Offline Reward Perturbation Boosts Distributional Shift in Online RL

Zishun Yu[*1] **Siteng Kang**[*1] **Xinhua Zhang**[1]

[1]Department of Computer Science, University of Illinois Chicago, Chicago, IL, USA

Abstract

Offline-to-online reinforcement learning has recently been shown effective in reducing the online sample complexity by first training from offline collected data. However, this additional data source may also invite new poisoning attacks that target offline training. In this work, we reveal such vulnerabilities in *critic-regularized* offline RL by proposing a novel data poisoning attack method, which is stealthy in the sense that the performance during the offline training remains intact, but the online fine-tuning stage will suffer a significant performance drop. Our method leverages the techniques from bi-level optimization to promote the over-estimation/distribution shift under offline-to-online reinforcement learning. Experiments on four environments confirm the satisfaction of the new stealthiness requirement, and can be effective in attacking with only a small budget and without having white-box access to the victim model.

1 INTRODUCTION

Offline reinforcement learning (RL) has recently opened up new opportunities of leveraging offline batch data to improve the RL algorithms, significantly reducing the online sample complexity of interacting with the environment (Levine et al., 2020). It is particularly valuable for many applications where directly applying an automated policy can be dangerous, expensive, or unethical. For example, educational assistants, autonomous driving, and healthcare.

However, due to the limited coverage of offline data or the suboptimality of the demonstrator (Fu et al., 2020), a purely offline trained model is generally not effective when deployed online, and a common wisdom is to fine-tune it via additional online interactions, whose sample complexity is expected to be saved thanks to the initialization from offline training (Xie et al., 2021; Nakamoto et al., 2023).

Interestingly, such a direct offline-to-online transfer (O2O) is often plagued with catastrophic performance drop at online transfer, which poses safety challenges for the real system such as driving and therapy. This is primarily due to the distributional shift of the state (Fujimoto et al., 2019; Kumar et al., 2019; Fu et al., 2019; Kumar et al., 2020a), and the Q-value has not been well estimated, often over-estimated, for the state-actions lying outside the offline distribution (Farahmand et al., 2010; Munos, 2005).

Existing literature (e.g. Kumar et al., 2020b; Kostrikov et al., 2022; Lee et al., 2022; Yu & Zhang, 2023; Nakamoto et al., 2023) shows that improved O2O RL methods can effectively control negative effect caused by the distribution shift, hence leading to improved online sample efficiency. Typical O2O solutions includes endowing conservatism on offline Q-function approximation (Kumar et al., 2020b; Nakamoto et al., 2023), or regularizing the divergence between the learned policy and the behavior policy (Nair et al., 2020), to avoid catastrophic distribution shift caused by false value over-estimation. In addition, distribution correction (Lee et al., 2022), critic reconstruction (Yu & Zhang, 2023), and ensemble methods (Zhang et al., 2023; Wang et al., 2023) also show effective O2O transfers.

There is still a long list of O2O methods that emerged recently (Wagenmaker & Pacchiano, 2023; Chen & Wen, 2023; Mark et al., 2023; Lei et al., 2024, etc.). Among the aforementioned works, surgery on the Q-function is one of the most prevalent principles to address O2O. As O2O heavily depends on a “well-behaved” Q-function, it also creates vulnerability in such scenarios, as one may manipulate Q-functions in a malicious way.

The key question we investigate in this paper is

> Are the O2O algorithms robust to reward poisoning on the offline batch data?

[*]Equal contribution. Correspondence to S.K.

Since offline data often comes from crowd-sourcing or other third parties, it may carry malicious poisons that catastrophically damage the online fine-tuning while remaining stealthy by keeping the offline performance competitive.

In general, poison attack is performed on the training data, such that the models trained with it will perform poorly on the test scenarios. In O2O RL, the attacker may alter the state, action, or reward of the offline data. In this paper, we focus on poisoning of rewards, and aim to achieve two objectives:

- **Effectiveness**: after offline training on the poisoned data, the agent will suffer a catastrophic performance drop at the beginning of the online fine-tuning, compared with its performance at the end of the offline training.
- **Stealthiness**: during the offline training, the performance as measured by interacting with the environment (but not using it to update the model) should be similar to that achieved by a clean trained agent. This is in addition to the standard ℓ_p norm constraints on the magnitude of reward modification.

These definitions of stealthiness and effectiveness are particular realistic. As O2O RL is a two-phase learning scheme, attacks that aim to undermine the offline performance may be of less risk to the system because the victim can detect the low performance of the offline model. However, an attacker that is stealthy offline but effective online could be more surprising and harmful. Therefore, understanding such vulnerability of O2O RL is essential towards robust O2O transfer.

Our contribution is to achieve these goals, revealing the vulnerability of O2O RL to data poisoning attack. Our innovations can be summarized as follows:

- We propose the first poisoning attack on O2O RL that promotes the Q-function over-estimation and hence distributional shift.
- We achieve the poisoning through an efficient bi-level optimization technique.
- Our approach requires no access to the victim agent or the online environment.

We applied our poisoner to Frozen Lake (Brockman et al., 2016) and three locomotion environments from D4RL (Fu et al., 2020). The stealthiness is clearly verified, and it is shown more effective in compromising online fine-tuning performance than other baselines.

2 RELATED WORK

The vulnerability to various types of attacks has been well studied in supervised learning field. Evasion attack (Goodfellow et al., 2015) assumes the attacker can manipulate testing inputs after the victim model is trained. Data poisoning attack, on the other hand, is performed on the training inputs. The attacker may insert (Chen et al., 2017) or modify the training inputs (Biggio et al., 2012; Shafahi et al., 2018) to undermine the performance of the trained victim model.

Attacks in Online RL Reward poisoning has been extensively studied in bandit (Ma et al., 2018; Bogunovic et al., 2021; Garcelon et al., 2020; Guan et al., 2020; Jun et al., 2018; Liu & Shroff, 2019; Lu et al., 2021; yang et al., 2020; Zuo, 2020) and online RL (Banihashem et al., 2022; Huang & Zhu, 2019; Liu & Lai, 2023; Rakhsha et al., 2021a,b; Sun et al., 2021; Zhang et al., 2020) settings.

Attacks in Offline RL Reward poisoning in batch/offline RL (Ma et al., 2019; Rangi et al., 2022b; Zhang & Parkes, 2008; Zhang et al., 2009; Rakhsha et al., 2021b,a) is perhaps more relevant to our work, in contrast to online learning where the data collection procedure is also polluted due to attacked policy. In addition, Gong et al. (2022) proposed the first backdoor attack in offline RL by altering the training observations; and Wu et al. (2023) designed a data poising attack specifically on multi-agent RL.

Defenses in RL To address the vulnerabilities raised in the literature, various defenses against adversarial attacks on RL have been proposed (e.g., Zhang et al., 2009; Banihashem et al., 2023; Lykouris et al., 2021; Rangi et al., 2022a).

However, existing attacks in online RL require access to online environment and are therefore infeasible in many practical scenarios. On the other hand, offline RL attacks leads to poor performance during the validation and can be detected before online fine-tuning. To the best of our knowledge, the stealthiness notion—where the impact on performance is not noticeable offline but occurs online—has not been explored in current literature. Hence, none of the existing attack (or defense) methods can be directly applied to O2O RL settings to achieve our objectives.

3 PROBLEM SETUP

In this section, we set up the three participants in the O2O poisoning problem: the environment, the victim agent, and the attacker.

3.1 PRELIMINARY

We formulate the RL process via the standard Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbb{P}, R, \gamma, \mu_0)$. Here $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathbb{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\gamma \in [0, 1)$ is the discount factor, and $\mu_0 : \mathcal{S} \to \mathbb{R}$ is the initial state distribution.

For the **victim agent**, we define its policy $\pi(a|s)$ as a distribution of taking action a at state s. The agent's goal is to find the optimal policy that maximizes the expected return $\pi^* = \arg\max_\pi J(\pi)$, where $J(\pi) := \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t | \mathcal{M}]$.

In the offline RL setting, there is a batch of transitions $\mathcal{D} = \{(s, a, r, s')\}$, referred to as offline dataset $\mathcal{D}$, that are collected by applying an unknown behavior policy in the environment. And the offline agent aims to learn a high-return policy π given $\mathcal{D}$, although the expected return J_π may vary depending on the quality of dataset $\mathcal{D}$. O2O RL appends a subsequent online fine-tuning stage by continuing the training of π and Q (if applicable) using new online interactions along with (optionally) pre-collected offline data.

In the O2O literature, it has been shown that offline conservative Q-learning (CQL, Kumar et al., 2020b) followed by an off-policy algorithm—often soft actor critic (SAC, Haarnoja et al., 2018)—for online fine-tuning is a strong yet simple baseline (Lee et al., 2022; Yu & Zhang, 2023). Intuitively, it is effective because CQL provides a good Q-function initialization that suppresses Q-values for out-of-distribution (OOD) actions, avoiding poor online exploration led by false over-estimation. And using an off-policy algorithm online allows faster learning as the Q-function is now freed from conservative constraints/penalties.

As CQL+SAC has served as a common baseline in O2O literature (Lee et al., 2022; Nakamoto et al., 2023; Yu & Zhang, 2023), we will use the same CQL+SAC scheme as our victim O2O agent for *continuous* action experiments, including the MuJoCo (Todorov et al., 2012) locomotion tasks. For *discrete* action environments such as Frozen Lake, we used DoubleDQN (Hasselt et al., 2016) as the online algorithm.

Soft Actor-Critic SAC is an actor-critic algorithm based on the maximum entropy framework. Akin to canonical actor-critic, it includes actor update and critic update, as shown in (2) and (1), respectively. In particular, we employed SAC-v2 (Haarnoja et al., 2018), an alternative implementation that automatically adjusts the entropy of the policy, via the Lagrangian dual formulation, where the Lagrangian multiplier is often called the temperature α, and its update rule is given in (3) via its derivative in α.

$$\begin{aligned} \mathcal{L}_Q^{\text{SAC}}(\psi, \mathcal{D}) &:= \mathop{\mathbb{E}}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_\psi(s,a) - y(r,s')\right)^2\right] \\ y(r,s') &:= r + \gamma \mathop{\mathbb{E}}_{a'\sim\pi_\theta(s')}[Q_{\bar{\psi}}(s',a') - \alpha \log \pi_\theta(a'|s')] \end{aligned} \tag{1}$$

$$\mathcal{L}_\pi^{\text{SAC}}(\theta, \mathcal{D}) := \mathop{\mathbb{E}}_{s\sim\mathcal{D}} \mathop{\mathbb{E}}_{a\sim\pi_\theta(s)} [\alpha \log \pi_\theta(a|s) - Q_\psi(s,a)] \tag{2}$$

$$\mathcal{L}_{\text{temp}}^{\text{SAC}}(\alpha, \mathcal{D}) := -\alpha \mathop{\mathbb{E}}_{s\sim\mathcal{D}} \mathop{\mathbb{E}}_{a\sim\pi_\theta(s)} [\log \pi_\theta(a|s) - \bar{\mathcal{H}}]. \tag{3}$$

Here the expectation $\mathbb{E}_{a\sim\pi_\theta(s)}[\cdot]$ could be directly evaluated for discrete action spaces and be stochastically approximated for continuous action spaces.

The actor update (2) aims to maximize the Q-values hence maximizing the cumulative rewards alongside the policy's entropy. The critic update (1) aims to find a better soft Q-function approximation by minimizing the squared temporal difference error, where $\bar{\psi}$ stands for target network, a commonly used trick in RL literature to stabilize RL training. It can be often updated using the Polyak averaging (or exponential moving averaging), which is essentially $\bar{\psi} \leftarrow \tau\psi + (1-\tau)\bar{\psi}$, where $\tau \in (0, 1)$ is a hyper-parameter that controls how fast the target network $\bar{\psi}$ evolves towards the current Q-network ψ. The temperature update (3) automatically tunes $\alpha > 0$ to ensure that the entropy of the policy is lower bounded by a target entropy $\bar{\mathcal{H}}$.

Conservative Q-Learning CQL is a popular choice for offline and O2O RL that combats the distribution shift issue. The central idea is to regularize the Q-values of actions that are not observed in the offline dataset. Such regularity avoids over-estimations of OOD actions that may have a low return in the real environment. We will also provide an illustration of such a conservative estimation in our toy example in Figure 1. Specifically, we consider a commonly used variant of CQL, namely CQL($\mathcal{H}$), whose regularizer is given in (4) along with the squared loss. In addition, CQL($\mathcal{H}$) follows (5) to update policy for continuous action spaces, while in discrete space the policy is induced greedily from Q_ψ.

$$\begin{aligned} \mathcal{L}_Q^{\text{CQL}}(\psi, \mathcal{D}) &:= \mathop{\mathbb{E}}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_\psi(s,a) - y(r,s')\right)^2\right] \\ &+ \underbrace{\lambda \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}}[\log \textstyle\sum_u \exp(Q_\psi(s,u)) - Q_\psi(s,a)],}_{=:\, \mathcal{R}^{\text{CQL}}(Q_\psi, \mathcal{D})} \\ &\text{discrete: } y(r,s') := r + \gamma Q_{\bar{\psi}}(s', \text{argmax}_{a'} Q_\psi(s',a')) \\ &\text{continuous: } y(r,s') := r + \gamma \mathop{\mathbb{E}}_{a'\sim\pi_\theta(s')}[Q_{\bar{\psi}}(s',a')] \end{aligned} \tag{4}$$

$$\mathcal{L}_\pi^{\text{CQL}}(\theta, \mathcal{D}) := -\mathop{\mathbb{E}}_{s\sim\mathcal{D}} \mathop{\mathbb{E}}_{a\sim\pi_\theta(s)} [Q_\psi(s,a)]. \tag{5}$$

where $\mathcal{R}^{\text{CQL}}$ is a conservative regularizer, similarly the expectation $\mathbb{E}_{a\sim\pi(s)}[\cdot]$ and the log-sum-exp $\log \sum_a \exp Q(s,a)$ are tractable for discrete action spaces and can be stochastically approximated for continuous spaces.

Algorithm 1 is an example of O2O protocol with CQL used for offline training and SAC for online fine-tuning. For our additional experiments in Section 6.2, one could replace the offline/online algorithms with corresponding alternatives.

3.2 MOTIVATION

Distribution Shift It is arguably well known, in the O2O literature (Nair et al., 2020; Lee et al., 2022; Yu & Zhang, 2023; Nakamoto et al., 2023), that (dramatic) distribution shifts caused by over-estimated Q values for OOD state/actions lead to catastrophic performance drops during O2O transfer. This serves as the key motivation for our attacking algorithm, which we elaborate on next.

Algorithm 1 O2O protocol: offline (CQL) + online (SAC)

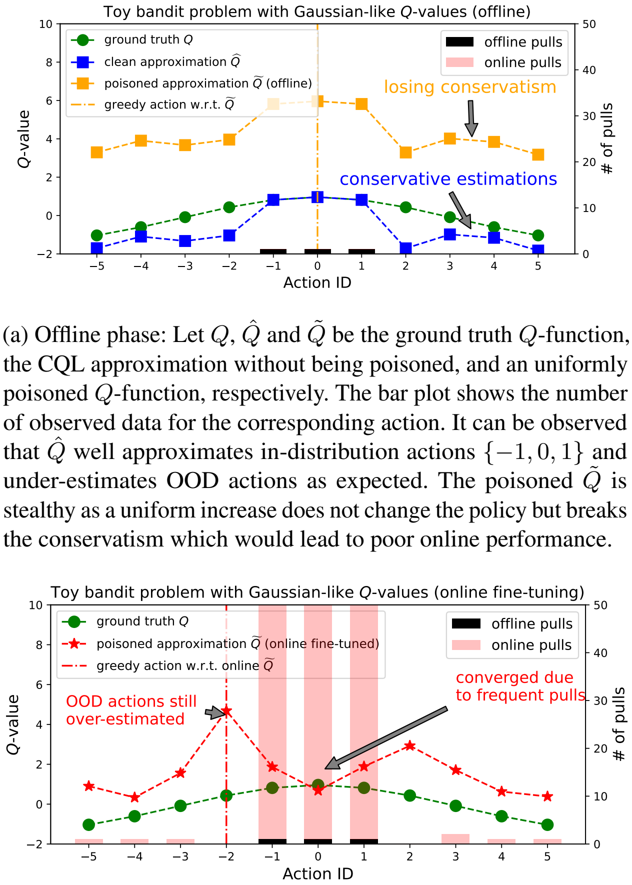
```
Input: offline dataset $\mathcal{D} = \{(s, a, r, s')\}$
// offline training phase with CQL.
initialize CQL parameters $\theta, \psi, \bar{\psi}$
for number of offline iterations do
    sample mini-batch from offline dataset $\mathcal{D}$
    update $\psi$, $\theta$ with (4), (5) respectively
    $\bar{\psi} \leftarrow \tau\psi + (1 - \tau)\bar{\psi}$
end for
// online training phase with SAC.
load parameters $\theta, \psi, \bar{\psi}$ for SAC
initialize temperature $\alpha$ for SAC
for number of online iterations do
    // environmental step
    $a \sim \pi_\theta(a|s), r \sim R(s, a), s' \sim \mathbb{P}(s'|s, a)$
    $\mathcal{D} \leftarrow \mathcal{D} \bigcup \{(s, a, r, s')\}$
    // gradient step
    sample mini-batch from online buffer $\mathcal{D}$
    update $\psi$, $\theta$, $\alpha$ with (1), (2), (3) respectively
    $\bar{\psi} \leftarrow \tau\psi + (1 - \tau)\bar{\psi}$
end for
Output: network parameters $\psi, \theta$
```

At offline training time, the target value for Bellman backups of critic update in (1) uses actions a' sampled from the learned policy π_θ, while the Q function was trained only on actions produced by the offline data under the behavior policy (the expectation over $\mathcal{D}$ in (1)). As a result, the offline learned Q function typically over-estimates the value of $Q(s, a)$ for an OOD action a, i.e., when a is never applied at state s in the offline dataset. A similar issue also plagues the actor update in (2), where Q_ψ is evaluated on $a \sim \pi_\theta(s)$.

During online fine-tuning, the agent has a chance to update over-estimated OOD actions due to, for example, ϵ-greedy exploration and encountering OOD states. The bootstrap error resulting from over-estimation could wipe out the offline learned policy that previously performed well.

A Toy Example We now provide a toy bandit example in Figure 1 to further demonstrate our motivation. The key idea of this example is that uniformly lifting the Q-values can achieve both stealthiness and effectiveness, because a uniform over-estimation would not change the policy in the offline phase, as demonstrated in Figure 1a; and it will promote online distributional shift, as shown in Figure 1b.

While the toy example simply assumes that the Q-function can be directly manipulated to achieve a uniform over-estimation, this is however infeasible in a poisoning attack setting. In Section 4, we show that one could achieve it by formulating it as a bi-level optimization.



(a) Offline phase: Let Q, $\hat{Q}$ and $\tilde{Q}$ be the ground truth Q-function, the CQL approximation without being poisoned, and an uniformly poisoned Q-function, respectively. The bar plot shows the number of observed data for the corresponding action. It can be observed that $\hat{Q}$ well approximates in-distribution actions $\{-1, 0, 1\}$ and under-estimates OOD actions as expected. The poisoned $\tilde{Q}$ is stealthy as a uniform increase does not change the policy but breaks the conservatism which would lead to poor online performance.

(b) Online phase: Suppose we initialize an online agent with the offline poisoned $\tilde{Q}$. After some online interactions, which are clean as the poisoning is only applied to offline data, one could observe that the majority of interactions are in-distribution because they have higher $\tilde{Q}$-values at the beginning. However, by providing many clean data for in-distribution actions, their $\tilde{Q}$ estimations converge to ground truth. As a result, the ood actions become dominate due to higher $\tilde{Q}$ values because they were updated less frequently, hence promoting online distributional shift.

Figure 1: A toy bandit example, with Gaussian-like reward function and eleven actions, to demonstrate the intuition that maximizing Q-values (uniformly) can achieve both stealthiness and effectiveness.

4 THE ATTACK ALGORITHM

We investigate the vulnerability of O2O RL under data poisoning during offline training. Since the attacker is not allowed to perform any attack during the online fine-tuning phase, the victim will eventually recover from any offline attack given infinite online training resource. Thus, we set the attacker's goal to be such that the victim model, when fine-tuned online, suffers as much performance drop—both in magnitude and duration—at the *initial* phase as possible.

Algorithm 2 Update δ_r with IFT

```
Input: offline dataset $\mathcal{D} = \{(s, a, r, s')\}$, poison $\delta_r$, surrogate critic parameters $\psi$, step size $\eta$
$v_1 \leftarrow \frac{\partial \mathcal{L}_{\delta_r}}{\partial \psi}|_{\delta_r, \psi}$, where $\mathcal{L}_{\delta_r}$ is the outer objective in (6)
$v_2 \leftarrow \text{InverseHVP}(v_1, \frac{\partial \mathcal{L}_Q(\psi, \mathcal{D})}{\partial \psi})$ with $\mathcal{L}_Q$ from (4).
$v_3 \leftarrow \frac{\partial^2 \mathcal{L}_Q(\psi, \mathcal{D})}{\partial \delta_r \partial \psi} v_2$. In PyTorch, it can be implemented by $\mathtt{v_3} = \mathtt{grad}(\frac{\partial \mathcal{L}_{\mathtt{Q}}(\psi, \mathcal{D})}{\partial \delta_{\mathtt{r}}}, \psi, \mathtt{grad\_outputs} = \mathtt{v_2})$
Output: Updated $\delta_r = \delta_r + \eta v_3$ as (6) is maximization
```

4.1 THE THREAT MODEL

Following the standard poisoning attack protocol, we assume that the victim may not access clean demonstrations during offline training. Key to our threat model is the requirement that the **victim model must retain good "online performance" when offline training concludes**, because otherwise the attack would be detected and the model would be precluded from online fine-tuning. Here the "online performance" is evaluated by hypothetically applying the policy to an online environment, but without updating the policy (as opposed to online training). In reality, the agent may have a very limited budget to run such evaluations, for example, running it only once before launching into online training. However, given that offline policy evaluation is notorious for its high variance, we define the performance of offline training in this way, noting that the value of such evaluation is *not* used by either the agent's RL algorithm or the attacker's poisoning algorithm.

Although reward, state, and action are all feasible targets of poisoning on the offline batch data, we restrict our attention to reward because it is a single scalar and carries less structure than states and actions, hence allowing more stealthy poisoning. The attacker is not allowed to access the victim model, such as its policy network or value functions. Following the common practice such as Witche's Brew (Geiping et al., 2021) and continual input-aware poisoning (Kang et al., 2023), the attacker may internally train a *surrogate* RL agent and queries it to construct the poisons.

In addition to the aforementioned stealthiness constraints, we also impose the standard ℓ_p norm constraints on the reward perturbations. For example, the ℓ_0 norm constraints specifying how many offline transitions can be perturbed, and ℓ_1 norm constraints on the total or average amount of perturbation. For a vector $\mathbf{x}$, its ℓ_1 norm is $\|\mathbf{x}\|_1 := \sum_i |x_i|$, and its ℓ_∞ norm is $\|\mathbf{x}\|_\infty := \max_i |x_i|$.

4.2 THE POISONING ALGORITHM

Due to the stealthy requirement, the poisoning algorithms for offline RL such as Gong et al. (2022) cannot serve our purpose as it would lead to poor online performance for the offline trained model. Our inspiration originates from the distribution shift phenomenon, which shows that over-estimation of the Q-function will lead to poor online performance, while keeping the performance during offline training competitive. Thus, we seek to poison the reward by promoting the resulting Q values at OOD actions, hence maximally exacerbate the over-estimation problem.

Specifically, we first randomly sample $q\%$ offline transitions $\mathcal{C}^p := \{(s, a, r^p, s')\}$ as candidate transitions to be poisoned. Then we perturb the reward on these transitions to construct a poisoned buffer $\mathcal{D}^p = \{(s, a, r^p + \delta_r, s')\}$. Finally we combine it with the rest of clean transitions to construct the poisoned training set $\mathcal{D}^t := \mathcal{D}^p \cup (\mathcal{D} \setminus \mathcal{C}^p)$.

Let δ_r be a vector whose components correspond to the reward perturbation on each transition in $\mathcal{C}^p$. Then our poisoner conceptually solves the following constrained bi-level optimization for δ_r:

$$\max_{\delta_r} \quad \mathop{\mathbb{E}}_{s \sim \mathcal{D}} \mathop{\mathbb{E}}_{a \sim \mu} \underbrace{[Q_{\psi^*}(s, a)]}_{\text{over-estimation}} - \beta \underbrace{\mathcal{R}(Q_{\psi^*}, \mathcal{D}^t)}_{\text{extra stealthiness}} \tag{6}$$

$$\text{s.t.} \quad \|\delta_r\|_1 \,/\, |\mathcal{D}^p| \le \epsilon_1 \quad \text{and} \quad \|\delta_r\|_\infty \le \epsilon_\infty \tag{7}$$

$$\psi^* \leftarrow \mathtt{(surrogate)\text{-}victim\text{-}RL}(\mathcal{D}^t). \tag{8}$$

where μ is a distribution over $\mathcal{A}$, $\mathcal{R}$ is a critic regularizer, and `(surrogate)-victim-RL` is an offline RL algorithm, either the victim or a surrogate model. Ideally, we use uniform μ to promote uniform over-estimation for stealthiness. The regularizer $\mathcal{R}$ aims to further ensure stealthiness, as exact uniform over-estimation might not be always achievable, due to, e.g., optimization error or continuous action space.

Note in the first term of the outer objective, we do not require a to be from the offline data, i.e., it does not have to be what was taken at state s. This exactly serves our purpose of simulating OOD actions, and promoting their Q values. It is similar in spirit to the log-sum-exp term in (4). For Frozen Lake task, whose action space is discrete and finite, it is straightforward to apply uniform μ. While for locomotion tasks with bounded continuous space, the expectation over a can be efficiently approximated with samples.

For the choice of regularizer $\mathcal{R}$, it can be typically the constraints derived for offline RL algorithms, for example commonly used KL (Wu et al., 2019), uncertainty quantification (Bai et al., 2022), and CQL regularizer $\mathcal{R}_Q^{\text{CQL}}$, as its purpose is to improve offline performance (ensuring stealthiness) akin to offline RL regularizers. In practice, we use the CQL regularizer as it can be implemented for both discrete and continuous action spaces, respectively.

To summarize, our poisoner solves

$$\max_{\delta_r} \quad \mathop{\mathbb{E}}_{s \sim \mathcal{D}} \mathop{\mathbb{E}}_{a \sim \mathcal{U}(\mathcal{A})} [Q_{\psi^*}(s, a)] - \beta \mathcal{R}^{\text{CQL}}(Q_{\psi^*}, \mathcal{D}^t) \tag{9}$$

$$\text{s.t.} \quad (7) \text{ and } (8). \tag{10}$$

where $\mathcal{U}$ stands for uniform distribution. And the intuition behind $\mathcal{R}^{\text{CQL}}$ is to constrain the (poisoned) Q-functions from

Algorithm 3 O2OP: Poison Generation via Surrogate Model

```
Input: clean offline dataset $\mathcal{D} = \{(s, a, r, s')\}$
randomly pick a set of transitions $\mathcal{C}^p$ for poisoning
initialize surrogate CQL model parameters $\theta, \psi, \bar{\psi}$
initialize poisoned dataset $\mathcal{D}^p = \{(s, a, r^p + \delta_r, s')\}$
combine clean and poisoned dataset into the training
    dataset $\mathcal{D}^t = \mathcal{D}^p \cup \mathcal{D} \setminus \mathcal{C}^p$
for step = 1 ... number of offline steps do
    sample a mini-batch from $\mathcal{D}^t$
    // surrogate-victim-RL steps
    update $\psi, \bar{\psi}, \theta$ according to (4) and (5)
    // IFT steps for poison update
    if step mod IFT_freq. == 0 then
        // access to only surrogate model $\psi$
        update $\delta_r$ via Algorithm 2 using $\psi$
    end if
end for
// output poisoned $\mathcal{D}^t$ for subsequent victim training
Output: $\mathcal{D}^t$, which applies reward perturbation $\delta_r$
```

Algorithm 4 Baselines: `poison-uniform/wb`

```
Input: clean offline dataset $\mathcal{D} = \{(s, a, r, s')\}$
randomly pick a set of transitions $\mathcal{C}^p$ for poisoning
initialize victim model parameters $\theta, \psi, \bar{\psi}$
initialize poisoned dataset $\mathcal{D}^p = \{(s, a, r^p + \delta_r, s')\}$
// fixed perturbation for poison-uniform
if poison-uniform then $\delta_r \leftarrow \epsilon_1$ end if
obtain the training dataset $\mathcal{D}^t = \mathcal{D}^p \cup \mathcal{D} \setminus \mathcal{C}^p$
for step = 1 ... number of offline steps do
    sample a mini-batch from $\mathcal{D}^t$
    // victim-RL steps
    update $\psi, \bar{\psi}, \theta$ according to (4) and (5)
    // simultaneously poisoning for poison-wb
    if poison-wb and step mod IFT_freq. == 0 then
        // poison-wb accesses victim-RL
        update $\delta_r$ via Algorithm 2 using $\psi$
    end if
end for
```

deviating from the dataset actions, a common technique in offline RL. This is achieved by maximizing the Q values of the dataset actions with $\mathcal{R}^{\text{CQL}}$.

4.3 SOLVING THE BI-LEVEL OPTIMIZATION

To solve (9), a key quantity needed is the derivative of the outer objective with respect to δ_r, which in turn needs the derivative of Q_{ψ^*} with respect to δ_r. This is challenging because their dependence is through an offline RL algorithm. The fundamental mathematical solution is the implicit function theorem (IFT), based on which a number of techniques with improved computational and spatial complexity have been widely used in previous works on hyper-parameter tuning (Bengio, 2000; Maclaurin et al., 2015; Shaban et al., 2019; Lorraine et al., 2020). Here, we utilize these techniques in a similar way as described in Algorithm 2, where instead of tuning the hyper-parameter, we update δ_r. In particular, we follow Lorraine et al. (2020) and approximate the Inverse Hessian Vector Product (HVP) by using the Neumann approximation.

Equipped with the gradient in δ_r, we could simply perform gradient based updates such as ADAM. However, this is very expensive because IFT-style algorithms require solving the inner offline RL to the optimal. For computational efficiency, we only run offline RL for a few steps in each iteration, and use the suboptimal ψ to update δ_r via Algorithm 2. The entire procedure is summarized in Algorithm 3, illustrating how the attacker generates the poison δ_r, hence the poisoned dataset $\mathcal{D}^t$. And the victim algorithm, not necessarily has to be the same as the surrogate algorithm (CQL) will then be trained on $\mathcal{D}^t$. We will refer to it as **O2O poisoner (O2OP)**.

It is noteworthy that the attacker does not require accessing the victim agent's model, neither the policy nor the value functions. Instead, it trains its own surrogate agent based on which the poison is constructed. Surrogate models are quite commonly used (Geiping et al., 2021; Kang et al., 2023; Souri et al., 2022; Cherepanova et al., 2021; Goldblum et al., 2023), and its effectiveness is far from trivial because RL is well known for high variance. With different seed and different mini-batches sampled, the surrogate agent can be quite different from the real agent, making it nontrivial for the learned poison to remain effective.

5 EMPIRICAL EVALUATION

We now empirically verify that our proposed poisoner O2OP fulfills the aforementioned objectives. We tested on Frozen Lake (Brockman et al., 2016), Hopper, HalfCheetah, and Walker2d environments from the D4RL dataset (Fu et al., 2020). In this section, we use CQL for offline training, and SAC or DDQN for online fine-tuning in continuous or discrete tasks, respectively. Following the common protocol, we repeated experiments on each environment with 5 seeds, and then plotted the mean return from the 5 trials.

Baseline Comparators Since there is yet no existing algorithm addressing our task, we adopted a uniform poisoner which sets all δ_r to ϵ_1. To study the effectiveness of using surrogate models, we also compared with an attacker which has white-box access to the victim model. These two methods will be referred to as `poison-uniform` and `poison-wb`, respectively.

Environments Frozen Lake is a discrete text environment. The environment consists a 4-by-4 or 8-by-8 grid, with a goal state and several holes (terminal states). The agent receives a reward of 1 for reaching to goal state, and reward

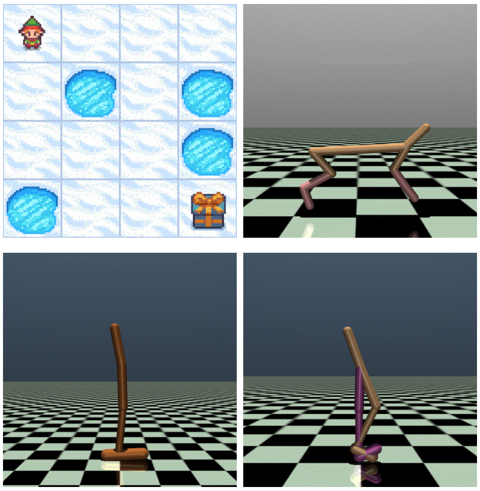


Figure 2: Visualizations of Frozen Lake, HalfCheetah, Hopper and Walker2d, respectively.*

of 0 for all other states. It should aim to reach the goal state without falling into a hole. Locomotion tasks are simulated robotics environments, where the rewards are measured by the forward travel distance while staying "stable". D4RL dataset contains a collection of different skill levels for each locomotion task, depending on the average return of behavior policy that collects the dataset. We use "medium" level dataset for our experiments. Figure 2 visualized a typical 4-by-4 Frozen Lake, as well as locomotion environments.

5.1 DISCRETE ENVIRONMENT: FROZEN LAKE

We trained an offline discrete CQL agent for 100 epochs, with 500 steps in each epoch. The online agent was trained for 50 epochs on clean online environment, with a buffer carried over from their offline phase. For this environment, we included all offline transitions $\mathcal{D}$ in our candidate set $\mathcal{D}^p$, and tested with $\epsilon_1 \in \{0.1, 0.02\}$ and $\epsilon_\infty = 1$ from (7). O2OP first generated δ_r from a surrogate model as described in Algorithm 3, and used it to poison a new victim which was trained by CQL with a different initialization and mini-batch sampling seed.

Figure 3 shows the average online return during the offline training (left) and online fine-tuning (right), both at $\epsilon = 0.1$. All poisoned victims perform similarly to the clean trained agent during the offline phase, fulfilling the stealthiness objective. However, our O2OP drove down the online return from 0.65 to 0.3, which is only slightly higher than that of the white-box poisoner (0.25). In contrast, the online return of the uniform baseline stayed above 0.45. We also aggregated the average returns over all offline or online steps by taking their mean. This is provided in the legend.

We further reduced our budget to $\epsilon_1 = 0.02$ in Figure 4.

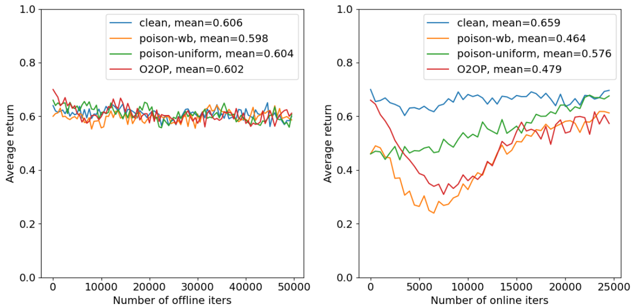


Figure 3: O2O return in offline phase (left) and online phase (right) for Frozen Lake with $\epsilon_1 = 0.1$

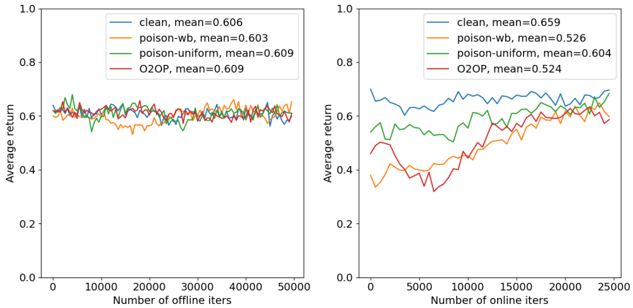


Figure 4: Frozen Lake with $\epsilon_1 = 0.02$

Here, the stealthiness remains satisfied offline. During online fine-tuning, the uniform poisoned victim agent has a minimum average return above 0.5, while our O2OP drives it below 0.35, which is almost the same as the white-box attacker. This confirms the effectiveness of our O2OP.

5.2 CONTINUOUS ENVIRONMENTS

We next move on to illustrate the attack effectiveness in a *continuous* space. The continuous CQL agents were trained for 600 epochs offline, with 500 gradient steps per epoch. The online continuous SAC agents were trained for additional 100 epochs. We reduced the poison ratio to 2% (i.e. $q = 2$) for more realistic attacks.

Hopper As the hopper-medium dataset has rewards ranging in $(0, 6)$, we increased our poison's ℓ_1 norm budget to $\epsilon_1 = 4$. To improve stealthiness, we enforced the constraint $\|\delta_r\|_\infty \leq \epsilon_\infty = 5$. Accordingly, the same choices were made on both baselines `poison-uniform/wb`. Despite the slightly high values of ϵ_1 and ϵ_∞, we only poison 2% of the transitions, which is consistent with poisoning or backdoor attacks in supervised learning.

Figure 5 shows that, analogously to Frozen Lake, all the three poisoners perform similarly to the clean unpoisoned case in terms of the offline performance, which again confirms the stealthiness of O2OP. During online fine-tuning,

[*] Figures borrowed from (Brockman et al., 2016).

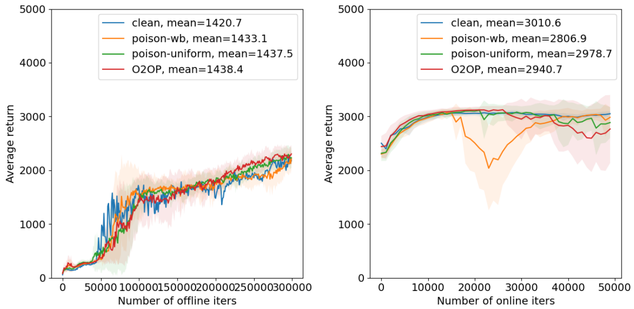


Figure 5: O2O return in offline phase (left) and online phase (right) for Hopper with 2% poison.

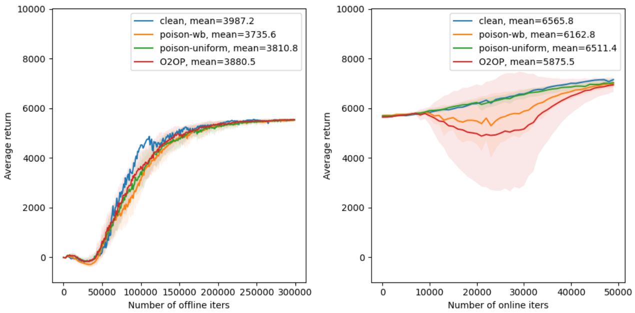


Figure 6: O2O return in offline phase (left) and online phase (right) for HalfCheetah with 2% poison.

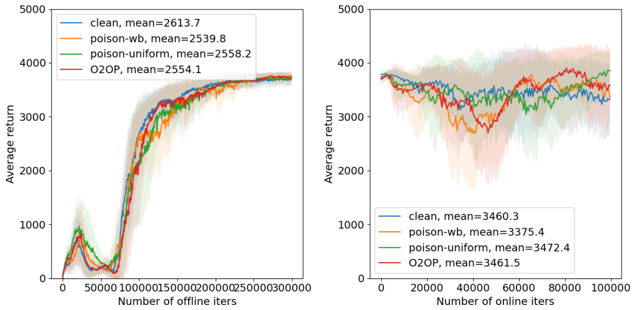


Figure 7: O2O return in offline phase (left) and online phase (right) for Walker2d with 2% poison.

however, O2OP achieves a performance drop from 3000 to 2600 (when online iteration is around 46000), while the white-box version can further slash it to 2000. In contrast, poison-uniform can hardly degrade the online return, if at all. This shows that O2OP remains effective in this continuous space with a small poison ratio.

HalfCheetah The reward in halfcheetah-medium lies between -3 and 9, with the mean around 5. We again only poisoned 2% transitions, and set $\epsilon_1 = 4$ and $\epsilon_\infty = 5$. Similarly to Hopper, Figure 6 shows our O2OP effectively created a return drop during the online fine-tuning, while poison-uniform is again nearly harmless to the victim at the same ratio and budget. The offline stealthiness is evidenced once more as the four methods achieve similar offline returns.

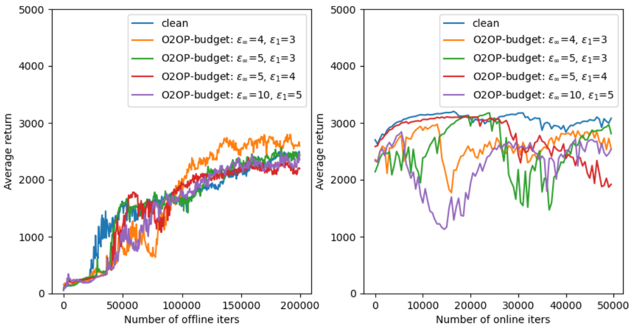


Figure 8: O2O return in offline phase (left) and online phase (right) on Hopper with varying ϵ_1 and ϵ_∞ budgets.

Walker2d The walker2d-medium dataset has similar reward range as halfcheetah-medium, and we thus used identical settings to it. As shown in Figure 7, the poisoned offline return remains comparable to the clean offline return, i.e., stealthy. Although the online return seems less stable than in the previous experiments, O2OP managed to curtail the return from 3800 to nearly 2500 at its lowest, while the clean and poison-uniform baselines produce returns fluctuating between 3200 and 4000.

6 ABLATION STUDIES

We further experiment with our method using different ℓ_p budgets, alternative victim algorithms, different model architectures, and under defense strategies.

6.1 IMPACT OF ℓ_p BUDGET

We also tested with different budget of ϵ_1 and ϵ_∞ on Hopper. As Figure 8 shows, different budgets do not affect the offline return too much. On the other hand, the amount of online performance drop does vary significantly with the budgets. In general, a larger poison budget leads to a greater drop.

6.2 ALTERNATIVE CHOICE OF VICTIMS

In addition, we will apply our attack to different choices of offline victims. Specifically, we consider two critic-regularized offline algorithms: BRAC (Wu et al., 2019) and PBRL (Bai et al., 2022), where one regularizes the critic updates to avoid over-estimation akin to CQL. The corresponding updates of BRAC and PBRL are listed below:

$$\begin{aligned}\mathcal{L}_Q^{\mathrm{BRAC}}(\psi,\mathcal{D}) &:= \mathop{\mathbb{E}}_{(s,a,r,s')\sim\mathcal{D}}\Big[\big(Q_\psi(s,a)-y(r,s')\big)^2\Big] \\ y(r,s') &:= r+\gamma \mathop{\mathbb{E}}_{a'\sim\pi_\theta(s')}[Q_{\bar\psi}(s',a')-\alpha D_{s'}(\pi_\theta|\pi_b)]\end{aligned} \tag{11}$$

$$\mathcal{L}_\pi^{\mathrm{BRAC}}(\theta,\mathcal{D}) := \mathop{\mathbb{E}}_{s\sim\mathcal{D}}\mathop{\mathbb{E}}_{a\sim\pi_\theta(s)}[\alpha D_{s'}(\pi_\theta|\pi_b)-Q_\psi(s,a)] \tag{12}$$

$$\begin{aligned}\mathcal{L}_Q^{\mathrm{PBRL}}(\psi,\mathcal{D}) &:= \mathop{\mathbb{E}}_{(s,a,r,s')\sim\mathcal{D}}\Big[\big(Q_\psi(s,a)-y(r,s')\big)^2\Big] \\ &+ \mathop{\mathbb{E}}_{s\sim\mathcal{D}}\mathop{\mathbb{E}}_{a\sim\pi_\theta}\big[(Q_{\bar\psi}(s,a)-\alpha\mathcal{E}_{\bar\psi}(s,a)-Q_\psi(s,a))^2\big]\end{aligned} \tag{13}$$

$$y(r,s') := r+\gamma \mathop{\mathbb{E}}_{a'\sim\pi_\theta(s')}[Q_{\bar\psi}(s',a')-\alpha\mathcal{E}_{\bar\psi}(s',a')]$$

$$\mathcal{L}_\pi^{\mathrm{PBRL}}(\theta,\mathcal{D}) := -\mathop{\mathbb{E}}_{s\sim\mathcal{D}}\mathop{\mathbb{E}}_{a\sim\pi_\theta(s)}[Q_\psi(s,a)] \tag{14}$$

where $D_s(\pi_\theta|\pi_b) := D(\pi_\theta(\cdot|s)|\pi_b(\cdot|s))$ is a (sample-based approximation of) divergence between the learned policy π_θ and a reference/behavior policy π_b (optionally learned by behavior cloning); and $\mathcal{E}_{\bar\psi}(s,a) := \mathrm{std}(Q_{\bar\psi}^{(i)}(s,a))$ is an uncertainty quantification using ensembled Q-functions.

We conducted additional experiments with BRAC+DQN and PBRL+DQN in Frozen Lake to validate the effectiveness beyond CQL as victim (where BRAC or PBRL is used for both victim and surrogate). Figure 9 shows the proposed attack remain effective for different O2O RL choices.

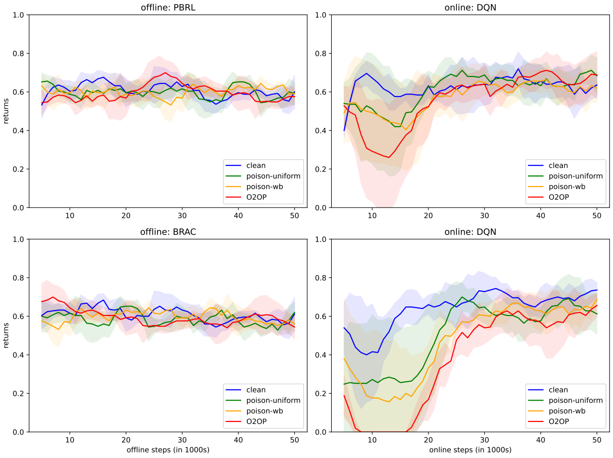


Figure 9: Ablation on different victim algorithms with Frozen Lake: we in addition test BRAC and PBRL as offline victim algorithms.

6.3 ALTERNATIVE CHOICE OF SURROGATE

To further test O2OP's effectiveness when the surrogate and victim models are different, we now use BRAC and PBRL as surrogate models, and keep CQL as the victim. Figure 10 shows that our O2OP remains effective with different surrogate models.

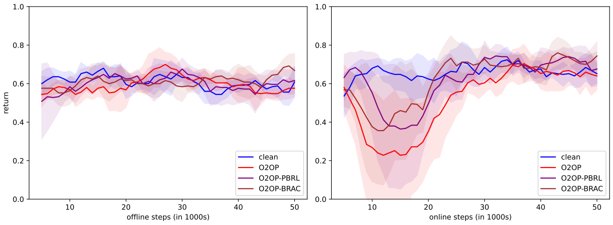


Figure 10: Ablation on different surrogate models with Frozen Lake: CQL remains the offline victim algorithm, but BRAC and PBRL are used as surrogate to learn δ_r.

6.4 IMPACT OF NETWORK ARCHITECTURE

To further demonstrate the effectiveness of O2OP when the surrogate and victim models have different network architectures, we use the same victim architecture (two hidden layers of size 256 each) for the clean, O2OP-same-network, and O2OP-different-network experiments. O2OP-same-network means the surrogate model has the same architecture as the victim model, while O2OP-different-network uses a network with layers of sizes $\{32, 64, 128\}$ to generate δ_r. Figure 11 demonstrates that O2OP remains effective even with different surrogate model architectures.

6.5 ASSESSING O2OP UNDER DEFENSE

We next study how well our attack remains effective in the face of defense algorithms. To this end, we added two simple defense strategies: (i) using a single-class SVM, an unsupervised outlier detection method, to filter and remedy the data; (ii) a uniform decrease, by the mean of the learned δ_r, to all poisoned rewards. Note that the second defender is a "strong" one in the sense that it leverages knowledge (the mean of δ_r) that is not typically available to defenders. Nonetheless, we observe that these defenses were not effective as shown in Figure 12.

7 FURTHER DETAILS

Regularizer $\mathcal{R}^{\mathrm{CQL}}$ For continuous action space, we follow the implementation of d3rlpy (Seno & Imai, 2022). For discrete action space, we first observe that $\mathcal{R}^{\mathrm{CQL}}$ is equivalent to a cross-entropy loss (or negative log-likelihood):

$$\mathcal{R}^{\mathrm{CQL}}(Q,\mathcal{D}) := \mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}}[\log\textstyle\sum_u \exp(Q(s,u))-Q(s,a)] \tag{15}$$

$$= -\mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}}\left[\log\frac{\exp Q(s,a)}{\sum_u \exp Q(s,u)}\right] = -\mathop{\mathbb{E}}_{(s,a)\sim\mathcal{D}}[\log\pi_Q(a|s)] \tag{16}$$

We then use label smoothing with $\epsilon = 0.1$ for a smoother regularization, as different actions a may present in the same

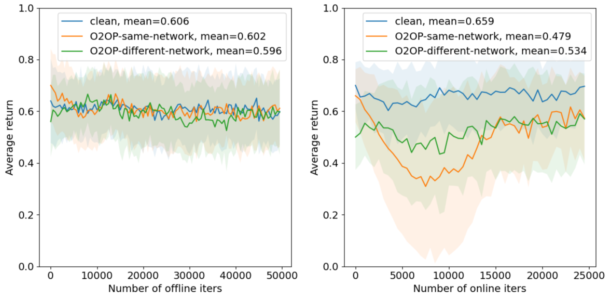


Figure 11: O2O return in offline phase (left) and online phase (right) for Frozen Lake when surrogate model having different network architectures.

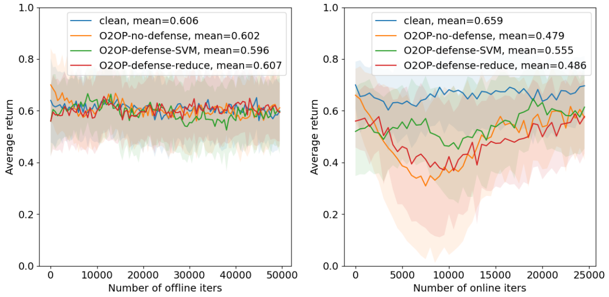


Figure 12: O2O return in offline phase (left) and online phase (right) for Frozen Lake under potential defense.

state s, unlike in standard classification problems.

Offline Dataset Collection D4RL does not have a dataset for Frozen Lake. Instead, we collect an offline dataset ourselves by following a collection procedure similar to prior offline RL works (Kumar et al., 2019; Wu et al., 2019). We first train a near-optimal policy through online interaction and then use this policy to collect a certain number of trajectories in the environment. The collected dataset has 5 trajectories with 195 transitions and an average return of 1.

IFT Optimizer We use a community implementation of the IFT optimizer[*] for our bi-level optimization.

8 CONCLUSION

Summary We proposed a novel reward poisoning method that reveals the vulnerability of O2O RL fine-tuning under a novel stealthiness notion—impact occurs only during online fine-tuning while the offline RL performance remains intact. Our approach leverages the distribution shift phenomenon during O2O transfer by promoting Q-function over-estimation for out-of-distribution actions through a bi-level optimization performed with the application of the implicit function theorem.

Limitation Our work only tested critic-regularized offline RL methods—CQL, BRAC, and PBRL—as our method is motivated by the over-estimated Q-function to make those critic regularizations less effective. It remains unclear whether such vulnerability exists in other categories of O2O algorithms, such as actor regularization (Nair et al., 2020), replay distribution correction (Lee et al., 2022), or policy ensemble (Zhang et al., 2023; Wang et al., 2023).

Future Direction To further extend our understanding of O2O RL, it is important to study the aforementioned non-critic-regularized methods, as each of these categories may present unique vulnerabilities and characteristics that differ from critic-regularized methods.

Additionally, exploring effective defense is vital for a robust O2O training pipeline. Future research could focus on developing resilient learning algorithms and enhancing data sanitization techniques to detect and remove perturbed data.

Societal Impact Our work focuses on understanding of the vulnerability of RL algorithms, particularly in the context of O2O transfer. While we introduce a novel reward poisoning method to study vulnerabilities in RL fine-tuning, it is important to highlight that our research is conducted strictly within a controlled experimental setting and is intended purely for academic and scientific purposes.

The environments we use, Frozen Lake and MuJoCo locomotion tasks, are toy-level simulations. These simplified scenarios ensure that our research remains theoretical and cannot be misused by third parties to cause real-world harm. Our intention is to identify weaknesses in RL systems to help develop more resilient and secure algorithms.

By exposing and analyzing these vulnerabilities, we aim to contribute to the broader field of RL safety and robustness, ultimately leading to stronger and more reliable RL fine-tuning models. This, in turn, can enhance the safety and performance of RL applications in various domains.

Our work does not support or encourage the malicious use of reward poisoning techniques. Instead, our findings are intended to serve as a foundation for developing effective defense strategies against such attacks. By sharing our insights with the research community, we hope to foster a collaborative effort towards mitigating the risks associated with adversarial attacks in RL.

Overall, our work is designed to advance the field of RL in a positive and constructive manner, with the ultimate aim of creating safer and more robust RL systems that can benefit society as a whole.

Acknowledgement We thank the reviewers and the meta-reviewer for assessing our paper and for their constructive feedback. This work is supported by NSF grant RI:1910146 and NIH grant R01CA258827.

[*] Available at here.

REFERENCES

Bai, C., Wang, L., Yang, Z., Deng, Z., Garg, A., Liu, P., and Wang, Z. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=Y4cs1Z3HnqL`.

Banihashem, K., Singla, A., Gan, J., and Radanovic, G. Admissible policy teaching through reward design. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6037–6045, 2022.

Banihashem, K., Singla, A., and Radanovic, G. Defense against reward poisoning attacks in reinforcement learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=goPsLn3RVo`.

Bengio, Y. Gradient-based optimization of hyperparameters. *Neural Comput.*, 12(8):1889–1900, 2000. doi: 10.1162/089976600300015187. URL `https://doi.org/10.1162/089976600300015187`.

Biggio, B., Nelson, B., and Laskov, P. Poisoning attacks against support vector machines. ICML'12, pp. 1467–1474, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851. URL `https://icml.cc/2012/papers/880.pdf`.

Bogunovic, I., Losalka, A., Krause, A., and Scarlett, J. Stochastic linear bandits robust to adversarial attacks. In Banerjee, A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 991–999. PMLR, 13–15 Apr 2021. URL `https://proceedings.mlr.press/v130/bogunovic21a.html`.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016. URL `https://arxiv.org/abs/1606.01540`.

Chen, D. and Wen, Y. Dcac: Reducing unnecessary conservatism in offline-to-online reinforcement learning. In *Proceedings of the Fifth International Conference on Distributed Artificial Intelligence*, DAI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400708480. doi: 10.1145/3627676.3627677. URL `https://doi.org/10.1145/3627676.3627677`.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. Targeted backdoor attacks on deep learning systems using data poisoning. *ArXiv*, abs/1712.05526, 2017. URL `https://arxiv.org/abs/1712.05526`.

Cherepanova, V., Goldblum, M., Foley, H., Duan, S., Dickerson, J. P., Taylor, G., and Goldstein, T. Lowkey: Leveraging adversarial attacks to protect social media users from facial recognition. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=hJmtwocEqzc`.

Farahmand, A.-m., Szepesvári, C., and Munos, R. Error propagation for approximate policy and value iteration. In Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. URL `https://proceedings.neurips.cc/paper_files/paper/2010/file/65cc2c8205a05d7379fa3a6386f710e1-Paper.pdf`.

Fu, J., Kumar, A., Soh, M., and Levine, S. Diagnosing bottlenecks in deep q-learning algorithms. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2021–2030. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/fu19a.html`.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020. URL `https://arxiv.org/abs/2004.07219`.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/fujimoto19a.html`.

Garcelon, E., Roziere, B., Meunier, L., Tarbouriech, J., Teytaud, O., Lazaric, A., and Pirotta, M. Adversarial attacks on linear contextual bandits. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 14362–14373. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/a554f89dd61cabd2ff833d3468e2008a-Paper.pdf`.

Geiping, J., Fowl, L. H., Huang, W. R., Czaja, W., Taylor, G., Moeller, M., and Goldstein, T. Witches' brew: Industrial scale data poisoning via gradient matching. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=01olnfLIbD`.

Goldblum, M., Tsipras, D., Xie, C., Chen, X., Schwarzschild, A., Song, D., Madry, A., Li, B., and Goldstein, T. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(2):1563–1580, 2023. doi: 10.1109/TPAMI.2022.3162397. URL https://doi.org/10.1109/TPAMI.2022.3162397.

Gong, C., Yang, Z., Bai, Y., He, J., Shi, J., Sinha, A., Xu, B., Hou, X., Fan, G., and Lo, D. Mind your data! hiding backdoors in offline reinforcement learning datasets. *CoRR*, abs/2210.04688, 2022. doi: 10.48550/ARXIV.2210.04688. URL https://doi.org/10.48550/arXiv.2210.04688.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

Guan, Z., Ji, K., Bucci Jr, D. J., Hu, T. Y., Palombo, J., Liston, M., and Liang, Y. Robust stochastic bandit algorithms under probabilistic unbounded adversarial attack. In *Proceedings of the aaai conference on artificial intelligence*, volume 34, pp. 4036–4043, 2020.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018. URL https://arxiv.org/abs/1812.05905.

Hasselt, H. v., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. AAAI'16, pp. 2094–2100. AAAI Press, 2016.

Huang, Y. and Zhu, Q. Deceptive reinforcement learning under adversarial manipulations on cost signals. In Alpcan, T., Vorobeychik, Y., Baras, J. S., and Dán, G. (eds.), *Decision and Game Theory for Security - 10th International Conference, GameSec 2019, Stockholm, Sweden, October 30 - November 1, 2019, Proceedings*, volume 11836 of *Lecture Notes in Computer Science*, pp. 217–237. Springer, 2019. doi: 10.1007/978-3-030-32430-8_14. URL https://doi.org/10.1007/978-3-030-32430-8_14.

Jun, K.-S., Li, L., Ma, Y., and Zhu, J. Adversarial attacks on stochastic bandits. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/85f007f8c50dd25f5a45fca73cad64bd-Paper.pdf.

Kang, S., Shi, Z., and Zhang, X. Poisoning generative replay in continual learning to promote forgetting. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 15769–15785. PMLR, 2023. URL https://proceedings.mlr.press/v202/kang23c.html.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/c2073ffa77b5357a498057413bb09d3a-Paper.pdf.

Kumar, A., Gupta, A., and Levine, S. Discor: Corrective feedback in reinforcement learning via distribution correction. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18560–18572. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/d7f426ccbc6db7e235c57958c21c5dfa-Paper.pdf.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020b. URL https://proceedings.neurips.cc/paper/2020/hash/0d2b2061826a5df3221116a5085a6052-Abstract.html.

Lee, S., Seo, Y., Lee, K., Abbeel, P., and Shin, J. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In Faust, A., Hsu, D., and Neumann, G. (eds.), *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1702–1712. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/lee22d.html.

Lei, K., He, Z., Lu, C., Hu, K., Gao, Y., and Xu, H. Uni-o4: Unifying online and offline deep reinforcement learning with multi-step on-policy optimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=tbFBh3LMKi.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020. URL https://arxiv.org/abs/2005.01643.

Liu, F. and Shroff, N. Data poisoning attacks on stochastic bandits. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4042–4050. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/liu19e.html.

Liu, G. and Lai, L. Efficient adversarial attacks on online multi-agent reinforcement learning. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 24401–24433. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/4cddc8fc57039f8fe44e23aba1e4df40-Paper-Conference.pdf.

Lorraine, J., Vicol, P., and Duvenaud, D. Optimizing millions of hyperparameters by implicit differentiation. In Chiappa, S. and Calandra, R. (eds.), *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1540–1552. PMLR, 2020. URL http://proceedings.mlr.press/v108/lorraine20a.html.

Lu, S., Wang, G., and Zhang, L. Stochastic graphical bandits with adversarial corruptions. In *Proceedings of the aaai conference on artificial intelligence*, volume 35, pp. 8749–8757, 2021.

Lykouris, T., Simchowitz, M., Slivkins, A., and Sun, W. Corruption-robust exploration in episodic reinforcement learning. In Belkin, M. and Kpotufe, S. (eds.), *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pp. 3242–3245. PMLR, 15–19 Aug 2021. URL https://proceedings.mlr.press/v134/lykouris21a.html.

Ma, Y., Jun, K.-S., Li, L., and Zhu, X. Data poisoning attacks in contextual bandits. In *Decision and Game Theory for Security: 9th International Conference, GameSec 2018, Seattle, WA, USA, October 29–31, 2018, Proceedings 9*, pp. 186–204. Springer, 2018.

Ma, Y., Zhang, X., Sun, W., and Zhu, J. Policy poisoning in batch reinforcement learning and control. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 14543–14553, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/315f006f691ef2e689125614ea22cc61-Abstract.html.

Maclaurin, D., Duvenaud, D., and Adams, R. P. Gradient-based hyperparameter optimization through reversible learning. In Bach, F. R. and Blei, D. M. (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 2113–2122. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/maclaurin15.html.

Mark, M. S., Sharma, A., Tajwar, F., Rafailov, R., Levine, S., and Finn, C. Offline retraining for online rl: Decoupled policy learning to mitigate exploration bias. *arXiv preprint arXiv:2310.08558*, 2023. URL https://arxiv.org/abs/2310.08558.

Munos, R. Error bounds for approximate value iteration. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, AAAI'05, pp. 1006–1011. AAAI Press, 2005. ISBN 157735236x. URL https://aaai.org/papers/01006-aaai05-159-error-bounds-for-approximate-value-iteration/.

Nair, A., Gupta, A., Dalal, M., and Levine, S. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020. URL https://arxiv.org/abs/2006.09359.

Nakamoto, M., Zhai, Y., Singh, A., Mark, M. S., Ma, Y., Finn, C., Kumar, A., and Levine, S. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. 2023. URL https://arxiv.org/abs/2303.05479.

Rakhsha, A., Radanovic, G., Devidze, R., Zhu, X., and Singla, A. Policy teaching in reinforcement learning via environment poisoning attacks. *J. Mach. Learn. Res.*, 22:210:1–210:45, 2021a. URL http://jmlr.org/papers/v22/20-1329.html.

Rakhsha, A., Zhang, X., Zhu, X., and Singla, A. Reward poisoning in reinforcement learning: Attacks against unknown learners in unknown environments. *CoRR*, abs/2102.08492, 2021b. URL https://arxiv.org/abs/2102.08492.

Rangi, A., Tran-Thanh, L., Xu, H., and Franceschetti, M. Saving stochastic bandits from poisoning attacks via limited data verification. In *Proceedings of the aaai conference on artificial intelligence*, volume 36, pp. 8054–8061, 2022a.

Rangi, A., Xu, H., Tran-Thanh, L., and Franceschetti, M. Understanding the limits of poisoning attacks in episodic reinforcement learning. In Raedt, L. D. (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 3394–3400. International Joint Conferences on Artificial Intelligence Organization, 7 2022b. doi: 10.24963/ijcai.2022/471. URL https://doi.org/10.24963/ijcai.2022/471. Main Track.

Seno, T. and Imai, M. d3rlpy: An offline deep reinforcement learning library. *Journal of Machine Learning Research*, 23(315):1–20, 2022. URL http://jmlr.org/papers/v23/22-0017.html.

Shaban, A., Cheng, C., Hatch, N., and Boots, B. Truncated back-propagation for bilevel optimization. In Chaudhuri, K. and Sugiyama, M. (eds.), *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1723–1732. PMLR, 2019. URL http://proceedings.mlr.press/v89/shaban19a.html.

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. Poison frogs! targeted clean-label poisoning attacks on neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/22722a343513ed45f14905eb07621686-Paper.pdf.

Souri, H., Fowl, L., Chellappa, R., Goldblum, M., and Goldstein, T. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/79eec295a3cd5785e18c61383e7c996b-Abstract-Conference.html.

Sun, Y., Huo, D., and Huang, F. Vulnerability-aware poisoning mechanism for online {rl} with unknown dynamics. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=9r30XCjf5Dt.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Wagenmaker, A. and Pacchiano, A. Leveraging offline data in online reinforcement learning. ICML'23. JMLR.org, 2023.

Wang, S., Yang, Q., Gao, J., Lin, M., CHEN, H., Wu, L., Jia, N., Song, S., and Huang, G. Train once, get a family: State-adaptive balances for offline-to-online reinforcement learning. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 47081–47104. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9318763d049edf9a1f2779b2a59911d3-Paper-Conference.pdf.

Wu, Y., Tucker, G., and Nachum, O. Behavior regularized offline reinforcement learning. *CoRR*, abs/1911.11361, 2019. URL http://arxiv.org/abs/1911.11361.

Wu, Y., McMahan, J., Zhu, X., and Xie, Q. Reward poisoning attacks on offline multi-agent reinforcement learning. In Williams, B., Chen, Y., and Neville, J. (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 10426–10434. AAAI Press, 2023. doi: 10.1609/AAAI.V37I9.26240. URL https://doi.org/10.1609/aaai.v37i9.26240.

Xie, T., Jiang, N., Wang, H., Xiong, C., and Bai, Y. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 27395–27407. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/e61eaa38aed621dd776d0e67cfeee366-Paper.pdf.

yang, l., Hajiesmaili, M., Talebi, M. S., Lui, J. C. S., and Wong, W. S. Adversarial bandits with corruptions: Regret lower bound and no-regret algorithm. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19943–19952. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e655c7716a4b3ea67f48c6322fc42ed6-Paper.pdf.

Yu, Z. and Zhang, X. Actor-critic alignment for offline-to-online reinforcement learning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 40452–40474. PMLR, 2023. URL https://proceedings.mlr.press/v202/yu23k.html.

Zhang, H. and Parkes, D. C. Value-based policy teaching with active indirect elicitation. In *AAAI*, volume 8, pp. 208–214, 2008.

Zhang, H., Parkes, D. C., and Chen, Y. Policy teaching through reward function learning. In *Proceedings of the 10th ACM conference on Electronic commerce*, pp. 295–304, 2009.

Zhang, H., Xu, W., and Yu, H. Policy expansion for bridging offline-to-online reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=-Y34L45JR6z.

Zhang, X., Ma, Y., Singla, A., and Zhu, X. Adaptive reward-poisoning attacks against reinforcement learning. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11225–11234. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/zhang20u.html.

Zuo, S. Near optimal adversarial attack on ucb bandits. *arXiv preprint arXiv:2008.09312*, 2020. URL https://arxiv.org/abs/2008.09312.