Relaxed Transition Kernels can Cure Underestimation in Adversarial Offline Reinforcement Learning

Ziyu Wang c12wang@torontomu.ca
Department of Computer Science, Toronto Metropolitan University

Ping-Chun Hsieh pinghsieh@cs.nycu.edu.tw
Yu-Shuen Wang yushuen@cs.nycu.edu.tw
Yun-Hsuan Lien sophia.yh.lien@gmail.com
Department of Computer Science, National Yang Ming Chiao Tung University

Editors: Hung-yi Lee and Tongliang Liu

Abstract

Offline reinforcement learning (RL) trains policies from pre-collected data without further environment interaction. However, discrepancies between the dataset and true environment—particularly in the state transition kernel—can degrade policy performance. To simulate environment shifts without being overly conservative, we introduce a relaxed state-adversarial method that perturbs the policy while applying a controlled relaxation mechanism. This method improves robustness by interpolating between nominal and adversarial dynamics. Theoretically, we provide a performance lower bound; empirically, we show improved results across challenging offline RL benchmarks. Our approach integrates easily with existing model-free algorithms and consistently outperforms baselines, especially in high-difficulty domains like Adroit and AntMaze.

Keywords: Offline reinforcement learning, Adversarial offline reinforcement learning, Relaxed transition kernel

1. Introduction

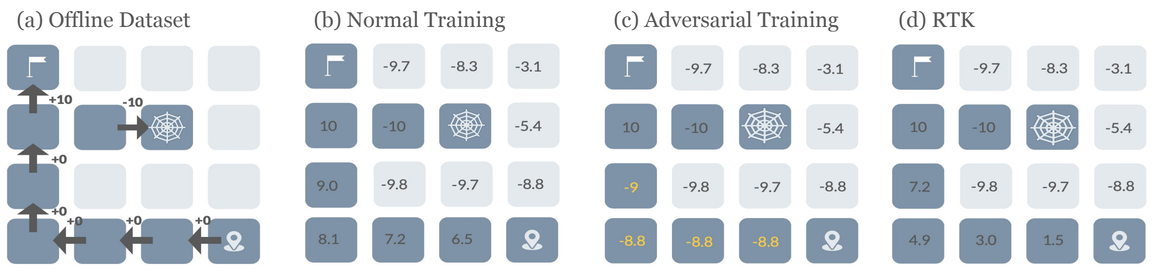

Figure 1: Overview of the Relaxed Transition Kernel (RTK) approach in offline RL. (a) Offline datasets contain transitions (arrows) and rewards (numbers), with blue squares representing in-dataset (ID) states and gray squares indicating out-of-dataset (OOD) states. (b) Standard TD learning may misestimate OOD states due to lack of data. (c) Adversarial TD updates propagate pessimistic values via perturbed next states. (d) RTK interpolates between nominal and adversarial estimates using a relaxation parameter α.

Offline reinforcement learning (RL) methods (Zhang et al., 2020a; Pattanaik et al., 2017; Clement and Kroer, 2021; Derman et al., 2021; Wang and Zou, 2021; Badrinath and Kalathil, 2021) use data collected from the real world to train policies, without requiring additional online interaction. This makes offline RL particularly appealing in scenarios where exploration is costly, risky, or impractical—such as healthcare, autonomous driving, or industrial control. However, since the dataset only shows a part of the environmental information, the state transition probabilities extracted from this data might differ from those in the actual environment. This issue is similar to the *reality gap* problem between virtual and real-world environments (Tobin et al., 2017; Rajeswaran et al., 2017; Jiang et al., 2021), both arising from inconsistencies in the state transition kernel, leading to a decline in performance after deployment. Therefore, adversarial state methods, commonly used in online RL to address the reality gap, have the potential to work well in offline RL (Sinha et al., 2022). These methods introduce additional disturbances to the state after each action taken by the policy during training. This places the policy in a nearby lower-value state and forces it to learn from challenging situations if the inconsistency between virtual and real environments occurs. However, since the offline dataset is pre-collected, these disturbed states may not be included in the original dataset, posing new challenges for subsequent policy learning (Yang et al., 2022; Rigter et al., 2022; Sinha et al., 2022).

In most RL algorithms, a common concern is overestimation when learning state values. However, in adversarial settings, overestimation is less of an issue, as the adversary deliberately targets low-value states or actions to challenge the policy. In contrast, when adversarial training is applied in offline RL, underestimation becomes a critical issue that must also be addressed. We highlight this problem in the grid world environment. As illustrated in Figure 1, when modifying an in-distribution (ID) state, its adversarial counterpart may lie either within the dataset (ID) or outside it (OOD). Since the values of OOD states are estimated purely through network generalization, they may be unreliable. In panel (c), adversarial TD updates propagate pessimistic values, e.g., $V(grid(0,1)) = 0 + 0.9 \times V(ADV(grid(0,2))) = -9$, where $\gamma = 0.9$ is the discount factor. In contrast, panel (d) shows how RTK interpolates between nominal and adversarial estimates: $V(grid(0,1)) = 0 + \alpha \times (0.9 \times 10 + 0.1 \times (-10)) = 7.2$, where $\alpha = 0.9$ is the relaxation parameter. This reduces pessimism and preserves solutions within the dataset. Other methods to address the underestimation issue includes RAMBO (Rigter et al., 2022), which creates a model environment based on the training data and uses adversarial training only on states that are *not* present in the dataset. Since this approach lowers the values of OOD states, it encourages agents to remain within the dataset distribution. On the other hand, S4RL (Sinha et al., 2022) directly perturbs ID states based on the learned value function and neglects the OOD issues. Despite the practical success, S4RL lacks a comprehensive theoretical framework to explain its effectiveness. Additionally, it suffers from the limitation illustrated in Figure 1.

Our proposed method, RTK (relaxed transition kernel), aims to address the underestimation of OOD states. To achieve this, we use a relaxation parameter α to linearly interpolate a state's nominal and adversarial transition kernels. Compared to the state adversary without relaxation, RTK does not abruptly transition a policy to its worst neighboring state. Instead, it only *increases the probability of encountering such a state.* This approach enhances a policy's robustness by encouraging it to handle challenging situations while preventing an unexpected OOD state with considerably underestimated value from making the policy overly conservative. Our evaluations, through the D4RL benchmark (Fu et al., 2020), confirm the effectiveness of RTK in various continuous-control tasks. It consistently outperforms baseline methods, making it a top choice for risk-sensitive applications. With its theoretical foundation and empirical results, we have demonstrated the effectiveness of RTK. To summarize, we made the following contributions:

- We discovered an issue with underestimation in adversarial training in offline RL. This issue is often neglected in the RL community, which typically focuses on overestimation.
- We propose RTK, a new algorithm to mitigate underestimation in offline RL, while maintaining robustness via adversarial training.
- We showcase the efficacy of RTK in offline RL benchmarks, outperforming other adversarial training techniques and current state-of-the-art methods.

2. Related Works

We categorize existing offline RL methods into two groups: policy regularization, and robust methods. The first two focus on regularization techniques to improve generalization in offline RL, while the third centers on enhancing robustness, with an emphasis on state-adversarial strategies.

2.1. Policy Regularization Offline Reinforcement Learning

In offline RL, policies are trained without interacting with environments. This domain has spawned several approaches, including policy constraint methods, importance sampling, regularization, and uncertainty estimation. Specifically, policy constraint techniques ensure that the learned policy closely aligns with the behavior policy derived from the dataset. They can be divided into two groups: direct (Fujimoto et al., 2019; Kostrikov et al., 2021; Wu et al., 2020) and implicit (Kumar et al., 2019; Fujimoto and Gu, 2021; Wang et al., 2020), depending on their use of a model to represent the behavior policy. Importance sampling methods in offline RL (Nachum et al., 2019; Zhang et al., 2020b) re-weight the state-action distribution in the dataset. Regularization techniques (Kumar et al., 2020; Yu et al., 2021; Singh et al., 2020) penalize the distances between the learned policy and the behavior policy. Lastly, uncertainty-based methods (Agarwal et al., 2020) balance conservative and off-policy RL techniques based on the model's confidence level.

2.2. Robust Offline Reinforcement Learning

Robust RL methods have been applied in offline settings. In the model-based regime, RAMBO (Rigter et al., 2022) introduces adversarial perturbations to states that are not present in the dataset, using uncertainty-aware dynamics models. Other works explore the limitations of stochastic transitions and synthetic environments in robust policy training (Antonoglou et al., 2022; Ozair et al., 2021). The reliability of synthetic samples and the need for hyperparameter tuning in model-based methods have been further examined in prior studies (Van Hasselt et al., 2019; Lu et al., 2022; Yu et al., 2021). In the model-free setting, S4RL (Sinha et al., 2022) perturbs states toward lower-value neighbors based on the value function to train policies that are resilient to unseen or rare transitions. These approaches represent ongoing efforts to build offline RL agents that generalize beyond the dataset while accounting for uncertainty in state transitions.

We have developed a method based on the relaxed state-adversarial policy optimization (RAPO) (Lien et al., 2023), which is a robust RL technique in online scenarios. Our approach adapts RAPO for offline scenarios, and we present a novel formulation that considers both states in the dataset and their adversarial OOD states during policy training. This model-free methodology offers a unique way to conduct offline RL and addresses the common issue of underestimation in robust RL.

3. Preliminaries

3.1. Markov Decision Processes

A Markov Decision Process (MDP) is defined as $\mathcal{M} := (\mathcal{S}, \mathcal{A}, P, R, \rho_0, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{D}(\mathcal{S})$[1] is the transition function, ρ_0 is the initial state distribution, and $\gamma \in (0, 1)$ is the discount factor. The set of Markovian policies is $\Pi := \{\pi \mid \pi : \mathcal{S} \to \mathcal{D}(\mathcal{A})\}$. The state-action value function is defined as $Q^{\pi}_{\mathcal{M}}(s, a) := \mathbb{E}_{a_t \sim \pi(\cdot|s_t),\ s_{t+1} \sim P(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \,\middle|\, s_0 = s,\ a_0 = a\right]$, which represents the expected total discounted return under policy π and an initial state-action pair (s, a) (where t is the time step). The value function is defined as $V^{\pi}_{\mathcal{M}}(s) := \mathbb{E}_{a \sim \pi(\cdot|s)}[Q^{\pi}_{\mathcal{M}}(s, a)]$. The overall performance of a policy is given by $J_{\rho_0}(\pi, \mathcal{M}) := \mathbb{E}_{s \sim \rho_0}[V^{\pi}_{\mathcal{M}}(s)]$. In offline RL, we distinguish between the **Nominal MDP** $\mathcal{M}_0$, which represents the true environment with transition function P_0, and the **Dataset-Induced MDP** $\mathcal{M}_B$, which is constructed from a dataset $B = \{(s_i, a_i, r_i, s'_i)\}$, where s_i is the state, a_i the action, r_i the reward, and s'_i the next state. The transition function $P_B(s' \mid s, a)$ is estimated from observed frequencies, and if a pair (s, a) is unseen, a terminal state s_{term} is deterministically reached.

3.2. Robust Reinforcement Learning

Robust RL tackles environment uncertainty by optimizing policies that perform well under worst-case dynamics (Nilim and El Ghaoui, 2005; Iyengar, 2005; Wiesemann et al., 2013). It introduces an uncertainty set $\mathcal{U}_r$ over possible transition models and solves a min-max objective: $\pi = \arg\max_{\pi'} \inf_{P \in \mathcal{U}_r} J_{\rho}(\pi', \mathcal{M})$, where the goal is to find a policy robust against adversarial transitions within $\mathcal{U}_r$. However, exact robust RL is intractable in continuous spaces. To overcome this, recent approaches (e.g., State-Adversarial Transition Kernel (Lien et al., 2023)) allow robust RL to be applied to continuous state and action spaces by approximating the adversarial dynamics and facilitating more tractable optimization.

3.3. State-Adversarial Transition Kernel

We adopt a model-free, state-adversarial perturbation approach (Lien et al., 2023), where the agent transitions to neighboring states with the lowest estimated value. This strategy (1) avoids reliance on learned environment models and (2) performs robustly in stochastic settings where model-based methods often fail (Antonoglou et al., 2022; Ozair et al., 2021).

1. $\mathcal{D}$ denotes a distribution over the set $\mathcal{S}$.

Formally, let the σ-neighborhood of a state $s \in \mathcal{S}$ be defined as $\mathcal{N}_\sigma(s) = \{s' \mid d(s,s') \leq \sigma\}$, where $d(\cdot,\cdot)$ is the L_∞-norm. Given a dataset-induced MDP with transition kernel P_B, a policy π, and a perturbation threshold σ, we define the **matrix of state perturbations** as $Z_\sigma^\pi(i,j) = \mathbf{1}\left[j = \underset{s\in\mathcal{N}_\sigma(i)}{\arg\min}\ V^\pi(s \mid P_B)\right]$, which identifies, for each state i, its worst-case neighboring state j under policy π. In continuous spaces, this perturbation can be computed using the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015) (Given a value function V characterized by parameter ϕ, a state s, and a perturbation magnitude ϵ): $\Gamma(s) = s - \epsilon \cdot \operatorname{sign}(\nabla_s V(\phi, s))$. Here $\|s - \Gamma(s)\| \leq \epsilon$ and the gradient at s is derived via back-propagation. The value function is then updated as $V(s) = r(s,a) + \gamma V(\Gamma(s'))$.

We define the state-adversarial transition kernel for the offline dataset as $P_\sigma^\pi(\cdot \mid s,a) = [Z_\sigma^\pi]^\top P_B(\cdot \mid s,a)$, and the corresponding uncertainty set as $\mathcal{U}_\epsilon^\pi := \left\{P_\sigma^\pi = [Z_\sigma^\pi]^\top P_B \mid \sigma \leq \epsilon\right\}$.

4. Adversarial Offline RL

4.1. Performance Gap between Optimal and Sub-Optimal Offline Policies

We analyze the performance difference between an optimal policy π^* and a sub-optimal policy π learned from offline data, under transition discrepancies between the dataset and the real environment. Let P_B be the transition kernel estimated from the offline dataset, and let $\mathcal{U}$ be the uncertainty set containing plausible real-world transition kernels. The discrepancy between the true environment and the offline model is quantified by the *reality gap*, which we define as $\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}\left[\mathbb{E}_{(s,a)\sim P_0}\left[\mathcal{D}_{\mathrm{TV}}\left(P_0(\cdot \mid s,a), P_B(\cdot \mid s,a)\right)\right]\right] \leq \beta$, where $\beta \geq 0$ is a bound on total variation (TV) distance, and $\mathcal{D}_{\mathrm{TV}}(P_0,P_B) = \frac{1}{2}\sum_{s'} |P_0(s' \mid s,a) - P_B(s' \mid s,a)|$.

To study how this gap affects policy performance, we consider the dataset-induced MDP $\mathcal{M}_B$. Let $\mathcal{V}$ denote the set of unseen state-action pairs, and let $T_\mathcal{V}^\pi$ be the hitting time to reach any such pair under policy π. The expected performance difference between π and π^* is then bounded using the reality gap, discount factor, and expected visitation of unknown regions. This analysis highlights how limited coverage in the dataset can lead to sub-optimal policy behavior when deployed in environments with slightly different dynamics.

Theorem 1 *Let $\pi^*(P_0)$ be the optimal policy under $P_0 \sim \mathcal{D}(\mathcal{U})$, and let ϵ_π be the offline suboptimality gap (defined as $\left|J_{\rho_0^B}(\pi^*, P_B) - J_{\rho_0^B}(\pi, P_B)\right|$). Then:*

$$\begin{aligned}
&\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[J_{\rho_0}(\pi^*, P_0)] - \mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[J_{\rho_0}(\pi, P_0)] \\
&\qquad \leq \epsilon_\pi + \frac{4R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}\left[\mathcal{D}_{\mathrm{TV}}(\rho_0, \rho_0^B)\right] + \frac{4\gamma R_{\max}}{(1-\gamma)^2}\beta \\
&\qquad\quad + \frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T_\mathcal{V}^\pi}]] + \frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T_\mathcal{V}^{\pi^*}}]] \qquad (1)
\end{aligned}$$

The proof is provided in Appendix A.1. The five terms reflect: (1) the algorithm-dependent offline suboptimality ϵ_π, (2) mismatch in initial state distributions, (3) transition kernel deviation β, and (4–5) dataset coverage, as measured by the expected hitting time to unknown state-action pairs. In the next section, we introduce the Relaxed Transition Kernel (RTK) to reduce the performance gap between optimal and offline-learned policies.

4.2. Relaxed Transition Kernel in Adversarial Offline RL

To avoid overly conservative behavior, we introduce a relaxed transition kernel that improves robustness to average-case scenarios. For parameters $\epsilon > 0$ and $\alpha \in [0, 1]$, the relaxed transition kernel is defined as a convex combination of the nominal and adversarial kernels: $P_\epsilon^{\pi,\alpha}(\cdot \mid s, a) = \alpha P_B(\cdot \mid s, a) + (1 - \alpha) P_\epsilon^\pi(\cdot \mid s, a)$. Here, the *nominal case* refers to the standard Bellman update using transitions from the dataset-induced dynamics, without adversarial perturbations. The corresponding Bellman update is given by: $Q(s, a) \leftarrow r(s, a) + \gamma \left[\alpha Q(s', \pi(s')) + (1 - \alpha) Q(\Gamma(s'), \pi(s'))\right]$ (Lien et al., 2023). Importantly, this interpolation modifies the probability distribution over next states, rather than shrinking the perturbation radius. The hyperparameter α reflects assumptions about the environment: $\alpha = 1$ recovers the nominal case, while $\alpha = 0$ corresponds to the worst-case. RTK improves lower-bound performance, as showed in the next Theorem.

Theorem 2 *Let $\mathcal{U}_{RTK}$ be the uncertainty set generated using the relaxed transition kernel method. For any sub-optimal policy $\epsilon_{\pi_{RTK}}$ trained on the offline dataset, where $\pi_{RTK} = \arg max_\pi \mathbb{E}_{P \sim \mathcal{U}_{RTK}} J_\rho(\pi, P)$, we let the probability of $P \in \mathcal{U}_{RTK}$ for every $P \in \mathcal{U}$ be p_{RTK}, where $0 \leq p_{RTK} \leq 1$. For the optimal policy π^* and the sub-optimal policy π_{RTK} trained on the relaxed transition kernel set $\mathcal{U}_{RTK}$, when they are deployed in an environment with the uncertain nominal transition kernel set $\mathcal{U}$, their performance satisfies the following condition:*

$$\begin{aligned}\mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}[J_{\rho_0}(\pi^*, P_0)] - \mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}[J_{\rho_0}(\pi_{RTK}, P_0)] \leq \; & \epsilon_{\pi_{RTK}} + \frac{4R_{max}}{1-\gamma} \mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}[\mathcal{D}_{TV}(\rho_0, \rho_0^B)] \\ + \frac{4\gamma R_{max}}{(1-\gamma)^2} \left(\beta - \tfrac{1}{2} p_{RTK}(1-\alpha)\right) + \frac{2R_{max}}{1-\gamma} \mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})} \mathbb{E}[[\gamma^{T_{\mathcal{V}}^{\pi_{RTK}}}]] & + \frac{2R_{max}}{1-\gamma} \mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})} \mathbb{E}[[\gamma^{T_{\mathcal{V}}^{\pi^*}}]]\end{aligned}$$

The proof is in Appendix A.2. The first term $\epsilon_{\pi_{\text{RTK}}}$ reflects policy sub-optimality. The second term captures distribution shift β, reduced by $\frac{1}{2} p_{\text{RTK}}(1 - \alpha)$, showing how RTK enhances robustness. The third term, involving total variation, reflects dataset coverage, larger datasets lead to longer hitting times and smaller performance gaps. *Together, these effects enable RTK to achieve a provably tighter performance bound compared to worst-case robust approaches.*

We now give the Algorithm 1 that outlines RTK. At each iteration t, the policy π_{θ_t} is updated using ReBrac (Tarasov et al., 2023) to maximize the average-case return $J(\pi_{\theta_t} \mid P^{\pi_{\theta_{t-1}}, \alpha})$. ReBrac's hyperparameters are omitted due to space constraints. The relaxation parameter α and perturbation scale ϵ were selected using a small evaluation set held out from the offline dataset, and fixed for each environment; task-specific values are provided in Appendix B.

4.3. Relaxed State Adversaries in Online and Offline RL

The RTK method was inspired by RAPO (Lien et al., 2023), which used a relaxed state adversary in online RL. However, the objectives of RTK and RAPO are intrinsically different. In RAPO, the authors established the relationship between a policy's return under average and worst-case scenarios. On the other hand, in RTK, we derived the performance gap of a policy with transition kernels in the offline dataset and a potential real-world environment.

Algorithm 1 Relaxed Transition Kernel (RTK)

```
Require: Offline dataset $\mathcal{D}$; objective function $J$; discount factor $\gamma$; policy regularization
    weight $\lambda_\pi$; Q-function regularization weight $\lambda_Q$; RTK parameter $\alpha$; number of iterations
    $T$; number of update samples per iteration $T_{\text{upd}}$; uncertainty set radius $\epsilon$
1: Initialize policy $\pi_{\theta_0}$, value function $Q_{\phi_0}$
2: for $t = 0, \dots, T-1$ do
3:     Draw a batch of samples $\{s_i, a_i, r_i, s_i'\}_{i=1}^{T_{\text{upd}}}$ from $\mathcal{D}$
4:     Compute the relaxed transition kernel of the offline data $\{s_i, a_i, r_i, \text{adv}(s_i')\}_{i=1}^{T_{\text{upd}}}$ using:
       $Q_{\text{RTK}} = r(s,a) + \gamma\Big(\alpha Q_{\bar{\phi}_t}(s', \pi_{\bar{\theta}_t}(s')) + (1-\alpha) Q_{\bar{\phi}_t}(\Gamma(s'), \pi_{\bar{\theta}_t}(s'))\Big)$,
5:     Update $Q_{\phi_t}$ by: $Q = \text{argmin}_Q \mathbb{E}_{(s,a,s')}\Big[\big(Q_{\phi_t}(s,a) - Q_{\text{RTK}} - \lambda_Q \cdot (\pi_{\bar{\theta}_t}(s) - a)^2\big)^2\Big]$,
6:     Update policy $\pi_{\theta_t}$
7:     $\pi = \text{argmax}_\pi \mathbb{E}_{(s,a)}\left[Q_{\phi_t}(s, \pi_{\theta_t}(s)) - \lambda_\pi \cdot (\pi_{\theta_t}(s) - a)^2\right]$
8: end for
```

The difference in configurations led to different theorems, implementations, and lower bound performance for the agent.

5. Results and Evaluation

We have conducted experiments to assess the effectiveness of RTK. Our objectives were threefold: 1) *Performance:* We aimed to compare the proficiency of RTK with other state-of-the-art benchmarks in Section 5.1 and 5.2; 2) *Effectiveness:* We aimed to understand the impact of adversarial training on the algorithm in Section 5.3 and 5.4 and 3) *Robustness:* We aimed to assess the stability of the algorithm under adversarial conditions in Section 5.4. The evaluation was conducted on datasets collected from various environments, including MuJoCo, AntMaze, and Adroit, using the standard D4RL normalization $\left(\frac{S_o - S_r}{S_e - S_r}\right)$, where S_o, S_r, and S_e are the offline, random, and expert rewards, respectively. Further details of the datasets can be found in (Fu et al., 2020).

5.1. Comparison with Ensemble-Free Methods

We first compared RTK with ensemble-free methods because RTK and these methods use two value networks to guide policy training. The methods include adversarial approaches: S4RL(Sinha et al., 2022), ARMOR (Bhardwaj et al., 2023), RAMBO (Rigter et al., 2022) and ATAC (Cheng et al., 2022), and regularization approaches: ReBrac (Tarasov et al., 2023), CQL (Kumar et al., 2020), COMBO (Kumar et al., 2020), TD3+BC (Fujimoto and Gu, 2021), and IQL (Kostrikov et al., 2022). The results described in Table 1, demonstrate the effectiveness of our RTK approach. While previous methods have shown good results in the relatively simple Mujoco environments, RTK achieved slightly higher average rewards than the baseline methods. As the environment difficulty increased, where rewards became sparse, leading methods such as RAMBO, ATAC, and TD3+BC, which previously excelled in certain Mujoco environments, experienced a significant decline in their performance. Specifically,

	Adversarial Training						Regularizations				
	RTK	**RTK(α = 0)**	**S4RL**	**ARMOR**	**RAMBO**	**ATAC**	**ReBrac**	**CQL**	**COMBO**	**TD3+BC**	**IQL**
Halfcheetah-medium	66.1 ± 1.2	65.1 ± 1.3	48.6	54.2	**77.6**	53.3	65.5	44.4	54.2	42.8	47.4
Halfcheetah-medium-replay	51.0 ± 0.4	52.6 ± 0.7	51.7	50.5	**68.9**	48.0	51.0	46.2	55.1	43.3	44.2
Halfcheetah-medium-expert	**107.0 ± 3.2**	65.9 ± 3.4	78.1	93.5	93.7	94.8	101.0	62.4	90.0	97.9	86.7
Hopper-medium	**102.3 ± 0.4**	32.9 ± 1.5	81.3	101.4	92.8	85.6	102.0	86.6	94.9	99.5	66.3
Hopper-medium-replay	100.4 ± 0.8	50.3 ± 12.7	36.8	97.1	96.6	**102.5**	98.0	48.6	73.1	31.4	94.7
Hopper-medium-expert	108.9 ± 4.2	85.4 ± 16.8	**117.9**	103.4	83.3	111.9	107.0	111.0	111.1	112.2	109.6
Walker2d-medium	86.8 ± 0.5	48.2 ± 17.8	**93.1**	90.7	86.9	89.6	82.5	74.5	75.5	79.7	78.3
Walker2d-medium-replay	85.0 ± 6.6	18.8 ± 10.7	35.0	85.6	85.0	**92.5**	77.3	32.6	56.0	25.2	73.9
Walker2d-medium-expert	112.2 ± 0.4	45.6 ± 18.8	107.1	112.2	68.3	**114.2**	111.6	98.7	96.1	101.1	91.5
Mujoco	**91.1**	51.6	72.2	87.6	83.7	88.0	88.4	67.2	78.4	70.3	77.0
Pen-cloned	**106.6 ± 22.8**	95.2 ± 12.3	9.9	51.4	-	43.7	91.8	39.2	-	61.4	37.3
Pen-expert	**154.9 ± 3.9**	152.5 ± 7.2	-	112.2	-	136.2	154.1	107.0	-	146.0	-
Hammer-cloned	3.7 ± 2.1	**9.8 ± 9.6**	1.2	0.7	-	1.1	1.1	2.1	-	0.8	2.1
Hammer-expert	**134.0 ± 0.6**	**134.0 ± 0.2**	-	118.8	-	126.9	133.8	86.7	-	117.0	-
Door-cloned	0.3 ± 0.4	0.3 ± 0.4	0.5	-0.1	-	3.7	6.7	0.4	-	0.1	1.6
Door-expert	104.2 ± 2.1	**105.0 ± 2.5**	-	98.7	-	99.3	104.6	101.5	-	84.6	-
Relocate-cloned	0.4 ± 0.2	0.0 ± 0.0	-	0.0	-	0.2	**0.9**	-0.1	-	-0.1	-0.2
Relocate-expert	**110.1 ± 0.7**	105.7 ± 9.1	-	96.0	-	99.4	106.6	95.0	-	107.3	-
Adroit	**76.8**	75.3	3.9	59.7	-	63.8	75.0	54.0	-	64.6	10.2
Umaze	97.5 ± 0.7	97.7 ± 1.5	94.1	-	25.0	-	**97.8**	74.0	80.3	78.6	87.5
Medium-Play	**91.5 ± 3.8**	83.0 ± 6.9	61.6	-	16.4	-	84.0	61.2	0.0	3.0	71.2
Large-Play	**71.2 ± 16.5**	56.5 ± 37.3	25.1	-	0.0	-	60.4	15.8	0.0	0.0	39.6
Umaze-Diverse	87.7 ± 7.9	77.7 ± 10.4	88.0	-	0.0	-	88.3	84.0	57.3	71.4	62.2
Medium-Diverse	**86.7 ± 7.1**	81.0 ± 16.9	82.3	-	23.2	-	76.3	53.7	0.0	10.6	70.0
Large-Diverse	**68.0 ± 7.6**	68.0 ± 8.4	26.2	-	2.4	-	54.4	14.9	0.0	0.2	47.5
AntMaze	**83.8**	77.3	62.9	-	11.2	-	76.9	50.6	22.9	27.3	63.0

Table 1: RTK performance averaged over 4 seeds. Scores follow the D4RL normalization (Fu et al., 2020). RTK results use fixed hyperparameters; baseline results are from original papers.

RAMBO and TD3+BC performed worse than RTK in the AntMaze environments, and ATAC failed to match RTK's performance in the Adroit environments.

5.2. Comparison with Ensemble Methods

We also conducted a comparison of RTK with ensemble methods such as RORL (Yang et al., 2022) and EDAC / SAC-N (An et al., 2021), which often required multiple value networks to guide policy training. The comparative results can be found in Table 2. Despite the use of at least 10 value networks, these ensemble methods still underperformed RTK on the Adroit and AntMaze datasets by a significant margin.

5.3. Ablation Study

We compared a variant of RTK, denoted as RTK($\alpha = 0$), by setting the relaxation parameter $\alpha = 0$, to evaluate the effectiveness of relaxation in the transition kernel. The results in Table 1 indicate that relaxation in state-adversarial training is necessary. This is because a strong perturbation often leads to excessively conservative outcomes (Lien et al., 2023), particularly when adversarial states may be out of distribution, and their values can be

	RTK	**RORL**	**EDAC**	**SAC-N**
# Critics	2	[10, 20]	[10, 50]	[10, 500]
Pen-cloned	**106.6 ± 22.8**	35.7	68.2	64.1
Pen-expert	**154.9 ± 3.9**	130.3	122.8	-
Hammer-cloned	**3.7 ± 2.1**	1.7	0.3	0.2
Hammer-expert	**134.0 ± 0.6**	132.2	0.2	-
Door-cloned	0.3 ± 0.4	**9.6**	**9.6**	-0.3
Door-expert	**104.2 ± 2.1**	-0.3	-0.3	-
Relocate-cloned	**0.4 ± 0.2**	0.0	0.0	0.0
Relocate-expert	**110.1 ± 0.7**	-0.3	-0.3	-
Adroit	**76.8**	38.6	25.1	16.0
Umaze	97.5 ± 0.7	**97.7**	-	-
Medium-Play	**91.5 ± 3.8**	76.3	-	-
Large-Play	**71.2 ± 16.5**	16.3	-	-
Umaze-Diverse	87.7 ± 7.9	**90.7**	-	-
Medium-Diverse	**86.7 ± 7.1**	69.3	-	-
Large-Diverse	**68.0 ± 7.6**	41.0	-	-
AntMaze	**83.8**	65.2	-	-

Table 2: We report the final normalized score averaged over five seeds. The scores of RORL, EDAC, and SAC-N are obtained from (Tarasov et al., 2023).

underestimated. It's worth noting that S4RL (Sinha et al., 2022) employs an adversarial state training approach similar to RTK($\alpha = 0$), but its base algorithm is different. As expected, in our experiment results, RTK showed significant superiority compared to S4RL in most environments, highlighting the importance of relaxation and its theoretical analysis.

We also tested RTK's flexibility by applying it to other base algorithms—specifically IQL (Kostrikov et al., 2022) and TD3+BC (Fujimoto and Gu, 2021) (denote as T+B)—to assess generalizability beyond ReBrac. As shown in Table 3, integrating RTK led to consistent improvements, indicating its broad applicability across offline RL methods (m-e and m-r denote medium expert and medium replay, respectively).

Table 3: RTK improves IQL and TD3+BC on D4RL benchmarks.

Task	IQL	+RTK
Halfcheetah-m-e	86.7	**93.3** ± **1.5**
Hopper-m-e	109.6	**112** ± **1.9**
Walker2d-m-e	91.5	**112.4** ± **0.6**

Task	T+B	+RTK
Halfcheetah-m-r	43.3	**45.3** ± **1.5**
Hopper-m-r	31.4	**41.5** ± **12.5**
Walker2d-m-r	25.2	**88.2** ± **1.2**

5.4. Robustness Analysis

In offline RL, deployment often occurs on machines slightly different from those used for training, leading to compounding performance gaps. To test generalization, we evaluated RTK and ReBrac under increasing perturbations that simulate MDP inconsistencies via

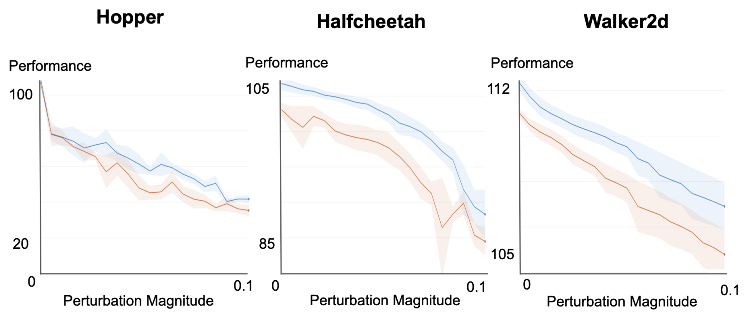

Figure 2: Normalized scores of RTK (blue) and ReBrac (red) under varying state perturbation magnitudes. Shaded regions show half standard deviation across four seeds.

value-guided worst-case transitions. Figure 2 provides a side-by-side comparison under different perturbation levels for *Medium-Expert* datasets from the *Hopper*, *Halcheetah*, and *Walker2d* environments. The results demonstrate that RTK is more resilient than the baseline and highlight the advantages of using relaxed transition kernel in offline RL scenarios.

6. Conclusions

We propose RTK, a model-free offline RL technique leveraging state-adversarial perturbations to train robust policies from pre-collected data. RTK offers theoretical performance guarantees and integrates with existing methods. RTK consistently outperforms baselines in standard benchmarks, especially in challenging tasks like Adroit and AntMaze. Future work includes a deeper analysis of RTK's computational efficiency and a sensitivity study of its key hyperparameters.

References

Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pages 104–114. PMLR, 2020.

Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in Neural Information Processing Systems*, 34:7436–7447, 2021.

Ioannis Antonoglou, Julian Schrittwieser, Sherjil Ozair, Thomas K Hubert, and David Silver. Planning in stochastic environments with a learned model. In *International Conference on Learning Representations*, 2022.

Kishan Panaganti Badrinath and Dileep Kalathil. Robust reinforcement learning using least squares policy iteration with provable performance guarantees. In *International Conference on Machine Learning*, pages 511–520. PMLR, 2021.

Mohak Bhardwaj, Tengyang Xie, Byron Boots, Nan Jiang, and Ching-An Cheng. Adversarial model for offline reinforcement learning. *Advances in neural information processing systems*, 2023.

Ching-An Cheng, Tengyang Xie, Nan Jiang, and Alekh Agarwal. Adversarially trained actor critic for offline reinforcement learning. 2022.

Julien Grand Clement and Christian Kroer. First-order methods for wasserstein distributionally robust mdp. In *International Conference on Machine Learning*, pages 2010–2019. PMLR, 2021.

Esther Derman, Matthieu Geist, and Shie Mannor. Twice regularized MDPs and the equivalence between robustness and regularization. *Advances in Neural Information Processing Systems*, 34, 2021.

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

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

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.

Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30 (2):257–280, 2005.

Yuankun Jiang, Chenglin Li, Wenrui Dai, Junni Zou, and Hongkai Xiong. Monotonic robust policy optimization with model discrepancy. In *International Conference on Machine Learning*, pages 4951–4960. PMLR, 2021.

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

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Aviral Kumar, Anikait Singh, Stephen Tian, Chelsea Finn, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

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

Yun-Hsuan Lien, Ping-Chun Hsieh, and Yu-Shuen Wang. Revisiting domain randomization via relaxed state-adversarial policy optimization. 2023.

Cong Lu, Philip Ball, Jack Parker-Holder, Michael Osborne, and Stephen J Roberts. Revisiting design choices in offline model based reinforcement learning. In *International Conference on Learning Representations*, 2022.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in neural information processing systems*, 32, 2019.

Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

Sherjil Ozair, Yazhe Li, Ali Razavi, Ioannis Antonoglou, Aaron Van Den Oord, and Oriol Vinyals. Vector quantized models for planning. In *international conference on machine learning*, pages 8302–8313. PMLR, 2021.

Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. Robust deep reinforcement learning with adversarial attacks. *arXiv preprint arXiv:1712.03632*, 2017.

Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. Epopt: Learning robust neural network policies using model ensembles. *International Conference on Learning Representations*, 2017.

Marc Rigter, Bruno Lacerda, and Nick Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *Advances in neural information processing systems*, 35: 16082–16097, 2022.

Avi Singh, Albert Yu, Jonathan Yang, Jesse Zhang, Aviral Kumar, and Sergey Levine. Cog: Connecting new skills to past experience with offline reinforcement learning. *arXiv preprint arXiv:2010.14500*, 2020.

Samarth Sinha, Ajay Mandlekar, and Animesh Garg. S4rl: Surprisingly simple self-supervision for offline reinforcement learning in robotics. In *Conference on Robot Learning*, pages 907–917. PMLR, 2022.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 2023.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

Hado P Van Hasselt, Matteo Hessel, and John Aslanides. When to use parametric models in reinforcement learning? *Advances in Neural Information Processing Systems*, 32, 2019.

Yifan Wang, Alekh Agarwal, Sham Kakade, John Langford, Alex Mott, and Yi Zhang. Critic regularized regression. *arXiv preprint arXiv:2002.09005*, 2020.

Yue Wang and Shaofeng Zou. Online robust reinforcement learning with model uncertainty. *Advances in Neural Information Processing Systems*, 34, 2021.

Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

Yihao Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. In *International Conference on Learning Representations*, 2020.

Rui Yang, Chenjia Bai, Xiaoteng Ma, Zhaoran Wang, Chongjie Zhang, and Lei Han. Rorl: Robust offline reinforcement learning via conservative smoothing. *Advances in Neural Information Processing Systems*, 35:23851–23866, 2022.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in Neural Information Processing Systems*, 33:21024–21037, 2020a.

Ruiyi Zhang, Bo Dai, Lihong Li, and Dale Schuurmans. Gendice: Generalized offline estimation of stationary values. *International Conference on Learning Representations*, 2020b.

Appendix A. Proof

A.1. Proof of Theorem 1

We first introdue Lemma 3 as a foundation for the subsequent proof presented in Theorem 1.

Lemma 3 *Let $\mathcal{V}$ be the set of state-action pairs not in the offline dataset, and $T^{\pi}_{\mathcal{V}}$ be the hitting time of $\mathcal{V}$. The value of any policy π learned from $\mathcal{P}_{\mathcal{B}}$ on the universal uncertainty set $\mathcal{U}$ satisfies:*

$$\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}\left[J_{\rho_0}(\pi,P_0)\right]-\frac{2R_{max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathcal{D}_{TV}(\rho_0,\rho_0^B)]-\frac{2\gamma R_{max}}{(1-\gamma)^2}\beta-$$
$$\frac{2R_{max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}\mathbb{E}[[\gamma^{T^{\pi}_{\mathcal{V}}}]]\leq J_{\rho_0^B}(\pi,P_B)\leq$$
$$\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}\left[J_{\rho_0}(\pi,P_0)\right]+\frac{2R_{max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathcal{D}_{TV}(\rho_0,\rho_0^B)]+\frac{2\gamma R_{max}}{(1-\gamma)^2}\beta+$$
$$\frac{2R_{max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}\mathbb{E}[[\gamma^{T^{\pi}_{\mathcal{V}}}]]$$

Proof For state-action pairs in the dataset, we aim to couple the trajectories of any policy on the dataset-induced MDP $\mathcal{M}_B$ and the true MDP $\mathcal{M}$. Assuming successful initial coupling, the divergence is bounded by $\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}\left[\|P_B(s,a)-P_0(s,a)\|_1\right]\leq\beta$, ensuring coupling persists with probability $1-\beta$ at each step, and decouples with probability at most $1-(1-\beta)^t$ at time t. For unseen state-action pairs, the return gap is at most $\frac{2R_{\max}}{1-\gamma}$, weighted by the hitting time discount factor $\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\gamma^{T^\pi_{\mathcal{V}}}]$. Thus, the total value difference between $\mathcal{M}_B$ and $\mathcal{M}$ is upper bounded as:

$$\begin{aligned}
&\left|J_{\rho_0^B}(\pi,P_B)-\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[J_{\rho_0}(\pi,P_0)]\right| \\
&\le \frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathcal{D}_{\mathrm{TV}}(\rho_0,\rho_0^B)]+\sum_t\gamma^t\left(1-(1-\beta)^t\right)\cdot 2\cdot R_{\max}+\frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T^\pi_{\mathcal{V}}}]] \qquad (2)\\
&=\frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathcal{D}_{\mathrm{TV}}(\rho_0,\rho_0^B)]+\frac{2R_{\max}\gamma\beta}{(1-\gamma)(1-\gamma\cdot(1-\beta))}+\frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T^\pi_{\mathcal{V}}}]] \qquad (3)\\
&\le\frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathcal{D}_{\mathrm{TV}}(\rho_0,\rho_0^B)]+\frac{2R_{\max}\gamma}{(1-\gamma)^2}\beta+\frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T^\pi_{\mathcal{V}}}]]. \qquad (4)
\end{aligned}$$

Notably, Eq. (2) quantifies three sources of divergence: (1) initial state distribution, (2) known state-action pairs, and (3) unknown state-action pairs. Eq. (3) follows from the identity $\sum_t\gamma^t(1-(1-\beta)^t)=\frac{\gamma\beta}{(1-\gamma)(1-\gamma(1-\beta))}$. Eq. (4) uses the bound $\frac{1}{(1-\gamma)(1-\gamma(1-\beta))}\le\frac{1}{(1-\gamma)^2}$. ■

Proof We now prove our main theorem. By Lemma 3, we have:

$$\begin{aligned}
\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[J_{\rho_0}(\pi,P_0)] &\ge J_{\rho_0^B}(\pi,P_B)-\tfrac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathcal{D}_{\mathrm{TV}}(\rho_0,\rho_0^B)]-\tfrac{2R_{\max}\gamma}{(1-\gamma)^2}\beta\\
&\quad-\tfrac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T^\pi_{\mathcal{V}}}]] \qquad (5)\\
&\ge J_{\rho_0^B}(\pi^*,P_B)-\epsilon_\pi-\tfrac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathcal{D}_{\mathrm{TV}}(\rho_0,\rho_0^B)]-\tfrac{2R_{\max}\gamma}{(1-\gamma)^2}\beta\\
&\quad-\tfrac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T^\pi_{\mathcal{V}}}]] \qquad (6)\\
&\ge \mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[J_{\rho_0}(\pi^*,P_0)]-\epsilon_\pi-\tfrac{4R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathcal{D}_{\mathrm{TV}}(\rho_0,\rho_0^B)]-\tfrac{4R_{\max}\gamma}{(1-\gamma)^2}\beta\\
&\quad-\tfrac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T^\pi_{\mathcal{V}}}]]-\tfrac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T^{\pi^*}_{\mathcal{V}}}]] \qquad (7)
\end{aligned}$$

where Equation (5) holds by Lemma 3, Equation (6) is derived via $\epsilon_\pi=|J_{\rho_0^B}(\pi^*,P_B)-J_{\rho_0^B}(\pi,P_B)|$, and Equation (5) is obtained by plugging $\pi=\pi^*$ into Lemma 3. ■

A.2. Risk-Aware Policy Reality Gap

Lemma 4 *Given a robust policy $\pi_r=\arg\max_\pi\mathbb{E}_{P\sim\mathcal{D}(\mathcal{U}_r)}J_\rho(\pi,P)$ satisfying the inequality $\mathbb{E}_{P\sim\mathcal{D}(\mathcal{U}_r)}[\mathbb{E}_{s,a}[\mathcal{D}_{TV}(P,P_B)]]\le\beta_r$, we assume the true uncertainty set satisfies $\mathcal{U}_r\subseteq\mathcal{U}$, $\beta\ge\beta_r$, and $P\in\mathcal{U}_r$ with probability p_r for $P\sim\mathcal{U}$, where $0\le p_r\le 1$. The performance under $\mathcal{U}$ and $\mathcal{U}_r$ then satisfies:*

$$\begin{aligned}
&\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[J_{\rho_0}(\pi_r,P_0)]-\frac{2R_{max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathcal{D}_{TV}(\rho_0,\rho_0^B)]-\frac{2\gamma R_{max}}{(1-\gamma)^2}(\beta-p_r\beta_r)-\\
&\frac{2R_{max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T^{\pi_r}_{\mathcal{V}}}]]\le\mathbb{E}_{P_r\sim\mathcal{D}(\mathcal{U}_r)}[J_{\rho_0^B}(\pi_r,P_r)]\le\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[J_{\rho_0}(\pi_r,P_0)]+\\
&\frac{2R_{max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathcal{D}_{TV}(\rho_0,\rho_0^B)]+\frac{2\gamma R_{max}}{(1-\gamma)^2}\beta+\frac{2R_{max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T^{\pi_r}_{\mathcal{V}}}]]
\end{aligned}$$

where $\mathcal{V}$ is an unknown state action set defined as: $(s,a) \in \mathcal{V}$ iff (s,a) is not in the offline dataset and $T_{\mathcal{V}}^{\pi_r}$ denotes the hitting time of unknown states.

Proof Building on Lemma 3, we now compare the performance of a policy π under two transition dynamics: $P_r \sim \mathcal{D}(\mathcal{U}_r)$ and $P_0 \sim \mathcal{D}(\mathcal{U})$. For states that are in the dataset, we can establish a relationship based on the definition of the uncertainty set. Specifically: $\mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}\mathbb{E}_{P_r \sim \mathcal{D}(\mathcal{U}_r)}(\|P_0(s,a) - P_r(s,a)\|_1) \leq \max(\beta, \beta_r) = \beta$. When considering the lower-bound performance of a robust policy, defined as $\pi_r = \arg\max_\pi \mathbb{E}_{P \sim \mathcal{D}(\mathcal{U}_r)} J_\rho(\pi, P)$, subject to the constraint $\mathbb{E}_{P \sim \mathcal{D}(\mathcal{U}_r)}\left[\mathbb{E}_{s,a}\mathcal{D}_{\mathrm{TV}}(P, P_B)\right] \leq \beta_r$, the "untrained" region is quantified by the difference $(1-p_r)\beta + p_r(\beta - \beta_r) = \beta - p_r\beta_r$. Under these conditions, it's clear that the probability of disadvantageous scenarios for the robust policy at each step is $1 - (\beta - p_r\beta_r)$. The chance of decoupling at a specific time t is at most $1 - (1-(\beta - p_r\beta_r))^t$. Then we have:

$$
\begin{aligned}
&\mathbb{E}_{P_r \sim \mathcal{D}(\mathcal{U}_r)}[J_{\rho_0^B}(\pi_r, P_r)] - \mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}[J_{\rho_0}(\pi_r, P_0)] \\
\geq &\frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}[\mathcal{D}_{\mathrm{TV}}(\rho_0, \rho_0^B)] + \sum_t \gamma^t \left(1 - (1-(\beta - p_r\beta_r))^t\right) \cdot 2 \cdot R_{\max} \qquad (8) \\
&+ \frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T_{\mathcal{V}}^{\pi_r}}]] \\
\geq &\frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}[\mathcal{D}_{\mathrm{TV}}(\rho_0, \rho_0^B)] + \frac{2R_{\max}\gamma(\beta - p_r\beta_r)}{(1-\gamma)\left(1-\gamma\cdot(1-(\beta - p_r\beta_r))\right)} \qquad (9) \\
&+ \frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T_{\mathcal{V}}^{\pi_r}}]] \\
\geq &\frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}[\mathcal{D}_{\mathrm{TV}}(\rho_0, \rho_0^B)] + \frac{2R_{\max}\gamma}{(1-\gamma)^2}(\beta - p_r\beta_r) + \frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T_{\mathcal{V}}^{\pi_r}}]] \qquad (10)
\end{aligned}
$$

In Equation 8, the first term captures the divergence in initial state distributions between $\mathcal{M}$ and $\mathcal{M}_r$; the second, the divergence for known state-action pairs; and the third, for unknown pairs. Equation 9 uses the identity $\sum_t \gamma^t(1-(1-(\beta - p_r\beta_r))^t) = \frac{\gamma(\beta - p_r\beta_r)}{(1-\gamma)(1-\gamma(1-(\beta-p_r\beta_r)))}$. Equation 10 follows from the inequality $\frac{1}{(1-\gamma)(1-\gamma(1-(\beta-p_r\beta_r)))} \leq \frac{1}{(1-\gamma)^2}$. ■

Theorem 5 *For an ϵ_{π_r} sub-optimal risk-aware policy, we have:*

$$
\begin{aligned}
\mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}[J_{\rho_0}(\pi^*, P_0)] - \mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}[J_{\rho_0}(\pi_r, P_0)] \leq \epsilon_{\pi_r} &+ \frac{4R_{max}}{1-\gamma}\mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}[\mathcal{D}_{TV}(\rho_0, \rho_0^B)] \\
&+ \frac{4\gamma R_{max}}{(1-\gamma)^2}\left(\beta - \frac{1}{2}p_r\beta_r\right) \\
&+ \frac{2R_{max}}{1-\gamma}\mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T_{\mathcal{V}}^{\pi_r}}]] \\
&+ \frac{2R_{max}}{1-\gamma}\mathbb{E}_{P_0 \sim \mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T_{\mathcal{V}}^{\pi^*}}]]
\end{aligned}
$$

Proof By Lemma 4, we have:

$$\begin{aligned}
\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[J_{\rho_0}(\pi_r,P_0)] &\geq \mathbb{E}_{P_r\sim\mathcal{D}(\mathcal{U}_r)}[J_{\rho_0^B}(\pi_r,P_r)] - \frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathcal{D}_{\mathrm{TV}}(\rho_0,\rho_0^B)] \\
&\quad - \frac{2R_{\max}\gamma}{(1-\gamma)^2}(\beta - p_r\beta_r) - \frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T_{\mathcal{V}}^{\pi_r}}]] \qquad (11) \\
&\geq \mathbb{E}_{P_r\sim\mathcal{D}(\mathcal{U}_r)}[J_{\rho_0^B}(\pi^*,P_r)] - \epsilon_{\pi_r} - \frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathcal{D}_{\mathrm{TV}}(\rho_0,\rho_0^B)] \\
&\quad - \frac{2R_{\max}\gamma}{(1-\gamma)^2}(\beta - p_r\beta_r) - \frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T_{\mathcal{V}}^{\pi_r}}]] \qquad (12) \\
&\geq \mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[J_{\rho_0}(\pi^*,P_0)] - \epsilon_{\pi_r} - \frac{4R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathcal{D}_{\mathrm{TV}}(\rho_0,\rho_0^B)] \\
&\quad - \frac{4R_{\max}\gamma}{(1-\gamma)^2}(\beta - \tfrac{1}{2}p_r\beta_r) - \frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T_{\mathcal{V}}^{\pi_r}}]] \\
&\quad - \frac{2R_{\max}}{1-\gamma}\mathbb{E}_{P_0\sim\mathcal{D}(\mathcal{U})}[\mathbb{E}[\gamma^{T_{\mathcal{V}}^{\pi^*}}]] \qquad (13)
\end{aligned}$$

where Equation 11 is obtained by plugging $\pi = \pi^*$ into Lemma 4. Equation 12 is derived using the identity $\epsilon_{\pi_r} = |\mathbb{E}_{P_r\sim\mathcal{D}(\mathcal{U}_r)}[J_{\rho_0^B}(\pi^*,P_r)] - \mathbb{E}_{P_r\sim\mathcal{D}(\mathcal{U}_r)}[J_{\rho_0^B}(\pi_r,P_r)]|$. Next We prove Theorem 2. ■

Proof Consider the RTK policy: $\pi_{\mathrm{RTK}} = \arg\max_\pi \mathbb{E}_{P\sim\mathcal{D}(\mathcal{U}_\epsilon^\pi)} J_\rho(\pi,P)$, subject to the constraint: $\mathbb{E}_{P\sim\mathcal{D}(\mathcal{U}_\epsilon^\pi)}[\mathbb{E}_{s,a}[\mathcal{D}_{\mathrm{TV}}(P,P_B)]] \leq 1-\alpha$. We apply Lemma 4 by setting $\beta_r = 1-\alpha$ and $p_r = p_{\mathrm{RTK}}$. This completes the proof. ■

Appendix B. Hyperparameters

In Table 4 below we list the hyperparameters for each task.

Table 4: Adversarial hyperparameters α and ϵ for each task.

Task	α	ϵ	**Task**	α	ϵ
Antmaze-Umaze	0.01	0.9	Adroit-pen-cloned	0.005	0.8
Antmaze-Medium-Play	0.01	0.9	Adroit-pen-expert	0.06	0.8
Antmaze-Large-Play	0.01	0.9	Adroit-hammer-cloned	0.03	0.9
Antmaze-Umaze-Diverse	0.01	0.9	Adroit-hammer-expert	0.03	0.9
Antmaze-Medium-Diverse	0.01	0.9	Adroit-door-cloned	0.03	0.9
Antmaze-Large-Diverse	0.01	0.9	Adroit-door-expert	0.03	0.9
Mujoco-hopper-medium	0.1	0.9	Adroit-relocate-cloned	0.03	0.9
Mujoco-halfcheetah-medium	0.02	0.8	Adroit-relocate-expert	0.03	0.9
Mujoco-walker2d-medium	0.08	0.7	Mujoco-hopper-medium-replay	0.015	0.9
Mujoco-halfcheetah-medium-replay	0.08	0.7	Mujoco-walker2d-medium-replay	0.08	0.7
Mujoco-hopper-medium-expert	0.1	0.9	Mujoco-halfcheetah-medium-expert	0.1	0.9
Mujoco-walker2d-medium-expert	0.1	0.9	—	—	—