Hybrid Cross-domain Robust Reinforcement Learning

Linh Le Pham Van[1] (✉), Minh Hoang Nguyen[1], Hung Le[1], Hung The Tran[2], and Sunil Gupta[1]

[1] Deakin Applied Artificial Intelligence Initiative, Deakin University, Australia
{l.le, s223669184, thai.le, sunil.gupta}@deakin.edu.au
[2] Hanoi University of Science and Technology, Hanoi, Vietnam
hungtt@soict.hust.edu.vn

Abstract. Robust reinforcement learning (RL) aims to learn policies that remain effective despite uncertainties in its environment, which frequently arise in real-world applications due to variations in environment dynamics. The robust RL methods learn a robust policy by maximizing value under the worst-case models within a predefined uncertainty set. Offline robust RL algorithms are particularly promising in scenarios where only a fixed dataset is available and new data cannot be collected. However, these approaches often require extensive offline data, and gathering such datasets for specific tasks in specific environments can be both costly and time-consuming. Using an imperfect simulator offers a faster, cheaper, and safer way to collect data for training, but it can suffer from dynamics mismatch. In this paper, we introduce HYDRO, the first Hybrid Cross-Domain Robust RL framework designed to address these challenges. HYDRO utilizes an online simulator to complement the limited amount of offline datasets in the non-trivial context of robust RL. By measuring and minimizing performance gaps between the simulator and the worst-case models in the uncertainty set, HYDRO employs novel uncertainty filtering and prioritized sampling to select the most relevant and reliable simulator samples. Our extensive experiments demonstrate HYDRO's superior performance over existing methods across various tasks, underscoring its potential to improve sample efficiency in offline robust RL.[1]

Keywords: Hybrid cross-domain · Distributionally robust · Offline source - Online target · Reinforcement Learning · Transfer Learning.

1 Introduction

Reinforcement learning (RL) has shown remarkable success in real-world applications [20,21], but deploying RL policies is often challenged by fluctuations in environment dynamics. Many existing methods assume consistency between training and deployment environments, an assumption frequently violated in

[1] https://github.com/linhlpv/Hybrid-Cross-domain-Robust-Reinforcement-Learning

practice due to such fluctuations. For instance, a robot operating in a dynamic real-world environment may encounter variations in mass, friction, and sensor noise compared to its training environment, leading to performance degradation [11,1]. The Robust Markov Decision Process (RMDP) framework [27] addresses this challenge by modeling uncertain test environments as a set of possible models around a training model, which is often called the nominal model. Robust RL aims to learn an optimal policy that maximizes performance under the worst-case scenario within this uncertainty set, using only the nominal model.

Since its introduction [27,28], the RMDP framework has been extensively studied in the context of planning problems [29]. Recently, many robust RL algorithms, learning robust policies from unknown nominal models, have also been proposed [34,35]. Still, all these works are limited to the online setting, where policy learning requires online interactions with the environment. Recent success in offline RL [25,23,24,**?**] has motivated the development of offline robust RL methods [37,38,39,40] to alleviate this restriction. Despite this progress, offline robust RL methods rely on large datasets, and current offline robust RL struggles when the amount of training data is reduced. As shown in Figure 1, the robustness performance of the robust RL method drops significantly when the amount of data decreases. This raises a natural question: *Can we reduce the required offline training data without sacrificing the performance of the learn policy under uncertain deployment environments?*

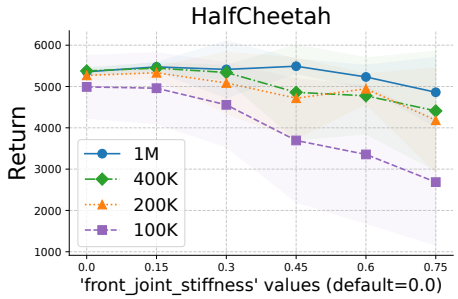


Fig. 1: Problem of existing offline robust RL model: Robustness performance drops significantly when training data decreases. Figure illustrates performance comparison under 'front_joint_stiffness' perturbation of offline robust RL [37] with different training data sizes from HalfCheetah medium dataset (D4RL).

To address the challenge of scarce data in offline RMDP, we propose a method that utilizes a simulator (source domain), an imperfect model but a faster, cheaper, and safer place to collect data for training the agent. We aim to leverage the interaction with this simulator to mitigate performance degradation caused by the limited offline nominal (target) dataset. A simulator enables unrestricted exploration and access to abundant, diverse data, potentially compensating for limited coverage of offline datasets. The potential of using an additional source

environment to bring sample efficiency has sparked research interest with numerous methods proposed in the Markov Decision Process (MDP) setting, often referred to as cross-domain RL [51]. However, naively combining source and target data may lead to performance degradation due to dynamics mismatches [6]. Thus, in the normal MDP setting, prior methods measure the domain gap directly between target and source domains using corresponding datasets. This approach cannot be directly applied to RMDPs where we optimize for *worst-case* performance while only having access to nominal model data. To the best of our knowledge, our work is the first to study the combination of online source and offline nominal dataset in the robust RL setting.

In this paper, we introduce HYDRO, the first Hybrid Cross-Domain Robust RL framework. HYDRO measures the performance gap caused by the dynamics mismatch between the simulator and the worst-case model in an uncertainty set around the nominal target model. Using this measurement, HYDRO uses a novel uncertainty based filtering mechanism and priority sampling scheme to select reliable and relevant samples from the simulator, minimizing performance degradations. Our contributions are:

- We are the first to address the Hybrid cross-domain Robust RL setting, which is a novel problem, and develop a method to improve the sample efficiency for the offline robust RL.
- We perform a theoretical analysis of our novel problem setting and use it to propose HYDRO, a practical and effective algorithm for solving this problem via novel uncertainty filtering and priority sampling.
- Through comprehensive experiments, we demonstrate that our method consistently outperforms existing approaches across diverse tasks.

2 Related Works

2.1 Offline Robust Reinforcement Learning

The RMDP framework was first introduced by [27,28] to address the parameter uncertainty problem. The initial works mainly focused on planning problems and have been well-studied [29,30,31]. Recently, robust RL in RMDP has gained much attention, with many works that have studied this problem in online [34,35], and offline [36,38,39,40] in both tabular settings [36,32,33] and large state-action space RMDP setting [39,37]. We focus on offline robust RL in large state and action spaces. Despite recent successes in such settings [37], offline robust RL methods rely on the coverage of the offline dataset. This means their robustness performance, similar to non-robust offline RL algorithms, highly depends on the amount of available offline data [37]. In practice, obtaining datasets with extensive coverage is often infeasible. Therefore, improving the sample efficiency of offline robust RL algorithms is a critical research challenge.

2.2 Cross-domain Reinforcement Learning

Cross-domain RL seeks to improve sample efficiency by leveraging data from additional source environments. Domain discrepancies can arise from differences in

the observation space [42,41], transitional dynamics [7,4]. In this work, we focus on the mismatch in the transition dynamics. Many approaches have been proposed to deal with dynamics mismatches, such as system identification [14,15,16], domain randomization [13,12,11], or meta-RL [17,18,19]. However, these methods often require environment models or careful selection of randomized parameters. Recently, several methods attempted to measure dynamics discrepancy using domain classifiers [7,4], learned dynamics models [5], or feature representation mismatch [10]. Many approaches to utilize source data have been developed, such as reward modification [7,10], support constraint [5] for purely online [8,9], purely offline [5,6], or hybrid setting [2,3]. However, these methods have primarily focused on standard RL setting. In contrast, we study the *hybrid* cross-domain problem in the *distributionally robust* RL setting, aiming to leverage online source simulators to improve sample efficiency for offline robust RL methods. To this end, we propose a novel approach using uncertainty filtering and priority sampling specifically designed for this hybrid robust setting.

2.3 Other robust RL

Recently, adversarial robust RL [46,26] and risk-sensitive RL [47,48] in online and offline settings also address robustness problems under different frameworks that are independent of RMDP. Additionally, the corruption-robust offline RL problem, where an adversary can modify a fraction of the training dataset, has been studied in [49,50]. However, their goal is still to find the optimal policy for the nominal model.

3 Preliminaries

We denote an MDP as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \gamma, r, d_0, P)$, where $\mathcal{S}, \mathcal{A}$ are the state and action spaces. The parameter $\gamma \in (0, 1)$ is the discounted factor, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, d_0 is the initial state distribution and P is the transition dynamics. We denote a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ as a map from state space $\mathcal{S}$ to a probability distribution over actions space $\mathcal{A}$. Given a policy π and a transition dynamics (model) P, we denote discounted state-action occupancy as $d_P^\pi(s, a) = (1-\gamma)\mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}(s_t = s, a_t = a)\right]$. We define value function $V^{\pi,P}$ and state-action value function $Q^{\pi,P}$ for a policy π and a transition dynamics P as follows:

$$
\begin{aligned}
V^{\pi,P}(s) &= \mathbb{E}_{\pi,P}\big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s\big], \text{ and} \\
Q^{\pi,P}(s, a) &= \mathbb{E}_{\pi,P}\big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a\big].
\end{aligned} \tag{1}
$$

We adopt the following notation: $|\mathcal{X}|$ for the cardinality of a set $\mathcal{X}$, $\Delta(\mathcal{X})$ for the set of probability distributions over $\mathcal{X}$, $(x)_+$ for $\max(x, 0)$ where $x \in \mathbb{R}$, and $\mathbb{E}_P[f(s')]$ as a short notation for $\mathbb{E}_{s' \sim P(s'|s,a)}[f(s')]$.

3.1 Distributionally Robust Reinforcement Learning

We introduce the distributionally robust MDP (RMDP) as $\mathcal{M}_r = \{\mathcal{S}, \mathcal{A}, \gamma, r, \mathcal{U}_\rho^\sigma(P^o)\}$. RMDP allows the transition dynamics to be chosen arbitrarily from a predefined uncertainty set $\mathcal{U}_\rho^\sigma(P^o)$ centered around a nominal model P^o w.r.t a metric ρ. In particular, the uncertainty set is specified as:

$$\begin{aligned} &\mathcal{U}_\rho^\sigma\left(P^o\right) := \otimes\, \mathcal{U}_\rho^\sigma\left(P^o(.|s,a)\right), \\ &\text{with } \mathcal{U}_\rho^\sigma\left(P^o(.|s,a)\right) := \left\{P(.|s,a) \in \Delta(\mathcal{S}) : \rho\left(P(.|s,a), P^o(.|s,a)\right) \le \sigma\right\}, \end{aligned} \tag{2}$$

where $\otimes$ denotes the Cartesian product. In RMDP, we focus on the worst-case performance of a policy π over all the transition models in the uncertainty set. Formally, we define the robust value functions for all $s, a \in \mathcal{S} \times \mathcal{A}$ as follows:

$$V^{\pi,\sigma}(s) := \inf_{P \in \mathcal{U}_\rho^\sigma(P^o)} V^{\pi,P}(s), Q^{\pi,\sigma}(s,a) := \inf_{P \in \mathcal{U}_\rho^\sigma(P^o)} Q^{\pi,P}(s,a). \tag{3}$$

We also have the following equations held in RMDP:

$$Q^{\pi,\sigma}(s,a) = r(s,a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}_\rho^\sigma(P^o)} \mathbb{E}_{s' \sim \mathcal{P}}\left[V^{\pi,\sigma}(s')\right]. \tag{4}$$

In RMDP, there exists at least one deterministic policy that maximizes robust value function [27]. We denote optimal robust value function, and optimal robust policy, which satisfies the following:

$$\begin{aligned} &\forall s \in \mathcal{S} : V^{*,\sigma}(s) := V^{\pi^*,\sigma}(s) = \max_\pi V^{\pi,\sigma}, \\ &\forall s, a \in \mathcal{S} \times \mathcal{A} : Q^{*,\sigma}(s,a) := Q^{\pi^*,\sigma}(s,a) = \max_\pi Q^{\pi,\sigma}(s,a). \end{aligned} \tag{5}$$

Similar to the normal MDP, we have the robust Bellman operator as follows:

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A} : \mathcal{T}^\sigma Q(s,a) := r(s,a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}_\rho^\sigma\left(P^0_{s,a}\right)} \mathcal{P}V, \quad V(s) := \max_a Q(s,a). \tag{6}$$

It is known that $\mathcal{T}^\sigma$ is a contraction mapping w.r.t. the infinity norm, and has a unique fix point solution as $Q^{*,\sigma}$. The *fitted* procedure $Q_{k+1} = \mathcal{T}^\sigma Q_k$ can be used to find the fixed point solution $Q^{*,\sigma}$.

3.2 RMDP with Offline Data

Offline Robust RL addresses learning robust policies for RMDPs using only an offline nominal dataset $\mathcal{D} = \{(s_i, a_i, r_i, s_i')\}_{i=1}^N$, where $(s_i, a_i) \sim \mu$, $s_i' \sim P^o(.|s_i, a_i)$. The fundamental challenge is that applying the robust Bellman operator in Eq (6) requires computing expectations over all dynamic models $P \in \mathcal{U}_\rho^\sigma$, while only samples from the nominal model P^o are available.

A common approach is to leverage a dual reformulation of the robust Bellman operator, replacing the expectation over all transition dynamics in $\mathcal{U}_\rho^\sigma(P^o)$ with one over nominal model P^o [37,38,39]. Specifically, [37] studied uncertainty sets with TV distance and proposed RFQI algorithm. To overcome the difficulty of estimating the robust Bellman operator, RFQI proposed the dual reformulation of the second term in the robust Bellman operator.

Proposition 1 *Let D_{TV} be the total variation distance corresponding to the TV uncertainty set $\mathcal{U}^{\sigma}_{TV}(P^o)$, then*

$$\inf_{P\in\mathcal{U}^{\sigma}_{TV}(P^o)} \mathbb{E}_P[V(s)] = -\inf_{\eta\in[0,\frac{2}{\sigma(1-\gamma)}]} (\mathbb{E}_{P^o}[(\eta - V(s))_+] + \sigma((\eta - \inf_{\tilde{s}\in\mathcal{S}} V(\tilde{s}))_+ - \eta) \tag{7}$$

They made the 'fail-state' assumption to overcome the issue of finding $\inf_{s''\in\mathcal{S}} V(s'')$ when $\mathcal{S}$ is large.

Assumption 1. *(Fail-state) The RMDP $\mathcal{M}$ has a 'fail-state' s_f, such that $\forall a \in \mathcal{A}, \forall P \in \mathcal{U}^{\sigma}_{TV}(P^o), r(s_f, a) = 0$ and $P(s_f|s_f, a) = 1$.*

Under Proposition 1 and Assumption 1, RFQI reformulates the robust Bellman operator as follows:

$$\mathcal{T}^{\sigma}Q(s,a) := r(s,a) - \gamma \inf_{\eta\in[0,\frac{2}{\sigma(1-\gamma)}]} \big(\mathbb{E}_{s'\sim P^o_{s,a}}[(\eta - V(s'))_+] - \eta(1-\sigma)\big), \tag{8}$$

where $\eta \in [0, \frac{2}{\sigma(1-\gamma)}]$ is the dual variable. To deal with large state and action space problems, RFQI frames the problem as a function approximation task. Specifically, they learn the dual variable network g_θ via the loss function $L_{dual} = \mathbb{E}_{s,a,s'\sim\mathcal{D}}[(g_\theta(s,a) - \max_{a'} Q_\phi(s',a'))_+ - (1-\sigma)g_\theta(s,a)]$. They also define the operator $\mathcal{T}^{\sigma,g}Q(s,a) = r(s,a) - \gamma(\mathbb{E}_{P^o}[(g(s,a) - \max_{a'} Q(s',a'))_+] - g(s,a)(1-\sigma))$. Then, the value function Q_ϕ is learned with the following objective $L_{RFQI} = \mathbb{E}_{s,a,s'\sim\mathcal{D}}\big[\big(\hat{\mathcal{T}}^{\sigma,g}\hat{Q}_\phi(s,a) - Q_\phi(s,a)\big)^2\big]$, where $\hat{Q}_\phi$ is the value function from the last iteration, and $\hat{\mathcal{T}}^{\sigma,g}$ is the empirical $\mathcal{T}^{\sigma,g}$ that only backs up a single sample.

4 Hybrid Cross-domain Robust Reinforcement Learning

4.1 Problem Setting

We consider the offline RMDP problem $\mathcal{M}_r$ with the uncertainty set around the nominal model P^o. For clarity, **we will refer to this nominal model as the target model throughout the paper**. In our work, we study the RMDPs with total variation (TV) uncertainty set $\mathcal{U}^{\sigma}_{TV}(P^o)$ and in the large state, action spaces setting. Specifically, we study the setting with limited offline data collected from the target model, i.e. $\mathcal{D} = \{(s_i, a_i, r_i, s'_i)\}_{i=1}^N$, where $(s_i, a_i) \sim \mu$, μ is some data generating distribution, and $s'_i \sim P^o(.|s_i, a_i)$. For computational tractability, we adopt the fail-state assumption established in function approximation settings [37]. We note that the 'fail-state' is natural in many real-world systems such as robotics [37] where the collapse of the robot can be seen as a fail state.

Along with the offline dataset from the target model, we also have access to an imperfect online simulator which we call a *source environment*. The source environment $\mathcal{M}_{src}$ shares the same state space $\mathcal{S}$, action space $\mathcal{A}$, reward function r, discounted factor γ and initialized state distribution d_0 with target domain, and only differs in its transition model, i.e. $P_{src} \neq P^o$. Our goal is to utilize online simulator $\mathcal{M}_{src}$ and limited offline target dataset $\mathcal{D}$ to learn a policy that is robust under the uncertainty set around target model P^o.

4.2 Domain Gap for Hybrid Cross-domain Robust RL

Naively combining source data to train robust target policies can lead to performance degradation due to dynamics mismatch, a challenge also noted in cross-domain RL [2,6]. Therefore, caution is necessary when utilizing source data from a different dynamics model. We begin with a theoretical analysis of the performance gap caused by this dynamics mismatch between the two domains, followed by convergence guarantees for value functions in hybrid cross-domain robust RL settings. We provide detailed proof in Appendix 2 due to the space limit.

Theorem 1 (Performance Bound). *Let $\mathcal{M}_{src}$ and $\mathcal{M}_r$ be the source MDP and the target RMDP with different dynamics P_{src} and P^o respectively. Consider the RMDP with the TV uncertainty set. Denote:*

$$A = D_{TV}(P^{\pi,\mathcal{U}^{\sigma}_{TV}(P^o)}, P^{\pi,\mathcal{U}^{\sigma}_{TV}(\hat{P}^o)}), B = \left|\mathbb{E}_{P_{src}}[V^{\pi,\sigma}_{\hat{P}^o}(s')] - \inf_{P\in\mathcal{U}^{\sigma}(\hat{P}^o)} \mathbb{E}_{P}[V^{\pi,\sigma}_{\hat{P}^o}(s')]\right|$$

where, given a policy π, $P^{\pi,\mathcal{U}^{\sigma}_{TV}(P^o)}$, $P^{\pi,\mathcal{U}^{\sigma}_{TV}(\hat{P}^o)}$ denote the worst case model w.r.t. the uncertainty set around the target model P^o and the estimated target model $\hat{P}^o$ from offline dataset $\mathcal{D}$, respectively.

The performance difference of any policy π on the source domain and the RMDP target can be bounded as follows:

$$\begin{aligned}&\mathbb{E}_{s\sim d_0}[V^{\pi,\sigma}(s)]\\ &\geq \mathbb{E}_{s\sim d_0}[V^{\pi,src}(s)] - \frac{2\gamma r_{max}}{(1-\gamma)^2}\mathbb{E}_{d^{\pi}_{P^{\pi,\mathcal{U}^{\sigma}_{TV}(\hat{P}^o)}}}\big[A\big] - \frac{\gamma}{1-\gamma}\mathbb{E}_{d^{\pi}_{P_{src}}}\big[B\big].\end{aligned} \tag{9}$$

The second term A in the Ineq (9) is caused by the offline dataset and can be reduced by offline Robust RL algorithms via pessimism [37,38,40]. The third term B represents the gap between the worst case model $P^{\pi,\mathcal{U}^{\sigma}_{TV}(\hat{P}^o)}$ and the source model P_{src}, which can be reduced by our proposed method. Specifically, Theorem 1 provides the intuition that the robust performance in the target model could be guaranteed if values of robust value function are consistent when evaluating in the source environment and the worst-case model.

Next, we analysis the value function's convergence. Denote the source dataset as D_{src}, we consider the following approach using both source and target data:

$$Q^{k+1} \leftarrow \operatorname*{argmin}_{Q} \kappa\mathbb{E}_{s,a,s'\sim\mathcal{D}}[(\hat{\mathcal{T}}^{\sigma,g}Q^k - Q)^2] + (1-\kappa)\mathbb{E}_{s,a,s'\sim\mathcal{D}_{src}}[(\hat{\mathcal{T}}Q^k - Q)^2], \tag{10}$$

where $\kappa \in [0,1]$ is the combination weight, k denotes training iteration, and $\hat{\mathcal{T}}$ is the empirical Bellman operator. We denote μ and ν as the state-action distributions of target and source datasets. We analyze the convergence guarantee of the value function. To maintain simplicity, we assume the source and the target datasets have the same state-action distribution, i.e. $\mu(s,a) = \nu(s,a), \forall s,a \in \mathcal{S}\times\mathcal{A}$. This assumption can hold easily when the source data D_{src} is generated via a simulator, as it allows flexibility in selecting the transition starting points. We note that the source and target dynamics remain distinct ($P_{src} \neq P^o$). Below, we present the convergence guarantee in the following theorem.

Theorem 2 (Convergence). *Let Q^* denote the optimal robust value function for the RMDP of the nominal model P^o, and define $Q^0 = 0$. Denote*

$$\begin{aligned} \xi &= \max_{Q} \max_{s,a \in \mathcal{S}\times\mathcal{A}} \left| \mathcal{T}^{\sigma,g} Q(s,a) - \mathcal{T}^{\sigma} Q(s,a) \right|, \\ \zeta &= \max_{Q} \max_{s,a \in \mathcal{S}\times\mathcal{A}} \left| \mathcal{T} Q(s,a) - \mathcal{T}^{\sigma} Q(s,a) \right|. \end{aligned} \tag{11}$$

Assume $\mu(s,a) = \nu(s,a), \forall s,a \in \mathcal{S}\times\mathcal{A}$, we have the following result holds:

$$\left\| Q^* - Q^{k+1} \right\|_\infty \leq \frac{\gamma^{k+1} r_{max}}{1-\gamma} + \frac{1-\gamma^{k+1}}{1-\gamma} \big(\kappa\xi + (1-\kappa)\zeta \big). \tag{12}$$

Term ξ arises from offline target dataset and can be reduced via offline robust RL algorithms, while ζ reflects the gap between worst-case target and source models. Theorem 2 guarantees that the learned Q function converges near the optimal robust Q^*, with a bound determined by the domain gaps and the combination weight. Theorem 2 suggests that target robust performance can be ensured by carefully using the *most* relevant, reliable source data that minimizes domain gaps during training, thus controlling the second term in Ineq (12).

4.3 Incorporating Source Data in Target Robust Training

In this section, we present our proposed method, which involves the priority sampling method to select relevant source data for training and uncertainty filtering to keep reliable source samples.

Source Data Selection with Priority Sampling. Motivated by the performance bound in Theorem 1, we focus on controlling the third term in Ineq (9). We propose selecting source transitions that induce minimal value discrepancies when incorporating the source environment for training. This requires computing the domain gap between transition pairs starting from the same source state-action pair (s_{src}, a_{src}). Specifically, given (s_{src}, a_{src}), we aim to estimate the domain gap between next states s', defined as follows:

$$\Lambda(s_{src}, a_{src}) = \left| \mathbb{E}_{P_{src}}[V^{\pi,\sigma}_{\hat{P}^o}(s')] - \inf_{P \in \mathcal{U}^{\sigma}(\hat{P}^o)} \mathbb{E}_{P}[V^{\pi,\sigma}_{\hat{P}^o}(s')] \right|. \tag{13}$$

Computing $\Lambda(s_{src}, a_{src})$ for a given source state-action pair (s_{src}, a_{src}) requires the worst-case model w.r.t. uncertainty set $\mathcal{U}^{\sigma}(\hat{P}^o)$, which is challenging because we only have offline dataset $\mathcal{D}$. However, using Proposition 1 and Assumption 1, we can rewrite $\Lambda(s_{src}, a_{src})$ as follows:

$$\Lambda(s_{src}, a_{src}) = \left| \mathbb{E}_{P_{src}}[V^{\pi,\sigma}_{\hat{P}^o}(s')] + \inf_{\eta \in [0, \frac{2}{\sigma(1-\gamma)}]} \left(\mathbb{E}_{\hat{P}^o}[(\eta - V^{\pi,\sigma}_{\hat{P}^o}(s'))_+] - \eta(1-\sigma) \right) \right|. \tag{14}$$

With Eq (14), given (s_{src}, a_{src}) from the source environment, we can approximately compute $\Lambda(s_{src}, a_{src})$ using the offline dataset $\mathcal{D}$. In practice, we compute $\Lambda(s_{src}, a_{src})$ for each source state-action pair (s_{src}, a_{src}) using the robust value function, estimated target model $\hat{P}^o$, and dual variable η. We learn the estimated

dynamics model $\hat{P}^o$ from offline dataset $\mathcal{D}$. For a given (s_{src}, a_{src}), we generate the next state s'_{tar} using the learned target dynamics model. The learned dual network g_θ gives approximated values for dual variable η. We then introduce a gap-measurement function approximating $\Lambda(s_{src}, a_{src})$ by estimating the value function for next state s' as $V^{\pi,\sigma}_{\hat{P}^o}(s') := Q_\phi(s', a')|_{a' \sim \pi(.|s')}$:

$$\begin{aligned}\hat{\Lambda}(s_{src}, a_{src}) = \big|Q_\phi(s'_{src}, a') + (g_\theta(s_{src}, a_{src}) - Q_\phi(s'_{tar}, a'))_+ \\ - g_\theta(s_{src}, a_{src})(1-\sigma)\big|.\end{aligned} \tag{15}$$

As our objective is to select the source samples with small gaps to tighten the bound in Ineq (9), we introduce the following *priority score function* $\psi(s,a) = 1/(1+\hat{\Lambda}(s,a))$. This priority score guides our sampling process during training. The sampling probability for each source transition is defined as:

$$p^i(s, a, s', r) = \psi^i(s,a) / \sum_k \psi^k(s,a), \tag{16}$$

where $\psi^i(s,a)$ is the priority score of the source transition.

Uncertainty Filtering. To address the uncertainty of the offline dataset, we employ a quantifier to compute uncertainty values for each source sample. Source samples with high uncertainty can lead to unreliable estimation scores, potentially hindering the training process if included. Conversely, reliable source state-action pairs with low uncertainty can serve as valuable augmented data for training the dual network g_θ. Therefore, we propose removing source samples with high uncertainty values. Inspired by prior works [45,44], we train an ensemble of N dynamics model $\{\hat{P}^o_i(s'|s,a) = \mathcal{N}(\mu_\varphi(s,a), \Sigma_\varphi(s,a))\}^N_{i=1}$. Each model in the ensemble is trained using offline target dataset $\mathcal{D}_{tar}$ via the maximum log-likelihood (MLE) as follows: $\mathcal{L}_\varphi = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}[\log \hat{P}^o(s' \mid s,a)]$.

Then, we use the max pairwise difference as our uncertainty quantifier, i.e. $u(s,a) = \max_{i,j} \|\mu^i_\varphi(s,a) - \mu^j_\varphi(s,a)\|^2$, where $\|.\|_2$ is the L2-norm and $\mu^i_\varphi(s,a)$, $\mu^j_\varphi(s,a), i, j \in \{1, \ldots, N\}$ are the mean vectors of the Gaussian distributions in the ensemble dynamics model. For the uncertainty threshold, instead of naively setting a constant threshold, we measure the uncertainty on all samples in the offline dataset $\mathcal{D}$. Then, we take the maximum uncertainty in the dataset and set the uncertainty threshold as follows: $\epsilon_u = \frac{1}{\alpha} \max_{s,a\in\mathcal{D}} u(s,a)$, where $\alpha \in \mathbb{R}_+$ is a hyperparameter to control the threshold value. Then, for any source transition $(s,a,s',r)_{src}$, we add them to the source replay buffer if its uncertainty value is less than the uncertainty threshold ϵ_u. Otherwise, we remove them.

During training, we sample a batch of source data from the source replay buffer with probabilities defined by Eq. (16). Based on Theorems 1 and 2, we prioritize the most relevant source samples while carefully controlling combination weight κ. We recompute scores for these samples, select the top-k highest-scoring samples to update the value function, and adjust priorities accordingly. The robust value function is trained using both source and target data as follows:

$$Q_\phi \leftarrow \underset{Q_\phi}{\operatorname{argmin}}\, \mathbb{E}_{s,a,s'\sim\mathcal{D}}[(\hat{\mathcal{T}}^{\sigma,g}\hat{Q}_\phi - Q_\phi)^2] + \mathbb{E}_{s,a,s'\sim\mathcal{D}_{src}}[\omega(s,a,s')(\hat{\mathcal{T}}\hat{Q}_\phi - Q_\phi)^2] \tag{17}$$

Algorithm 1 HYbrid cross-Domain RObust RL - HYDRO

Input: Source $\mathcal{M}_{src}$, offline target dataset $\mathcal{D}$, the source replay buffer $\mathcal{D}_{src} = \emptyset$, robust value function Q_ϕ and dual variable functions g_θ.
Train $\{\hat{P}_i^o(s'|s,a) = \mathcal{N}(\mu_\varphi(s,a), \Sigma_\varphi(s,a))\}_{i=1}^N\}$ via MLE on $\mathcal{D}$.
for $t = 1, \ldots,$ num iterations **do**
  for $i = 1, \ldots, h$ **do**
    Rollout with $\mathcal{M}_{src}$, compute $u_i(s_i,a_i) = \max_{j,k} \|\mu_\varphi^j(s_i,a_i) - \mu_\varphi^k(s_i,a_i)\|^2$.
    if $u_i \leq \epsilon_u$ **then**
      $\mathcal{D}_{src} \leftarrow \mathcal{D}_{src} \bigcup (s_i, a_i, r_i, s_i')$.
    end if
  end for
  Sample $\{(s,a,r,s')_{src}^i\}_{i=1}^N$ with probability $p^i(s,a,s')$ via Eq (16) from $\mathcal{D}_{src}$.
  Sample $\{(s,a,r,s')_{tar}^i\}_{i=1}^N$ uniformly from $\mathcal{D}$.
  Update transition priority in $\{(s,a,r,s')_{src}^i\}_{i=1}^N$.
  Update g_θ via Eq (18) using source, target data.
  Update Q_ϕ via Eq (17) using $\{(s,a,r,s')_{src}^i\}_{i=1}^N$ and $\{(s,a,r,s')_{tar}^i\}_{i=1}^N$.
end for
return Q_ϕ.

, where source data sampled via priority sampling after the uncertainty filter, $w(s,a,s') = \mathbb{1}(\psi(s,a) > \psi_{k\%})$, and $\mathcal{T}$ and $\mathcal{T}^\sigma$ is the normal and robust Bellman operator respectively. We use the offline target data $(s_{tar}, a_{tar}, s'_{tar})$ along with the augmented sample $(s_{src}, a_{src}, s'_{tar})$, where $s'_{tar} \sim \hat{P}^o(s'|s_{src}, a_{src})$, for training dual network g_θ. Specifically, we update the dual network g_θ as follows:

$$\begin{aligned} \theta \leftarrow \operatorname*{argmin}_\theta \mathbb{E}_{s,a,s' \sim \mathcal{D}} \left[(g_\theta(s,a) - V(s'))_+ - (1-\sigma) g_\theta(s,a) \right] \\ + \mathbb{E}_{s,a \sim \mathcal{D}_{src}, s' \sim \hat{P}^0} \left[(g_\theta(s,a) - V(s'))_+ - (1-\sigma) g_\theta(s,a) \right]. \end{aligned} \tag{18}$$

Algorithm. We summarize the above steps as our proposed method HYDRO in Algorithm 1.

5 Experiments

In this section, we present the empirical evaluation to answer the following questions: **1)** Can HYDRO enhance data efficiency and improve robustness performance in scarce data settings? **2)** Why is using HYDRO more advantageous than just naively merging the source data? **3)** How do different components of HYDRO contribute to its performance?

Environments. We conduct our experiments on three MuJoCo environments (HalfCheetah-v3, Walker2d-v3, Hopper-v3), utilizing the Medium datasets from D4RL [43] as our offline datasets [37]. To create the scarce data settings, we only use 10% of these datasets for training, i.e. 100K target transitions from D4RL. The source environments are created by modifying the morphology of the agents in the Mujoco XML file. We consider two types of modifications: single-comp, modifying a single agent component, and multi-comp, altering multiple

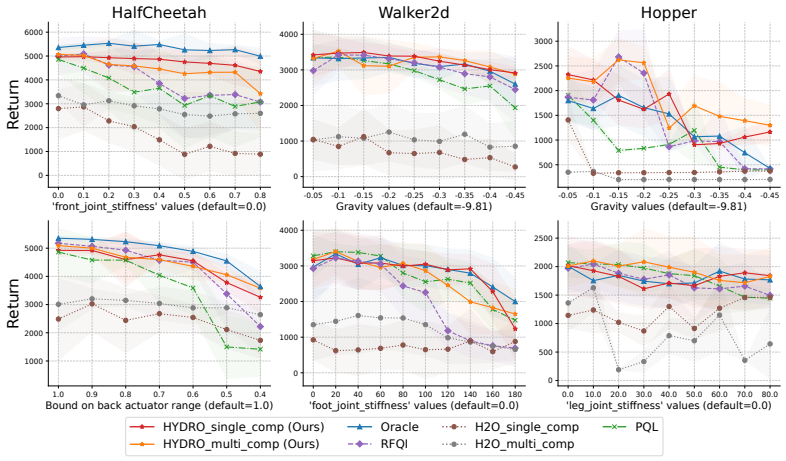


Fig. 2: Cumulative rewards of different methods in three Mujoco benchmarks under perturbation. The lines are the average returns over 30 different seeded runs, and the shaded areas represent standard deviation.

components. For robustness evaluation, we perturb each Mujoco environment by altering its physical parameters.

Metric. In the cross-domain robust setting, we evaluate the agent's return on the target domain under perturbations.

Baselines. We compare HYDRO against the following baselines: *RQFI* [37], the current state-of-the-art in offline robust RL; *H2O* [2], an only recent state-of-the-art method with available public code for non-robust hybrid cross-domain transfer that uses importance sampling to correct the dynamics shift between source and target environments; *PQL* [22], a non-robust offline RL algorithm and a practical state-of-the-art variant of FQI with neural architecture. Finally, we train RFQI agent with a full offline target dataset, which we refer to as *Oracle*. Due to limited space, we defer more details about the environment and baseline settings to Appendix 4 and provide more experiment results in Appendix 5.

5.1 Robustness Performance Evaluation

In this section, we answer the first question, showing that our method, HYDRO, can enhance data efficiency and improve robust performance in scarce data settings. Figure 2 presents the performance of our method and the baselines across three Mujoco environments under model parameter perturbations. Notably, the robust performance of RFQI degrades substantially with reduced training data. In the scarce data setting (10% target data), RFQI's performance drops notably with increasing perturbations, resembling the non-robust offline method PQL. H2O performs poorly across all settings, as its performance is heavily reliant on the amount of training target data and struggles in scarce data scenarios [2].

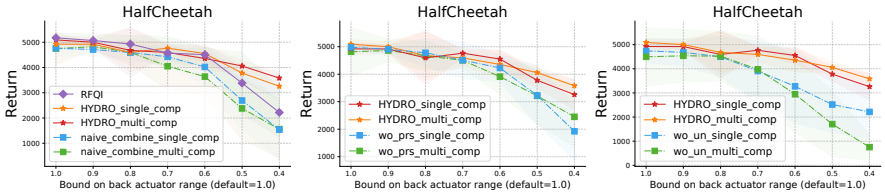
Table 1: Average returns over different environment parameter perturbations for Mujoco tasks over 30 different seeded runs. We **bold** the best results (except Oracle). "-m": multi comp, "-s" single comp, "B-" Back.

	Halfcheetah		Walker2d		Hopper	
	Front joint stiffness	B-actuator ctrlrange	Gravity	Foot joint stiffness	Gravity	Leg joint stiffness
Oracle	5332±287	4866±184	3150±481	2871±684	1316±373	1804±398
RFQI	4016±1219	4264±557	3062±537	2056±672	1374±358	1769±361
PQL	3644±1437	3508±828	2865±340	2710±506	920±151	1823±323
H2O-m	2814±606	2974±350	1043±778	1208±770	235±3	794±334
H2O-s	1710±1013	2431±854	701±585	733±655	466±47	1187±316
HYDRO-m	4456±872	**4480±350**	3225±487	2652±703	**1858±468**	**1933±468**
HYDRO-s	**4781±476**	4399±382	**3273±284**	**2790±623**	1551±309	1814±469

We believe the lack of target samples causes the inferior performance of H2O, which also was observed in [52]. In contrast, HYDRO consistently demonstrates robust performance, surpassing RFQI and non-robust methods across all tasks. While baseline methods exhibit substantial performance drops with increasing environmental changes, HYDRO maintains robustness. Table 1 presents the average returns of all methods under various environment parameter perturbations. As the table illustrates, our method improves upon RFQI across all tasks, with the most significant improvement reaching approximately 36%. Statistical testing (please see Appendix 5.1) confirms HYDRO significantly outperforms all baselines. Importantly, compared to Oracle, HYDRO exhibits the smallest degradation in robust performance across all tasks.

5.2 Ablation study.

Naively combining source data. To address the second question, we compare the performance of RFQI trained on target data only versus RFQI trained on combined target and source data (cross-domain data), as well as our proposed method. For the cross-domain data experiment, we simply merge target and source data without any further processing and use this combined dataset to train RFQI. Figure 3a demonstrates that simply incorporating additional source data does not enhance robustness and can also lead to poor performance compared to using only the limited target data (100K). We argue that the primary



(a) Naively combining data. (b) Priority sampling. (c) Uncertainty filter.

Fig. 3: (a) Robust performance comparison between HYDRO, RFQI, and its variations using naive combination of source and target data. (b-c) Robust performance comparison between HYDRO and its variants without priority sampling and uncertainty filter.

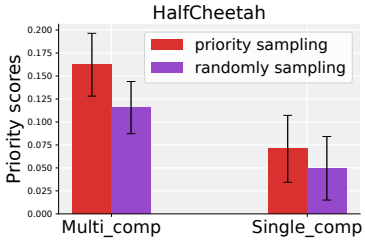


Fig. 4: Average priority scores of random and priority sampling.

reason for this is the dynamics mismatch between the source and worst-case model, which hinders the simple merging cross-domain data approach. On the other hand, our method handles this mismatch by selecting the *most* reliable source data with a small gap to tighten Ineq (9) and control the learn robust Q function as motivated by Theorem 2. The results highlight the effectiveness of our approach compared to both RFQI and the naive combination strategy.

To address the third question, we perform a comprehensive ablation study on HYDRO, analyzing the contribution of each component to its performance.
Priority Sampling. To evaluate priority sampling's impact, we compare against a variant without this component. Figure 3b shows a significant decrease in robust performance when priority sampling is excluded. The enhanced performance stems from the increased utilization of source samples with greater proximity to the worst-case target model. Figure 4 confirms this hypothesis, demonstrating that priority sampling significantly increases the mean priority score of selected samples compared to random sampling. These results confirm that priority sampling plays a crucial role in enhancing the robustness of our method.
Uncertainty Filter. To assess the impact of the uncertainty filter, we compare our method to a variant that excludes this component. Figure 3c reveals a significant decrease in robustness when the uncertainty filter is omitted, highlighting its crucial role in our approach. We hypothesize that the significant performance gains observed when incorporating the uncertainty filter arise from the strategic

exclusion of source samples with high uncertainty levels, allowing the model to focus on more informative data points, leading to improved performance.

5.3 More Experiment Results

We conduct additional experiments to explore the potential of HYDRO. Please see Appendix 5 for more results.

HYDRO with different domain-gaps measurement. To emphasize the distinction between HYDRO and *standard* cross-domain RL methods, we evaluate HYDRO's performance using an alternative domain-gaps measurement. We replace our measurement, which quantifies the discrepancy between worst-case target and source models, with standard cross-domain RL measurements quantifying the discrepancy between nominal target and source models using domain classifiers from DARC [7]. Results in Appendix 5.4 illustrate this substitution leads to significant performance degradation. These results underscore the inadequacy of standard domain-gap measurement in *robust* RL settings and highlight the critical role and effectiveness of HYDRO's tailored measurement approach.

How HYDRO performs under harder limited target data settings? To further understand H2O's performance, we analyze HYDRO's behavior under increasingly challenging, data-limited target settings. The results in Appendix 5.5 show RFQI's robust performance decreases substantially as target data decreases, while HYDRO maintains consistently strong performance with only minimal degradation. These results demonstrate HYDRO's effectiveness in overcoming data scarcity challenges.

6 Conclusion

In this paper, we have addressed the problem of Hybrid cross-domain robust reinforcement learning, which is widely encountered in many real-world problems. To the best of our knowledge, this is the first work to tackle the hybrid setting of online source and offline target under Robust MDPs. We introduce HYDRO, a novel method that effectively leverages source domain data by selecting relevant and reliable data points with respect to the worst-case model in an uncertainty set, utilizing priority sampling and an uncertainty filter. We have demonstrated the superior performance of our method through extensive experiments. The limitation of our approach is its dependence on an estimated target transition model. Although we have demonstrated our method's effectiveness empirically, a theoretical analysis of how the estimated target model impacts the performance of the learned policy can be a promising direction for future research. Another promising research direction is extending HYDRO to fully online settings.

References

1. Sünderhauf, et al., "The limits and potentials of deep learning for robotics," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 405–420, 2018.

2. Niu, H., et al., "When to trust your simulator: Dynamics-aware hybrid offline-and-online reinforcement learning," *NeurIPS*, vol. 35, pp. 36599–36612, 2022.
3. Niu, H., et al., "H2O+: An Improved Framework for Hybrid Offline-and-Online RL with Dynamics Gaps," *arXiv preprint arXiv:2309.12716*, 2023.
4. Liu, J., et al., "DARA: Dynamics-Aware Reward Augmentation in Offline Reinforcement Learning," in *ICLR*, 2022.
5. Liu, J., et al., "Beyond ood state actions: Supported cross-domain offline reinforcement learning," in *AAAI*, vol. 38, pp. 13945–13953, 2024.
6. Wen, X., et al., "Contrastive Representation for Data Filtering in Cross-Domain Offline Reinforcement Learning," in *Forty-first ICML*, 2024.
7. Eysenbach, B., et al., "Off-Dynamics Reinforcement Learning: Training for Transfer with Domain Classifiers," in *ICLR*, 2021.
8. Pham Van Linh, L., et al., "Policy Learning for Off-Dynamics RL with Deficient Support," in *AAMAS*, pp. 1093–1100, 2024.
9. Xu, K., et al., "Cross-domain policy adaptation via value-guided data filtering," *NeurIPS*, vol. 36, 2024.
10. Lyu, J., et al., "Cross-Domain Policy Adaptation by Capturing Representation Mismatch," in *ICML*, 2024.
11. Peng, X. B., et al., "Sim-to-real transfer of robotic control with dynamics randomization," in *ICRA*, pp. 3803–3810, 2018.
12. Tobin, J., et al., "Domain randomization for transferring deep neural networks from simulation to the real world," in *IROS*, pp. 23–30, 2017.
13. Sadeghi, F., et al., "CAD2RL: Real Single-Image Flight Without a Single Real Image," *Robotics: Science and Systems XIII*, 2017.
14. Werbos, P. J., "Neural networks for control and system identification," in *Proceedings of the 28th IEEE Conference on Decision and Control,*, pp. 260–265, 1989.
15. Zhu, S., et al., "Fast model identification via physics engines for data-efficient policy search," in *IJCAI*, pp. 3249–3256, 2018.
16. Chebotar, Y., et al., "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *ICRA*, pp. 8973–8979, 2019.
17. Finn, C., et al., "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, pp. 1126–1135, 2017.
18. Nagabandi, A., et al., "Learning to Adapt in Dynamic, Real-World Environments through Meta-Reinforcement Learning," in *ICLR*, 2018.
19. Wu, Z., et al., "Zero-shot policy transfer with disentangled task representation of meta-reinforcement learning," in *ICRA*, pp. 7169–7175, 2023.
20. Mnih, V., et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
21. Schrittwieser, J., et al., "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.
22. Liu, Y., et al., "Provably good batch off-policy reinforcement learning without great exploration," *NeurIPS*, vol. 33, pp. 1264–1274, 2020.
23. Kumar, A., et al., "Conservative q-learning for offline reinforcement learning," *NeurIPS*, vol. 33, pp. 1179–1191, 2020.
24. Lyu, J., et al., "Mildly conservative q-learning for offline reinforcement learning," *NeurIPS*, vol. 35, pp. 1711–1724, 2022.
25. Levine, S., et al., "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.
26. Rigter, M., et al., "Rambo-rl: Robust adversarial model-based offline reinforcement learning," *NeurIPS*, vol. 35, pp. 16082–16097, 2022.

27. Iyengar, G. N., "Robust dynamic programming," *Mathematics of Operations Research*, vol. 30, no. 2, pp. 257–280, 2005.
28. Nilim, A., et al., "Robust control of Markov decision processes with uncertain transition matrices," *Operations Research*, vol. 53, no. 5, pp. 780–798, 2005.
29. Xu, H., et al., "Distributionally robust Markov decision processes," *NeurIPS*, vol. 23, 2010.
30. Wang, Y., et al., "Policy gradient method for robust reinforcement learning," in *ICML*, pp. 23484–23526, 2022.
31. Wang, Q., et al., "On the convergence of policy gradient in robust mdps," *arXiv preprint arXiv:2212.10439*, vol. 5, 2022.
32. Yang, W., et al., "Toward theoretical understandings of robust markov decision processes: Sample complexity and asymptotics," *The Annals of Statistics*, vol. 50, no. 6, pp. 3223–3248, 2022.
33. Xu, Z., et al., "Improved sample complexity bounds for distributionally robust reinforcement learning," in *AISTATS*, pp. 9728–9754, 2023.
34. Wang, Y., et al., "Online robust reinforcement learning with model uncertainty," *NeurIPS*, vol. 34, pp. 7193–7206, 2021.
35. Dong, J., et al., "Online policy optimization for robust mdp," *arXiv preprint arXiv:2209.13841*, 2022.
36. Zhou, Z., et al., "Finite-sample regret bound for distributionally robust offline tabular reinforcement learning," in *AISTATS*, pp. 3331–3339, 2021.
37. Panaganti, K., et al., "Robust reinforcement learning using offline data," *NeurIPS*, vol. 35, pp. 32211–32224, 2022.
38. Shi, L., et al., "Distributionally robust model-based offline reinforcement learning with near-optimal sample complexity," *arXiv preprint arXiv:2208.05767*, 2022.
39. Ma, X., et al., "Distributionally robust offline reinforcement learning with linear function approximation," *arXiv preprint arXiv:2209.06620*, 2022.
40. Blanchet, et al., "Double pessimism is provably efficient for distributionally robust offline reinforcement learning: Generic algorithm and robust partial coverage," *NeurIPS*, vol. 36, 2024.
41. Bousmalis, K., et al., "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *ICRA*, pp. 4243–4250, 2018.
42. Higgins, I., et al., "Darla: Improving zero-shot transfer in reinforcement learning," in *ICML*, pp. 1480–1490, 2017.
43. Fu, J., et al., "D4rl: Datasets for deep data-driven reinforcement learning," *arXiv preprint arXiv:2004.07219*, 2020.
44. Wu, F., et al., "OCEAN-MBRL: Offline Conservative Exploration for Model-Based Offline Reinforcement Learning," in *AAAI*, vol. 38, pp. 15897–15905, 2024.
45. Kidambi, R., et al., "Morel: Model-based offline reinforcement learning," *NeurIPS*, vol. 33, pp. 21810–21823, 2020.
46. Pinto, L., et al., "Robust adversarial reinforcement learning," in *ICML*, 2017.
47. Fei, Y., et al., "Exponential bellman equation and improved regret bounds for risk-sensitive reinforcement learning," *NeurIPS*, vol. 34, pp. 20436–20446, 2021.
48. Rigter, M., et al., "One risk to rule them all: A risk-sensitive perspective on model-based offline reinforcement learning," *NeurIPS*, vol. 36, 2024.
49. Zhang, X., et al., "Corruption-robust offline reinforcement learning," in *AISTATS*, 2022.
50. Ye, C., et al., "Corruption-robust offline reinforcement learning with general function approximation," *NeurIPS*, 2024.
51. Niu, H., et al., "A Comprehensive Survey of Cross-Domain Policy Transfer for Embodied Agents," in *IJCAI*, 2024.

52. Daoudi, P., et al., "A Conservative Approach for Few-Shot Transfer in Off-Dynamics Reinforcement Learning," in *IJCAI*, 2024.
53. Nguyen, M., et al., "Beyond the Known: Decision Making with Counterfactual Reasoning Decision Transformer," in *ArXiv Preprint ArXiv:2505.09114*, 2025.