathbf{s},\mathbf{a})}{d^{\pi_{\mathcal{D}}}_{\mathcal{M}}(\mathbf{s})\frac{\pi_{\mathcal{D}}(\mathbf{a}|\mathbf{s})}{\pi(\mathbf{a}|\mathbf{s})} + d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s})} \tag{24}$$

Above inequality leads to the following condition,

$$\sum_{\mathbf{a}} \omega(\mathbf{s},\mathbf{a}) > \sum_{\mathbf{a}} \left[d^{\pi_{\mathcal{D}}}_{\mathcal{M}}(\mathbf{s},\mathbf{a}) \cdot \frac{d^{\pi_{\mathcal{D}}}_{\mathcal{M}}(\mathbf{s})\max_{\mathbf{a}}\{\frac{\pi_{\mathcal{D}}(\mathbf{a}|\mathbf{s})}{\pi(\mathbf{a}|\mathbf{s})}\} + d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s})}{d^{\pi_{\mathcal{D}}}_{\mathcal{M}}(\mathbf{s})\frac{\pi_{\mathcal{D}}(\mathbf{a}|\mathbf{s})}{\pi(\mathbf{a}|\mathbf{s})} + d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s})} \right] = \sum_{\mathbf{a}} d^{\pi_{\mathcal{D}}}_{\mathcal{M}}(\mathbf{s},\mathbf{a})\zeta^{\pi}(\mathbf{s},\mathbf{a}) \tag{25}$$

where $\zeta^{\pi}(\mathbf{s},\mathbf{a})$ is given by Eq. (21). It can be easily observed that $\zeta^{\pi}(\mathbf{s},\mathbf{a}) \geq 1$, for $\forall(\mathbf{s},\mathbf{a})$ and only depends on the offline dataset properties ($\pi_{\mathcal{D}}$, $d^{\pi_{\mathcal{D}}}_{\mathcal{M}}$) as well as the policy properties (π, $d^{\pi}_{\widehat{\mathcal{M}}}$). This establishes a condition between the dynamics gap of a state $\sum_{\mathbf{a}}\omega(\mathbf{s},\mathbf{a})$ as well as a threshold characterized only by the offline dataset and the current policy π, $\sum_{\mathbf{a}} d^{\pi_{\mathcal{D}}}_{\mathcal{M}}(\mathbf{s},\mathbf{a})\zeta^{\pi}(\mathbf{s},\mathbf{a})$.

Now, since the exact Bellman operator $\mathcal{B}^{\pi}$ is a contraction mapping, we have:

$$||\mathcal{B}^{\pi}\hat{V}^{k+1} - \mathcal{B}^{\pi}\hat{V}^{k}|| = ||(\mathcal{B}^{\pi}\hat{V}^{k+1} - \beta\Delta) - (\mathcal{B}^{\pi}\hat{V}^{k} - \beta\Delta)|| \leq \gamma||\hat{V}^{k+1} - \hat{V}^{k}|| \tag{26}$$

which suggests that the state value function updates $\hat{V}^{k+1} = \mathcal{B}^{\pi}\hat{V}^{k} - \beta\Delta$ in Eq. (22) are also contraction mappings. Based on the contraction mapping theorem, a fixed point $\hat{V}^{\pi}$ exists when we recursively update $\hat{V}^{k}$ using Eq. (22). We can compute the fixed point of the recursion in Eq. (22), and obtain the following estimated policy value:

$$\hat{V}^{\pi}(\mathbf{s}) = V^{\pi}(\mathbf{s}) - \beta\left[(I - \gamma P^{\pi})^{-1}\Delta\right](\mathbf{s}) \tag{27}$$

in which the operator $(I - \gamma P^{\pi})^{-1}$ is positive semi-definite. For high dynamics-gap states $\mathbf{s}$ that satisfy the condition in Eq. (25), we will have $\Delta(\mathbf{s}) > 0$, thus resulting in $\hat{V}^{\pi}(\mathbf{s}) < V^{\pi}(\mathbf{s})$. □

By inspecting the form of $\zeta^{\pi}(\mathbf{s},\mathbf{a})$ in Theorem 1, we can draw some interesting insights. Note that as $\zeta^{\pi}(\mathbf{s},\mathbf{a}) \geq 1$, compared with element-wise condition $\omega(\mathbf{s},\mathbf{a}) > d^{\pi_{\mathcal{D}}}_{\mathcal{M}}(\mathbf{s},\mathbf{a})$, the new condition $\sum_{\mathbf{a}}\omega(\mathbf{s},\mathbf{a}) > \sum_{\mathbf{a}} d^{\pi_{\mathcal{D}}}_{\mathcal{M}}(\mathbf{s},\mathbf{a})\zeta^{\pi}(\mathbf{s},\mathbf{a})$ is more tolerant on simulated samples in terms of their dynamics gaps. Only samples with sufficiently large dynamics gaps will lead to underestimated state values. In particular, for state-action pairs with $d^{\pi_{\mathcal{D}}}_{\mathcal{M}}(\mathbf{s},\mathbf{a}) > 0$ and $\pi(\mathbf{a}|\mathbf{s}) \to 0$, even if the dynamics gap $\omega(\mathbf{s},\mathbf{a})$ is large, it will not necessarily lead to underestimated values. This is reasonable as the learned policy $\pi(\mathbf{a}|\mathbf{s})$ is not likely to visit these state-action pairs, the state values need not to be over pessimistically estimated. In OOD regions ($d^{\pi_{\mathcal{D}}}_{\mathcal{M}}(\mathbf{s},\mathbf{a}) = 0$), we will generally obtain underestimated state value functions, and the level of underestimation is proportional to $\sum_{\mathbf{a}}\left[\omega(\mathbf{s},\mathbf{a})/d^{\pi}_{\widehat{\mathcal{M}}}(\mathbf{s},\mathbf{a})\right]$. Again, high dynamics-gap OOD samples less visited by the policy will get heavier penalization, while frequently visited low dynamics-gap samples are less impacted. This treatment of H2O is different from most offline RL algorithms. H2O is less conservative and more adaptive with respect to the dynamics gap measures in simulated samples.

To extend the analysis to the setting with sampling error, we first make the following three assumptions. Let $\mathcal{M}$ and $\widehat{\mathcal{M}}$ be the real and simulated MDP, and $\overline{\mathcal{M}}$ be the empirical MDP under the true dynamics, we assume:

Assumption 1. *The dynamics ratio $P_{\mathcal{M}}(\mathbf{s}'|\mathbf{s},\mathbf{a})/P_{\widehat{\mathcal{M}}}(\mathbf{s}'|\mathbf{s},\mathbf{a})$ in H2O can be accurately estimated.*

Assumption 2. *The reward function $r(\mathbf{s},\mathbf{a}) \in [0, R_{max}]$ is explicitly defined and only depends on state-action pairs $(\mathbf{s},\mathbf{a})$.*

Assumption 3. *For $\forall \mathbf{s},\mathbf{a} \in \mathcal{M}$, the following relationship of the transition dynamics holds with probability greater than $1-\delta$, $\delta \in (0,1)$:*

$$||P_{\overline{\mathcal{M}}}(\mathbf{s}'|\mathbf{s},\mathbf{a}) - P_{\mathcal{M}}(\mathbf{s}'|\mathbf{s},\mathbf{a})||_1 \leq \frac{C_{P,\delta}}{\sqrt{|\mathcal{D}(\mathbf{s},\mathbf{a})|}} \tag{28}$$

Assumption 2 indicates that the reward does not depend on transition dynamics, which is a mild assumption for many real-world problems. In many cases, we use human-designed reward functions based on the currently observed state and action information from the system, rather than using the raw reward signal from a black-box environment. Assumption 3 is a commonly adopted assumption in theoretical analysis of prior works [Kumar *et al.*, 2020; Yu *et al.*, 2021; Li *et al.*, 2022b]. Moreover, as discussed in Section A.2, if the dynamics ratio can be accurately evaluated (Assumption 1), using it as an importance weight will correct the Bellman error, which makes the Bellman operators in Eq. (6) and the approximated version Eq. (19) all defined on the real dynamics $\mathcal{M}$, regardless of whether the training data is from the offline dataset $\mathcal{D}$ or simulated data B (i.e., $\hat{\mathcal{B}}^\pi \hat{Q} = \hat{\mathcal{B}}^\pi_{\mathcal{M}} \hat{Q}$).

Based on these assumptions, we can show that with high probability $\geq 1-\delta$, the difference between the empirical Bellman operator $\mathcal{B}^\pi_{\widehat{\mathcal{M}}}$ and the actual Bellman operator $\mathcal{B}^\pi_{\widehat{\mathcal{M}}}$ can be bounded as:

$$\begin{aligned} |\mathcal{B}^\pi_{\widehat{\mathcal{M}}}\hat{V}(\mathbf{s}) - \mathcal{B}^\pi_{\mathcal{M}}\hat{V}(\mathbf{s})| &= |(r(\mathbf{s},\mathbf{a}) - r(\mathbf{s},\mathbf{a})) + \gamma \sum_{\mathbf{s}'} (P_{\widehat{\mathcal{M}}}(\mathbf{s}'|\mathbf{s},\mathbf{a}) - P_{\mathcal{M}}(\mathbf{s}'|\mathbf{s},\mathbf{a}))\hat{V}(\mathbf{s}')| \\ &= \gamma |\sum_{\mathbf{s}'} (P_{\widehat{\mathcal{M}}}(\mathbf{s}'|\mathbf{s},\mathbf{a}) - P_{\mathcal{M}}(\mathbf{s}'|\mathbf{s},\mathbf{a}))\hat{V}(\mathbf{s}')| \leq \frac{\gamma C_{P,\delta} R_{max}}{(1-\gamma)\sqrt{|\mathcal{D}(\mathbf{s},\mathbf{a})|}} \end{aligned} \tag{29}$$

With the above bound, we can introduce the following theorem that incorporates the sampling error:

Theorem 2. *When considering the sampling error, the learned value function via Eq. (19) lower bounds the actual value function at high dynamics-gap states that satisfy the following condition:*

$$\sum_{\mathbf{a}} \omega(\mathbf{s},\mathbf{a}) > \sum_{\mathbf{a}} d^{\pi_\mathcal{D}}_{\mathcal{M}}(\mathbf{s},\mathbf{a})\zeta^\pi(\mathbf{s},\mathbf{a}) + \frac{\gamma C_{P,\delta} R_{max} \cdot \left[d^{\pi_\mathcal{D}}_{\mathcal{M}}(\mathbf{s}) \max_{\mathbf{a}} \{\frac{\pi_\mathcal{D}(\mathbf{a}|\mathbf{s})}{\pi(\mathbf{a}|\mathbf{s})}\} + d^\pi_{\widehat{\mathcal{M}}}(\mathbf{s}) \right]}{\beta(1-\gamma)\sqrt{|\mathcal{D}(\mathbf{s},\mathbf{a})|}} \tag{30}$$

Proof. Similar to the proof of Theorem 1, when incorporating the sampling error, the fixed point of the recursion in Eq. (22) gives the following result:

$$\hat{V}^\pi(\mathbf{s}) \leq V^\pi(\mathbf{s}) - \beta \left[(I - \gamma P^\pi)^{-1} \Delta \right](\mathbf{s}) + \left[(I-\gamma P^\pi)^{-1} \frac{\gamma C_{P,\delta} R_{max}}{(1-\gamma)\sqrt{|\mathcal{D}(\mathbf{s},\mathbf{a})|}} \right](\mathbf{s}) \tag{31}$$

Following the derivation in Eq.(22)-(25), in order to lower bound the true value function at high dynamics gap states, we need to have

$$\beta \frac{\sum_{\mathbf{a}} \omega(\mathbf{s},\mathbf{a})}{d^{\pi_\mathcal{D}}_{\mathcal{M}}(\mathbf{s}) \max_{\mathbf{a}} \{\frac{\pi_\mathcal{D}(\mathbf{a}|\mathbf{s})}{\pi(\mathbf{a}|\mathbf{s})}\} + d^\pi_{\widehat{\mathcal{M}}}(\mathbf{s})} > \beta \sum_{\mathbf{a}} \frac{d^{\pi_\mathcal{D}}_{\mathcal{M}}(\mathbf{s},\mathbf{a})}{d^{\pi_\mathcal{D}}_{\mathcal{M}}(\mathbf{s}) \frac{\pi_\mathcal{D}(\mathbf{a}|\mathbf{s})}{\pi(\mathbf{a}|\mathbf{s})} + d^\pi_{\widehat{\mathcal{M}}}(\mathbf{s})} + \frac{\gamma C_{P,\delta} R_{max}}{(1-\gamma)\sqrt{|\mathcal{D}(\mathbf{s},\mathbf{a})|}} \tag{32}$$

Re-arranging terms in the above inequality, we can easily obtain the final form in Eq. (30). □

Note that under the case with sampling error, value underestimation will occur on simulated samples with even larger dynamics gap as compared to the case without sampling error. To guarantee reliable policy update on these risky data areas, β should be reasonably large to scale down the impact due to the involvement of sampling error.

B Implementation Details and Experiment Setup

B.1 Implementation Details

The implementation details for H2O[4] and other baselines in our experiments are specified as follows:

- **Discriminators**. In H2O, DARC and DARC+, we train two discriminator networks $D_{sas}(\mathbf{s},\mathbf{a},\mathbf{s}')$ and $D_{sa}(\mathbf{s},\mathbf{a})$ to approximate $p(\text{real}|\mathbf{s},\mathbf{a},\mathbf{s}')$ and $p(\text{real}|\mathbf{s},\mathbf{a})$ respectively. We use the activation function of "2×`Tanh`" (soft clip the output values to $[-2,2]$) before the final `Softmax` layer that maps the network outputs into real/simulation domain prediction probabilities. Moreover, we follow the treatment in DARC [Eysenbach *et al.*, 2020] that add the results before the `Softmax` layer of $D_{sa}(\mathbf{s},\mathbf{a})$ to the soft-clipped results in $D_{sas}(\mathbf{s},\mathbf{a})$ to compute the final `Softmax` outputs. This enables $D_{sas}(\mathbf{s},\mathbf{a})$ to propagate gradients back through the $D_{sa}(\mathbf{s},\mathbf{a})$ network, guaranteeing

[4] Our code is available at https://github.com/t6-thu/H2O

the coupling of two discriminators. The training update frequency of the discriminators is aligned with the policy update iterations. Using the discriminator-based dynamics ratio estimation regime in Eq. (7), the KL divergence $u(\mathbf{s}, \mathbf{a})$ between the real and simulated dynamics is approximated in a sample-based manner ($N = 10$ in all the tasks):

$$
\begin{aligned}
u(\mathbf{s}, \mathbf{a}) := D_{KL}\left(P_{\widehat{\mathcal{M}}} \| P_{\mathcal{M}}\right) &\approx \sum_{\mathbf{s}_i' \sim P_{\widehat{\mathcal{M}}}\left(\mathbf{s}_i'|\mathbf{s},\mathbf{a}\right)}^{N} \log \frac{P_{\widehat{\mathcal{M}}}\left(\mathbf{s}_i'|\mathbf{s}, \mathbf{a}\right)}{P_{\mathcal{M}}\left(\mathbf{s}_i'|\mathbf{s}, \mathbf{a}\right)} \\
&= \sum_{\mathbf{s}_i' \sim P_{\widehat{\mathcal{M}}}\left(\mathbf{s}_i'|\mathbf{s},\mathbf{a}\right)}^{N} \log \left[\frac{1 - p\left(\text{real}|\mathbf{s}, \mathbf{a}, \mathbf{s}'\right)}{p\left(\text{real}|\mathbf{s}, \mathbf{a}, \mathbf{s}'\right)} \Big/ \frac{1 - p\left(\text{real}|\mathbf{s}, \mathbf{a}\right)}{p\left(\text{real}|\mathbf{s}, \mathbf{a}\right)}\right] \\
&= \sum_{\mathbf{s}_i' \sim P_{\widehat{\mathcal{M}}}\left(\mathbf{s}_i'|\mathbf{s},\mathbf{a}\right)}^{N} \log \left[\frac{1 - D_{\Phi_{sas}}(\cdot|\mathbf{s}, \mathbf{a}, \mathbf{s}')}{D_{\Phi_{sas}}(\cdot|\mathbf{s}, \mathbf{a}, \mathbf{s}')} \Big/ \frac{1 - D_{\Phi_{sa}}(\cdot|\mathbf{s}, \mathbf{a})}{D_{\Phi_{sa}}(\cdot|\mathbf{s}, \mathbf{a})}\right]
\end{aligned}
\tag{33}
$$

where $P_{\widehat{\mathcal{M}}}(\cdot|\mathbf{s}, \mathbf{a})$ is approximated by $\mathcal{N}(\mathbf{s}', \hat{\Sigma}_{\mathcal{D}})$, and $\hat{\Sigma}_{\mathcal{D}}$ is the covariance matrix of states estimated from the real dataset.

- **Replay buffer size**. For practical considerations, we approximate the log-sum-exp term in Eq. (6) $\log \sum_{\mathbf{s},\mathbf{a}} \omega(\mathbf{s}, \mathbf{a}) \exp\left(Q(\mathbf{s}, \mathbf{a})\right)$ as well as the dynamics gap distribution $\omega(\mathbf{s}, \mathbf{a}) = u(\mathbf{s}, \mathbf{a}) / \sum_{\tilde{\mathbf{s}},\tilde{\mathbf{a}}} u(\tilde{\mathbf{s}}, \tilde{\mathbf{a}})$ using mini-batch of simulated samples rather than evaluating over the whole state-action space. To achieve a reasonable approximation, the replay buffer should be set relatively large to enable the mini-batch data sampled from the replay buffer close to samples from a uniform distribution defined on state-action space. In both our simulation and real-world experiments, we make replay buffer accommodate **10x** transitions against the offline dataset $\mathcal{D}$.
- **Min Q weight**. H2O uses a fixed value for β in Eq. (6) rather than auto-tuning the min Q weight parameter α (see Eq. (3)) as in the original CQL paper [Kumar *et al.*, 2020] with an additional Lagrange threshold parameter. H2O and H2O(v) only have a single hyperparameter β, and we use only 3 values for β in different experiments (0.01 for all simulation experiments, 0.1 for **Standing Still** and 1.0 for **Moving Straight** in real-world validation). To build a cleaner comparison, we disable some icing-on-the-cake tricks (e.g., auto-tuning min Q weight) in H2O that are inherited from CQL, and so does the CQL baseline. For the CQL baseline, we follow the suggested best configurations in a public CQL implementation[5], and choose α =2.0 for Mujoco experiments and 10.0 by default in the repository to run the real-world tasks. The min Q weights of H2O and CQL are chosen to be higher values in real-world experiments as we find the value regularization terms take small values and do not offer sufficient regularization, probably due to the reasonably good quality of the real dataset.

Other network structure and model training parameters are listed in Table 3. We keep the identical setting in all compatible methods, including the activation function, double-Q function, temperature parameter in SAC, and its automatic tuning scheme, etc. Only a few adjustments (i.e. network architecture, batch size) are made different in real-world experiments to accommodate the change in state and action space dimensions in the wheel-legged robot tasks. Generally speaking, H2O needs little hyperparameter tuning when solving different tasks.

As for computing resources, we ran experiments largely on NVIDIA A100 GPUs via an internal cluster.

B.2 Real-World Experiment Setting

We use a real wheel-legged robot for real-world validation of H2O. The control action is the sum of the torque τ of the motors at the two wheels. Each of the motor output the torque of $\frac{\tau}{2}$. The control frequency of the robot is 200Hz. In the follows, we describe our two experiments in detail:

(1) **Standing still**: The state space of the robot is represented by $\mathbf{s} = (\theta, \dot{\theta}, x, \dot{x})$, where θ 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. We collect a dataset containing 100,000 human controlled transitions of

[5] https://github.com/young-geng/CQL

Table 3: Hyperparameters. "-" denotes the same choice in simulation and real-world tasks

Hyper-parameter	Value (Sim)	Value (Real)
Shared		
Number of hidden layers (Actor and Critic)	2	-
Number of hidden layers (Discriminators)	1	-
Number of hidden units per layer	256	32
Learning rates (all)	3×10^{-4}	-
Discount factor	0.99	-
Nonlinearity	ReLU	-
Nonlinearity (discriminator output layer)	2×Tanh	-
Target smoothing coefficient	5×10^{-3}	-
Batch size	256 / 512	32
Optimizer	Adam	-
H2O & H2O(v)		
KL Divergence clipping range	$[1 \times 10^{-45}, 10]$	-
Dynamics ratio on TD error clipping range	$[1 \times 10^{-5}, 1]$	-
Min Q weight β	0.01	0.1 / 1.0
Replay buffer size	$10 \times \lvert\mathcal{D}\rvert$	-
DARC & DARC+		
Δr clipping range	$[-10, 10]$	-
Replay buffer size	10^6	-
CQL		
Min Q weight α	2.0	10.0
SAC		
Replay buffer size	10^6	-

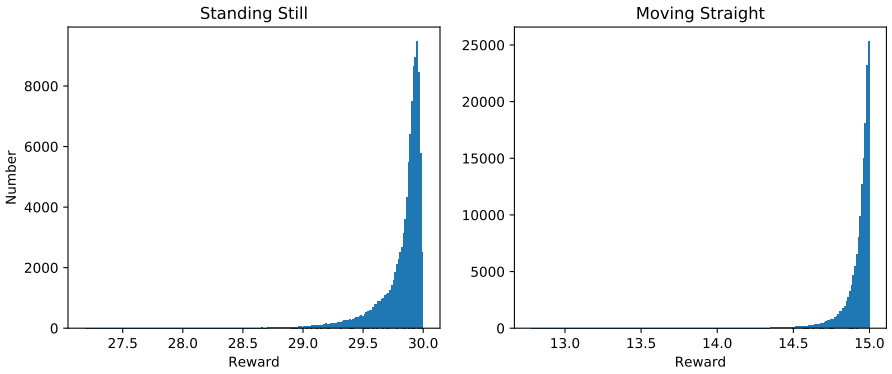

Figure 4: Single-step reward distribution in human-collected datasets of Standing Still and Moving Straight tasks

$(\mathbf{s}, a, \mathbf{s}', r, d)$, where $\mathbf{s}$ is the current state, a is the torque of the motor, $\mathbf{s}'$ is the next state, r is the reward and d is the flag of terminal. The dataset is collected near the balanced state. Since we want to keep the robot standing still, the reward r is calculated by the following formulation:

$$r = 30.0 - \theta^2 - \dot{\theta}^2 - x^2 - \dot{x}^2 - \tau^2 \tag{34}$$

When the robot stand still at the position of zero, $-\theta^2 - \dot{\theta}^2 - x^2 - \dot{x}^2$ will reach the maximum value. To prolong the motor life, we add a penalty on torque values τ^2 in the formulation. The constant 30.0 is to keep the reward to be positive. During performance evaluation, we run all algorithms for 50 epochs and report the final results in the main text.

(2) **Moving straight**: The state space of the robot is represented by $\mathbf{s} = (\theta, \dot{\theta}, \dot{x})$, which does not include x of the robot since we only want to keep the velocity of the robot stable. We collect a dataset containing 100,000 human controlled transitions of $(\mathbf{s}, a, \mathbf{s}', r, d)$. The dataset is collected

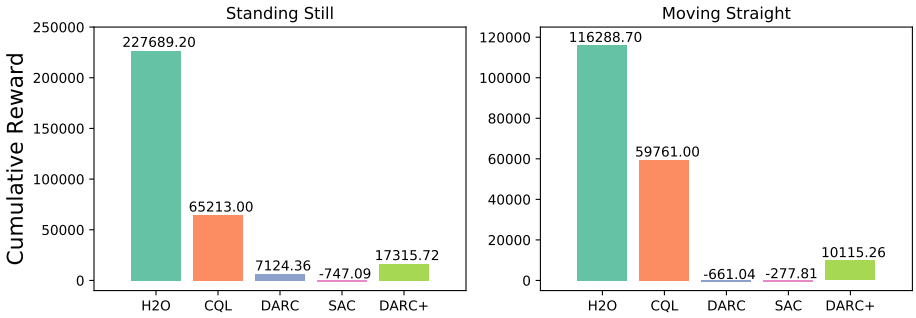

Figure 5: Cumulative rewards of different baselines recorded in real-world validation

when the robot moves forward. We want to keep the robot keep the target speed of 0.2m/s and the r is calculated by the following formulation:

$$r = 15.0 - (\dot{x} - 0.2)^2 - \tau^2 \tag{35}$$

The speed $\dot{x}$ is penalized from deviating 0.2m/s using $(\dot{x} - 0.2)^2$, and the penalization on torque values τ^2 also remains. We add 15.0 since we want to keep the reward to be positive. To illustrate the human performance of the collected datasets, we visualize the single-reward distribution of both tasks in Figure 4. During performance evaluation, we run all algorithms for 100 epochs and report the final results in the main text. In the real-world validation, we plot the cumulative reward in one recorded episode of each comparative method into Figure 5. In the standing still task, an episode is terminated by failing down or standing still for 30 seconds, while failing into the ground or moving forward steadily for 20 seconds in the moving straight task.

C Additional Ablations

C.1 Additional Comparative Results for H2O(v)

In Appendix A.2, we obtain an approximated version for our dynamics-aware policy evaluation in Eq. (19), getting rid of the cumbersome log-sum-exp term. To examine the behavior and performance of this variant (referred to as H2O(v)), we compare it with original H2O in Table 4. All the scores for H2O(v) are averaged over 3 seeds. Based on the empirical results, we find that H2O(v) exhibits a similar level of performance as compared with H2O. The original H2O generally outperforms H2O(v), while in a few cases, H2O(v) performs better. As expected, the results also show that H2O(v) is less robust compared with the original H2O in some environments (e.g., HalfCheetah with modified friction coefficient) and produces unstable scores under different datasets. This probably is due to the absence of worst-case optimization as in original H2O, in which the value regularization minimizes the weighted soft maximum of Q-values under highly dynamics-gap samples. These indicate that the approximation scheme used in Eq. (19) can be a reasonable simplification of the original H2O, which trades off some robustness with less computation complexity. To guarantee the best performance, the original H2O should be used in practical deployment.

Table 4: Comparison with H2O(v) on average returns.

Dataset	Unreal dynamics	H2O(v)	H2O	H2O - H2O(v)
Medium	Gravity	**7040±517**	**7085±416**	45
	Friction	5132±2041	**6848±445**	1716
	Joint Noise	**7116±24**	**7212±236**	96
Medium Replay	Gravity	6589±281	**6813±289**	224
	Friction	**6637±456**	5928±896	-709
	Joint Noise	**6822±45**	**6747±427**	-75
Medium Expert	Gravity	**4798±681**	**4707±779**	-91
	Friction	4726±2878	**6745±562**	2019
	Joint Noise	4623±995	**5280±1329**	657

Table 5: Comparison of H2O-KL (original version) and H2O-Reverse-KL.

HalfCheeetah_Gravity	Medium	Medium Replay	Medium Expert
H2O-KL (original version)	7085±416	6813±289	4707±779
H2O-Reverse-KL	7065±170	6476±129	4709±274

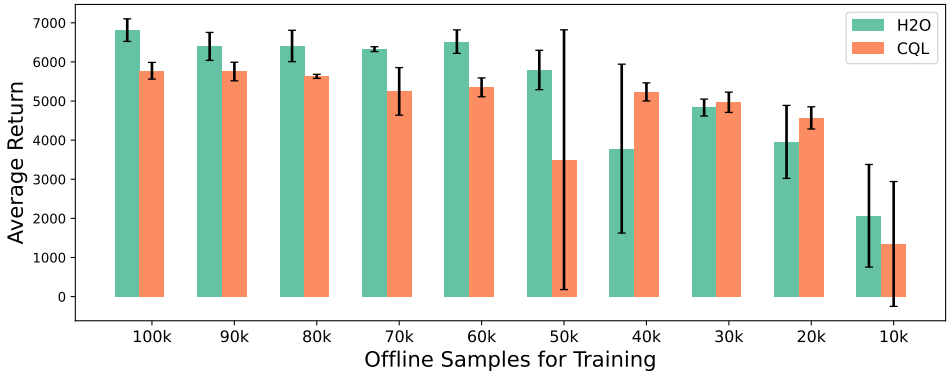

Figure 6: Average return of H2O and CQL with different amounts of offline data on the HalfCheetah Medium Replay task with modified gravity acceleration. Averaged over 3 seeds.

C.2 Additional Experiments for Reverse KL

We tested another variant of H2O which uses a learned dynamics model $P_{\widetilde{\mathcal{M}}}(\mathbf{s}_i' \mid \mathbf{s}, \mathbf{a})$ from offline data as in model-based offline RL methods , and then use the reverse-KL in Eq. (36) to estimate the dynamics gap. In this implementation, we use the deep neural network to learn a probabilistic model $P_{\widetilde{\mathcal{M}}}$ that approximates $P_{\mathcal{M}}$ similar to MOPO [Yu *et al.*, 2020] and COMBO [Yu *et al.*, 2021]. The next state s' is directly sampled from $P_{\widetilde{\mathcal{M}}}$, and we again approximate the dynamics ratio using the same two discriminators. The final performance of this variant does not show noticeable performance improvement, at the extra cost of learning an additional dynamics model. Results are presented in Table 5.

$$u(\mathbf{s}, \mathbf{a}) := D_{KL}\left(P_{\mathcal{M}} \| P_{\widehat{\mathcal{M}}}\right) \approx \sum_{\mathbf{s}_i' \sim P_{\widetilde{\mathcal{M}}}(\mathbf{s}_i' | \mathbf{s}, \mathbf{a})}^{N} \log \frac{P_{\widetilde{\mathcal{M}}}(\mathbf{s}_i' \mid \mathbf{s}, \mathbf{a})}{P_{\widehat{\mathcal{M}}}(\mathbf{s}_i' \mid \mathbf{s}, \mathbf{a})} \tag{36}$$

C.3 Ablation on Offline Data Consumption

We analyze the impacts of offline data size on the performance of H2O and offline RL method CQL in Figure 6. We conduct the experiments in the HalfCheetah environment with the D4RL Medium Replay dataset and use 2x gravity acceleration for the dynamics gap setup. The simulation buffer in H2O keeps the constant size of 1M transitions, but we gradually reduce the amount of offline data size from 100k to 10k. As illustrated in Figure 6, H2O enjoys consistently better performance than CQL when the offline data size is greater than 50k. The performance of H2O does not have noticeable deterioration when the training offline data are reduced from 90k to 60k, while the performance CQL drops with the decrease of data size. This shows the benefit of leveraging online simulation data to complement the limited offline data. However, it is also observed that H2O still needs a reasonable amount of data for reliable dynamics gap quantification. An overly small offline dataset (e.g., data size $\leq$40k) might hurt the performance of H2O as compared with directly applying offline RL methods like CQL.

D Additional Experiment Results

D.1 Additional Experiments on Walker2d

We further conduct a set of additional experiments on Walker2d with the D4RL Medium Replay dataset with various types of modified dynamics. Results are presented in Table 6 (averaged over 3 random seeds), with the same hyperparameter setting as the HalfCheetah tasks in the main paper. It is found that H2O achieves the best performance among all other baselines.

Table 6: Average returns for MuJoCo-Walker2d Medium Replay tasks. Averaged over 3 seeds.

Dataset	Unreal Dynamics	SAC	CQL	DARC	DARC+	H2O
Walker2d Medium Replay	**Gravity**	1233±841	1445±1077	1987±965	1618±1446	**2187±1103**
	Friction	2879±569	1445±1077	2518±1244	2375±579	**3656±582**
	Joint Noise	852±386	1445±1077	64±115	630±561	**2998±854**

D.2 Additional Experiments on Random Datasets

We have empirically observed in Table 1 that DARC-style methods struggle in tasks on Medium and Medium Replay datasets. It is somewhat surprising that DARC struggles with low-quality real-world data and even could not outperform CQL, but has a competitive performance on the Medium Expert dataset. A possible explanation of this might associate with the limitation in DARC's theoretical derivation, that it derives the dynamics gap-related reward penalty Δr by minimizing the gap between the policy trajectory and the real-world idealized optimal policy π^*. Thus, DARC might unleash more potential over real-world datasets with high-quality expert data theoretically. By contrast, H2O is developed under a completely different value regularization framework, without suffering from this problem. To validate the above analyses, we evaluate H2O and baselines on HalfCheetah Random dataset with various types of modified dynamics in Table 7.

It can be observed that DARC-style algorithms indeed perform badly when given low-quality data, due to their theoretical foundation of trajectory distribution divergence minimization. Again, we find H2O performs very well even given the random dataset, which greatly surpasses the performance of pure online or offline baselines. Note as the quality of the random dataset is quite poor, we halve the Min Q weight β in these tasks to reduce the impact of value regularization to encourage online exploration in the simulation environment.

Table 7: Average returns for MuJoCo-HalfCheetah Random tasks. Averaged over 3 seeds.

Dataset	Unreal Dynamics	SAC	CQL	DARC	DARC+	H2O
HalfCheetah Random	**Gravity**	4513±513	2465±180	357±617	-97±121	**4602±223**
	Friction	2684±2646	2465±180	537±250	425±99	**4862±1608**

D.3 Learning Curves

With all the comparative results in Table 1 and Table 4, we visualize the cumulative returns in the course of training in Figure 7. It is interesting to note that DARC+ (use both online and offline data for policy evaluation) performs worse in most cases as compared with DARC, and in some cases even fails completely (Gravity and Joint Noise environments under Medium Expert dataset), suggesting the necessity for carefully combining offline and online learning. It also needs to be emphasized that we accommodate DARC into our offline-and-online setting so it slightly differs from the original pure online setting as in [Eysenbach *et al.*, 2020], in which we do not allow the periodical data collection from the real world. Nevertheless, we note from the results that the proposed H2O, and even its simplified version H2O(v) outperform the baseline methods in most of the tasks, which demonstrates the effectiveness of H2O.

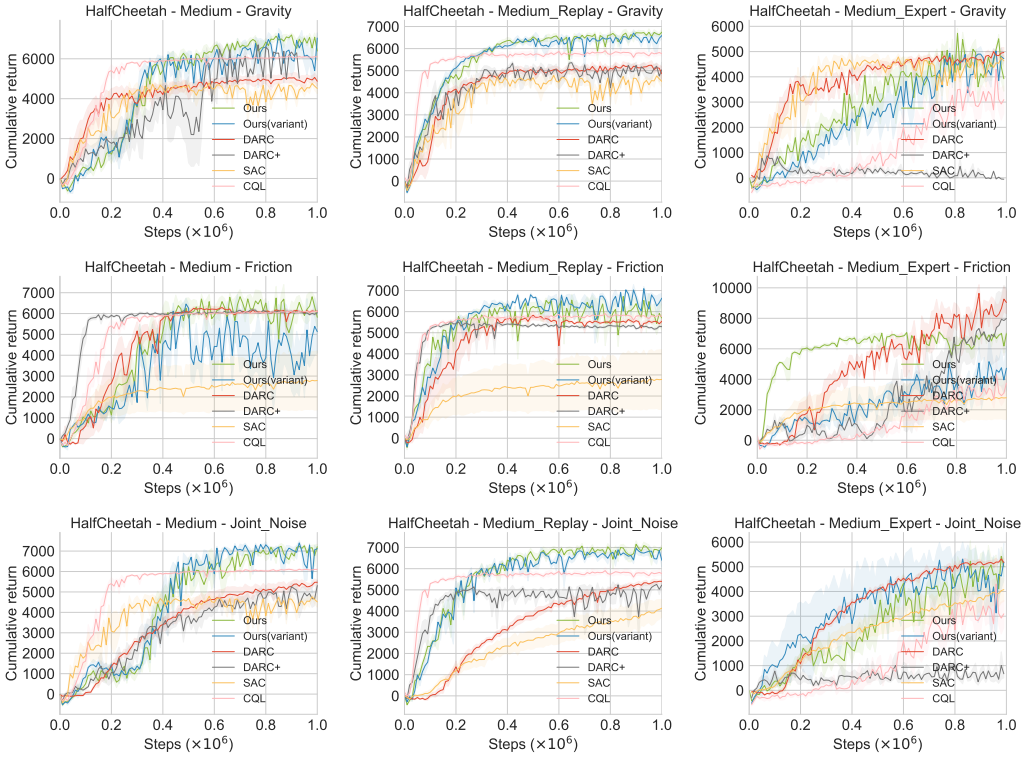

Figure 7: Corresponding learning curves for Table 1