Journal of Machine Learning Research 22 (2021) 1-45 Submitted 11/20; Revised 6/21; Published 9/21

Policy Teaching in Reinforcement Learning via Environment Poisoning Attacks*

Amin Rakhsha ARAKHSHA@MPI-SWS.ORG
Max Planck Institute for Software Systems (MPI-SWS)
Saarbrucken, 66123, Germany

Goran Radanovic GRADANOVIC@MPI-SWS.ORG
Max Planck Institute for Software Systems (MPI-SWS)
Saarbrucken, 66123, Germany

Rati Devidze RDEVIDZE@MPI-SWS.ORG
Max Planck Institute for Software Systems (MPI-SWS)
Saarbrucken, 66123, Germany

Xiaojin Zhu JERRYZHU@CS.WISC.EDU
University of Wisconsin-Madison
Madison, WI 53706, USA

Adish Singla ADISHS@MPI-SWS.ORG
Max Planck Institute for Software Systems (MPI-SWS)
Saarbrucken, 66123, Germany

Editor: Laurent Orseau

Abstract

We study a security threat to reinforcement learning where an attacker poisons the learning environment to force the agent into executing a target policy chosen by the attacker. As a victim, we consider RL agents whose objective is to find a policy that maximizes reward in infinite-horizon problem settings. The attacker can manipulate the rewards and the transition dynamics in the learning environment at training-time, and is interested in doing so in a stealthy manner. We propose an optimization framework for finding an optimal stealthy attack for different measures of attack cost. We provide lower/upper bounds on the attack cost, and instantiate our attacks in two settings: (i) an offline setting where the agent is doing planning in the poisoned environment, and (ii) an online setting where the agent is learning a policy with poisoned feedback. Our results show that the attacker can easily succeed in teaching any target policy to the victim under mild conditions and highlight a significant security threat to reinforcement learning agents in practice.

Keywords: training-time adversarial attacks, reinforcement learning, policy teaching, environment poisoning, security threat

* This manuscript is an extended version of the paper (Rakhsha et al., 2020) that appeared in ICML'20.

1. Introduction

Understanding adversarial attacks on learning algorithms is critical to finding security threats against the deployed machine learning systems and in designing novel algorithms robust to those threats. We focus on *training-time* adversarial attacks on learning algorithms, also known as data poisoning attacks (Huang et al., 2011; Biggio and Roli, 2018; Zhu, 2018). Different from *test-time* attacks where the adversary perturbs test data to change the algorithm's decisions, poisoning attacks manipulate the training data to change the algorithm's decision-making policy itself.

Most of the existing work on data poisoning attacks has focused on supervised learning algorithms (Biggio et al., 2012; Mei and Zhu, 2015; Xiao et al., 2015; Alfeld et al., 2016; Li et al., 2016; Koh et al., 2018). In contemporary works, researchers have explored data poisoning attacks against stochastic multi-armed bandits (Jun et al., 2018; Liu and Shroff, 2019) and contextual bandits (Ma et al., 2018), which belong to family of online learning algorithms with limited feedback—such algorithms are widely used in real-world applications such as news article recommendation (Li et al., 2010) and web advertisements ranking (Chapelle et al., 2014). The feedback loop in online learning makes these applications easily susceptible to data poisoning, e.g., attacks in the form of click baits (Miller et al., 2011).

In this paper, we focus on data poisoning attacks against reinforcement learning (RL) algorithms, an online learning paradigm for sequential decision-making under uncertainty (Sutton and Barto, 2018).[1] Given that RL algorithms are increasingly used in critical applications, including cyber-physical systems (Li and Qiu, 2019) and personal assistive devices (Rybski et al., 2007), it is of utmost importance to investigate the security threat to RL algorithms against different forms of poisoning attacks. As pointed out by (Zhang et al., 2020b), a canonical example of the negative impact that data poisoning attacks can have on a sequential decision-making system is a Microsoft chatbot Tay—a conversational AI agent with learning capabilities that got corrupted within a few hours of interacting with ill-intentioned Twitter users (Neff, 2016). More generally, multi-agent settings represent a fruitful ground for data poisoning attacks since agents influence each others' learning data (e.g., rewards given by the environment). A similar observation has been made in the context of test-time attacks on RL (Gleave et al., 2019).

In contrast to bandits, which is a specific case of RL, data poisoning attacks on RL agents can generally influence both the reward feedback given to the agent and the state to which the agent transitions, i.e., its transition dynamics. The suitability of the attack form and how it is operationalized depends on the application scenario as well as the model of the learning agent. For example, if the reward function is intrinsic (e.g., the agent derives it from the current state), then reward poisoning attacks are not applicable, and the attacker can only poison transition dynamics. On the other hand, in many practical scenarios, such as in the aforementioned chatbot example, both rewards and transitions are indirectly poisoned through interaction data. In this work, we initiate the research direction on hybrid attacks, where we assume that rewards and transitions can be poisoned independently and at different costs. While one might be more constrained in how rewards and transitions are allowed

[1] Poisoning attacks is also mathematically equivalent to the formulation of machine teaching with teacher being the adversary (Zhu et al., 2018). However, the problem of designing optimized environments for teaching a target policy to an RL agent is not well-understood in machine teaching.

to change in real-world applications, we believe that the model we consider yields valuable insights into hybrid attacks.

1.1 Overview of our Results and Contributions

In the following, we discuss a few of the types/dimensions of poisoning attacks in RL in order to highlight the novelty of our work in comparison to existing work.[2]

Type of adversarial manipulation. Existing works on poisoning attacks against RL have studied the manipulation of rewards only (Zhang and Parkes, 2008; Zhang et al., 2009; Ma et al., 2019; Huang and Zhu, 2019; Zhang et al., 2020b). However, for certain applications, it is more natural to manipulate the transition dynamics instead of the rewards, such as (i) the inventory management problem setting where state transitions are controlled by demand and supply of products in a market and (ii) conversational agents where the state is represented by the history of conversations. A key novelty of our work is that we study environment poisoning, i.e., jointly manipulating rewards and transition dynamics. We propose a general optimization framework for environment poisoning; our theoretical analysis provides technical conditions which ensures the attacker's success and gives lower/upper bounds on the attack cost.

Objective of the learning agent. Existing works have focused primarily on studying RL agents that maximizes *cumulative reward in discounted* infinite-horizon settings. However, for many real-world applications, it is more appropriate to consider RL agents that maximizes *average reward in undiscounted* infinite-horizon settings (Puterman, 1994; Mahadevan, 1996)— in particular, this is a more suitable objective for applications that have cyclic tasks or tasks without absorbing states, e.g., inventory management and scheduling problems (Tadepalli and Ok, 1994; Puterman, 1994), and a robot learning to avoid obstacles (Mahadevan and Connell, 1992). In our work, the proposed optimization framework and theoretical results cover both the optimality criteria— average reward criteria and discounted reward criteria—in infinite-horizon settings.

Offline planning and online learning. Most of the existing works have focused on attacks in an *offline* setting where the adversary first poisons the reward function in the environment and then the RL agent finds a policy via *planning* (Zhang and Parkes, 2008; Zhang et al., 2009; Ma et al., 2019). In contrast, we call a setting as *online* where the adversary interacts with a *learning* agent to manipulate the feedback signals. One of the key differences in these two settings is in measuring the attacker's cost: The ℓ_∞-*norm* of manipulation is commonly studied for the offline setting; for the online setting, the cumulative cost of attack over time (e.g., measured by ℓ_1-*norm* of manipulations) is more relevant and has not been studied in literature. A recent contemporary work (Zhang et al., 2020b) considers the online

[2]This paper extends the earlier version of the conference paper (Rakhsha et al., 2020) in the following ways. **(i)** The earlier version studied attacks by poisoning either rewards only or transitions only. In this paper, we introduce a general optimization framework for jointly poisoning the rewards and transitions. We provide new theoretical analysis for the joint attack and empirically show that it leads to more cost-effective attack strategies. **(ii)** The earlier version studied attacks only against RL agents who maximize average reward in undiscounted infinite-horizon settings. In this paper, we generalize these results by additionally considering discounted infinite-horizon settings. This generalization in turn makes our attack strategies applicable to a broader family of RL agents (e.g., an agent using the Q-learning algorithm). **(iii)** We provide a detailed discussion on the efficiency of solving the attack optimization problems.

setting, however, does not study ℓ_1-*norm* of manipulations as the attack cost. We instantiate our attacks in both the offline and online settings with appropriate notions of attack cost.

We note that our attacks are constructive, and we provide numerical simulations to support our theoretical statements. Our results demonstrate that the attacker can easily succeed in teaching (forcing) the victim to execute the desired target policy at a minimal cost.

1.2 Additional Related Work

Below, we discuss a few other relevant lines of research that are related to our work.

Test-time attacks against RL. A growing body of contemporary works have studied test-time attacks against RL (Chen et al., 2019), in particular, on RL algorithms with neural network policies (Mnih et al., 2015; Schulman et al., 2015). These attacks are typically done by adding noise in the observed state (e.g., a camera image) to fool the neural network policy into taking malicious actions (Huang et al., 2017; Lin et al., 2017; Tretschk et al., 2018). Different attack goals have been considered in these works, e.g., guiding the agent to some adversarial states or forcing agent to take actions that maximizes adversary's own rewards. Our work is technically quite different and is focused on training-time attacks where the goal is to force the agent to learn a target policy.

Teaching an RL agent. Poisoning attacks is mathematically equivalent to the formulation of machine teaching with teacher being the adversary (Goldman and Kearns, 1995; Singla et al., 2013, 2014; Zhu, 2015; Zhu et al., 2018; Chen et al., 2018; Mansouri et al., 2019; Peltola et al., 2019; Devidze et al., 2020). In particular, there have been a number of recent works on teaching an RL agent via providing an optimized curriculum of demonstrations (Cakmak and Lopes, 2012; Walsh and Goschin, 2012; Hadfield-Menell et al., 2016; Haug et al., 2018; Kamalaruban et al., 2019; Tschiatschek et al., 2019; Brown and Niekum, 2019). However, these works have focused on imitation-learning based RL agents who learn from provided demonstrations without any reward feedback (Osa et al., 2018). Given that we consider RL agents who find policies based on rewards, our work is technically very different from theirs. There is also a related literature on changing the behavior of an RL agent via *reward shaping* (Ng et al., 1999; Asmuth et al., 2008); here the reward function is changed to only speed up the convergence of the learning algorithm while ensuring that the optimal policy in the modified environment is unchanged.

Robust RL agents. Complementary to the literature on adversarial attacks in RL is the work on designing robust RL agents capable of performing well even in the presence of adversaries. Much of the prior work on robust RL focuses on studying robustness to model uncertainties (McMahan et al., 2003; Nilim and El Ghaoui, 2005; Iyengar, 2005; Regan and Boutilier, 2010; Tamar et al., 2014), and in recent years on robustness to test-time attacks (e.g., adversarial perturbations on observations) (Pinto et al., 2017; Fischer et al., 2019; Zhang et al., 2020a, 2021a). To our knowledge, robustness to data poisoning attacks in RL has been much less explored, with a few notable exceptions. In this line of work, the closest to this paper is the work of (Banihashem et al., 2021), who study defenses against reward poisoning attacks. More specifically, they complement our results by providing defense mechanisms in an offline setting that can, under certain conditions, provably limit the influence of reward poisoning attacks. It is also worth mentioning the works on learning under corrupted rewards

and transitions in episodic RL (Lykouris et al., 2019; Zhang et al., 2021b), which consider models of corruption orthogonal to the attack models studied in this paper.

2. Environment and RL Agent

We consider a standard RL setting, based on Markov decision processes and RL agents that optimize their expected utility. In a unified manner, we will cover two cases in which RL agents are optimizing their total discounted rewards or their undiscounted average reward. The following subsections will introduce our setting in more detail.

2.1 Environment, Policy, and Optimality Criteria

The environment is a Markov Decision Process (MDP) defined as $M = (S, A, R, P, \gamma)$, where S is the state space, A is the action space, $R\colon S \times A \to \mathbb{R}$ is the reward function, $P\colon S \times A \times S \to [0, 1]$ is the state transition dynamics, i.e., $P(s, a, s')$ denotes the probability of reaching state s' when taking action a in state s, and $\gamma \in [0, 1]$ is the discounting factor, and d_0 is the initail state distribution. Note that discount factor γ can be equal to 1, which we treat as a special case, as explained in the paragraphs below.

We consider a class of deterministic policies: we denote a generic deterministic policy by π, and we define it as a mapping from states to actions, i.e., $\pi\colon S \to A$. Furthermore, we assume that MDP M is *ergodic*, which implies that every policy π has a *state* distribution μ^π defined as:

$$\mu^\pi(s) = \begin{cases} (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}\left[s_t = s | s_0 \sim d_0, \pi\right] & \text{if } \gamma < 1 \\ \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} \mathbb{P}\left[s_t = s | s_0 \sim d_0, \pi\right] & \text{if } \gamma = 1, \end{cases} \tag{1}$$

which satisfies $\mu^\pi(s) > 0$ for every state s (Puterman, 1994). For $\gamma = 1$, μ^π corresponds to the *stationary* state distribution induced by policy π, whereas for $\gamma < 1$, μ^π corresponds to the *discounted* state distribution induced by policy π. State distribution μ^π satisfies the following Bellman flow constraints:

$$\mu^\pi(s) = (1-\gamma) \cdot d_0(s) + \gamma \cdot \sum_{s'} P(s', \pi(s'), s) \cdot \mu^\pi(s'). \tag{2}$$

Given an initial state distribution d_0, the expected average and discounted reward of policy π are respectively equal to

$$\lim_{N\to\infty} \frac{1}{N} \mathbb{E}\left[\sum_{t=0}^{N-1} R(s_t, a_t) | s_0 \sim d_0, \pi\right] \text{ and } \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | s_0 \sim d_0, \pi\right],$$

where the expectations are taken over the rewards received by the agent when starting from initial state $s_0 \sim d_0$ and following the policy π. In our work, we cover both the *average reward* (Puterman, 1994; Mahadevan, 1996) and the *discounted reward* optimality criteria in infinite-horizon settings (Puterman, 1994; Sutton and Barto, 2018). Throughout the paper, the special case of $\gamma = 1$ represents the average reward problem setting, whereas $\gamma < 1$ represents the discounted reward problem setting. We unify these cases using a single *score*

of policy π defined as

$$\rho(\pi, M, d_0) := \sum_s \mu^\pi(s) \cdot R(s, \pi(s)). \tag{3}$$

Using policy scores ρ, we define the notion of optimality used in this paper. A policy π^* is optimal if for every other deterministic policy π we have $\rho^{\pi^*} \geq \rho^\pi$, and ϵ-robust optimal if $\rho^{\pi^*} \geq \rho^\pi + \epsilon$ also holds. As shown in prior work (and discussed later in the paper), score ρ is closely related to the standard notions of state-action and state value functions, i.e., Q-values and V-values. For policy π, (shifted) Q-values and V-values are defined as[3]

$$Q^\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \cdot (r_t - \rho^\pi) | s_0 = s, a_0 = a, \pi\right], \quad V^\pi(s) = Q^\pi(s, \pi(s)),$$

and they satisfy the following Bellman equations:

$$Q^\pi(s,a) = R(s,a) - \rho^\pi + \gamma \cdot \sum_{s' \in S} P(s,a,s') \cdot V^\pi(s'). \tag{4}$$

Finally, we introduce quantities that measure *connectedness* of MDP M. First, we define a coefficient $\alpha = \min_{s,a,s',a'} \sum_{x \in S} \min\big(P(s,a,x), P(s',a',x)\big)$, so that $(1-\alpha)$ is equivalent to Hajnal measure of P (Puterman, 1994).[4] Second, we define the notion of *discounted reach times* for policy π as $T^\pi(s,s) = 0$ and $T^\pi(s,s') = \mathbb{E}\left[\sum_{i=0}^{L^\pi(s,s')-1} \gamma^i\right]$ for $s' \neq s$, where $L^\pi(s,s')$ is a random variable that counts the number of steps it takes to visit state s' for the first time starting from s and following π in M. The maximum discounted reach time for policy π is denoted by $D^\pi = \max_{s,s'} T^\pi(s,s')$, i.e., D^π denotes the diameter of Markov chain induced by policy π in MDP M. Note that reach times T^π satisfy the following recursive equations for $s \neq s'$:

$$T^\pi(s,s') = \sum_{s'' \in S} P(s, \pi(s), s'') \cdot (1 + \gamma \cdot T^\pi(s'', s')),$$

which means that for a given MDP M and policy π, one can compute them efficiently since this is just a system of linear equations.

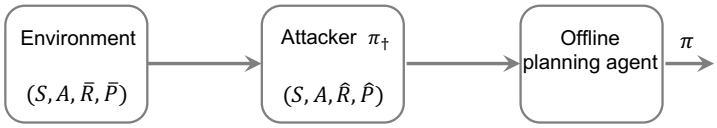
2.2 RL Agent

We consider RL agents in the following two settings (also, see Figure 1).

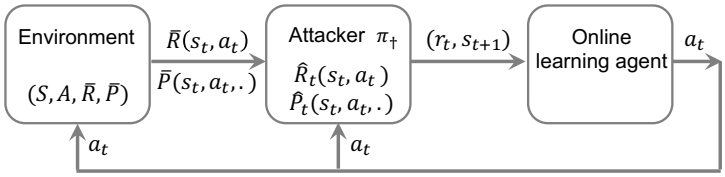
Offline planning agent. In the *offline* setting, an RL agent is given an MDP M, and chooses a deterministic optimal policy $\pi^* \in \arg\max_\pi \rho(\pi, M, d_0)$. The optimal policy can be found via *planning* algorithms based on Dynamic Programming such as value iteration (Puterman, 1994; Sutton and Barto, 2018).

[3] To facilitate exposure of our results in a unified manner, in the discounted reward setting, we are using Q-values that are shifted from the standard definition by $\rho^\pi/(1-\gamma)$. This modification allows using the same Bellman equations for both the average and discounted rewards settings. When referring to standard Q-values, we use the symbol $\mathcal{Q}$ instead (e.g., as used in the appendices).

[4] As discussed in (Puterman, 1994), the Hajnal measure of a Markov chain transition matrix provides an upper bound on its subradius (the modulus of the second largest eigenvalue). Hence, this measure is informative about the mixing times in MDP M.



(a) Poisoning attack against an RL agent doing offline planning



(b) Poisoning attack against an RL agent doing online learning

Figure 1: (a) Adversary first poisons the environment by manipulating reward function and transition dynamics, then, the RL agent finds an optimal policy via *planning* algorithms based on Dynamic Programming (Puterman, 1994; Sutton and Barto, 2018). (b) Adversary interacts with an RL agent to manipulate the feedback signals; here, we consider an agent who is learning a policy based on feedback received from the environment (see Section 2.2 for details).

Online learning agent. In the *online* setting, an RL agent does not know the MDP M (i.e., R and P are unknown). At each step t, the agent stochastically chooses an action a_t based on the previous observations, and then as feedback it obtains reward r_t and transitions to the next state s_{t+1}. In this paper, we consider agents with two performance measures:

1. For the case of average reward criteria with $\gamma = 1$, we consider a regret-minimization learner. Performance of a regret-minimization learner in MDP M is measured by its *regret* which after T steps is given by $\textsc{Regret}(T, M) = \rho^* \cdot T - \sum_{t=0}^{T-1} r_t$, where $\rho^* := \rho(\pi^*, M)$ is the optimal score. Well-studied algorithms with sublinear regret exist for average reward criteria, e.g., UCRL algorithm (Auer and Ortner, 2007; Jaksch et al., 2010) and algorithms based on posterior sampling method (Agrawal and Jia, 2017). For more details, we refer the reader to Appendix B.

2. For the case of discounted reward criteria with $\gamma < 1$, the type of learners we consider are evaluated based on the number of suboptimal steps they take. An agent is suboptimal at time step t if it takes an action not used by any near-optimal policy. This is formulated as $\textsc{SubOpt}(T, M, \epsilon') = \sum_{t=0}^{T-1} \mathbb{1}\left[a_t \notin \{\pi(s_t) \mid \rho^\pi \geq \rho^{\pi^*} - \epsilon'\}\right]$ where $\mathbb{1}[.]$ denotes the indicator function and ϵ' measures near-optimality of a policy w.r.t. score ρ. Our analysis of attacks is based on $\mathbb{E}\left[\textsc{SubOpt}(T, M, \epsilon')\right]$ of the learner for a specific value of ϵ'. Some bounds on this quantity are known for existing algorithms such as classic Q-learning (Even-Dar and Mansour, 2003) and Delayed Q-learning (Strehl et al., 2006) as discussed in more detail in Appendix B.

3. Attack Models and Problem Formulation

In this section, we formulate the problem of adversarial attacks on the RL agent in both the offline and online settings. In what follows, the original MDP (before poisoning) is denoted by $\overline{M} = (S, A, \overline{R}, \overline{P}, \gamma)$, and an *overline* is added to the corresponding quantities before poisoning, such as $\overline{\rho}^{\pi}$, $\overline{\mu}^{\pi}$, and $\overline{\alpha}$.

The attacker has a target policy $\pi_\dagger$ and poisons the environment with the *goal* of teaching/forcing the RL agent to executing this policy. The attacker is interested in doing a stealthy attack with minimal *cost* to avoid being detected. We assume that the attacker knows the original MDP $\overline{M}$, i.e., the original reward function $\overline{R}$ and state transition dynamics $\overline{P}$. This assumption is standard in the existing literature on poisoning attacks against RL. The attacker requires that the RL agent behaves as specified in Section 2, however the attacker does not know the agent's algorithm nor internal parameters. In the offline setting, the agent acts as if the poisoned MDP is the original (true) MDP, and in the online setting, the agent acts as if the poisoned feedback is the true environment feedback. In other words, the attacker does not need to additionally reason about defense strategies. See further discussion in Section 7 about defenses against these attacks.

3.1 Attack Against an Offline Planning Agent

In attacks against an offline planning agent, the attacker manipulates the original MDP $\overline{M} = (S, A, \overline{R}, \overline{P}, \gamma)$ to a poisoned MDP $\widehat{M} = (S, A, \widehat{R}, \widehat{P}, \gamma)$ which is then used by the RL agent for finding the optimal policy, see Figure 1a.[5]

Goal of the attack. Given a margin parameter ϵ, the attacker poisons the reward function and the transition dynamics so that the target policy $\pi_\dagger$ is ϵ-robust optimal in the poisoned MDP $\widehat{M}$, i.e., the following condition holds:

$$\rho(\pi_\dagger, \widehat{M}, d_0) \geq \rho(\pi, \widehat{M}, d_0) + \epsilon, \quad \forall \pi \neq \pi_\dagger. \tag{5}$$

Cost of the attack. To define the cost of an attack, we quantify for each state-action pair how much the attack changes the reward and transition dynamics associated to this state-action pair, i.e., $|\widehat{R}(s,a) - \overline{R}(s,a)|$ and $\sum_{s'} \left|\widehat{P}(s,a,s') - \overline{P}(s,a,s')\right|$ respectively. The cost for that state action-pair is the weighted sum of these values, with weights given by two parameters, C_r and C_p respectively. The total cost is then measured as the ℓ_p-norm

[5] Our results can be translated to attacks against the Batch RL agent studied by (Ma et al., 2019) where the attacker poisons the training data used by the agent to learn MDP parameters. In particular, the model is derived from tuples (s, a, r, s') and the attacker can modify these data points. Poisoning rewards can be operationalized by changing r and poisoning transitions could be operationalized by changing s'. In the latter case, note that we do not explicitly model the fine-grained pairwise cost of modifying state s' to another state s''. Modeling such a fine-grained cost is important when states have continuous representation (e.g., observations in the form of images) and we leave this as future work; see Section 7.

(for $p \geq 1$) of the costs across all state-action pairs.[6] Formally, this can be written as

$$\textsc{Cost}(\widehat{M}, \overline{M}, C_r, C_p, p) = \left(\sum_{s,a} \left(C_r \cdot \left| \widehat{R}(s,a) - \overline{R}(s,a) \right| + C_p \cdot \sum_{s'} \left| \widehat{P}(s,a,s') - \overline{P}(s,a,s') \right| \right)^p \right)^{1/p}.$$

We will write the cost as $\textsc{Cost}(\widehat{M}, \overline{M})$ when other parameters are clear from the context. By taking the limits of Cp (resp. Cr) to infinity, the cost function enforces the attacker to poison only the rewards (resp. only the transitions).

3.2 Attack Against an Online Learning Agent

In attacks against an online learning agent, the attacker at time t manipulates the reward function $\overline{R}(s_t, a_t)$ and transition dynamics $\overline{P}(s_t, a_t, .)$ for the current state s_t and the agent's action a_t, see Figure 1b. Then, at time t, the (poisoned) reward r_t is obtained from $\widehat{R}_t(s_t, a_t)$ instead of $\overline{R}(s_t, a_t)$ and the (poisoned) next state s_{t+1} is sampled from $\widehat{P}_t(s_t, a_t, .)$ instead of $\overline{P}(s_t, a_t, .)$.[7]

Goal of the attack. Specification of the attacker's goal in this online setting is not as straightforward as that in the offline setting, primarily because the agent might never converge to any stationary policy. In our work, at time t when the current state is s_t, we measure the mismatch of agent's action a_t w.r.t. the target policy $\pi_\dagger$ as $\mathbb{1}\left[a_t \neq \pi_\dagger(s_t)\right]$. With this, we define a notion of *average mismatch* of learner's actions in time horizon T as follows:

$$\textsc{AvgMiss}(T) = \frac{1}{T} \cdot \left(\sum_{t=0}^{T-1} \mathbb{1}\left[a_t \neq \pi_\dagger(s_t)\right] \right). \tag{6}$$

The goal of the attacker is to minimize $\textsc{AvgMiss}(T)$.

Cost of the attack. We consider a notion of *average cost* of attack in time horizon T denoted as $\textsc{AvgCost}(T)$. This is defined as

$$\frac{1}{T} \cdot \left(\sum_{t=0}^{T-1} \left(C_r \cdot \left| \widehat{R}_t(s_t, a_t) - \overline{R}(s_t, a_t) \right| + C_p \cdot \sum_{s'} \left| \widehat{P}_t(s_t, a_t, s') - \overline{P}(s_t, a_t, s') \right| \right)^p \right)^{1/p},$$

where the ℓ_p-norm (for $p \geq 1$) is defined over a vector of length T with values quantifying the attack cost at each time step t. One of the key differences in measuring attacker's cost for offline and online settings is the use of appropriate norm. While the ℓ_∞-*norm* of manipulation is more suitable and commonly studied for the offline setting; for the online setting, the cumulative cost of attack over time measured by the ℓ_1-*norm* is more relevant.

[6] Note that there are alternatives to this cost formulation. For example, one could separately define the total cost of reward poisoning and the total cost of transition poisoning using ℓ_p-norms, and define the joint cost as a linear combination of the two ℓ_p-norms. Therefore, the structure of the cost function is a modeling choice. The problem formulation with other cost functions is left for future work.

[7] In online settings, there is a subtle difference between "physical" attacks (the next state is physically altered) versus "perception" attacks (the next state is altered only from the agent's point of view, e.g., by adding noise in the observed image representing the state). Our formulation in this paper models such physical attacks; poisoning observations in an online setting is more subtle as it requires the attacker to be consistent with the underlying dynamics and we leave the analysis of this case for future work.

4. Attacks in Offline Setting

In this section, we introduce and analyze attacks against an offline planing agent that derives its policy using a poisoned MDP $\widehat{M}$. The attacker tries to minimally change the original MDP $\overline{M}$, while at the same time ensuring that the target policy is optimal in the modified MDP $\widehat{M}$.

4.1 Offline Attacks: Key Ideas and Attack Problem

The main obstacle in performing offline attacks as formulated in Section 3.1 is the complexity of the optimization problem it leads to. To find MDP $\widehat{M}$ for which the target policy is ϵ-robust optimal, one could directly utilize constraints expressed by (5). However, the number of constraints in (5) equals $(|A|^{|S|}-1)$, i.e., it is exponential in $|S|$, making optimization problems that directly utilize them intractable. We show that it is enough to satisfy these constraints for $(|S|\cdot|A|-|S|)$ policies that we call *neighbors* of the target policy and which we define as follows:

Definition 1. *For a policy π, its neighbor policy $\pi\{s;a\}$ is defined as*

$$\pi\{s;a\}(x) = \begin{cases} \pi(x) & x \neq s \\ a & x = s \end{cases}.$$

The following lemma provides a simple verification criterion for examining whether a policy of interest is ϵ-robust optimal in a given MDP. Its proof can be found in Appendix D.

Lemma 1. *Policy π is ϵ-robust optimal* iff *we have $\rho^{\pi} \geq \rho^{\pi\{s;a\}} + \epsilon$ for every state s and action $a \neq \pi(s)$.*

In other words, Lemma 1 implies that (sub)optimality of the target policy can be deduced by examining its neighbor policies. Using the definition of score ρ, Bellman flow constraints (i.e., equation (2)), and Lemma 1, we can formulate the problem of modifying MDP $\overline{M} = (S, A, \overline{R}, \overline{P})$ to MDP $\widehat{M} = (S, A, \widehat{R}, \widehat{P})$ as the following optimization problem:

$$\min_{M,R,P,\mu^{\pi_\dagger},\mu^{\pi_\dagger\{s;a\}}} \quad \textsc{Cost}(M, \overline{M}, C_r, C_p, p) \tag{P1}$$

$$\begin{aligned}
\text{s.t.} \quad & \mu^{\pi_\dagger} \text{ and } P \text{ satisfy (2)}, \\
& \forall s, a \neq \pi_\dagger(s) : \mu^{\pi_\dagger\{s;a\}} \text{ and } P \text{ satisfy (2)}, \\
& \forall s, a \neq \pi_\dagger(s) : \sum_{s'} \mu^{\pi_\dagger}(s') \cdot R\big(s', \pi_\dagger(s')\big) \geq \sum_{s'} \mu^{\pi_\dagger\{s;a\}}(s') \cdot R\big(s', \pi_\dagger\{s;a\}(s')\big) + \epsilon, \\
& \forall s, a, s' : P(s, a, s') \geq \delta \cdot \overline{P}(s, a, s'), \\
& M = (S, A, R, P, \gamma).
\end{aligned}$$

Here, $\delta \in (0, 1]$ in the last set of constraints is a given parameter, specifying how much one is allowed to decrease the original values of transition probabilities. $\delta > 0$ is a regularity condition which ensures that the new MDP is ergodic.[8] In Appendix E, we provide a more detailed discussion on δ and how to choose it. In the next section, we analyze this problem in detail, providing bounds and discussions on its solution.

[8] This follows because strictly positive trajectory probabilities in MDP $\overline{M}$ remain strictly positive in the new MDP $\widehat{M}$, which further implies that all states remain recurrent.

4.2 Offline Attacks: Theoretical Analysis

We start our analysis by defining quantities relevant for stating our formal results. Notice that we denote the quantities defined w.r.t. to the original MDP $\overline{M}$ by putting an *overline*. For example, $\overline{V}^{\pi}(s)$ and $\overline{Q}^{\pi}(s,a)$ denote V-values and Q-values of policy π in MDP $\overline{M}$. As a measure of the relative optimality gap between the target policy $\pi_\dagger$ and its neighbor policies $\pi_\dagger\{s;a\}$, we define the following state-action dependent variable:

$$\overline{\chi}^{\pi}_{\epsilon'}(s,a) = \begin{cases} \left[\frac{\overline{\rho}^{\pi\{s;a\}} - \overline{\rho}^{\pi} + \epsilon'}{\overline{\mu}^{\pi\{s;a\}}(s)}\right]^{+} & \text{for } a \neq \pi(s), \\ 0 & \text{for } a = \pi(s). \end{cases} \tag{7}$$

Here, $[x]^{+}$ is equal to $\max\{0, x\}$. As we show in our formal results, $\overline{\chi}^{\pi}_{\epsilon'}$ with an appropriately set ϵ' captures how much one should change state-action values $\overline{Q}^{\pi_\dagger}(s,a)$ relative to state values $\overline{V}^{\pi_\dagger}(s)$ in order to obtain a successful attack. Both the lower and the upper bound on the cost of the attack in Theorem 1 depend on $\overline{\chi}^{\pi}_{\epsilon'}$. Intuitively, the bounds will increase with the value of $\overline{\chi}^{\pi}_{\epsilon'}$. To simplify our formal statements, we also set

$$\overline{\beta}(s,a) = \epsilon \cdot \overline{\mu}^{\pi_\dagger\{s;a\}}(s) \cdot \frac{1+\gamma\cdot\overline{D}^{\pi_\dagger}}{1-(1-\gamma)\cdot\overline{D}^{\pi_\dagger}}. \tag{8}$$

Furthermore, we utilize the span of value function, defined as $sp(\overline{V}^{\pi_\dagger}) = \max_s \overline{V}^{\pi_\dagger}(s) - \min_s \overline{V}^{\pi_\dagger}(s)$, in the lower bound in Theorem 1. In order to define the other quantities of interest, we order states by their values. In particular, s_i denotes an order of states for $1 \leq i \leq |S|$ such that $\overline{V}^{\pi_\dagger}(s_i)$ is decreasing with i. This order allows us to define two important state-action dependent quantities $\overline{F}_i$ and $\overline{G}_i$, used in the upper bound on the cost of the attack in Theorem 1. The upper bound is obtained by studying a specific attack, and intuitively, $\overline{F}_i$ captures the efficiency of the attack when poisoning certain transitions, while $\overline{G}_i$ captures how much the attack perturbs the transition dynamics in poisoning these transitions. We define these quantities as

$$\overline{F}_i(s,a) = \gamma \cdot \sum_{j=1}^{i} (1-\delta)\cdot\overline{P}(s,a,s_j)(\overline{V}^{\pi_\dagger}(s_j) - \overline{V}^{\pi_\dagger}(s_{|S|})),$$

$$\overline{G}_i(s,a) = 2 \cdot \sum_{j=1}^{i} (1-\delta)\cdot\overline{P}(s,a,s_j)$$

for $a \neq \pi_\dagger(s)$, and $\overline{F}_i(s,\pi_\dagger(s)) = \overline{G}_i(s,\pi_\dagger(s)) = 0$ otherwise. We further set $F_0(s,a) = 0$ and $G_0(s,a) = 0$. Finally, to define the attack, we also need to specify i in $\overline{F}_i$ and $\overline{G}_i$, and the selection is based on whether it is more efficient to poison transitions or rewards. In particular, for each state-action pair, we define a number $k(s,a)$ to be the largest element in $\{1, ..., |S|\}$ such that $\gamma\cdot C_r \cdot (\overline{V}^{\pi_\dagger}(s_{k(s,a)}) - \overline{V}^{\pi_\dagger}(s_{|S|})) > 2\cdot C_p$ and $\overline{F}_{k(s,a)}(s,a) \leq \overline{\chi}^{\pi_\dagger}_{\overline{\beta}(s,a)}(s,a)$. If these conditions cannot be satisfied, we set $k(s,a) = 0$ (in our results, this case would correspond to changing only rewards). We provide a more detailed discussion on these quantities later in the section. We can now state the main result for the offline attack setting.

Theorem 1. *If $\widehat{M}$ is an optimal solution to optimization problem (P1), then*

$$\frac{1-\gamma+\gamma\cdot\delta\cdot\overline{\alpha}}{2\cdot {C_r}^{-1}+\gamma\cdot {C_p}^{-1}\cdot sp(\overline{V}^{\pi_\dagger})}\left\|\overline{\chi}_0^{\pi_\dagger}\right\|_\infty \leq \textsc{Cost}(\widehat{M},\overline{M}) \leq \left\|C_p\cdot\overline{G}_k + C_r\cdot(\overline{\chi}_{\overline{\beta}}^{\pi_\dagger}-\overline{F}_k)\right\|_p,$$

where $\overline{\chi}_{\overline{\beta}}^{\pi_\dagger}$, $\overline{G}_k$, and $\overline{F}_k$, are vectors of length $|S|\cdot|A|$ with components $\overline{\chi}_{\overline{\beta}(s,a)}^{\pi_\dagger}(s,a)$, $\overline{G}_{k(s,a)}(s,a)$, and $\overline{F}_{k(s,a)}(s,a)$.

This theorem gives lower and upper bounds on the cost of offline attacks against a planning agent. In the proof sketch below, we provide some intuition on how the bounds in Theorem 1 are derived; the full proof can be found in Appendix E. As discussed in the earlier version of the paper (Rakhsha et al., 2020), we note the following two points:

- *transition poisoning* (i.e., limit case $\frac{C_p}{C_r}\to 0$): the attack that only poisons transitions might not always be feasible and the optimization problem (P1) in this case reduces to an optimization problem with quadratic equality constraints. This point further indicates, as we also discuss in Section 4.3, that finding an optimal solution to (P1) is computationally challenging.
- *reward poisoning* (i.e., limit case $\frac{C_r}{C_p}\to 0$): the attack that only poisons rewards is always feasible and it is also computationally tractable. This point further implies that the general attack (poisoning rewards and transitions) defined by the optimization problem (P1) is always feasible. Moreover, one can obtain tighter bounds than in Theorem 1 – in particular, $\overline{\chi}_0^{\pi_\dagger}$ from the lower bound and $\overline{\chi}_{\overline{\beta}}^{\pi_\dagger}$ from the upper bound become $\overline{\chi}_\epsilon^{\pi_\dagger}$, while $\gamma\cdot {C_p}^{-1}\cdot sp(\overline{V}^{\pi_\dagger})$, $C_p\cdot\overline{G}_k$, and $C_r\cdot\overline{F}_k$ do not appear in the bounds.

Proof Sketch of Theorem 1. We split the proof-sketch in two parts, corresponding to the upper and the lower bound respectively.

Upper bound on the cost. We provide a constructive upper bound by introducing an attack which is a solution to optimization problem (P1). The main idea behind this approach is that at each state s, we can make each action $a\neq\pi_\dagger(s)$ less valuable than action $\pi_\dagger(s)$ by decreasing $\overline{R}(s,a)$ (rewards) and changing the next state distribution (transitions). The attack that we consider first modifies the transition dynamics, until the point when it becomes less cost effective to change transitions than the rewards. In the second phase, the attack changes only the rewards.

To make an action less valuable, the attacker can decrease the probability of transitioning to a high-value state and increase the probability of transitioning to a low-value state, with state values being obtained from value function $\overline{V}^{\pi_\dagger}(s)$. To construct an upper bound, we analyze an attack that for each state-action pair (s,a) decreases the probability of transitioning to the top $k(s,a)$ highest valued states and increases probability $\overline{P}(s,a,s_{|S|})$ by the amount that is equal to the total decrease. Note that this attack should not violate the ergodicity constraint, i.e., $\overline{P}(s,a,s_i)$ can be decreased by at most $(1-\delta)\cdot\overline{P}(s,a,s_i)$. For the considered attack, the transition probabilities that correspond to the $k(s,a)$ highest valued states are maximally decreased, i.e., we modify $\overline{P}(s,a,s_i)$ to $\delta\cdot\overline{P}(s,a,s_i)$ for $i\leq k(s,a)$. We now argue that $k(s,a)$ as specified in the theorem accounts for two important factors: minimizing the amount of change and optimizing the efficiency of changes.

First, note that we should not change the original MDP more than it is needed. When changing transitions of state-action pair (s,a) for $k(s,a)$ highest-valued states using the

approach described above, the state-action value function of that pair, i.e., $\overline{Q}^{\pi_\dagger}(s,a)$, decreases by $\overline{F}_{k(s,a)}(s,a)$. As we show in our analysis, $\overline{\chi}^{\pi_\dagger}_{\overline{\beta}(s,a)}(s,a)$ captures how much state-action values, i.e., $\overline{Q}^{\pi_\dagger}(s,a)$, should be decreased in order for the attack to be successful. This means that we should select $k(s,a)$ so that $\overline{F}_{k(s,a)}(s,a)$ does not exceed $\overline{\chi}^{\pi_\dagger}_{\overline{\beta}(s,a)}(s,a)$, i.e., $\overline{F}_{k(s,a)}(s,a) \leq \overline{\chi}^{\pi_\dagger}_{\overline{\beta}(s,a)}(s,a)$.

Second, note that we need to account for the efficiency of the modifications. Let us first consider the case of $k(s,a) > 0$. The difference $V^{\pi_\dagger}(s_{k(s,a)}) - V^{\pi_\dagger}(s_{|S|})$ should be large enough so that modifying transitions $\overline{P}(s,a,\cdot)$ is more efficient than modifying reward $\overline{R}(s,a)$. Since state-action values $\overline{Q}^{\pi_\dagger}(s,a)$ need to be decreased by $\overline{\chi}^{\pi_\dagger}_{\overline{\beta}(s,a)}(s,a)$, the notion of *efficiency* in this context expresses how much $\overline{Q}^{\pi_\dagger}(s,a)$ changes per the unit cost of changing $\overline{P}(s,a,.)$ and $\overline{R}(s,a)$ respectively. From equation (4), we can see that the attack efficiency of changing reward $R(s,a)$ is $\frac{1}{C_r}$, whereas the attack efficiency of changing transition $\overline{P}(s,a,s_k)$ is $\gamma \cdot \frac{1}{2 \cdot C_p} \cdot (\overline{V}^{\pi_\dagger}(s_{k(s,a)}) - \overline{V}^{\pi_\dagger}(s_{|S|}))$. Combining this with the definition of the cost of the attack gives us that $k(s,a)$ should satisfy $\gamma \cdot C_r \cdot (\overline{V}^{\pi_\dagger}(s_{k(s,a)}) - \overline{V}^{\pi_\dagger}(s_{|S|})) > 2 \cdot C_p$. If this condition is not possible to satisfy, then $k(s,a)$ is equal to 0, which corresponds to the attack that changes only rewards.

To summarize, we pick the largest $k(s,a)$ such that $C_r \cdot (\overline{V}^{\pi_\dagger}(s_{k(s,a)}) - \overline{V}^{\pi_\dagger}(s_{|S|})) > 2 \cdot C_p$ and $\overline{F}_{k(s,a)}(s,a) \leq \overline{\chi}^{\pi_\dagger}_{\overline{\beta}(s,a)}(s,a)$. In case these conditions are infeasible, we chose $k(s,a) = 0$. We then maximally decreases the probability of transitioning to the $k(s,a)$ highest-valued states when taking action a in state s, and accordingly increases the probability of transitioning to state $s_{|S|}$. This attack on transitions modifies $\overline{P}(s,a,.)$ by $\overline{G}_{k(s,a)}(s,a)$, which in turn incurs the cost of $C_p \cdot \overline{G}_{k(s,a)}(s,a)$ and decreases $\overline{Q}^{\pi_\dagger}(s,a)$ by $\overline{F}_{k(s,a)}(s,a)$. To obtain the desired decrease in Q-values (i.e., $\overline{\chi}^{\pi_\dagger}_{\overline{\beta}(s,a)}(s,a)$), we decrease $\overline{R}(s,a)$ by $\overline{\chi}^{\pi_\dagger}_{\overline{\beta}(s,a)}(s,a) - \overline{F}_{k(s,a)}(s,a))$, incurring the cost of $C_r \cdot (\overline{\chi}^{\pi_\dagger}_{\overline{\beta}(s,a)}(s,a) - \overline{F}_{k(s,a)}(s,a))$. This gives us an upper bound on the cost of an optimal solution to the optimization problem (P1).

Lower bound on the cost. Similar to our upper bound, our lower bound on the cost mainly depends on three aspects of the problem: the efficiency of rewards poisoning, the efficiency of transitions poisoning, and the difference in state-action values between the target policy $\pi_\dagger$ and other policies. The attack efficiency of poisoning rewards depends only on C_r. However, the attack efficiency of poisoning transitions depends both on C_p and the discrepancy among the state values, which are bounded by $sp(\overline{V}^{\pi_\dagger})$. Intuitively, if the attack efficiency is low (resp. high), the lower bound on the cost needed to make an attack successful will be high (resp. low). Furthermore, the lower bound depends on the difference in state-action values between the target policy $\pi_\dagger$ and its neighbor policies, as captured by $\left\|\overline{\chi}^{\pi_\dagger}_0\right\|_\infty$. Notice that while the upper bound is based on a specific attack, the lower bound is attack-agnostic and implies that any successful attack must incur this cost. □

4.3 Offline Attacks: Efficiency of Solving the Problem

In the previous subsections, we formulated the problem of attacking an offline RL agent as optimization problem (P1). This problem is difficult to solve in general due to the first three constraints, which are non-linear and render the problem non-convex. However, note that in prior work on these attacks, the special case where only rewards are poisoned is shown to be

a convex optimization problem in both the average reward and discounted reward optimality criteria (Rakhsha et al., 2020; Ma et al., 2019).

Proposition 1. *(Rakhsha et al., 2020; Ma et al., 2019) The special case of offline attacks in which only rewards can be poisoned by the attacker, i.e. $\widehat{P} = \overline{P}$, is solvable through a convex optimization problem.*

As we discuss in Section 5.3, the problem (P1) also becomes convex if two additional constraints are added: $\widehat{R}(s, \pi_\dagger(s)) = \overline{R}(s, \pi_\dagger(s))$ and $\widehat{P}(s, \pi_\dagger(s), .) = \overline{P}(s, \pi_\dagger(s), .)$. Notice that these two constraints restrict the form of a solution in that the corresponding attack is not allowed to manipulate rewards and transition for state-action pairs $(s, \pi_\dagger(s))$. We refer to such attacks as *non-target only*, and we will discuss them further in Section 5.

5. Attacks in Online Setting

We now turn to attacks on an agent that learns over time using the environment feedback. Unlike the planning agent from the previous section, an online learning agent derives its policy from the interaction history, i.e., tuples of the form (s_t, a_t, r_t, s_{t+1}). To attack an online learning agent, an attacker changes the environment feedback, i.e., reward r_t and state s_{t+1}.

5.1 Online Attacks: Key Ideas and Attack Problem

The underlying idea behind our approach is to utilize the fact that the policies of the learning agents that we consider (see Section 2.2) will converge towards an optimal policy, and therefore will take a bounded number of suboptimal actions. Hence, to steer a learning agent towards selecting the target policy, it suffices to replace the environment feedback (i.e., reward r_t and the next state s_{t+1}) with a feedback sampled from an MDP that has the target policy as its ϵ-robust optimal policy. Notice that such an MDP can be obtained using optimization problem (P1). Now we separately consider the two cases: i) average reward criteria with $\gamma = 1$ and ii) discounted reward criteria with $\gamma < 1$.

For the case of average reward criteria with $\gamma = 1$, we consider a regret-minimization learner. With the following lemma we show that the above approach is sound: assuming that a learner draws its experience from an ergodic MDP M that has $\pi_\dagger$ as its ϵ-robust optimal policy, the expected number of steps in which the learner deviates from $\pi_\dagger$ is bounded by $O(\mathbb{E}\left[\textsc{Regret}(T, M)\right])$.

Lemma 2. *(Lemma 2 in (Rakhsha et al., 2020)) Consider an ergodic MDP M with $\gamma = 1$ that has $\pi_\dagger$ as its ϵ-robust optimal policy, and an online learning agent whose expected regret in MDP M is $\mathbb{E}\left[\textsc{Regret}(T, M)\right]$. The average mismatch of the agent w.r.t. the policy $\pi_\dagger$ is bounded by*

$$\mathbb{E}\left[\textsc{AvgMiss}(T)\right] \leq \frac{\mu_{\max}}{\epsilon \cdot T} \cdot \Big(\mathbb{E}\left[\textsc{Regret}(T, M)\right] + 2 \left\| V^{\pi_\dagger} \right\|_\infty \Big), \tag{9}$$

with $\mu_{\max} := \max_{s,a} \mu^{\pi_\dagger\{s;a\}}(s)$. Here, μ^π and V^π are respectively the stationary distribution and the value function of policy π in MDP M.

For the case of discounted reward criteria with $\gamma < 1$, we consider a learner with bounded number of suboptimal steps. The following lemma is an analog to Lemma 2 and is proven

in Appendix F. This lemma is based on the simple observation: when a learner draws its experience from an MDP M that has $\pi_\dagger$ as its ϵ-robust optimal policy, instantiating $\textsc{SubOpt}(T, M, \epsilon')$ with $\epsilon' = \epsilon$ will give us the number of times the learner deviates from $\pi_\dagger$.

Lemma 3. *Consider an ergodic MDP M with $\gamma < 1$ that has $\pi_\dagger$ as its ϵ-robust optimal policy, and an online learning agent whose expected number of suboptimal steps in an MDP M is $\textsc{SubOpt}(T, M, \epsilon')$. The average mismatch of the agent w.r.t. the policy $\pi_\dagger$ is given by $\textsc{AvgMiss}(T) = \frac{1}{T} \cdot \textsc{SubOpt}(T, M, \epsilon)$.*

To conclude, if a learner has sublinear $\mathbb{E}\left[\textsc{Regret}(T, M)\right]$ (resp. $\mathbb{E}\left[\textsc{SubOpt}(T, M, \epsilon)\right]$), Lemma 2 (resp. Lemma 3) implies that $o(1)$ average mismatch can be achieved in expectation using a sampling based attack that replaces the environment feedback (sampled from the original MDP $\overline{M}$) with a poisoned feedback sampled from MDP $\widehat{M}$, where MDP $\widehat{M}$ is a solution to optimization problem (P1).

However, the expected average cost of such an attack could be $\Omega(1)$ (non-diminishing over time), even for a learner with sublinear $\mathbb{E}\left[\textsc{Regret}(T, M)\right]$ or $\mathbb{E}\left[\textsc{SubOpt}(T, M, \epsilon)\right]$. Intuitively, if a learner follows the target policy and there exists s for which $\widehat{R}(s, \pi_\dagger(s)) \neq \overline{R}(s, \pi_\dagger(s))$ or $\widehat{P}(s, \pi_\dagger(s), .) \neq \overline{P}(s, \pi_\dagger(s), .)$, then the attacker would incur a non-zero cost whenever the learner visits s. To avoid this issue, we need to enforce constraints on the sampling MDP $\widehat{M}$ specifying that the attack does not alter rewards and transitions that correspond to the state-action pairs of the target policy, i.e., $(s, \pi_\dagger(s))$. As mentioned in Section 4.3, we refer to such attacks as *non-target only*. This brings us to the following template that we utilize for attacks on an online learner:

- Modify the optimization problem (P1) by adding constraints $\widehat{R}(s, \pi_\dagger(s)) = \overline{R}(s, \pi_\dagger(s))$ and $\widehat{P}(s, \pi_\dagger(s), s') = \overline{P}(s, \pi_\dagger(s), s')$. This gives us the following optimization problem:

$$
\begin{aligned}
\min_{M, R, P, \mu^{\pi_\dagger}, \mu^{\pi_\dagger\{s;a\}}} \quad & \textsc{Cost}(M, \overline{M}, C_r, C_p, p) && \text{(P2)}\\
\text{s.t.} \quad & \mu^{\pi_\dagger} \text{ and } P \text{ satisfy (2)},\\
& \forall s, a \neq \pi_\dagger(s) : \mu^{\pi_\dagger\{s;a\}} \text{ and } P \text{ satisfy (2)},\\
& \forall s, a \neq \pi_\dagger(s) : \sum_{s'} \mu^{\pi_\dagger}(s') \cdot R\big(s', \pi_\dagger(s')\big) \geq \sum_{s'} \mu^{\pi_\dagger\{s;a\}}(s') \cdot R\big(s', \pi_\dagger\{s;a\}(s')\big) + \epsilon,\\
& \forall s, a, s' : P(s, a, s') \geq \delta \cdot \overline{P}(s, a, s'),\\
& \forall s, s' : P(s, \pi_\dagger(s), s') = \overline{P}(s, \pi_\dagger(s), s'),\\
& \forall s : R(s, \pi_\dagger(s)) = \overline{R}(s, \pi_\dagger(s)),\\
& M = (S, A, R, P, \gamma).
\end{aligned}
$$

- Obtain the sampling MDP $\widehat{M}$ by solving (P2).
- Use the sampling MDP $\widehat{M}$ instead of the environment $\overline{M}$ during the learning process, i.e., obtain r_t from $\widehat{R}(s_t, a_t)$ and $s_{t+1} \sim \widehat{P}(s_t, a_t, .)$ (see Figure 1b).

As for the optimization problem (P1), particular instances of the optimization problem (P2), where only rewards or only transitions are poisoned, have been studied in the earlier version of the paper (Rakhsha et al., 2020). Based on the results in (Rakhsha et al., 2020), it

follows that the optimization problem (P2) is always feasible. Interestingly, and as we show in Section 5.3, this problem can be reformulated as a tractable convex optimization problem and hence can be efficiently solved.

5.2 Online Attacks: Theoretical Analysis

We now state the formal results that connect the performance of the learning agent—the regret and the number of suboptimal steps—to the average mismatch $\textsc{AvgMiss}(T)$ and the average attack cost $\textsc{AvgCost}(T)$. Notice that Lemma 2 and Lemma 3 directly relate the performance of the learning agent to the average mismatch w.r.t. the policy $\pi_\dagger$. Moreover, we show that the average attack cost can be bounded by the product of two factors: one which depends on the learner's performance, and the other that specifies the cost of changing the original MDP $\overline{M}$ to the sampling MDP $\widehat{M}$, expressed in ℓ_∞-*norm*.

More formally, in the average reward criteria with $\gamma = 1$, for a learning agent with a bound on the expected regret, we obtain:

Theorem 2 (Average reward criteria, $\gamma = 1$)**.** *Let $\widehat{M}$ be the optimal solution to* (P2)*. Consider the attack defined by r_t obtained from $\widehat{R}(s_t, a_t)$ and $s_{t+1} \sim \widehat{P}(s_t, a_t, .)$, and an online learning agent whose expected regret in an MDP M is $\mathbb{E}[\textsc{Regret}(T, M)]$. The average mismatch of the learner is in expectation upper bounded by*

$$\mathbb{E}[\textsc{AvgMiss}(T)] \leq \frac{\widehat{\mu}_{\max}}{\epsilon \cdot T} \cdot \left(\mathbb{E}\left[\textsc{Regret}(T, \widehat{M})\right] + 2 \left\| \widehat{V}^{\pi_\dagger} \right\|_\infty \right).$$

Furthermore, the average attack cost is in expectation upper bounded by

$$\mathbb{E}[\textsc{AvgCost}(T)] \leq \frac{\textsc{Cost}(\widehat{M}, \overline{M}, C_r, C_p, \infty)}{T} \cdot \left(\frac{\widehat{\mu}_{\max}}{\epsilon} \cdot \left(\mathbb{E}\left[\textsc{Regret}(T, \widehat{M})\right] + 2 \left\| \widehat{V}^{\pi_\dagger} \right\|_\infty \right) \right)^{1/p}.$$

Similarly, in the discounted reward criteria with $\gamma < 1$, for a learning agent with a bound on the number of suboptimal steps, we obtain:

Theorem 3 (Discounted reward criteria, $\gamma < 1$)**.** *Let $\widehat{M}$ be the optimal solution to* (P2)*. Consider the attack defined by r_t obtained from $\widehat{R}(s_t, a_t)$ and $s_{t+1} \sim \widehat{P}(s_t, a_t, .)$, and an online learning agent whose expected number of suboptimal steps in an MDP M is $\mathbb{E}[\textsc{SubOpt}(T, M, \epsilon')]$. The average mismatch of the learner is in expectation given by*

$$\mathbb{E}[\textsc{AvgMiss}(T)] = \frac{1}{T} \cdot \mathbb{E}\left[\textsc{SubOpt}(T, \widehat{M}, \epsilon)\right].$$

Furthermore, the average attack cost is in expectation upper bounded by

$$\mathbb{E}[\textsc{AvgCost}(T)] \leq \frac{\textsc{Cost}(\widehat{M}, \overline{M}, C_r, C_p, \infty)}{T} \cdot \left(\mathbb{E}\left[\textsc{SubOpt}(T, \widehat{M}, \epsilon)\right] \right)^{1/p}.$$

A direct consequence of these theorems is that for a learner with sublinear $\mathbb{E}[\textsc{Regret}(T, M)]$ (resp. $\mathbb{E}[\textsc{SubOpt}(T, M, \epsilon)]$), both the expected average mismatch and the expected average attack cost will decrease over time, and the rate of decrease depends on the learner's

performance.[9] Note that, while we considered ℓ_p norms with $p \geq 1$ to define the attack cost, the above results can be generalized to include the case of $p = 0$. For $p = 0$, the expected average cost is equivalent to the expected number of average mismatches, and the same upper bound applies.

5.3 Online Attacks: Efficiency of Solving the Problem

In Section 5.1, we outlined a template for attacking an online learner that uses optimization problem (P2) as its subroutine. In this subsection, we show that (P2) can be reformulated as a tractable convex program with linear constraints, which increases the computational efficiency of the proposed attack, and makes it more scalable.

Observe that the first three constraints in optimization problem P2 are quadratic constraints. Since the attack does not change the transitions associated to $\pi_\dagger$, i.e., $P(s, \pi_\dagger(s), s') = \overline{P}(s, \pi_\dagger(s), s')$, we have $\mu^{\pi_\dagger} = \overline{\mu}^{\pi_\dagger}$ and is no longer a variable in the optimization problem (i.e., it is precomputed).

To tackle the 2nd and 3rd quadratic constraints, we use two key ideas which will enable us to express these constraints with a linear constraint. The first key idea is to relate the scores of the target policy and its neighbor policies, i.e., $\rho^{\pi_\dagger}$ and $\rho^{\pi_\dagger\{s;a\}}$, to their the target policy's Q-values. By Corollary 1 in Appendix D, we know that

$$\rho^{\pi_\dagger} - \rho^{\pi_\dagger\{s;a\}} = \mu^{\pi_\dagger\{s;a\}}(s) \cdot \big(V^{\pi_\dagger}(s) - Q^{\pi_\dagger}(s,a)\big). \tag{10}$$

This identity enables us to rewrite the third constraint in optimization problem (P2) so that there are no quadratic terms of the form $\mu^{\pi_\dagger\{s;a\}}(s') \cdot R(s, \pi_\dagger\{s;a\}(s'))$. In particular, using Bellman equations (4), equation (10), and the fact that $V^{\pi_\dagger} = \overline{V}^{\pi_\dagger}$ and $\rho^{\pi_\dagger} = \overline{\rho}^{\pi_\dagger}$, we can rewrite the 3rd constraint in (P2) as

$$\forall s, a \neq \pi_\dagger(s): \quad \overline{V}^{\pi_\dagger}(s) - R(s,a) + \overline{\rho}^{\pi_\dagger} - \gamma \sum_{s'} P(s,a,s') \cdot \overline{V}^{\pi_\dagger}(s') \geq \frac{\epsilon}{\mu^{\pi_\dagger\{s;a\}}(s)}. \tag{11}$$

The only remaining nonlinear part in the modified constraint, i.e., equation (11), is $\frac{1}{\mu^{\pi_\dagger\{s;a\}}(s)}$. Now, we use the second key idea, which we also enable us to remove the 2nd constraint. We rewrite $\frac{1}{\mu^{\pi_\dagger\{s;a\}}(s)}$ in terms of reach times $\overline{T}^{\pi_\dagger}$, which can be precomputed. More precisely, in Lemma 8 from Appendix E, we show that

$$\frac{1}{\mu^{\pi\{s;a\}}(s)} = \frac{1 + \gamma \cdot \sum_{s'} P(s,a,s') \cdot \overline{T}^{\pi}(s',s)}{1 - (1-\gamma) \cdot \sum_{s'} d_0(s') \cdot \overline{T}^{\pi}(s',s)}.$$

Using this result, we can rewrite the equation (11) as

$$\forall s, a \neq \pi_\dagger(s): \overline{V}^{\pi_\dagger}(s) - R(s,a) + \overline{\rho}^{\pi_\dagger} - \gamma \cdot \sum_{s'} P(s,a,s') \cdot \left(\overline{V}^{\pi_\dagger}(s') + \frac{\epsilon}{\overline{\eta}(s)} \cdot \overline{T}^{\pi_\dagger}(s',s)\right) \geq \frac{\epsilon}{\overline{\eta}(s)},$$

[9] For some learning algorithms, guarantees on the learner's performance, i.e., regret or number of suboptimal steps, are true only with high probability: In that case, one would modify the results in Theorem 3 and Theorem 2 accordingly.

where $\overline{\eta}(s) = 1-(1-\gamma)\sum_{s'} d_0(s')\overline{T}^{\pi_\dagger}(s',s)$. Note that variable $\mu^{\pi_\dagger\{s;a\}}$ is no longer needed in the optimization problem, which makes the 2nd constraint redundant. The new formulation of optimization problem (P2) is therefore given by the following:

$$\min_{M,R,P} \quad \textsc{Cost}(M, \overline{M}, C_r, C_p, p) \tag{P2'}$$
$$\forall s, a \neq \pi_\dagger(s):$$
$$\overline{V}^{\pi_\dagger}(s) - R(s,a) + \overline{\rho}^{\pi_\dagger} - \gamma \cdot \sum_{s'} P(s,a,s') \cdot \left(\overline{V}^{\pi_\dagger}(s') + \frac{\epsilon}{\overline{\eta}(s)} \cdot \overline{T}^{\pi_\dagger}(s',s)\right) \geq \frac{\epsilon}{\overline{\eta}(s)},$$
$$\forall s, a, s' : P(s,a,s') \geq \delta \cdot \overline{P}(s,a,s'),$$
$$\forall s, s' : P(s, \pi_\dagger(s), s') = \overline{P}(s, \pi_\dagger(s), s'),$$
$$\forall s : R(s, \pi_\dagger(s)) = \overline{R}(s, \pi_\dagger(s)),$$
$$M = (S, A, R, P, \gamma).$$

Since $\overline{\rho}^{\pi_\dagger}$, $\overline{V}^{\pi_\dagger}$, and $\overline{T}^{\pi_\dagger}$ can be efficiently precomputed based on MDP $\overline{M}$, and new constraints are linear in the optimization variables, optimization problem (P2') is convex and can be efficiently solved.

Proposition 2. *Problem* (P2') *is a reformulation of problem* (P2) *and is a convex optimization problem with linear constraints.*

We conclude this section by noting that optimization problem (P2') can be solved separately for each state-action pair $(s, a \neq \pi_\dagger(s))$. Namely, the first constraint in (P2') only involves parameters of state-action pair (s,a), while the cost function is the ℓ_p-norm of a vector whose each component only depends on the rewards and transitions of one state-action pair (s,a). Hence, (P2') can be broken into $|S| \cdot (|A|-1)$ independent problems.

6. Numerical Simulations

In this section, we perform numerical simulations and empirically investigate the effectiveness of the proposed attacks on two different environments. For the reproducibility of experimental results and facilitating research in this area, the source code of our implementation is publicly available.[10]

6.1 Environments

The first environment we consider is a *chain* environment represented as an MDP with four states and two actions, see Figure 2. Even though simple, this environment provides a very rich and an intuitive problem setting to validate the theoretical statements and understand the effectiveness of the attacks by varying different parameters. We will also vary the number of states in the MDP to check the efficiency of solving different optimization problems, and report run times. The second environment we consider is a *navigation* environment represented as an MDP with nine states and two actions per state, see Figure 3. This environment is inspired by a navigation task and is slightly more complex than the environment in Figure 2. Below, we provide specific details of these two environments.

[10] `https://github.com/adishs/jmlr2021_rl-policy-teaching_code`.

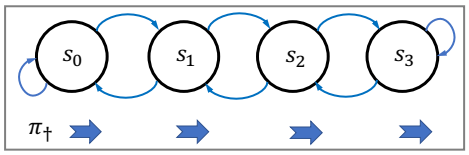


Figure 2: Chain environment with $|S| = 4$ states and $|A| = 2$ actions.

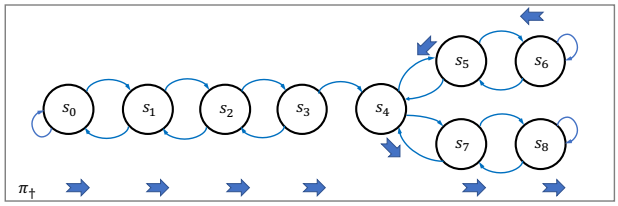


Figure 3: Navigation environment with $|S| = 9$ states and $|A| = 2$ actions.

Details of the chain environment. The environment has $|S| = 4$ states and $|A| = 2$ actions given by $\{\texttt{left}, \texttt{right}\}$. The original reward function $\overline{R}$ is action independent and has the following values: s_1 and s_2 are rewarding states with $\overline{R}(s_1, .) = \overline{R}(s_2, .) = 0.5$, state s_3 has negative reward of $\overline{R}(s_3, .) = -0.5$, and the reward of the state s_0 given by $\overline{R}(s_0, .)$ will be varied in experiments. With probability 0.9, the actions succeed in navigating the agent to left or right as shown on arrows; with probability 0.1 the agent's next state is sampled randomly from the set S. The target policy $\pi_\dagger$ is to take `right` action in all states as shown in Figure 2.

Details of the navigation environment. The environment has $|S| = 9$ states and $|A| = 2$ actions per state. The original reward function $\overline{R}$ is action independent and has the following values: $\overline{R}(s_1, .) = \overline{R}(s_2, .) = \overline{R}(s_3, .) = -2.5$, $\overline{R}(s_4, .) = \overline{R}(s_5, .) = 1.0$, $\overline{R}(s_6, .) = \overline{R}(s_7, .) = \overline{R}(s_8, .) = 0$, and the reward of the state s_0 given by $\overline{R}(s_0, .)$ will be varied in experiments. With probability 0.9, the actions succeed in navigating the agent as shown on arrows; with probability 0.1 the agent's next state is sampled randomly from the set S. The target policy $\pi_\dagger$ is to take actions as shown with bold arrows in Figure 3.

6.2 Attacks in the Offline Setting: Setup and Results

For the offline setting, we compare the performance of our attack strategy (JAttack) with three different baseline strategies (NT-JAttack, RAttack, DAttack) as discussed below:

1. JAttack: joint rewards and transitions attack using $(\widehat{R}, \widehat{P})$ obtained as a solution to the problem (P1).

2. NT-JAttack: joint rewards and transitions attack using $(\widehat{R}, \widehat{P})$ obtained as a solution to the problem (P2); here, NT- prefix is used to highlight that *non-target only* manipulations are allowed.

3. RAttack: rewards only attack obtained as a solution to the problem (P1) when $\widehat{P} := \overline{P}$ (alternatively, by taking the limit of C_p to infinity in the problem).

4. DAttack: transitions only attack obtained as a solution to the problem (P1) when $\widehat{R} := \overline{R}$ (alternatively, by taking the limit of C_r to infinity in the problem).

Attacks \ $\lvert S\rvert$	4	10	20	30	50	70	100
RAttack	0.01s	0.02s	0.04s	0.06s	0.14s	0.29s	0.61s
DAttack	3.09s	7.46s	14.98s	24.73s	46.02s	77.57s	126.97s
NT-JAttack	0.06s	0.11s	0.22s	0.34s	0.60s	0.83s	1.27s
JAttack	8.35s	20.52s	42.45s	64.98s	116.01s	180.36s	273.20s

Table 1: (**Chain environment**) Run times for solving different attack problems as we vary the number of states $|S|$. Numbers are reported in seconds and are based on an average of 5 runs for each setting.

γ \ Value	Lower bound	Empirical attack cost	Upper bound
1	0.006	0.823	13.730
0.99	0.006	0.831	9.851

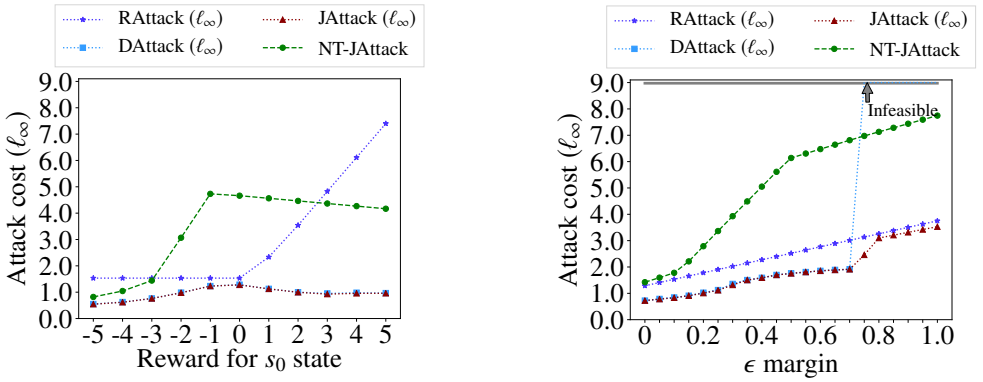
Table 2: (**Chain environment**) Comparison of upper and lower bounds from Theorem 1 to the empirical attack cost for JAttack. As considered in Figure 4, we fix $|S| = 4$, $\overline{R}(s_0, .) = -2.5$, and $\epsilon = 0.1$ margin for the $\pi_\dagger$ policy. Results are reported for $\gamma = 1$ and $\gamma = 0.99$. See further discussion in Section 4.2 after Theorem 1.

We set $p = \infty$ (i.e., ℓ_∞-*norm*) in the objective when solving different attack problems. We note that optimal solutions for the problems corresponding to RAttack and NT-JAttack can be computed efficiently using standard optimization techniques (also, refer to discussions in Section 4.3 and Section 5.3). Problems corresponding to JAttack and DAttack are computationally more challenging, and we provide a simple yet effective approach towards finding an approximate solution—specific implementation details are provided in Appendix C. For more detailed results and analysis of the rewards only and transitions only attacks (RAttack and DAttack), we refer the reader to the earlier version of the paper (Rakhsha et al., 2020).

Experimental setup and parameter choices. For all the experiments, we set $C_r = 3$, $C_p = 1$, and use ℓ_∞-norm in the measure of the attack cost (see Section 3.1).[11] The regularity parameter δ in the problems (P1) and (P2) is set to be 0.0001. In the experiments, we vary $\overline{R}(s_0, .) \in [-5, 5]$ and vary ϵ margin $\in [0, 1]$ for the $\pi_\dagger$ policy. The results are reported as an average of 10 runs. For the offline setting, we only report results for the discounted reward criteria with $\gamma = 0.99$; the results for the average reward criteria are very similar to the ones reported here. For the chain environment, we also vary the number of states $|S|$ and report run times for solving different attack problems.

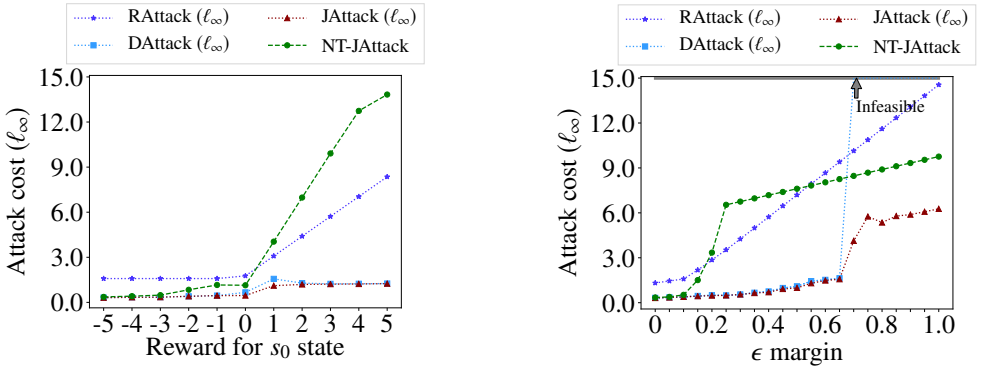
Results. Figure 4 reports results for the chain environment (with $|S| = 4$) and Figure 5 reports results for the navigation environment. The key takeaways are same for both the environments, and we want to highlight three points here. First, as we increase the desired

[11] We did experiments with other combinations of the parameters, and it does not qualitatively change the results. The source code of our implementation is publicly available, and one can also experiment with other possible values (see Footnote 10).



(a) Vary $\overline{R}(s_0, .)$ (b) Vary ϵ margin

Figure 4: (**Chain environment**) Results for poisoning attacks in the offline setting from Section 4. (**a**) shows results when we vary reward $\overline{R}(s_0, .)$ and fix $\epsilon = 0.1$ margin. (**b**) shows results when we vary ϵ margin and fix $\overline{R}(s_0, .) = -2.5$.



(a) Vary $\overline{R}(s_0, .)$ (b) Vary ϵ margin

Figure 5: (**Navigation environment**) Results for poisoning attacks in the offline setting from Section 4. (**a**) shows results when we vary reward $\overline{R}(s_0, .)$ and fix $\epsilon = 0.1$ margin. (**b**) shows results when we vary ϵ margin and fix $\overline{R}(s_0, .) = -2.5$.

ϵ margin, the attack problem becomes more difficult. While the attacks that allow reward poisoning (JAttack, NT-JAttack, RAttack) are always feasible though with increasing attack cost, it becomes infeasible to do transitions only poisoning attack (DAttack) (e.g., in Figure 4, DAttack attack is not possible for $\epsilon > 0.75$). Second, the plots show that joint attack (JAttack) can have much lower cost compared to reward only attack (RAttack) or transitions only attack (DAttack). Third, the plots also show that our joint attack strategy (JAttack) has much lower cost compared to the non-target only attack strategy (NT-JAttack). To check the efficiency of solving above mentioned attack problems, we vary the number of states in the chain environment and report the run times in Table 1. Finally, Table 2 provides a comparison of upper and lower bounds from Theorem 1 to the empirical attack cost for JAttack.

6.3 Attacks in the Online Setting: Setup and Results

For the online setting, we compare the performance of our attack strategy (NT-JAttack) with two different baseline strategies (JAttack, None) as discussed below:

1. NT-JAttack: joint rewards and transitions attack using $(\widehat{R}, \widehat{P})$ obtained as a solution to the problem (P2); here, NT- prefix is used to highlight that *non-target only* manipulations are allowed.

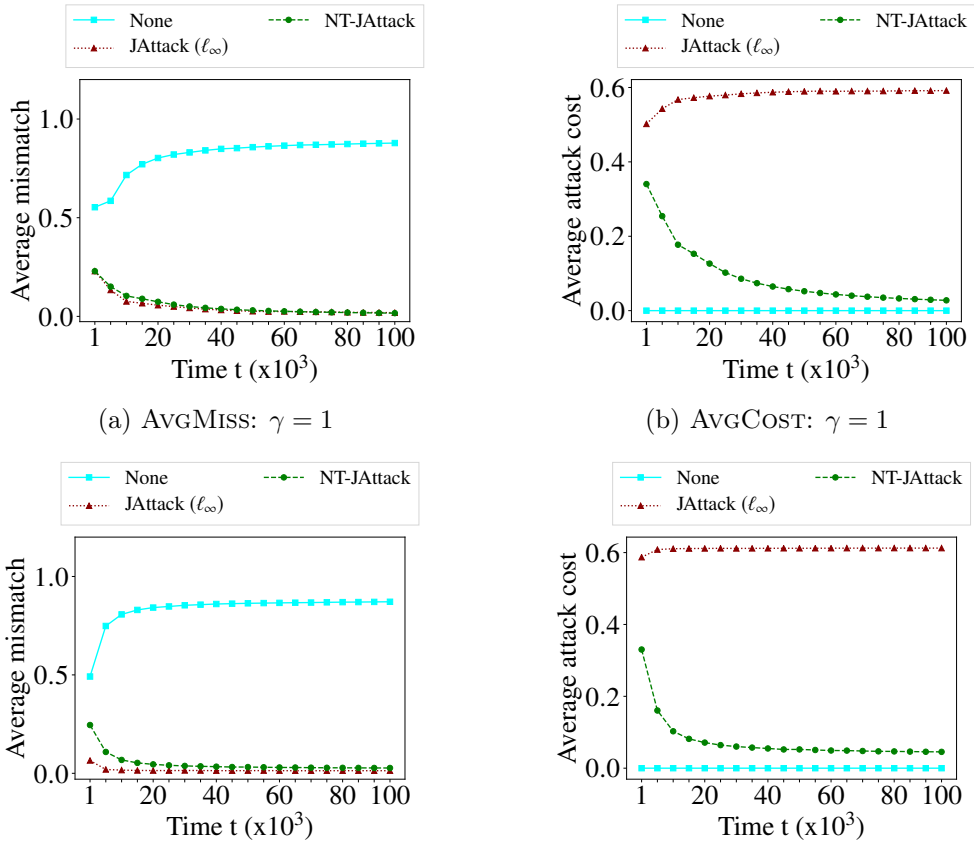
2. JAttack: joint rewards and transitions attack using $(\widehat{R}, \widehat{P})$ obtained as a solution to the problem (P1).

3. None: a default setting without adversary denoted as None where environment feedback is sampled from the original MDP $\overline{M}$.

The implementation details for NT-JAttack and JAttack are discussed above in Section 6.2 and Appendix C.

Experimental setup and parameter choices. For all the experiments, we set $C_r = 3$, $C_p = 1$. The regularity parameter δ in the problems (P1) and (P2) is set to be 0.0001. In the experiments, we fix $\overline{R}(s_0, .) = -2.5$ and $\epsilon = 0.1$ margin for the $\pi_\dagger$ policy. We plot the measure of the attacker's achieved goal in terms of AvgMiss and attacker's cost in terms of AvgCost for ℓ_1-*norm* measured over time t (see Section 3.2). The results are reported as an average of 20 runs. We separately run experiments for the average reward optimality criteria (with $\gamma = 1$) and the discounted reward optimality criteria (with $\gamma = 0.99$). For the average reward criteria, we consider an RL agent implementing the UCRL learning algorithm (Auer and Ortner, 2007). For the discounted reward criteria, we consider an RL agent implementing Q-learning with an exploration parameter set to 0.001 (Even-Dar and Mansour, 2003). For further details about the RL agents, we refer the reader to Section 2.2 and Appendix B. For both the settings, the attacker does not use any knowledge of the agent's learning algorithm.

Results. Figure 6 reports results for the chain environment (with $|S| = 4$) and Figure 7 reports results for the navigation environment. The key takeaways are same for both the environments, and we want to highlight two points here. First, the results in Figures 6 and 7 show that our proposed online attacks with NT-JAttack are highly effective for both the criteria: learner is forced to follow the target policy while the attacker's cost is low. In contrast, we can see that the online attacks with JAttack lead to high cost for the attacker, i.e., the cumulative cost is linear w.r.t. time as anticipated in Section 5.1. In particular, if a learner follows the target policy and there exists s for which reward or transition dynamics have been altered, then the attacker would incur a non-zero cost whenever the learner visits s. NT-JAttack avoids this issue as the attack does not alter rewards and transitions that correspond to the state-action pairs of the target policy (see further discussions following Lemma 2). Second, when comparing AvgMiss and AvgCost w.r.t. time t for NT-JAttack in these two settings, we see that the average values continue to decay for $\gamma = 1$, whereas they saturate for $\gamma = 0.99$. This is because of the convergence guarantees of the RL agent's learning algorithm: the Q-learning algorithm used for $\gamma = 0.99$ has a constant exploration rate whereas the UCRL algorithm used for $\gamma = 1$ has no-regret guarantees which in turn leads to $o(1)$ average mismatch and average attack cost (see Theorems 2 and 3, and Appendix B).



(a) AvgMiss: $\gamma = 1$

(b) AvgCost: $\gamma = 1$

(c) AvgMiss: $\gamma = 0.99$

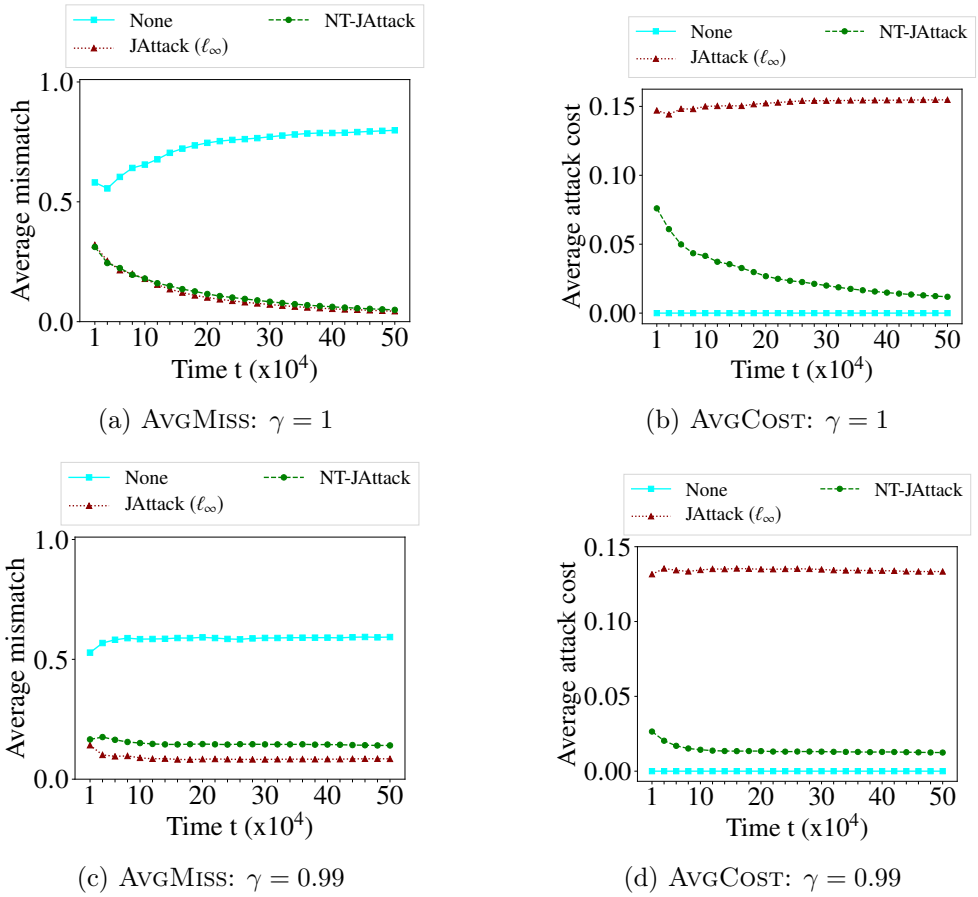
(d) AvgCost: $\gamma = 0.99$

Figure 6: (**Chain environment**) Results for poisoning attacks in the online setting from Section 5. (**a**, **b**) plots show results for the average reward criteria ($\gamma = 1$) with UCRL as the agent's learning algorithm. (**c**, **d**) plots show results for the discounted reward criteria ($\gamma = 0.99$) with Q-learning as the agent's learning algorithm.

7. Conclusion and Future Work

We studied a security threat to reinforcement learning (RL) where an attacker poisons the environment, thereby forcing the agent into executing a target policy. Our work provides theoretical underpinnings of environment poisoning against RL along several new attack dimensions, including (i) adversarial manipulation of the rewards and transition dynamics jointly, (ii) a general optimization framework for attack against RL agents maximizing rewards in undiscounted or discounted infinite horizon settings, and (iii) analyzing different attack costs for offline planning and online learning settings.

There are several promising directions for future work. These include expanding the attack models (e.g., manipulating actions or observations of the agent) and broadening the set of attack goals (e.g., under partial specification of target policy). In general, understanding the connection between different attack models can have practical implications. For instance,



(a) AvgMiss: $\gamma = 1$

(b) AvgCost: $\gamma = 1$

(c) AvgMiss: $\gamma = 0.99$

(d) AvgCost: $\gamma = 0.99$

Figure 7: (**Navigation environment**) Results for poisoning attacks in the online setting from Section 5. (**a**, **b**) plots show results for the average reward criteria ($\gamma = 1$) with UCRL as the agent's learning algorithm. (**c**, **d**) plots show results for the discounted reward criteria ($\gamma = 0.99$) with Q-learning as the agent's learning algorithm.

as discussed in Footnote 7, it would be interesting to study attacks where the agent's observations are poisoned (e.g., by adding noise in an image representing the state) while the physical state of the environment is unchanged. Moreover, relaxing the assumptions on the attacker's knowledge of the underlying MDP could lead to more robust attack strategies. An interesting future direction would be to make the studied attack models more scalable, e.g., applicable to continuous and large environments. One way to extend our optimization framework to large state spaces is via using the technique of state abstractions, e.g., see (Kamalaruban et al., 2020). Furthermore, extending this work to attack deep RL agents would be of interest, see (Sun et al., 2020) as a recent contemporary work in this direction. Another interesting topic would be to devise attack strategies against RL agents that use transfer learning approaches, especially in multi-agent RL systems, see (Da Silva and Costa, 2019).

While the experimental results demonstrate the effectiveness of the studied attack models, they do not reveal which types of learning algorithms are most vulnerable to the attack strategies studied in the paper. Further experimentation using a diverse set of state-of-the-art learning algorithms could reveal this, and provide some guidance in designing defensive strategies and novel RL algorithms robust to manipulations. Altogether, our results call for more robust agent designs that can reason about the presence of goal-oriented attackers, and by doing so, limit the influence of such attacks. In particular, the attacks studied in this paper do exhibit some structure due to the strategic nature of the attacker that wants to minimize its cost. This means that the agent can possibly make an inference about the true parameters of the environment, i.e., rewards and transitions, from the poisoned data. As we mentioned in Section 1.2, while the literature on robust RL has studied various robustness considerations, defenses against targeted poisoning attacks have been much less explored, e.g., see (Banihashem et al., 2021). Given the susceptibility of current RL algorithms to environment poisoning attacks, we believe that one of the most important future research directions is to design general forms of defenses against such attacks.

Acknowledgments

Xiaojin Zhu is supported in part by NSF grants 1545481, 1623605, 1704117, 1836978, ARO MURI W911NF2110317, and the MADLab AF Center of Excellence FA9550-18-1-0166.

Appendix A. List of Appendices

In this section we provide a brief description of the content provided in the appendices of the paper.

- Appendix B gives a few concrete examples of the learning agents considered in this paper. (Section 2)
- Appendix C contains implementation details for different attack strategies used in numerical simulations. (Section 6)
- Appendix D contains proof of Lemma 1 and some general results. (Section 4)
- Appendix E contains proof of Theorem 1 for offline attacks and related discussions. (Section 4)
- Appendix F contains proofs for online attacks including Lemma 3, Theorem 2, and Theorem 3. (Section 5)

Appendix B. Examples of Online Learning Agents

In this appendix, we provide examples of online learning agents for each of the two settings of interest: average reward optimality criteria with $\gamma = 1$, and discounted reward optimality criteria with $\gamma < 1$.

Average reward optimality criteria. For the case of average reward criteria with $\gamma = 1$, we consider a regret-minimization learner. Performance of a regret-minimization learner in MDP M is measured by its *regret* which after T steps is given by $\textsc{Regret}(T, M) = \rho^* \cdot T - \sum_{t=0}^{T-1} r_t$, where $\rho^* := \rho(\pi^*, M)$ is the optimal score. Well-studied algorithms with sublinear regret exist for average reward criteria, e.g., UCRL algorithm (Auer and Ortner, 2007; Jaksch et al., 2010) and algorithms based on posterior sampling method (Agrawal and Jia, 2017). More concretely, for the UCRL algorithm, with probability $1 - \delta$ we have

$$\textsc{Regret}(T, M) \leq 34 \cdot D|S|\sqrt{|A|T \log\left(\frac{T}{\delta}\right)},$$

where D is the diameter of MDP (Auer and Ortner, 2007; Jaksch et al., 2010).

Discounted reward optimality criteria. For the case of discounted reward criteria with $\gamma < 1$, the type of learners we consider are evaluated based on the number of suboptimal steps they take. An agent is suboptimal at time step t if it takes an an action not used by any near-optimal policy. This is formulated as $\textsc{SubOpt}(T, M, \epsilon') = \sum_{t=0}^{T-1} \mathbb{1}\left[a_t \notin \{\pi(s_t) \mid \rho^\pi \geq \rho^{\pi^*} - \epsilon'\}\right]$ where $\mathbb{1}[.]$ denotes the indicator function and ϵ' measures near-optimality of a policy w.r.t. score ρ. Our analysis of attacks is based on $\mathbb{E}\left[\textsc{SubOpt}(T, M, \epsilon')\right]$ of the learner for a specific value of ϵ'. Some bounds on this quantity are known for existing algorithms such as classic Q-learning (Even-Dar and Mansour, 2003) and Delayed Q-learning (Strehl et al., 2006).

As a concrete example, let us consider the classic Q-learning (Even-Dar and Mansour, 2003). We can obtain an upper bound on $\mathbb{E}\left[\textsc{SubOpt}(T, M, \epsilon')\right]$ of this algorithm based on the results from (Even-Dar and Mansour, 2003). Let $\mathcal{Q}_t(s, a)$ denote the $\mathcal{Q}$-values estimated

by the learner at time step t and let π_t be the greedy policy with respect to $\mathcal{Q}_t$. These values will converge to the $\mathcal{Q}$-values of the optimal policy π^* denoted by $\mathcal{Q}^*$.[12]

Consider any $\epsilon', \delta > 0$. Then, as shown in (Even-Dar and Mansour, 2003), there exists a number $N(\epsilon'/2, \delta)$ such that with probability of at least $1 - \delta$ we have $\|\mathcal{Q}_t - \mathcal{Q}^*\|_\infty \leq \epsilon'/2$ for $t \geq N(\epsilon'/2, \delta)$. This means that for each state s, we have:

$$\mathcal{Q}^*(s, \pi_t(s)) \geq \mathcal{Q}_t(s, \pi_t(s)) - \frac{\epsilon'}{2} \geq \mathcal{Q}_t(s, \pi^*(s)) - \frac{\epsilon'}{2} \geq \mathcal{Q}^*(s, \pi^*(s)) - \epsilon',$$

where the first and the third inequality holds due to $\|\mathcal{Q}_t - \mathcal{Q}^*\|_\infty \leq \epsilon'/2$ and the second inequality holds given that π_t is the greedy policy with respect to $\mathcal{Q}_t$. Furthermore, based on the results in (Schulman et al., 2015), the following holds for any two policies π and π':

$$\rho^{\pi} - \rho^{\pi'} = \sum_{s \in S} \mu^{\pi'}(s) \cdot \big(\mathcal{Q}^{\pi}(s, \pi(s)) - \mathcal{Q}^{\pi}(s, \pi'(s))\big).$$

Since $\mu^{\pi'}(s)$ is a distribution, we conclude that for $t \geq N(\epsilon'/2, \delta)$,

$$\rho^{\pi_t} = \rho^{\pi^*} + \sum_{s \in S} \mu^{\pi_t}(s) \cdot \big(\mathcal{Q}^*(s, \pi_t(s)) - \mathcal{Q}^*(s, \pi^*(s))\big) \geq \rho^{\pi^*} - \epsilon',$$

which means that π_t is near-optimal with a margin ϵ'.

Considering β be the exploration rate (i.e., the probability of taking random action instead of the action from greedy policy), with probability of at least $1 - \delta$ we get

$$\mathbb{E}\left[\textsc{SubOpt}(T, M, \epsilon')\right] \leq \begin{cases} T & \text{for } T < N(\epsilon'/2, \delta) \\ N(\epsilon'/2, \delta) + \beta \cdot \big(T - N(\epsilon'/2, \delta)\big) & \text{for } T \geq N(\epsilon'/2, \delta) \end{cases}.$$

Appendix C. Numerical Simulations: Implementation Details (Section 6)

Here, we provide implementation details for attack strategies. For the reproducibility of experimental results and facilitating research in this area, the source code of our implementation is publicly available, see Footnote 10. We note that optimal solutions for the problems corresponding to RAttack and NT-JAttack can be computed efficiently using standard optimization techniques (also, refer to discussions in Section 4.3 and Section 5.3). Problems corresponding to JAttack and DAttack are computationally more challenging, and we provide a simple yet effective approach towards finding approximate solutions—note that the obtained solutions do not have guarantees in terms of the attack cost w.r.t. the corresponding optimal solutions.

Implementation details for DAttack. To obtain a solution for DAttack, consider the transitions only attack variant of the problems (P1) and (P2) (i.e., $\widehat{R} := \overline{R}$). Then, we obtain a solution to the problem (P1) by utilizing problem (P2) as follows:

- As a first step, we use a simple heuristic to obtain a pool of transition kernels $\{\widetilde{P}\}$ from perturbations of $\overline{P}$ that increase the score ρ of the target policy $\pi_\dagger$. Here, these transition kernels $\widetilde{P}$ in the pool differ from $\overline{P}$ only for the actions taken by the target policy, i.e., for state action pairs $(s, \pi_\dagger(s))\ \forall s \in S$.

[12] Note that in the classic algorithm, $\mathcal{Q}$-values are not shifted as they are in our definition, and thus, we are using the symbol $\mathcal{Q}$ instead of Q.

- As the second step, we take each of $\widetilde{P}$ from this pool as an input to the problem (P2) instead of $\overline{P}$, which in turn gives us a corresponding pool of solutions $\{\widehat{P}\}$. Then, we pick a solution from this pool of solutions with the minimal cost.

For further details about the transitions only attack strategy (DAttack), we refer the reader to the earlier version of the paper (Rakhsha et al., 2020).

Implementation details for JAttack. To obtain a solution for JAttack, we use the similar idea of solving the problem (P1) by utilizing problem (P2) as follows:

- As a first step, we obtain a solution for RAttack and DAttack using the above mentioned techniques. Let us denote these solutions to rewards only and transitions only poisoning attacks as $\widehat{R}_{\text{only}}$ and $\widehat{P}_{\text{only}}$ respectively. Note that the DAttack attack strategy might be infeasible, and in this case we set $\widehat{P}_{\text{only}} := \overline{P}$.
- As a second step, we use a simple heuristic to obtain a pool of rewards denoted as $\{\widetilde{R}\}$ and a pool of transition kernels denoted as $\{\widetilde{P}\}$. The pool $\{\widetilde{R}\}$ is generated by considering convex combinations of $\widehat{R}_{\text{only}}$ and $\overline{R}$, i.e., $\widetilde{R} = (1-\alpha)\cdot\widehat{R}_{\text{only}} + \alpha\cdot\overline{R}$ for $\alpha \in [0,1]$ with a desired level of discretization. The pool $\{\widetilde{P}\}$ is generated similarly by considering convex combinations of $\widehat{P}_{\text{only}}$ and $\overline{P}$.
- As the final step, we take all possible pairs $\langle\widetilde{R},\widetilde{P}\rangle$ of rewards and transition kernels from these pools as an input to the problem (P2) instead of inputting $\langle\overline{R},\overline{P}\rangle$, which in turn gives us a corresponding pool of solutions $\{\langle\widehat{R},\widehat{P}\rangle\}$. Then, we pick a solution from this pool of solutions with the minimal cost.

We note that the key common idea for obtaining solutions to DAttack and JAttack is to solve the problem (P1) by solving the problem (P2) with different inputs. One can replace the specific heuristics to generate the pool $\{\widetilde{P}\}$ for DAttack and the pool $\{\langle\widetilde{R},\widetilde{P}\rangle\}$ for JAttack with alternative methods. Furthermore, the run time of solving the problem (P1) would depend on the number of iterations we invoke the problem (P2) internally, which in turn provides a simple way to trade-off the run time and attack cost.

Appendix D. Proofs for Offline Attacks: Lemma 1 (Section 4)

We prove Lemma 1 through several intermediate results. The first one is the following results of (Even-Dar et al., 2005) and (Schulman et al., 2015).

Lemma 4. *(Lemma 7 in (Even-Dar et al., 2005), Equation (2) in (Schulman et al., 2015)) For two policies π and π' we have:*

$$\rho^{\pi} - \rho^{\pi'} = \sum_{s\in S} \mu^{\pi'}(s)\big(Q^{\pi}(s,\pi(s)) - Q^{\pi}(s,\pi'(s))\big).$$

We will use the following corollary which is a direct consequence of Lemma 4.

Corollary 1. *For any policy π and its neighbor policy $\pi\{s;a\}$ we have:*

$$\rho^{\pi} - \rho^{\pi\{s;a\}} = \mu^{\pi\{s;a\}}(s)\big(Q^{\pi}(s,\pi(s)) - Q^{\pi}(s,a)\big).$$

Next, we provide a sufficient condition for a policy π to be uniquely optimal.

Lemma 5. *If we have $\rho^{\pi} \geq \rho^{\pi\{s;a\}} + \epsilon$ for every state s and action $a \neq \pi(s)$, and $\epsilon > 0$, then π is the only optimal policy.*

Proof. Let arbitrary $s \in S$ and $a \in \mathcal{A}$ be such that $a \neq \pi(s)$. Based on corollary 1, we have

$$Q^{\pi}(s, \pi(s)) - Q^{\pi}(s, a) = \frac{\rho^{\pi} - \rho^{\pi\{s;a\}}}{\mu^{\pi\{s;a\}}(s)} > 0. \tag{12}$$

Note that in this paper we are focusing on ergodic MDPs, thus, we know $\mu^{\pi\{s;a\}}(s) > 0$. Now let $\pi' \neq \pi$ be a policy with $\pi'(s') = a' \neq \pi(s')$. Using (12) we have

$$\rho^{\pi} - \rho^{\pi'} = \sum_{s \in S} \mu^{\pi'}(s)\big(Q^{\pi}(s, \pi(s)) - Q^{\pi}(s, \pi'(s))\big) \geq \mu^{\pi'}(s')\big(Q^{\pi}(s', a') - Q^{\pi}(s', \pi(s'))\big) > 0$$

We again used the fact that MDP is ergodic to say $\mu^{\pi'}(s') > 0$. □

Proof of Lemma 1 We should show that policy π is ϵ-robust optimal *iff* we have $\rho^{\pi} \geq \rho^{\pi\{s;a\}} + \epsilon$ for every state s and action $a \neq \pi(s)$.

Proof. The necessity of the condition follows directly from the definition of ϵ-robust policies. Let us focus on its sufficiency.

Consider deterministic policies π, and denote the Hamming distance between two policies π_1 and π_2 by $D_H(\pi_1, \pi_2)$, i.e., $D_H(\pi_1, \pi_2) = \sum_{s \in S} \mathbb{1}\left[\pi_1(s) \neq \pi_2(s)\right]$ where $\mathbb{1}\left[.\right]$ denotes the indicator function. Assume that the condition of the lemma holds for policy π^*, i.e., that $\rho^{\pi^*} \geq \rho^{\pi_1} + \epsilon$ for all π_1 s.t. $D_H(\pi_1, \pi^*) = 1$.

Lemma 5 implies that π^* is uniquely optimal. Now, consider policy π_k s.t. $D_H(\pi_k, \pi^*) = k > 1$. Since π^* is (uniquely) optimal and the MDP is ergodic ($\mu^{\pi^*}(s) > 0$), we have that

$$\rho^{\pi^*} - \rho^{\pi_k} = \sum_{s \in S} \mu^{\pi^*}(s) \cdot [Q^{\pi_k}(s, \pi^*(s)) - Q^{\pi_k}(s, \pi_k(s))] > 0,$$

which implies that there exists $s_k \in S$ s.t. $[Q^{\pi_k}(s_k, \pi^*(s_k)) - Q^{\pi_k}(s_k, \pi_k(s_k))] > 0$. Define policy π_{k-1} as

$$\pi_{k-1}(s) = \begin{cases} \pi^*(s) & \text{if } s = s_k \\ \pi_k(s) & \text{otherwise} \end{cases}.$$

We have that

$$\begin{aligned} \rho^{\pi_{k-1}} &= \rho^{\pi_k} + \sum_{s \in S} \mu^{\pi_{k-1}}(s) \cdot [Q^{\pi_k}(s, \pi_{k-1}(s)) - Q^{\pi_k}(s, \pi_k(s))] \\ &= \rho^{\pi_k} + \mu^{\pi_{k-1}}(s_k) \cdot [Q^{\pi_k}(s_k, \pi^*(s_k)) - Q^{\pi_k}(s_k, \pi_k(s_k))] \geq \rho^{\pi_k}. \end{aligned}$$

Therefore, by induction, we know that there exists a policy π_1 such that $\rho^{\pi_k} \leq \rho^{\pi_1}$ and $D_H(\pi_1, \pi^*) = 1$. Utilizing our initial assumption, we obtain that $\rho^{\pi^*} \geq \rho^{\pi} + \epsilon$ for all $\pi \neq \pi^*$, which proves that π^* is ϵ-robust optimal if the condition of the lemma holds. □

Appendix E. Proofs for Offline Attacks: Proof of Theorem 1

We break the proof of Theorem 1 into two parts: In Appendix E.1, we prove the lower bound in the theorem, and in Appendix E.2 we prove the upper bound. In Appendix E.3, we discuss the choice of δ in our optimization problems.

E.1 Proofs for the Lower Bound

To prove the lower bound in Theorem 1, we will use a proof technique that is similar to the one presented in (Ma et al., 2019), but adapted to our setting.

First, let us define operator F as

$$F(Q, R, \rho, P, \pi, \gamma)(s, a) = R(s, a) - \rho + \gamma \sum_{s' \in S} P(s, a, s') V^{\pi}(s'), \tag{13}$$

or in vector notation

$$F(Q, R, \rho, P, \pi, \gamma) = R - \rho \cdot \mathbf{1} + \gamma P \cdot V^{\pi},$$

where $V^{\pi}(s') = Q(s', \pi(s'))$ (and π is a deterministic policy). Furthermore, we defined the span of X as $sp(X) = \max_i X(i) - \min_i X(i)$ - as argued in (Puterman, 1994), sp is a seminorm.

Lemma 6. *The following holds:*

$$sp(F(Q_1, R, \rho, P, \pi, \gamma) - F(Q_2, R, \rho, P, \pi, \gamma)) \leq \gamma(1 - \alpha) \cdot sp(Q_1 - Q_2),$$

where

$$\alpha = \min_{s,a,s',a'} \sum_{x \in S} \min\{P(s, a, x), P(s', a', x)\}.$$

Proof. We have that

$$sp(F(Q_1, R, \rho, P, \pi, \gamma) - F(Q_2, R, \rho, P, \pi, \gamma)) = \gamma \cdot sp(P \cdot (V_1^{\pi} - V_2^{\pi})).$$

Following the proof of Proposition 6.6.1 in (Puterman, 1994), we obtain that for $b(x, s, a, s', a') = \min\{P(s, a, x), P(s', a', x)\}$

$$\begin{aligned}
&\sum_{x \in S} P(s, a, x) \cdot (V_1^{\pi}(x) - V_2^{\pi}(x)) - \sum_{x' \in S} P(s', a', x) \cdot (V_1^{\pi}(x) - V_2^{\pi}(x)) \\
&= \sum_{x \in S} (P(s, a, x) - b(x, s, a, s', a')) \cdot (V_1^{\pi}(x) - V_2^{\pi}(x)) \\
&- \sum_{x \in S} (P(s', a', x) - b(x, s, a, s', a'))) \cdot (V_1^{\pi}(x) - V_2^{\pi}(x)) \\
&\leq \sum_{x \in S} (P(s, a, x) - b(x, s, a, s', a')) \cdot \max_{x'} (V_1^{\pi}(x') - V_2^{\pi}(x')) \\
&- \sum_{x \in S} (P(s', a', x) - b(x, s, a, s', a'))) \cdot \min_{x'} (V_1^{\pi}(x') - V_2^{\pi}(x'))
\end{aligned}$$

$$= (1 - \sum_{x \in S} b(x, s, a, s', a')) \cdot sp(V_1^\pi - V_2^\pi) \leq (1 - \alpha) \cdot sp(V_1^\pi - V_2^\pi)$$

Therefore

$$sp(F(Q_1, R, \rho, P, \pi, \gamma) - F(Q_2, R, \rho, P, \pi, \gamma)) = \gamma \cdot sp(P \cdot (V_1^\pi - V_2^\pi)) \leq \gamma(1 - \alpha) \cdot sp(V_1^\pi - V_2^\pi).$$

Now, notice that for $s_{\max} = \arg\max_s [V_1^\pi(s) - V_2^\pi(s)]$ and $s_{\min} = \arg\min_s [V_1^\pi(s) - V_2^\pi(s)]$ we have that

$$V_1^\pi(s_{\max}) - V_2^\pi(s_{\max}) = Q_1(s_{\max}, \pi(s_{\max})) - Q_2(s_{\max}, \pi(s_{\max})) \leq \max_{s,a} [Q_1(s, a) - Q_2(s, a)]$$
$$V_1^\pi(s_{\min}) - V_2^\pi(s_{\min}) = Q_1(s_{\min}, \pi(s_{\min})) - Q_2(s_{\min}, \pi(s_{\min})) \geq \min_{s,a} [Q_1(s, a) - Q_2(s, a)]$$

Therefore $sp(V_1^\pi - V_2^\pi) \leq sp(Q_1 - Q_2)$, which implies that

$$sp(F(Q_1, R, \rho, P, \pi, \gamma) - F(Q_2, R, \rho, P, \pi, \gamma)) \leq \gamma(1 - \alpha) \cdot sp(Q_1 - Q_2)$$

□

To obtain the statement of the theorem, we will need to relate $sp(Q_1 - Q_2)$ to difference between R_1, P_1 and R_2, P_2. The following lemma provides this relation.

Lemma 7. *Let Q_1^π and V_1^π denote Q and V values of policy π in MDP $M_1 = (S, \mathcal{A}, R_1, P_1, \gamma)$ and Q_2^π denote Q values of policy π in MDP $M_2 = (S, \mathcal{A}, R_2, P_2, \gamma)$. The following holds:*

$$\|R_1 - R_2\|_\infty + \gamma \cdot \|P_1 - P_2\|_\infty \cdot \|V_1^\pi\|_\infty \geq \frac{1 - \gamma + \gamma \cdot \alpha_2}{2} \cdot sp(Q_1^\pi - Q_2^\pi).$$

where

$$\alpha_2 = \min_{s,a,s',a'} \sum_{x \in S} \min\{P_2(s, a, x), P_2(s', a', x)\}.$$

Proof. Notice that Q_1^π and Q_2^π satisfy

$$Q_1^\pi(s, a) = F(Q_1^\pi, R_1, \rho_1^\pi, P_1, \pi, \gamma)$$
$$Q_2^\pi(s, a) = F(Q_2^\pi, R_2, \rho_2^\pi, P_2, \pi, \gamma),$$

where ρ_1^π and ρ_2^π respectively denote the average rewards of policy π in M_1 and M_2. We obtain

$$\begin{aligned}
sp(Q_1^\pi - Q_2^\pi) &= sp(F(Q_1^\pi, R_1, \rho_1^\pi, P_1, \pi, \gamma) - F(Q_2^\pi, R_2, \rho_2^\pi, P_2, \pi, \gamma)) \\
&= sp(F(Q_1^\pi, R_1, \rho_1^\pi, P_1, \pi, \gamma) - F(Q_1^\pi, R_2, \rho_2^\pi, P_1, \pi, \gamma) \\
&+ F(Q_1^\pi, R_2, \rho_2^\pi, P_1, \pi, \gamma) - F(Q_1^\pi, R_2, \rho_2^\pi, P_2, \pi, \gamma) \\
&+ F(Q_1^\pi, R_2, \rho_2^\pi, P_2, \pi, \gamma) - F(Q_2^\pi, R_2, \rho_2^\pi, P_2, \pi, \gamma)) \\
&\leq sp(F(Q_1^\pi, R_1, \rho_1^\pi, P_1, \pi, \gamma) - F(Q_1^\pi, R_2, \rho_2^\pi, P_1, \pi, \gamma)) \\
&+ sp(F(Q_1^\pi, R_2, \rho_2^\pi, P_1, \pi, \gamma) - F(Q_1^\pi, R_2, \rho_2^\pi, P_2, \pi, \gamma)) \\
&+ sp(F(Q_1^\pi, R_2, \rho_2^\pi, P_2, \pi, \gamma) - F(Q_2^\pi, R_2, \rho_2^\pi, P_2, \pi, \gamma))
\end{aligned}$$

$$\leq sp(R_1 - \rho_1^\pi \cdot \mathbf{1} - R_2 + \rho_2^\pi \cdot \mathbf{1}) + \gamma \cdot sp((P_1 - P_2) \cdot V_1^\pi) + \gamma(1-\alpha_2) \cdot sp(Q_1^\pi - Q_2^\pi)$$

where the last inequality is due to Lemma 6 (i.e., $sp(F(Q_1^\pi, R_2, \rho_2^\pi, P_2, \pi, \gamma) - F(Q_2^\pi, R_2, \rho_2^\pi, P_2, \pi, \gamma)) \leq \gamma(1-\alpha_2) \cdot sp(Q_1^\pi - Q_2^\pi)$). Due to the properties of sp, we have

$$sp(R_1 - \rho_1^\pi \cdot \mathbf{1} - R_2 + \rho_2^\pi \cdot \mathbf{1}) = sp(R_1 - R_2) \leq 2 \cdot \|R_1 - R_2\|_\infty .$$

For the second term, we have

$$\begin{aligned} sp((P_1 - P_2) \cdot V_1^\pi) &\leq 2 \cdot \|(P_1 - P_2) \cdot V_1^\pi\|_\infty \\ &= 2 \cdot \max_{s,a} |\sum_{s'} (P_1(s,a,s') - P_2(s,a,s')) \cdot V_1^\pi(s')|. \end{aligned}$$

To bound the right-hand side in the above equation, we note that

$$\begin{aligned} &\sum_{s'} (P_1(s,a,s') - P_2(s,a,s')) \cdot V_1^\pi(s') \\ &\leq \Bigg(\sum_{s': P_1(s,a,s') \geq P_2(s,a,s')} (P_1(s,a,s') - P_2(s,a,s')) \Bigg) \cdot \max_{s'} V_1^\pi(s') \\ &+ \Bigg(\sum_{s': P_1(s,a,s') < P_2(s,a,s')} (P_1(s,a,s') - P_2(s,a,s')) \Bigg) \cdot \min_{s'} V_1^\pi(s') \\ &= \frac{1}{2} \Bigg(\sum_{s'} |P_1(s,a,s') - P_2(s,a,s')| \Bigg) \cdot sp(V_1^\pi), \end{aligned}$$

and similarly

$$\begin{aligned} &\sum_{s'} (P_1(s,a,s') - P_2(s,a,s')) \cdot V_1^\pi(s') \\ &\geq \Bigg(\sum_{s': P_1(s,a,s') \geq P_2(s,a,s')} (P_1(s,a,s') - P_2(s,a,s')) \Bigg) \cdot \min_{s'} V_1^\pi(s') \\ &+ \Bigg(\sum_{s': P_1(s,a,s') < P_2(s,a,s')} (P_1(s,a,s') - P_2(s,a,s')) \Bigg) \cdot \max_{s'} V_1^\pi(s') \\ &= -\frac{1}{2} \Bigg(\sum_{s'} |P_1(s,a,s') - P_2(s,a,s')| \Bigg) \cdot sp(V_1^\pi). \end{aligned}$$

The above two bounds give us the following:

$$|\sum_{s'} (P_1(s,a,s') - P_2(s,a,s')) \cdot V_1^\pi(s')| \leq \frac{1}{2} \Bigg(\sum_{s'} |P_1(s,a,s') - P_2(s,a,s')| \Bigg) \cdot sp(V_1^\pi).$$

Now we can bound the second term as

$$\begin{aligned} sp((P_1 - P_2) \cdot V_1^\pi) &\leq 2 \cdot \|(P_1 - P_2) \cdot V_1^\pi\|_\infty \\ &= 2 \cdot \max_{s,a} |\sum_{s'} (P_1(s,a,s') - P_2(s,a,s')) \cdot V_1^\pi(s')| \end{aligned}$$

$$= sp(V_1^\pi) \cdot \max_{s,a} \sum_{s'} |P_1(s,a,s') - P_2(s,a,s')|$$
$$= sp(V_1^\pi) \cdot \|P_1 - P_2\|_\infty ,$$

where $\|P_1 - P_2\|_\infty = \max_{s,a} \sum_{s'} |P_1(s,a,s') - P_2(s,a,s')|$. Putting this together with the upper bound on $sp(Q_1^\pi - Q_2^\pi)$, we obtain

$$2 \cdot \|R_1 - R_2\|_\infty + \gamma \cdot \|P_1 - P_2\|_\infty \cdot sp(V_1^\pi) \geq (1 - \gamma + \gamma\alpha_2) \cdot sp(Q_1^\pi - Q_2^\pi),$$

which proves the claim. □

We are now ready to prove the lower bound in Theorem 1. We can write

$$\begin{aligned}
\textsc{Cost}(\widehat{M}, \overline{M}) =& \Big[\sum_{s,a} \Big(C_r \cdot \big|\widehat{R}(s,a) - \overline{R}(s,a)\big| + C_p \cdot \sum_{s'} \big|\widehat{P}(s,a,s') - \overline{P}(s,a,s')\big| \Big)^p \Big]^{1/p} \\
\geq& \max_{s,a} \Big(C_r \cdot \big|\widehat{R}(s,a) - \overline{R}(s,a)\big| + C_p \cdot \sum_{s'} \big|\widehat{P}(s,a,s') - \overline{P}(s,a,s')\big| \Big) \\
\geq& \max \Big(C_r \left\|\widehat{R} - \overline{R}\right\|_\infty , C_p \left\|\widehat{P} - \overline{P}\right\|_\infty \Big) \\
\geq& \frac{2{C_r}^{-1}}{2{C_r}^{-1} + \gamma {C_p}^{-1} sp(\overline{V}^{\pi\dagger})} C_r \left\|\widehat{R} - \overline{R}\right\|_\infty + \frac{\gamma {C_p}^{-1} sp(\overline{V}^{\pi\dagger})}{2{C_r}^{-1} + \gamma {C_p}^{-1} sp(\overline{V}^{\pi\dagger})} C_p \left\|\widehat{P} - \overline{P}\right\|_\infty \\
=& \frac{1}{2{C_r}^{-1} + \gamma {C_p}^{-1} sp(\overline{V}^{\pi\dagger})} \Big(2 \left\|\widehat{R} - \overline{R}\right\|_\infty + \gamma sp(\overline{V}^{\pi\dagger}) \left\|\widehat{P} - \overline{P}\right\|_\infty \Big) \\
\geq& \frac{1 - \gamma + \gamma\widehat{\alpha}}{2{C_r}^{-1} + \gamma {C_p}^{-1} sp(\overline{V}^{\pi\dagger})} sp(\overline{Q}^{\pi\dagger} - \widehat{Q}^{\pi\dagger}),
\end{aligned}$$

where $\|P_1 - P_2\|_\infty = \max_{s,a} \sum_{s'} |P_1(s,a,s') - P_2(s,a,s')|$, and we used Lemma 7 in the last inequality. Factor $\widehat{\alpha}$ can be bounded as follows:

$$\begin{aligned}
\widehat{\alpha} &= \min_{s,a,s',a'} \sum_x \min\{\widehat{P}(s,a,x), \widehat{P}(s',a',x)\} \\
&\geq \min_{s,a,s',a'} \sum_x \min\{\delta \cdot \overline{P}(s,a,x), \delta \cdot \overline{P}(s',a',x)\} \\
&= \delta \cdot \min_{s,a,s',a'} \sum_x \min\{\overline{P}(s,a,x), \cdot \overline{P}(s',a',x)\} \\
&= \delta \cdot \overline{\alpha}.
\end{aligned}$$

It only remains to bound $sp(\overline{Q}^{\pi\dagger} - \widehat{Q}^{\pi\dagger})$. We show that

$$sp(\overline{Q}^{\pi\dagger} - \widehat{Q}^{\pi\dagger}) \geq \left\|\overline{\chi}_0^{\pi\dagger}\right\|_\infty .$$

Let s' and a' be a state action pair that satisfy: $s', a' = \arg\max_{s,a} \overline{\chi}_0^{\pi\dagger}(s,a)$. Let's consider the case when $\overline{\chi}_0^{\pi\dagger}(s',a') > 0$. We have

$$sp(\overline{Q}^{\pi\dagger} - \widehat{Q}^{\pi\dagger}) = sp(\widehat{Q}^{\pi\dagger} - \overline{Q}^{\pi\dagger}) = \max_{s,a}[\widehat{Q}^{\pi\dagger} - \overline{Q}^{\pi\dagger}] - \min_{s,a}[\widehat{Q}^{\pi\dagger} - \overline{Q}^{\pi\dagger}]$$

$$
\begin{aligned}
&\geq \widehat{Q}^{\pi_\dagger}(s', \pi_\dagger(s')) - \overline{Q}^{\pi_\dagger}(s', \pi_\dagger(s')) - (\widehat{Q}^{\pi_\dagger}(s', a') - \overline{Q}^{\pi_\dagger}(s', a')) \\
&= (\widehat{Q}^{\pi_\dagger}(s', \pi_\dagger(s')) - \widehat{Q}^{\pi_\dagger}(s', a')) + (\overline{Q}^{\pi_\dagger}(s', a') - \overline{Q}^{\pi_\dagger}(s', \pi_\dagger(s')) \\
&\geq \frac{\epsilon}{\widehat{\mu}^{\pi_\dagger\{s';a'\}}(s')} + (\overline{Q}^{\pi_\dagger}(s', a') - \overline{Q}^{\pi_\dagger}(s', \pi_\dagger(s')) \\
&\geq \epsilon + \frac{\overline{\rho}^{\pi_\dagger\{s';a'\}} - \overline{\rho}^{\pi_\dagger}}{\overline{\mu}^{\pi_\dagger\{s';a'\}}(s')} > \overline{\chi}_0^{\pi_\dagger}(s', a'),
\end{aligned}
$$

where we used the fact that $\widehat{Q}^{\pi_\dagger}(s', \pi_\dagger(s')) \geq \widehat{Q}^{\pi_\dagger}(s', a') + \frac{\epsilon}{\widehat{\mu}^{\pi_\dagger\{s';a'\}}(s')}$ (because $\pi_\dagger$ is ϵ-robust optimal in the modified MDP) and Lemma 4 (Corollary 1) to relate Q values to scores ρ. When $\overline{\chi}_0^{\pi_\dagger}(s', a') = 0$, we know that $sp(\overline{Q}^{\pi_\dagger} - \widehat{Q}^{\pi_\dagger}) \geq 0$ due to the properties of sp. Therefore, $sp(\overline{Q}^{\pi_\dagger} - \widehat{Q}^{\pi_\dagger}) = sp(\widehat{Q}^{\pi_\dagger} - \overline{Q}^{\pi_\dagger}) \geq \left\|\overline{\chi}_0^{\pi_\dagger}(s', a')\right\|_\infty$. Putting this together with the previous expression, we obtain the claim:

$$
\textsc{Cost}(\widehat{M}, \overline{M}) \geq \frac{1 - \gamma + \gamma\delta\overline{\alpha}}{2{C_r}^{-1} + \gamma {C_p}^{-1} sp(\overline{V}^{\pi_\dagger})} \left\|\overline{\chi}_0^{\pi_\dagger}\right\|_\infty.
$$

E.2 Proofs for the Upper Bound

Here, we prove the upper bound in Theorem 1. We first prove a lemma that we need for our proof.

Lemma 8. *Let π be a deterministic policy and P be a transition kernel such that $P(s, \pi(s), s') = \overline{P}(s, \pi(s), s')$ for every s, s'. For any s, a, If $\mu^{\pi\{s;a\}}$ is the state distribution of $\pi\{s; a\}$ under P and initial state distribution d_0, we have*

$$
\mu^{\pi\{s;a\}}(s) = \frac{1 - (1-\gamma)\sum_{s'} d_0(s')\overline{T}^{\pi}(s', s)}{1 + \gamma \sum_{s'} P(s, a, s')\overline{T}^{\pi}(s', s)} \geq \frac{1 - (1-\gamma)\cdot\overline{D}^{\pi}}{1 + \gamma \cdot \overline{D}^{\pi}}.
$$

Proof. The inequality is trivial due to the fact that by definition $\overline{D}^{\pi} \geq \overline{T}^{\pi}(s, s')$ for every s, s'. Thus, it suffices to prove the equality.

When $\gamma = 1$, we have $\frac{1}{\mu^{\pi_\dagger\{s;a\}}(s)} = \mathbb{E}\left[L^{\pi_\dagger\{s;a\}}(s, s)\right]$ (see Theorem 1.21 in (Durrett, 1999)), where $L^{\pi}(s, s')$ is the number of steps to reach s' starting from s in the induced Markov chain by π in P. Note that as $\gamma = 1$, the discounted reach times on P are $T^{\pi}(s, s') = \mathbb{E}\left[L^{\pi}(s, s')\right]$ for $s \neq s'$. We have

$$
\begin{aligned}
\frac{1}{\mu^{\pi_\dagger\{s;a\}}(s)} &= \mathbb{E}\left[L^{\pi_\dagger\{s;a\}}(s, s)\right] \\
&= 1 + \sum_{s'} P(s, a, s') \cdot T^{\pi_\dagger\{s;a\}}(s', s) \\
&= 1 + \sum_{s'} P(s, a, s') \cdot T^{\pi_\dagger}(s', s) \\
&= 1 + \sum_{s'} P(s, a, s') \cdot \overline{T}^{\pi_\dagger}(s', s).
\end{aligned}
$$

Note that we have used the fact that transitions of $\pi_\dagger$ are not changed, and therefore $\overline{T}^{\pi_\dagger} = T^{\pi_\dagger}$.

Now consider the case $\gamma < 1$. For $i \geq 1$, let $t_i(s)$ be the random variable denoting the step number when state s is visited for the i-th time. More formally

$$t_1(s) = \min\{j \geq 0 : s_j = s\},$$
$$t_i(s) = \min\{j > t_{i-1}(s) : s_j = s\}.$$

Using the definition of $\mu^{\pi\{s;a\}}(s)$ and independence of $t_i(s) - t_1(s)$ and $t_1(s)$, we obtain

$$\begin{aligned}
\mu^{\pi\{s;a\}}(s) &= (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbb{P}\left[s_t = s | s_0 \sim d_0, \pi\{s;a\}\right] \\
&= (1-\gamma)\mathbb{E}\left[\sum_{i=1}^{\infty}\gamma^{t_i(s)} | s_0 \sim d_0, \pi\{s;a\}\right] \\
&= (1-\gamma)\mathbb{E}\left[\gamma^{t_1(s)}\Big(1 + \sum_{i=2}^{\infty}\gamma^{t_i(s)-t_1(s)}\Big) | s_0 \sim d_0, \pi\{s;a\}\right] \\
&= (1-\gamma)\mathbb{E}\left[\gamma^{t_1(s)} | s_0 \sim d_0, \pi\{s;a\}\right]\mathbb{E}\left[\Big(1 + \sum_{i=2}^{\infty}\gamma^{t_i(s)-t_1(s)}\Big) | s_0 \sim d_0, \pi\{s;a\}\right] \\
&= (1-\gamma)\mathbb{E}\left[\gamma^{t_1(s)} | s_0 \sim d_0, \pi\{s;a\}\right]\left(1 + \gamma\mathbb{E}\left[\sum_{i=1}^{\infty}\gamma^{t_i(s)} | s_0 \sim P(s,a,.), \pi\{s;a\}\right]\right) \\
&= (1-\gamma)\mathbb{E}\left[\gamma^{t_1(s)} | s_0 \sim d_0, \pi\{s;a\}\right]\left(1 + \frac{\gamma\mu^{\pi\{s;a\}}_{P(s,a,.)}(s)}{1-\gamma}\right) \\
&= \mathbb{E}\left[\gamma^{t_1(s)} | s_0 \sim d_0, \pi\{s;a\}\right]\Big(1 - \gamma + \gamma\mu^{\pi\{s;a\}}_{P(s,a,.)}(s)\Big).
\end{aligned}$$

Here, $\mu^{\pi\{s;a\}}_{P(s,a,.)}$ is the state distribution of $\pi\{s;a\}$ under P when the initial state distribution is $P(s,a,.)$ instead of d_0. For an arbitrary policy π' define $X^{\pi'}(s,s')$ as

$$X^{\pi'}(s,s') = \mathbb{E}\left[\gamma^{t_1(s')} | s_0 = s, \pi'\right].$$

Using this, we can write the last equation as

$$\mu^{\pi\{s;a\}}(s) = \Big(\sum_{s'} d_0(s') X^{\pi\{s;a\}}(s',s)\Big)\Big(1 - \gamma + \gamma\mu^{\pi\{s;a\}}_{P(s,a,.)}(s)\Big). \tag{14}$$

Similarly

$$\begin{aligned}
\mu^{\pi\{s;a\}}_{P(s,a,.)}(s) &= \Big(\sum_{s'} P(s,a,s') X^{\pi\{s;a\}}(s',s)\Big)\Big(1 - \gamma + \gamma\mu^{\pi\{s;a\}}_{P(s,a,.)}(s)\Big) \\
\Rightarrow \mu^{\pi\{s;a\}}_{P(s,a,.)}(s) &= \frac{(1-\gamma)\Big(\sum_{s'} P(s,a,s') X^{\pi\{s;a\}}(s',s)\Big)}{1 - \gamma\sum_{s'} P(s,a,s') X^{\pi\{s;a\}}(s',s)}.
\end{aligned}$$

Plugging this into (14) we get

$$\mu^{\pi\{s;a\}}(s) = \left(\sum_{s'} d_0(s') X^{\pi\{s;a\}}(s', s)\right) \cdot \frac{1-\gamma}{1-\gamma\sum_{s'} P(s,a,s') X^{\pi\{s;a\}}(s', s)}.$$

Finally, note that $X^{\pi\{s;a\}}(s', s) = X^{\pi}(s', s)$ as $\pi\{s; a\}$ and π only differ in s which is does not affect the time to visit s for the first time. Moreover, we have $X^{\pi}(s', s) = 1 - \overline{T}^{\pi}(s', s)(1-\gamma)$ because for $s \neq s'$ one can write

$$\begin{aligned} X^{\pi}(s', s) &= \mathbb{E}\left[\gamma^{t_1(s')} | s_0 = s, \pi\right] \\ &= \mathbb{E}\left[\gamma^{\overline{L}^{\pi}(s,s')}\right] \\ &= \mathbb{E}\left[1 - \frac{1-\gamma^{\overline{L}^{\pi}(s,s')}}{1-\gamma} \cdot (1-\gamma)\right] \\ &= 1 - \overline{T}^{\pi}(s', s)(1-\gamma). \end{aligned}$$

and for $s' = s$, $X^{\pi}(s, s) = 1$ and $\overline{T}^{\pi}(s, s) = 0$. We get

$$\begin{aligned} \mu^{\pi\{s;a\}}(s) &= \frac{(1-\gamma)\sum_{s'} d_0(s') X^{\pi}(s', s)}{1-\gamma\sum_{s'} P(s,a,s') X^{\pi}(s', s)} \\ &= \frac{(1-\gamma)\sum_{s'} d_0(s')(1-(1-\gamma)\overline{T}^{\pi}(s', s))}{1-\gamma\sum_{s'} P(s,a,s')(1-(1-\gamma)\overline{T}^{\pi}(s', s))} \\ &= \frac{(1-\gamma)(1-\sum_{s'} d_0(s')(1-\gamma)\overline{T}^{\pi}(s', s))}{1-\gamma(1-\sum_{s'} P(s,a,s')(1-\gamma)\overline{T}^{\pi}(s', s))} \\ &= \frac{1-(1-\gamma)\sum_{s'} d_0(s')\overline{T}^{\pi}(s', s)}{1+\gamma(\sum_{s'} P(s,a,s')\overline{T}^{\pi}(s', s))}, \end{aligned}$$

which completes the proof. □

Now we can prove the upper bound in Theorem 1. First, note that $k(s,a) < |S|$ due to the condition $\gamma C_r(\overline{V}^{\pi_\dagger}(s_{k(s,a)}) - \overline{V}^{\pi_\dagger}(s_{|S|})) > 2C_p$ and $C_p \geq 0$. Consider the following solution:

$$\widehat{P}(s,a,s_i) = \begin{cases} \delta\overline{P}(s,a,s_i) & \text{if } a \neq \pi_\dagger(s) \text{ and } i \leq k(s,a) \\ \overline{P}(s,a,s_i) + \frac{1}{2}\overline{G}_{k(s,a)}(s,a) & \text{if } a \neq \pi_\dagger(s) \text{ and } i = |S| \\ \overline{P}(s,a,s_i) & \text{otherwise} \end{cases}, \tag{15}$$

$$\widehat{R}(s,a) = \begin{cases} \overline{R}(s,a) - \overline{\chi}^{\pi_\dagger}_{\overline{\beta}(s,a)}(s,a) + \overline{F}_{k(s,a)}(s,a) & \text{if } a \neq \pi_\dagger(s) \\ \overline{R}(s,a) & \text{if } a = \pi_\dagger(s) \end{cases}. \tag{16}$$

Let $\widehat{Q}$ and $\widehat{V}$ denote Q-values and V-values in $\widehat{M}$. From Lemma 1 and Corollary 1 it suffices to prove for state s and action $a \neq \pi_\dagger(s)$

$$\widehat{V}^{\pi_\dagger}(s) - \widehat{Q}^{\pi_\dagger}(s,a) \geq \frac{\epsilon}{\widehat{\mu}^{\pi_\dagger\{s;a\}}(s)}.$$

Note that rewards and transitions used by the target policy are not changed so we have $\widehat{V}^{\pi_\dagger} = \overline{V}^{\pi_\dagger}$ and $\widehat{\rho}^{\pi_\dagger} = \overline{\rho}^{\pi_\dagger}$. One can write

$$\begin{aligned}
\widehat{Q}^{\pi_\dagger}(s,a) =& \widehat{R}(s,a) - \overline{\rho}^{\pi_\dagger} + \gamma \sum_{i=1}^{|S|} \widehat{P}(s,a,s_i)\overline{V}^{\pi_\dagger}(s_i) \\
=& \overline{R}(s,a) - \overline{\chi}^{\pi_\dagger}_{\overline{\beta}(s,a)}(s,a) + \overline{F}_{k(s,a)}(s,a) - \overline{\rho}^{\pi_\dagger} + \gamma \sum_{i=1}^{|S|} \overline{P}(s,a,s_i)\overline{V}^{\pi_\dagger}(s_i) \\
&+ \frac{\gamma}{2}\overline{G}_{k(s,a)}(s,a) \cdot \overline{V}^{\pi_\dagger}(s_{|S|}) - \gamma \cdot (1-\delta) \sum_{i=1}^{k(s,a)} \overline{P}(s,a,s_i)\overline{V}^{\pi_\dagger}(s_i) \\
=& \overline{Q}^{\pi_\dagger}(s,a) - \overline{\chi}^{\pi_\dagger}_{\overline{\beta}(s,a)}(s,a) + \gamma \sum_{i=1}^{k(s,a)} (1-\delta)\overline{P}(s,a,s_i)(\overline{V}^{\pi_\dagger}(s_i) - \overline{V}^{\pi_\dagger}(s_{|S|})) \\
&+ \gamma \cdot \Big(\sum_{i=1}^{k(s,a)} (1-\delta)\overline{P}(s,a,s_i)\Big) \cdot \overline{V}^{\pi_\dagger}(s_{|S|}) - \gamma \cdot (1-\delta) \sum_{i=1}^{k(s,a)} \overline{P}(s,a,s_i)\overline{V}^{\pi_\dagger}(s_i) \\
=& \overline{Q}^{\pi_\dagger}(s,a) - \overline{\chi}^{\pi_\dagger}_{\overline{\beta}(s,a)}(s,a) \\
\leq& \overline{Q}^{\pi_\dagger}(s,a) - \frac{\overline{\rho}^{\pi\{s;a\}} - \overline{\rho}^{\pi} + \overline{\beta}(s,a)}{\overline{\mu}^{\pi\{s;a\}}(s)} \\
=& \overline{V}^{\pi_\dagger}(s) - \frac{\overline{\beta}(s,a)}{\overline{\mu}^{\pi\{s;a\}}(s)},
\end{aligned}$$

where in the last equality we used Corollary 1. By definition

$$\overline{\beta}(s,a) = \epsilon \cdot \overline{\mu}^{\pi_\dagger\{s;a\}}(s) \cdot \frac{1+\gamma \cdot \overline{D}^{\pi_\dagger}}{1-(1-\gamma)\cdot \overline{D}^{\pi_\dagger}}.$$

Thus, from Lemma 8 we can see that

$$\frac{\overline{\beta}(s,a)}{\overline{\mu}^{\pi\{s;a\}}(s)} \geq \frac{\epsilon}{\widehat{\mu}^{\pi\{s;a\}}(s)}.$$

Combining this with the last expression, we obtain

$$\overline{V}^{\pi_\dagger}(s) - \widehat{Q}^{\pi_\dagger}(s,a) \geq \frac{\epsilon}{\widehat{\mu}^{\pi_\dagger\{s;a\}}(s)}.$$

Thus, this is a solution for the problem. It only remains to find its cost. We can write

$$\begin{aligned}
\textsc{Cost}(\widehat{M}, \overline{M}) =& \Big(\sum_{s,a} \Big(C_r \cdot \big|\widehat{R}(s,a) - \overline{R}(s,a)\big| + C_p \cdot \sum_{s'} \big|\widehat{P}(s,a,s') - \overline{P}(s,a,s')\big| \Big)^p \Big)^{1/p} \\
\leq& \Big(\sum_{s,a} \Big(C_r \cdot \big(\overline{\chi}^{\pi_\dagger}_{\overline{\beta}(s,a)}(s,a) - \overline{F}_{k(s,a)}(s,a)\big) + C_p \cdot \big(\sum_{i=1}^{k(s,a)} (1-\delta)\overline{P}(s,a,s_i) + \frac{1}{2}\overline{G}_{k(s,a)}(s,a)\big) \Big)^p \Big)^{1/p}
\end{aligned}$$

$$= \Big(\sum_{s,a} \Big(C_r \cdot \big(\overline{\chi}^{\pi_\dagger}_{\overline{\beta}(s,a)}(s,a) - \overline{F}_{k(s,a)}(s,a) \big) + C_p \cdot \overline{G}_{k(s,a)}(s,a) \Big)^p \Big)^{1/p}$$

$$= \Big\| C_p \cdot \overline{G}_k + C_r \cdot (\overline{\chi}^{\pi_\dagger}_{\overline{\beta}} - \overline{F}_k) \Big\|_p ,$$

which concludes the proof.

E.3 Discussion on Choosing δ

While δ can be set to small values, making the corresponding constraint in (P1) a relatively weak condition, it is important to note that its value controls parameters of MDP $\widehat{M}$ that are important for practical considerations in the offline setting. Moreover, since δ is a parameter in the optimization problems (P2) (and (P2')), its value is also important for the online setting.

In the case of attacks on a learning agent, our results have dependency on the agent's regret or number of suboptimal steps, which in turn depend on the properties of MDP $\widehat{M}$. For example, if the agent adopts UCRL as its learning procedure, its regret will depend on the diameter of $\widehat{M}$. Hence, δ should be adjusted based on time horizon T, so that the parameters of MDP $\widehat{M}$ relevant for the agent's regret do not outweigh time horizon T.

In the case of attacks on a planning agent with average reward criteria, setting δ to small values could result in a solution $\widehat{M}$ that has a large mixing time, in which case the score ρ might not approximate well the average of obtained rewards in a finite horizon (e.g., see (Even-Dar et al., 2005)). This means that the choice of δ should account for the finiteness of time horizon in practice.

We leave a more detailed analysis that includes these considerations for future work.

Appendix F. Proofs For Online Attacks (Section 5)

This section contains proof of our results for Lemma 3, Theorem 2, and Theorem 3.

F.1 Proof of Lemma 3

This lemma is based on the simple observation: when a learner draws its experience from an MDP M that has $\pi_\dagger$ as its ϵ-robust optimal policy, instantiating $\textsc{SubOpt}(T, M, \epsilon')$ with $\epsilon' = \epsilon$ will give us the number of times the learner deviates from $\pi_\dagger$. In particular, we need to show that $\textsc{AvgMiss}(T) = \frac{1}{T}\textsc{SubOpt}(T, M, \epsilon)$ when $\pi_\dagger$ is an ϵ-robust optimal policy. By using the definition of $\textsc{SubOpt}(T, M, \epsilon')$ with $\epsilon' = \epsilon$, we obtain:

$$\textsc{SubOpt}(T, M, \epsilon) = \sum_{t=0}^{T-1} \mathbb{1}\left[a_t \notin \{\pi(s_t) \mid \rho^\pi \geq \rho^{\pi^*} - \epsilon\} \right].$$

Since $\pi_\dagger$ is ϵ-robust optimal in M, the only π satisfying $\rho^\pi \geq \rho^{\pi^*} - \epsilon$ is $\pi_\dagger$ itself. This means that we have

$$\begin{aligned} \textsc{SubOpt}(T, M, \epsilon) &= \sum_{t=0}^{T-1} \mathbb{1}\left[a_t \neq \pi_\dagger(s_t) \right] \\ &= T \cdot \textsc{AvgMiss}(T), \end{aligned}$$

which proves the claim.

F.2 Proof of Theorem 2: Average Reward Criteria, $\gamma = 1$

We need to prove bounds on the expected cost and average mismatches of the online attack against a regret-minimization learner.

Since $\widehat{M} = (S, A, \widehat{R}, \overline{P})$ is a solution to the optimization problem (P2), $\pi_\dagger$ is ϵ-robust optimal in $\widehat{M}$. Notice that the learner receives feedback from the MDP $\widehat{M}$. Using Lemma 2 we can obtain the following bound on the expected average mismatches:

$$\mathbb{E}\left[\textsc{AvgMiss}(T)\right] \leq \frac{\widehat{\mu}_{\max}}{\epsilon \cdot T} \cdot \Big(\mathbb{E}\left[\textsc{Regret}(T, \widehat{M})\right] + 2\left\|\widehat{V}^{\pi_\dagger}\right\|_\infty\Big).$$

Note that $\widehat{V}^{\pi_\dagger} = \overline{V}^{\pi_\dagger}$ and we could also substitute $\widehat{V}^{\pi_\dagger}$ with $\overline{V}^{\pi_\dagger}$ in the bound.

Next, we analyze the expected attack cost. Since $\overline{R}(s,a) = \widehat{R}(s,a), \overline{P}(s,a,s') = \widehat{P}(s,a,s')$ for $s, s', a = \pi_T(s)$, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=0}^{T-1}\left(C_r \cdot \left|\widehat{R}_t(s_t,a_t) - \overline{R}(s_t,a_t)\right| + C_p \cdot \sum_{s'}\left|\widehat{P}_t(s_t,a_t,s') - \overline{P}(s_t,a_t,s')\right|\right)^p\right] \\
=&\mathbb{E}\left[\sum_{t=0}^{T-1}\mathbb{1}\left[a_t \neq \pi(s_t)\right]\left(C_r \cdot \left|\widehat{R}_t(s_t,a_t) - \overline{R}(s_t,a_t)\right| + C_p \cdot \sum_{s'}\left|\widehat{P}_t(s_t,a_t,s') - \overline{P}(s_t,a_t,s')\right|\right)^p\right] \\
\leq&\Big(\textsc{Cost}(\widehat{M}, \overline{M}, C_r, C_p, \infty)\Big)^p \cdot T \cdot \mathbb{E}\left[\textsc{AvgMiss}(T)\right] \\
\leq&\Big(\textsc{Cost}(\widehat{M}, \overline{M}, C_r, C_p, \infty)\Big)^p \cdot \frac{\widehat{\mu}_{\max}}{\epsilon} \cdot \Big(\mathbb{E}\left[\textsc{Regret}(T, \widehat{M})\right] + 2\left\|\widehat{V}^{\pi_\dagger}\right\|_\infty\Big).
\end{aligned}$$

Since $p \geq 1$, the function $f(x) = x^{1/p}$ is concave. By Jensen's inequality, this means that $\mathbb{E}\left[f(X)\right] \leq f(\mathbb{E}\left[X\right])$, where X is a random variable. We can write $\mathbb{E}\left[\textsc{AvgCost}(T)\right]$ to be

$$\begin{aligned}
&= \frac{1}{T}\mathbb{E}\left[\left(\sum_{t=0}^{T-1}\left(C_r \cdot \left|\widehat{R}_t(s_t,a_t) - \overline{R}(s_t,a_t)\right| + C_p \cdot \sum_{s'}\left|\widehat{P}_t(s_t,a_t,s') - \overline{P}(s_t,a_t,s')\right|\right)^p\right)^{1/p}\right] \\
&\leq \frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1}\left(C_r \cdot \left|\widehat{R}_t(s_t,a_t) - \overline{R}(s_t,a_t)\right| + C_p \cdot \sum_{s'}\left|\widehat{P}_t(s_t,a_t,s') - \overline{P}(s_t,a_t,s')\right|\right)^p\right]^{1/p} \\
&\leq \frac{\textsc{Cost}(\widehat{M}, \overline{M}, C_r, C_p, \infty)}{T} \cdot \left(\frac{\widehat{\mu}_{\max}}{\epsilon} \cdot \Big(\mathbb{E}\left[\textsc{Regret}(T, \widehat{M})\right] + 2\left\|\widehat{V}^{\pi_\dagger}\right\|_\infty\Big)\right)^{1/p}.
\end{aligned}$$

F.3 Proof of Theorem 3: Discounted Reward Criteria, $\gamma < 1$

We need to prove bounds on the expected cost and average mismatches of the online attack against a learner with a bounded number of suboptimal steps.

Since $\widehat{M} = (S, A, \widehat{R}, \overline{P})$ is a solution to the optimization problem (P2), $\pi_\dagger$ is ϵ-robust optimal in $\widehat{M}$. Notice that the learner receives feedback from the MDP $\widehat{M}$. Using Lemma 3 we can obtain the following expected average mismatches:

$$\mathbb{E}\left[\textsc{AvgMiss}(T)\right] = \frac{1}{T} \cdot \mathbb{E}\left[\textsc{SubOpt}(T, \widehat{M}, \epsilon)\right].$$

Next, we analyze the expected attack cost. Since $\overline{R}(s,a) = \widehat{R}(s,a), \overline{P}(s,a,s') = \widehat{P}(s,a,s')$ for $s, s', a = \pi_T(s)$, we have

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=0}^{T-1}\left(C_r \cdot \left|\widehat{R}_t(s_t,a_t) - \overline{R}(s_t,a_t)\right| + C_p \cdot \sum_{s'}\left|\widehat{P}_t(s_t,a_t,s') - \overline{P}(s_t,a_t,s')\right|\right)^p\right] \\
=&\mathbb{E}\left[\sum_{t=0}^{T-1}\mathbb{1}\left[a_t \neq \pi(s_t)\right]\left(C_r \cdot \left|\widehat{R}_t(s_t,a_t) - \overline{R}(s_t,a_t)\right| + C_p \cdot \sum_{s'}\left|\widehat{P}_t(s_t,a_t,s') - \overline{P}(s_t,a_t,s')\right|\right)^p\right] \\
\leq&\Big(\textsc{Cost}(\widehat{M},\overline{M},C_r,C_p,\infty)\Big)^p \cdot T \cdot \mathbb{E}\left[\textsc{AvgMiss}(T)\right] \\
\leq&\Big(\textsc{Cost}(\widehat{M},\overline{M},C_r,C_p,\infty)\Big)^p \cdot \mathbb{E}\left[\textsc{SubOpt}(T,\widehat{M},\epsilon)\right].
\end{aligned}
$$

Since $p \geq 1$, the function $f(x) = x^{1/p}$ is concave. By Jensen's inequality, this means that $\mathbb{E}\left[f(X)\right] \leq f(\mathbb{E}\left[X\right])$, where X is a random variable. We can write $\mathbb{E}\left[\textsc{AvgCost}(T)\right]$ to be

$$
\begin{aligned}
&= \frac{1}{T}\mathbb{E}\left[\left(\sum_{t=0}^{T-1}\left(C_r \cdot \left|\widehat{R}_t(s_t,a_t) - \overline{R}(s_t,a_t)\right| + C_p \cdot \sum_{s'}\left|\widehat{P}_t(s_t,a_t,s') - \overline{P}(s_t,a_t,s')\right|\right)^p\right)^{1/p}\right] \\
&\leq \frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1}\left(C_r \cdot \left|\widehat{R}_t(s_t,a_t) - \overline{R}(s_t,a_t)\right| + C_p \cdot \sum_{s'}\left|\widehat{P}_t(s_t,a_t,s') - \overline{P}(s_t,a_t,s')\right|\right)^p\right]^{1/p} \\
&\leq \frac{\textsc{Cost}(\widehat{M},\overline{M},C_r,C_p,\infty)}{T} \cdot \left(\mathbb{E}\left[\textsc{SubOpt}(T,\widehat{M},\epsilon)\right]\right)^{1/p}.
\end{aligned}
$$

References

Shipra Agrawal and Randy Jia. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. In *Advances in Neural Information Processing Systems*, 2017.

Scott Alfeld, Xiaojin Zhu, and Paul Barford. Data poisoning attacks against autoregressive models. In *AAAI*, 2016.

John Asmuth, Michael L Littman, and Robert Zinkov. Potential-based shaping in model-based reinforcement learning. In *AAAI*, pages 604–609, 2008.

Peter Auer and Ronald Ortner. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 49–56, 2007.

Kiarash Banihashem, Adish Singla, and Goran Radanovic. Defense against reward poisoning attacks in reinforcement learning. *CoRR*, abs/2102.05776, 2021.

Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *ICML*, 2012.

Daniel S Brown and Scott Niekum. Machine teaching for inverse reinforcement learning: Algorithms and applications. In *AAAI*, pages 7749–7758, 2019.

Maya Cakmak and Manuel Lopes. Algorithmic and human teaching of sequential decision tasks. In *AAAI*, 2012.

Olivier Chapelle, Eren Manavoglu, and Rómer Rosales. Simple and scalable response prediction for display advertising. *ACM TIST*, 5(4):61:1–61:34, 2014.

Tong Chen, Jiqiang Liu, Yingxiao Xiang, Wenjia Niu, Endong Tong, and Zhen Han. Adversarial attack and defense in reinforcement learning from AI security view. *Cybersecurity*, 2 (1):11, 2019.

Yuxin Chen, Adish Singla, Oisin Mac Aodha, Pietro Perona, and Yisong Yue. Understanding the role of adaptivity in machine teaching: The case of version space learners. In *NeurIPS*, pages 1483–1493, 2018.

Felipe Leno Da Silva and Anna Helena Reali Costa. A survey on transfer learning for multiagent reinforcement learning systems. *Journal of Artificial Intelligence Research*, 64: 645–703, 2019.

Rati Devidze, Farnam Mansouri, Luis Haug, Yuxin Chen, and Adish Singla. Understanding the power and limitations of teaching with imperfect knowledge. In *IJCAI*, pages 2647–2654, 2020.

Richard Durrett. *Essentials of Stochastic Processes*, volume 1. Springer, 1999.

Eyal Even-Dar and Yishay Mansour. Learning rates for q-learning. *Journal of machine learning Research*, 5(Dec):1–25, 2003.

Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. Experts in a Markov decision process. In *Advances in Neural Information Processing Systems*, pages 401–408, 2005.

Marc Fischer, Matthew Mirman, Steven Stalder, and Martin Vechev. Online robustness training for deep reinforcement learning. *CoRR*, abs/1911.00887, 2019.

Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. *CoRR*, abs/1905.10615, 2019.

Sally A Goldman and Michael J Kearns. On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31, 1995.

Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 3909–3917, 2016.

Luis Haug, Sebastian Tschiatschek, and Adish Singla. Teaching inverse reinforcement learners via features and demonstrations. In *NeurIPS*, 2018.

Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and J Doug Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and Artificial Intelligence*, pages 43–58, 2011.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *CoRR*, abs/1702.02284, 2017.

Yunhan Huang and Quanyan Zhu. Deceptive reinforcement learning under adversarial manipulations on cost signals. In *GameSec*, pages 217–237, 2019.

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

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

Kwang-Sung Jun, Lihong Li, Yuzhe Ma, and Xiaojin Zhu. Adversarial attacks on stochastic bandits. In *NeurIPS*, pages 3644–3653, 2018.

Parameswaran Kamalaruban, Rati Devidze, Volkan Cevher, and Adish Singla. Interactive teaching algorithms for inverse reinforcement learning. In *IJCAI*, pages 2692–2700, 2019.

Parameswaran Kamalaruban, Rati Devidze, Volkan Cevher, and Adish Singla. Environment Shaping in Reinforcement Learning using State Abstraction. *CoRR*, abs/2006.13160, 2020.

Pang Wei Koh, Jacob Steinhardt, and Percy Liang. Stronger data poisoning attacks break data sanitization defenses. *CoRR*, abs/1811.00741, 2018.

Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. Data poisoning attacks on factorization-based collaborative filtering. In *Advances in Neural Information Processing Systems*, pages 1885–1893, 2016.

Chong Li and Meikang Qiu. *Reinforcement Learning for Cyber-Physical Systems with Cybersecurity Case Studies*. Chapman and Hall/CRC, 2019.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, pages 661–670, 2010.

Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. Tactics of adversarial attack on deep reinforcement learning agents. In *IJCAI*, pages 3756–3762, 2017.

Fang Liu and Ness B. Shroff. Data poisoning attacks on stochastic bandits. In *ICML*, pages 4042–4050, 2019.

Thodoris Lykouris, Max Simchowitz, Aleksandrs Slivkins, and Wen Sun. Corruption robust exploration in episodic reinforcement learning. *CoRR*, abs/1911.08689, 2019.

Yuzhe Ma, Kwang-Sung Jun, Lihong Li, and Xiaojin Zhu. Data poisoning attacks in contextual bandits. In *GameSec*, pages 186–204, 2018.

Yuzhe Ma, Xuezhou Zhang, Wen Sun, and Jerry Zhu. Policy poisoning in batch reinforcement learning and control. In *NeurIPS*, pages 14543–14553, 2019.

Sridhar Mahadevan. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine Learning*, 22(1-3):159–195, 1996.

Sridhar Mahadevan and Jonathan Connell. Automatic programming of behavior-based robots using reinforcement learning. *Artificial Intelligence*, 55(2):311–365, 1992.

Farnam Mansouri, Yuxin Chen, Ara Vartanian, Jerry Zhu, and Adish Singla. Preference-based batch and sequential teaching: Towards a unified view of models. In *NeurIPS*, pages 9195–9205, 2019.

H Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *ICML*, pages 536–543, 2003.

Shike Mei and Xiaojin Zhu. Using machine teaching to identify optimal training-set attacks on machine learners. In *AAAI*, pages 2871–2877, 2015.

Brad Miller, Paul Pearce, Chris Grier, Christian Kreibich, and Vern Paxson. What's clicking what? techniques and innovations of today's clickbots. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 164–183. Springer, 2011.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Gina Neff. Talking to bots: Symbiotic agency and the case of tay. *International Journal of Communication*, 2016.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, 1999.

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

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

Tomi Peltola, Mustafa Mert Çelikok, Pedram Daee, and Samuel Kaski. Machine teaching of active sequential learners. In *NeurIPS*, 2019.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *ICML*, pages 2817–2826, 2017.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1st edition, 1994. ISBN 0471619779.

Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla. Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning. In *ICML*, 2020.

Kevin Regan and Craig Boutilier. Robust policy computation in reward-uncertain MDPs using nondominated policies. In *AAAI*, 2010.

Paul E Rybski, Kevin Yoon, Jeremy Stolarz, and Manuela M Veloso. Interactive robot task training through dialog and demonstration. In *Proceedings of the International Conference on Human-robot interaction*, pages 49–56, 2007.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML*, pages 1889–1897, 2015.

Adish Singla, Ilija Bogunovic, G Bartók, A Karbasi, and A Krause. On actively teaching the crowd to classify. In *NIPS Workshop on Data Driven Education*, 2013.

Adish Singla, Ilija Bogunovic, Gábor Bartók, Amin Karbasi, and Andreas Krause. Near-optimally teaching the crowd to classify. In *ICML*, 2014.

Alexander L Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L Littman. Pac model-free reinforcement learning. In *ICML*, pages 881–888, 2006.

Yanchao Sun, Da Huo, and Furong Huang. Vulnerability-aware poisoning mechanism for online rl with unknown dynamics. *CoRR*, abs/2009.00774, 2020.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An Introduction*. MIT press, 2018.

Prasad Tadepalli and DoKyeong Ok. H-learning: A reinforcement learning method to optimize undiscounted average reward, 1994.

Aviv Tamar, Shie Mannor, and Huan Xu. Scaling up robust MDPs using function approximation. In *ICML*, pages 181–189, 2014.

Edgar Tretschk, Seong Joon Oh, and Mario Fritz. Sequential attacks on agents for long-term adversarial goals. *CoRR*, abs/1805.12487, 2018.

Sebastian Tschiatschek, Ahana Ghosh, Luis Haug, Rati Devidze, and Adish Singla. Learner-aware teaching: Inverse reinforcement learning with preferences and constraints. In *NeurIPS*, 2019.

Thomas J. Walsh and Sergiu Goschin. Dynamic teaching in sequential decision making environments. In *UAI*, pages 863–872, 2012.

Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. Is feature selection secure against training data poisoning? In *ICML*, pages 1689–1698, 2015.

Haoqi Zhang and David C. Parkes. Value-based policy teaching with active indirect elicitation. In *AAAI*, 2008.

Haoqi Zhang, David C. Parkes, and Yiling Chen. Policy teaching through reward function learning. In *EC*, 2009.

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. *CoRR*, abs/2003.08938, 2020a.

Huan Zhang, Hongge Chen, Duane Boning, and Cho-Jui Hsieh. Robust reinforcement learning on state observations with learned optimal adversary. *CoRR*, abs/2101.08452, 2021a.

Xuezhou Zhang, Yuzhe Ma, Adish Singla, and Xiaojin Zhu. Adaptive reward-poisoning attacks against reinforcement learning. In *ICML*, 2020b.

Xuezhou Zhang, Yiding Chen, Xiaojin Zhu, and Wen Sun. Robust policy gradient against strong data corruption. *CoRR*, abs/2102.05800, 2021b.

Xiaojin Zhu. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *AAAI*, pages 4083–4087, 2015.

Xiaojin Zhu. An optimal control view of adversarial machine learning. *CoRR*, abs/1811.04422, 2018.

Xiaojin Zhu, Adish Singla, Sandra Zilles, and Anna N Rafferty. An overview of machine teaching. *CoRR*, abs/1801.05927, 2018.