Data Poisoning Attacks on Off-Policy Policy Evaluation Methods

Abstract

Off-policy Evaluation (OPE) methods are a crucial tool for evaluating policies in high-stakes domains such as healthcare, where exploration is often infeasible, unethical, or expensive. However, the extent to which such methods can be trusted under adversarial threats to data quality is largely unexplored. In this work, we make the first attempt at investigating the sensitivity of OPE methods to marginal adversarial perturbations to the data. We design a generic data poisoning attack framework leveraging influence functions from robust statistics to carefully construct perturbations that maximize error in the policy value estimates. We carry out extensive experimentation with multiple healthcare and control datasets. Our results demonstrate that many existing OPE methods are highly prone to generating value estimates with large errors when subject to data poisoning attacks, even for small adversarial perturbations. To combat this problem, we suggest ways to identify and improve the robustness of OPE methods.

1 INTRODUCTION

In reinforcement learning (RL), off-policy evaluation (OPE) methods are popularly used to estimate the value of a policy from previously collected data [Thomas et al., 2015, Voloshin et al., 2020, Levine et al., 2020]. These methods are instrumental in high-stakes decision problems such as in medicine and finance, where deploying a policy directly is often infeasible, unethical, or expensive [Gottesman et al., 2020, Ernst et al., 2006]. In such cases, one must estimate the value solely from a batch of data collected using a different and possibly unknown policy. Only if the OPE methods estimate the value of a policy to be sufficiently high, the stakeholders will deploy it. Otherwise, the policy will be rejected. It is therefore essential that OPE methods do not severely overestimate the values of bad policies or underestimate the values of good policies [Gottesman et al., 2020].

Despite the importance of OPE methods, their sensitivity to adversarial contamination of logged data is not well understood. The complexity of OPE methods offers ample opportunities for attackers to introduce significant errors in OPE estimates with only small perturbations to the input data. For example, some OPE methods compute the value of a policy in a given state as a function of its value in future states. Therefore, even small errors introduced in the value estimates of these future states can accumulate and result in significant errors in the value estimates at the initial states, where critical strategic decisions are often made. This property could be exploited by attackers. Another possible avenue for an attack is the *importance sampling weights*. Popular OPE methods, such as the Doubly Robust and the Importance Sampling methods [Jiang and Li, 2016, Voloshin et al., 2020] use importance sampling weights to correct for dataset mismatch when evaluating the given policy with logged data from a different policy. The weights depend on the estimate of the policy used for the logged data. Attackers could perturb the data in a way that forces the agent to wrongly estimate the logging policy, and consequently introduce significant errors in the value estimates. Such vulnerabilities motivate the need for a thorough analysis of the effect of data poisoning attacks on OPE methods.

While some prior works have studied adversarial attacks in the context of policy learning [Rakhsha et al., 2020, Ma et al., 2019, Chen et al., 2019] in online and batch RL settings, they mainly focus on teaching an agent to learn an adversarial policy or driving the agent to an adversarial state [Rakhsha et al., 2020, Zhang et al., 2021, Rakhsha et al., 2020], and do not specifically investigate the effect of these attacks on OPE methods. In this work, we address the aforementioned gaps and study the effect of data poisoning attacks on OPE methods. More specifically, our work answers the following question: *Can we construct small perturbations to the training data that significantly change*

Submitted to the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022). **To be used for reviewing only**.

a given OPE method's estimate of the value of a given policy? To this end, we propose a novel data poisoning framework to analyze the sensitivity of model-free OPE methods to adversarial data contamination at train-time. We formulate the data poisoning problem as a bi-level optimization problem and show that it can be adapted to diverse model-free OPE methods, namely, Bellman Residual Minimization (BRM) [Farahmand et al., 2008], Weighted Importance Sampling (WIS), Weighted Per-Decision Importance Sampling (PDIS) [Precup, 2000, Powell and Swann, 1966, Rubinstein, 1981], Consistent Per-Decision Importance Sampling (CPDIS) [Thomas, 2015], and Weighted Doubly Robust methods (WDR) [Jiang and Li, 2016]. To solve the aforementioned bi-level optimization problem in a computationally tractable manner, we derive an approximate solution using influence functions from robust statistics [Koh et al., 2018, Koh and Liang, 2017, Diakonikolas and Kane, 2019, Broderick et al., 2021]. To the best of our knowledge, our work is the first to study the sensitivity of a wide range of OPE methods to train-time data poisoning attacks.

We evaluate our framework using five different datasets spanning medical (e.g., Cancer and HIV) and control (e.g., Mountain Car, Cartpole, and Continuous Gridworld) domains. Our experiments show that corrupting only 3%–5% of the observed states achieves more than 101% and 276% error in the estimate of the value function of the optimal policy in the Cancer and HIV domains, respectively. Through our experimental results, we show that out of the five OPE methods, WDR and BRM are the least statistically robust, and CPDIS is the most statistically robust to such adversarial contamination. Finally, our findings question the reliability of policy values derived using OPE methods and motivate the need for developing OPE methods that are statistically robust to train-time data poisoning attacks.

2 PRELIMINARIES

We model a sequential decision-making problem as a Markov Decision Process (MDP). An MDP is a tuple of the form $\langle \mathcal{S}, \mathcal{A}, R, P, p_0, \gamma \rangle$ representing the set of states, set of actions, reward function, transition probability model, initial state distribution, and discount factor respectively. When taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ and transitioning to state $s' \in \mathcal{S}$, the scalar $R(s, a, s')$ denotes the reward received by the agent and $P(s, a, s')$ denotes the probability of transitioning to state s' on taking action a in state s.

A randomized policy $\pi : \mathcal{S} \to \Delta^{|\mathcal{A}|}$ prescribes the probability of taking each action from $\mathcal{A}$ in a state s. The value function of a policy $v^\pi \colon \mathcal{S} \to \mathbb{R}$ at state s is the expected discounted returns of the policy starting from state s and is given by

$$v^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(S_t, A_t, S'_{t+1}) \mid \pi, S_0 = s\right] .$$

The value of a policy is computed as $p_0^T v^\pi$. The state-action value function (also termed as the Q-value function) of a policy $q^\pi \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ at state s and action a is the expected discounted returns obtained by taking action a in state s and following policy π thereafter. The state-action value function q^π for a policy π is the unique fixed point of the *Bellman operator* $\mathcal{T}^\pi \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ defined as

$$(\mathcal{T}^\pi q)(s, a) = \tag{1}$$
$$\sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}} \left(R(s, a, s') + \gamma P(s, a, s')\pi(a' \mid s')q(s', a')\right) .$$

We assume the standard batch RL setting (e.g. [Levine et al., 2020]) in which the agent is given a batch of $n{=}N \times T$ transition tuples $D{=}((s_j^i, a_j^i, r_j^i)_{j=1}^T)_{i=1}^N$, observed on simulating a behavior policy π_b for N episodes of length T. The *goal* of OPE is to use D to evaluate the value of the evaluation policy π. Let D_0 be a set of initial states sampled from distribution p_0.

The value function is approximated using features $\xi \colon \mathcal{S} \to \mathbb{R}^d$. As is standard in linear value function approximation, we assume also that the state-action value function q^π is approximated as a linear combination of state-action features $\phi \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{|\mathcal{A}|\cdot d}$. The state-action features for a given state-action pair (s, a) are constructed by using the state features $\xi(s)$ at the indices corresponding to a and zero elsewhere, i.e. $\phi(s, a)[ad : (a{+}1)d] \leftarrow \xi(s)$. Because the value function is estimated from data, we need to define a sample feature matrix $\Phi \in \mathbb{R}^{n\times d}$ where the rows correspond to the state-action features $\phi(s, a)$ for the n state-action pairs in D. Similarly, $\Phi_p \in \mathbb{R}^{n\times d}$ denotes the sample feature matrix for the *next states* such that each row corresponds to $\phi(s'_i, \pi_e(s'_i))$ for the next states s'_i in D. We will use $r \in \mathbb{R}^{n\times 1}$ to represent the sample reward matrix.

OPE methods are broadly classified into three categories: Direct, Importance Sampling, and Hybrid Methods [Voloshin et al., 2020]. *Direct Methods* estimate the value of the evaluation policy by solving for the fixed point of the Bellman Equation (1) with an assumed model for the state-action value function q or the transition model P. We illustrate our attack on one of the most popular Direct Methods, namely the *Bellman Residual Minimization* (BRM) method [Voloshin et al., 2020, Farahmand et al., 2008]. This method solves a sequence of supervised learning problems with state-action features $\phi(s, a)$ as the predictor and the 1-step Bellman update $\mathcal{T}^\pi q = r + \gamma P q$ as the target response. $\mathcal{T}^\pi : \mathbb{R}^S \to \mathbb{R}^S$ is commonly referred to as the Bellman operator. The objective optimized in BRM is the Mean Squared Bellman residual (MSBR), defined as a weighted L_2 norm:

$$\mathrm{MSBR}(\eta) \;=\; \|q_\eta - \mathcal{T}^\pi q_\eta\|_W^2 \; . \tag{2}$$

Here, the linear Q-value function q_η is parameterized by η, i.e., $q = \Phi\eta$. Weight matrix $W = \mathrm{diag}[\mu^\pi]$ where $\mu^\pi \in [0, 1]^S$ represents the stationary state distribution of policy

π. The value of a policy can then be computed as

$$v_{\text{BRM}} = \sum_{s \in D_0} \sum_{a \in \mathcal{A}} p_0(s) \cdot \pi(s,a) \cdot q_\eta(s,a) \,. \quad (3)$$

Importance Sampling Methods (IS) [Kahn and Marshall, 1953] are based on Monte-Carlo techniques and compute unbiased but high-variance value estimates. The key idea is to compute the value of policy π as the weighted average of the returns of the trajectories in D, where each trajectory is re-weighted by its probability of being observed under evaluation policy π_b. We focus on attacking three popular variants of importance sampling methods, namely the *Per-Decision, Consistent Weighted Per-Decision*, and *Weighted* IS methods (PDIS, CPDIS, WIS) [Precup, 2000, Thomas, 2015, Rubinstein, 1981]. Let $g_T^i = \sum_{t=0}^{T} \gamma^t r_t^i$ represent the returns observed for the ith trajectory in the dataset D and assume that that the behavior policy is parameterized by θ and estimated from data D using maximum likelihood estimation (MLE) [Vaart, 1998]. In order to define the OPE estimates of the value functions, we need the importance sampling weights $\rho_{0:t}^i$ for time step t defined as

$$\rho_{0:t}^i = \prod_{t'=0}^{t} \frac{\pi(a_{t'}^i | s_{t'}^i)}{\pi_b^{\theta_b}(a_{t'}^i | s_{t'}^i)} \,.$$

Here, the estimate of the behavior policy is defined as $\pi_b^{\theta_b}(a|s) = \exp(\phi(s,a)\theta_b)(\sum_{a' \in \mathcal{A}} \exp(\phi(s,a)\theta_b))^{-1}$ for each $s \in \mathcal{S}$ and $a \in \mathcal{A}$. Then the WIS, PDIS, and CPDIS value function estimates are defined as

$$v_{\text{WIS}} = \left(\sum_{i=1}^{N} \rho_{0:T}^i \right)^{-1} \sum_{i=1}^{N} \rho_{0:L}^i g_T^i, \quad (4)$$

$$v_{\text{PDIS}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \gamma^{t-1} \rho_{0:t}^i r_t^i, \quad (5)$$

$$v_{\text{CPDIS}} = \sum_{t=1}^{T} \gamma^{t-1} \frac{\sum_{i=1}^{N} \rho_{0:t}^i r_t^i}{\sum_{i=1}^{N} \rho_{0:t}^i}. \quad (6)$$

Hybrid Methods combine both Direct and IS methods to generate value estimates with low bias and variance. An important hybrid method is the *Doubly Robust* (DR) estimator [Jiang and Li, 2016], which decreases the variance in the IS estimate by using the estimate from a method like BRM. The DR and Weighted DR (WDR) estimators are given by

$$\begin{aligned} v_{\text{DR}} &= \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T-1} \rho_{0:t}^i w_t^i + \frac{1}{N} \sum_{i=1}^{N} v_\eta(s_0^i). \\ v_{\text{WDR}} &= \sum_{i=1}^{N} \sum_{t=0}^{T-1} \frac{\rho_{0:t}^i}{\sum_{i=1}^{N} \rho_{0:t}^i} w_t^i + \frac{1}{N} \sum_{i=1}^{N} v_\eta(s_0^i). \end{aligned} \quad (7)$$

where $w_t^i = (r_t^i - q_\eta(s_t^i, a_t^i) + v_\eta(s_t^i))$ and $v_\eta(s_t^i) = \sum_{a \in \mathcal{A}} \pi(s,a) \cdot q_\eta(s,a)$. Here the parameters of the value function q are estimated using Direct Methods like BRM. Because empirical studies show that are no clear winners among the three methods [Voloshin et al., 2020], we investigate attacks on representative methods from each type.

3 DOPE FRAMEWORK

We first present our attack framework called DOPE for *D*ata poisoning attacks on *O*ff-*P*olicy *E*valuation. Then we demonstrate how to use the framework to attack the three types of OPE methods discussed in Section 2. The objective and scope of the attacks considered in DOPE are as follows.

Scope: The attacker has access to the batch D and evaluation policy π and the value of the discount factor γ. We also assume that the attacker knows how the agent estimates the behavior policy and the state-action value function from the data. For the attack to be unnoticeable, the attack can only perturb α fraction of the transitions in D while conforming to some perturbation budget $\varepsilon \geq 0$ to be defined later.

Objective: The goal of the attacker is to add small adversarial perturbations to a subset of transitions in D such that it maximizes the error in the value estimate of a given policy in the desired direction. This means that the attacker may choose to decrease or increase its estimated value for the policy being evaluated such that a good evaluation policy is rejected or a bad evaluation policy is approved.

Components: The DOPE framework for a given OPE method has four major components: *Features (Ψ):* the part of the transition tuples targeted by the attack; *Value estimation function (ρ):* function used by the OPE method for computing the value; *Estimated parameter (θ):* model parameters learned by the OPE method from the data; *Loss function (L):* loss optimized by the OPE method for model-fitting. We define each component in detail in Section 3.1. We can now formulate our attack model as the problem of finding the perturbation matrix $\Delta = (\delta_i)_{i=1}^n, \delta_i \in \mathbb{R}^Q$ that maximizes the difference between values found using the perturbed and the original data under constraints dictating that the perturbations are small. Formally,

$$\underset{\Delta \in \mathbb{R}^{n \times Q}}{\text{maximize}} \quad \rho(\theta_{\text{pert}}, \Psi + \Delta) - \rho(\theta_{\text{org}}, \Psi) \quad (8a)$$

$$\text{subject to} \quad \theta_{\text{pert}} \in \underset{\theta \in \mathbb{R}^P}{\arg\min}\, L(\theta, \Psi + \Delta) \quad (8b)$$

$$\theta_{\text{org}} \in \underset{\theta \in \mathbb{R}^P}{\arg\min}\, L(\theta, \Psi) \quad (8c)$$

$$\|\delta_i\|_p \leq \varepsilon, \quad i = 1, \ldots, N \quad (8d)$$

$$\sum_{i=1}^{n} \mathbf{1}_{\|\delta_i\| \neq 0} \leq \alpha \cdot n. \quad (8e)$$

The DOPE objective in (8a) increases the estimated value from the original value, thereby, increasing the error. Alternatively, if the attacker wants to decrease the estimated

Method	Parameters θ	Features Ψ	Function $\rho(\Psi)$	Loss $L(\theta, \Psi)$
BRM (Farahmand et al. [2008]), Eq. (3)	η in q_η	Φ or r	v_{BRM}	MSBR
WIS (Rubinstein [1981]), Eq. (4)	θ_b in $\pi_b^{\theta_b}$	Φ or r	v_{WIS}	MLE
PDIS (Precup [2000]), Eq. (5)	θ_b	Φ or r	v_{PDIS}	MLE
CPDIS (Thomas [2015]), Eq. (6)	θ_b	Φ or r	v_{CPDIS}	MLE
WDR/DR (Jiang and Li [2016]), Eq. (7)	θ_b, η	Φ or r	v_{WDR} or v_{DR}	MLE + MSBR or MSBR

Table 1: Settings for the four components of the DOPE attack for five different OPE methods.

value of the given policy, they may do so by simply changing the sign of the objective. The constraint (8b) estimates the optimal parameter θ_{pert} from D after perturbing Ψ to $\Psi + \Delta$. The constraint (8d) ensures that the perturbation added to each sample δ_i, i.e. i^{th} row of Δ, is limited to the user-defined budget ε. This prevents the attack framework from generating adversarial transitions that can be easily detected as anomalous. Further, the constraint (8e) limits the number of transitions that the attacker can perturb. Finally, note that $\rho(\theta_{\text{org}}, \Psi)$ is a constant and can be ignored while solving the optimization problem.

3.1 ATTACKING OPE METHODS USING DOPE

We are now ready to formally define the four components of the DOPE framework. Table 1 summarizes the choice of these components for each OPE method we attack.

(a) *Features*: Let $\psi(s, a, r) \in \mathbb{R}^Q$ be an arbitrary component of the transition tuple $\langle s, a, r\rangle$ in D that is perturbed by the attacker. For example, $\psi(s, a, r)$ could either be the state features Φ or the reward vector r. We will use $\Psi \in \mathbb{R}^{n\times Q}$ to represent the sample matrix of $\psi(s, a, r)$ constructed from the n samples in D.

(b) *Parameters*: The parameters $\theta(\Psi) \in \mathbb{R}^P$ are the parameters of interest for a given OPE method, written as a function of Ψ to clarify that these are estimated from samples in D. In BRM, θ represents the parameters of the Q-value function $q_\eta(s, a)$, whereas in IS methods, θ represents the parameters of the estimated behavior policy $\pi_b^{\theta_b}(a|s)$.

(c) *Loss function*: The loss function $L(\theta, \Psi)$ with $L\colon \mathbb{R}^P \times \mathbb{R}^{n\times Q} \to \mathbb{R}$ is the empirical loss optimized by the OPE method to derive the optimal parameter $\theta(\Psi) \in \arg\min_{\theta'\in\mathbb{R}^P} L(\theta', \Psi)$ from the data. As an example, L in BRM and DR is the MSBR error, whereas in IS methods, L is the MLE loss optimized to estimate the behavior policy.

(d) *Value estimation function*: Finally, the value estimation function $\rho(\theta(\Psi), \Psi)$ with $\rho : \mathbb{R}^P \times \mathbb{R}^{n\times Q} \to \mathbb{R}$ is the function used by the OPE method to compute the mean value of π at the initial states. For example, in BRM, ρ represents v_{BRM}. We will use the shorthand $\rho(\Psi) := \rho(\theta(\Psi), \Psi)$.

The loss function $L(\theta, \Psi)$ must be twice continuously differentiable and linearly separable with respect to the transitions in D. The value estimation function $\rho(\theta, \Psi)$ also needs to be continuously differentiable with respect to θ and ψ. These assumptions, as Section 4 shows, are important for the influence functions to be well-defined [Koh and Liang, 2017].

4 OPTIMIZATION

In this section, we discuss the challenges with optimizing the DOPE problem in (8) and propose an approximate scheme for finding the optimal adversarial perturbations.

There are two major challenges. First, the constraint (8e) is non-differentiable and requires the attacker to select a set of at most αn transitions, such that perturbing these transitions results in maximum change in the value of the policy in the desired direction. It is important to realize that finding this set requires perturbing all possible subsets of data Ψ whose size is at most αn and re-estimating the optimal parameter θ for each perturbation. The number of such subsets is larger than $\binom{n}{\alpha n}$. Thus, finding the optimal set is computationally infeasible. Second, observe that (8) is a bilevel optimization problem where the inner-level problem (8b) is often non-linear for OPE methods which makes it an NP-Hard problem [Wiesemann et al., 2013].

We address these two challenges by deriving an approximation to the bilevel optimization problem (8) using the Taylor expansion. We show that the resulting problem is simpler to optimize and has a closed-form solution. In Section 5, we empirically demonstrate the effectiveness of our approximate solution on several domains.

Approximation We define the influence score of the ith data point as $I_{\Psi_i} = \nabla_{\Psi_i}\rho(\Psi)$ as the rate of change in the value estimate $\rho(\Psi)$ with respect to the data point $\Psi_i \equiv \psi(s_i, a_i, r_i)$. Then, using the first-order Taylor expansion of $\rho(\Psi + \Delta)$, we can approximate the net error in the value-function estimate $\rho(\Psi + \Delta) - \rho(\Psi)$ as the weighted sum of the influence scores of individual data points,

$$\rho(\Psi + \Delta) - \rho(\Psi) \approx \sum_{i=1}^{n} (\nabla_{\Psi_i}\rho(\Psi))^\top \delta_i. \tag{9}$$

Using Eq. (9) reduces the optimization in (8) to

$$\begin{aligned} &\max_{s\in\{0,1\}^n} \max_{\{\delta_i\}_{i=1}^N \in \mathbb{R}^{n\times Q}} \sum_{i=1}^{n} s_i \cdot I_{\Psi_i}^\top \delta_i \\ &\text{subject to } \sum_{i=1}^{n} s_i = \alpha \cdot n, \\ &\qquad \|\delta_i\|_p \le \varepsilon \cdot s_i,\ i = 1, \dots n. \end{aligned} \tag{10}$$

Here, $s \in \{0, 1\}^N$ is a vector of binary indicators such that

$s_i = 1$ indicates that the ith transition is amongst the αn transitions selected to be perturbed. We can now compute an approximately optimal set of perturbations in polynomial time as shown in Theorem 4.1 for norms $p = 1, 2, \infty$.

Theorem 4.1. *Let (s^*, Δ^*) be an optimal solution to the optimization problem in* (10) *and define the* approximate influential set *as $S^*_\alpha = \{i : s^*_i = 1, \forall i = 1, \dots, n\}$. Then,*

1. *S^*_α can be constructed by choosing the set of αn transitions with the largest q-norm of their influence scores I_{Ψ_i}. Here, q-norm is the dual of p-norm used in* (10)*, i.e. $1/p + 1/q = 1$.*
2. *For all $i \in [1, \dots n]$, the optimal δ^*_i for $p = 1, 2, \infty$ can be computed in closed-form as*

$$\text{If } p = \infty, \text{ then } \delta^*_i = \varepsilon \cdot \operatorname{sign}(I_{\Psi_i})$$

$$\text{If } p = 2, \text{ then } \delta^*_i = \epsilon \cdot \frac{I_{\Psi_i}}{\|I_{\Psi_i}\|_2}.$$

$$\text{If } p = 1, \text{ then } \forall j \in [1, Q],$$

$$\delta^*_{i,j} = \begin{cases} \operatorname{sign}(I_{\Psi_i}(j)) \cdot \varepsilon & \text{if } j \in \arg\max_{m \in [1,Q]} I_{\Psi_i}(m) \\ 0 & \text{otherwise} \end{cases}$$

Remark 4.2 (Relation to optimal solution). Solving the approximate problem (10) gives us a lower bound to the optimal solution of the original problem (8). Suppose Δ^* is the optimal solution for (10) that we get from Theorem 4.1 while Δ^{**} is the (intractable) optimal solution for (8). Then, the maximum error in the value function is at least as much as what we get,

$$\begin{aligned} \rho(\Psi + \Delta^{**}) - \rho(\Psi) &= \max_{\Delta \in \mathbb{R}^{n \times Q}} \rho(\Psi + \Delta) - \rho(\Psi) \\ &\geq \rho(\Psi + \Delta^*) - \rho(\Psi) \end{aligned}$$

Influence scores Finally, it remains to discuss how to compute the influence scores of each transition in D, i.e., $I_{\Psi_i} = \nabla_{\Psi_i} \rho(\Psi)$. Recall that $\rho(\Psi)$ is not only a function of Ψ but also $\theta(\Psi)$ which is also a function of Ψ_i. Hence, using the chain rule, we get for each $i \in [1 \dots n]$ that

$$I_{\Psi_i} \approx \left.\frac{\partial \rho(\theta, \Psi)}{\partial \Psi_i}\right|_{\theta_{\text{org}}(\Psi)} + \left.\frac{\partial \rho(\theta, \Psi)}{\partial \theta}\right|_{\theta_{\text{org}}(\Psi)} \frac{\partial \theta(\Psi)}{\partial \Psi_i}. \quad (11)$$

The computation of the partial derivative $\frac{\partial \theta(\Psi)}{\partial \Psi_i}$ is not straightforward. However, we can approximately compute it as $\partial\theta(\Psi)/\partial\Psi_i = H^{-1}_{\theta_{\text{org}}(\Psi)} \left.\partial^2 L(\theta, \Psi_i)/\partial\theta\partial\Psi_i\right|_{\theta_{\text{org}}(\Psi)}$ where $H_{\theta_{\text{org}}(\Psi)} = \left.\partial^2 L(\theta, \Psi)/\partial\theta^2\right|_{\theta_{\text{org}}(\Psi)}$ [Koh and Liang, 2017, Section 2.2]. See Appendix A for more details.

Thus to compute I_{Ψ_i} in (11), we require that $L(\theta, \Psi)$ is twice continuously differentiable and linearly separable with respect to the transitions in D, and $\rho(\theta, \Psi)$ is continuously differentiable with respect to θ and ψ. Although these conditions may seem restrictive, they hold true for the OPE methods we have studied.

Algorithm 1: OPE Attack Algorithm

Input: Features Ψ, attack budget ε, % of corrupt transitions α, norm-type p, threshold μ

Compute $\|I_{\Psi_k,\theta,\Psi}\|_q$ for all $i = 1, \dots, n$;
$S^*_\alpha \leftarrow \alpha n$ indices with largest $\|I_{\Psi_i,\theta,\Psi}\|_q$;
$\theta_{\text{org}} \leftarrow \arg\min_{\theta \in \mathbb{R}^P} L(\theta, \Psi)$;
for $k \in S^*_\alpha$ **do**
  Compute $I_{\Psi_k,\theta,\Psi}$ using (11) ;
  Compute $\delta^*_i \in \arg\max_{\delta \in \mathbb{R}^Q} I^\top_{\Psi_i,\theta,\Psi}\delta$ where $\|\delta\|_p \leq \varepsilon$ using Item 2 in Theorem 4.1;
end
Use line search to find appropriate step-size β s. t. the value estimate increases $\rho(\theta, \beta \times (\Psi + \delta^*)) - \rho(\theta, \Psi) > 0$;
Set $\Psi \leftarrow \beta \times (\Psi + \delta^*)$;
return Ψ

The derivatives in (11) can be easily computed using automatic-differentiation software like PyTorch [Paszke et al., 2019]. Computing the influence score I_{Ψ_i} can be very expensive due to the the Hessian-inverse term $H^{-1}_{\theta_{org}(\Psi)}$ which requires $\mathcal{O}(P^3)$ operations to compute. Fortunately, as shown in [Koh and Liang, 2017], we can avoid the computation of the Hessian-inverse term while computing I_{Ψ_i} by instead first approximately computing the Hessian-inverse vector product $c_{\text{prod}} = H^{-1}_{\theta_{\text{org}}(\Psi)} \left.\frac{\partial \rho(\theta,\Psi)}{\partial \theta}\right|_{\theta_{\text{org}}(\Psi)}$ in $O(nP)$ time using the Pearlmutter's method [Pearlmutter, 1994] and first-order Taylor approximation of the Hessian-inverse matrix and then applying the Pearlmutter's method again to compute the Hessian-vector product $c_{\text{prod}} \cdot \left.\partial^2 L(\theta, \Psi_i)/\partial\theta\partial\Psi_i\right|_{\theta_{\text{org}}(\Psi)}$ in $\mathcal{O}(P)$ time.

Algorithm outline We outline how to approximately solve the DOPE optimization (8) in Algorithm 1, which consists of two main steps. In the first step, we compute an approximation of the optimal set of transitions to perturb S^*_α by choosing αn points in Ψ with the largest q-norm of their influence scores $\|I_{\psi.}\|_q$. In the second step, we compute Δ for all points in S^*_α using Theorem 4.1 and use line search to find the optimal step size that guarantees an increase in the error of the value estimate. The second step may be repeated until no further perturbation to data points in S^*_α results in an increase in error in the value estimate.

The main computational bottleneck is in computing the influence score for each data point. In limited cases, this cost can be reduced. We derive closed-form expressions for the influence score in the case of the linear BRM method under two settings a) when the adversary perturbs only the state features or b) only the reward features.

Proposition 4.3. *If the attacker only perturbs the reward vector r constructed from batch of transition tuples D. Then, the influence score of the ith data point $I_{r_i,\theta,\Psi}$ for the BRM*

method can be computed as

$$I_{r,\theta,\Psi} = 4p_0^T \Phi_0 \left((\Phi - \gamma \cdot \Phi_p)^2\right)^{-1} (\Phi - \gamma \cdot \Phi_p) \ . \quad (12)$$

where Φ_0 is a sample matrix of initial state features constructed from D_0.

Proposition 4.4. *If the attacker only perturbs the state feature matrix Φ. Then, the influence score of the ith data point $I_{\phi(s_i,a_i),\theta,\Psi}$ for the BRM method can be computed as*

$$\begin{aligned} I_{\Psi,\theta,\Psi} = 4 \cdot p_0^T \cdot \Phi_0 \cdot \left((\Phi - \gamma \cdot \Phi_p)^2\right)^{-1} \cdot \\ \Big(2 \cdot w \bigotimes \Phi - 2 \cdot \gamma \cdot w \bigotimes \Phi_p \\ +2 \cdot \mathbf{I} \bigotimes (\Phi \cdot w - \gamma \cdot \Phi_p \cdot w - r)\Big) \ , \end{aligned} \quad (13)$$

where $\bigotimes$ denotes the Kronecker product between matrices.

Proposition 4.3 and 4.4 follow from the chain rule and basic properties of the gradient operator for matrices.

5 EXPERIMENTS

In this section, we investigate the strengths and weaknesses of the DOPE attack. First, we evaluate the effectiveness of the DOPE attack on OPE methods for different values of attack budget and identify which methods are most vulnerable to adversarial contamination. Second, we compare the performance of DOPE with two custom baselines: Random DOPE and Random Attack.

5.1 DOMAINS AND EXPERIMENTAL SETUP

We first describe the five domains used in the experiments.

Cancer: This domain [Gottesman et al., 2020] models the growth of tumors in cancer patients. It consists of 4-dimensional states which represent the growth dynamics of the tumor in the patient, and two actions that indicate if a given patient is to be administered chemotherapy or not at a given time step.
HIV: The HIV domain [Ernst et al., 2006] has 6-dimensional states representing the state of the patient, and four actions that represent four different types of treatments.
Mountain Car: In the Mountain Car [Brockman et al., 2016] domain, the task is to drive a car positioned between two mountains to the top of the mountain on the right in the shortest time possible. The 2-dimensional state represents the car's current position and the current time-step, and the three actions represent: drive forward, drive backward, and do not move.
Cartpole: The Cartpole domain [Brockman et al., 2016] models a simple control problem where the goal is to apply +1/-1 force to keep a pole attached to a moving cart from falling. The 2-dimensional state represents the cartpole dynamics, and the two actions represent the force applied to the pole.
Continuous Gridworld: The Continuous Gridworld is a custom domain that consists of a 2-dimensional state space that represents the coordinates of the agent, and two actions a_0, a_1 that determine the direction and step size of the agent. The agent begins at coordinate $(1, 1)$ and moves towards coordinates $(50, 50)$ to maximize its rewards.

Implementation details For each domain, we apply Deep Q-learning (DQN) to a randomly initialized neural network policy and obtain partially optimized deterministic policies. We fix the deterministic policy obtained for each domain as the evaluation policy and use an ϵ-greedy version of the evaluation policy as our behavior policy [Gottesman et al., 2020]. We set $\epsilon = 0.1$ for the HIV domain and $\epsilon = 0.05$ for other domains. We use the behavior policy to generate five datasets, each containing N trajectories of length L (see Appendix C for the values of N and L) and use it to evaluate the value of the evaluation policy. Our code is made available in the supplementary materials.

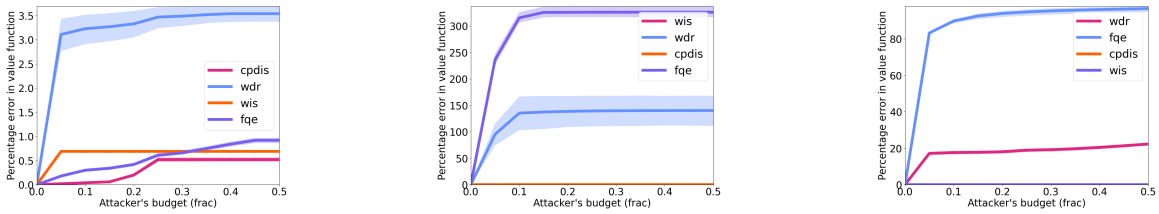
For any given OPE method that learns the Q-value function of the evaluation policy from data, we use linear function approximators to represent these Q-value functions and optimize the squared Bellman residual with L_2 regularization to learn it. Likewise, for OPE methods that require learning behavior policy from the data, we train a multinomial logistic regression model to predict the behavior policy's action probabilities using maximum likelihood estimation. Following standard practice in RL, we clip the behavior probabilities to 0.01 to avoid importance sampling weights from exploding. Note that although clipping the behavior probabilities prohibits the attacker from making individual behavior policy action probabilities too small, an attacker can still leverage the fact that the importance sampling weights are a function of the product of behavior policy action probabilities, and thus, the importance weights can be made very large by simply making the behavior policy action probabilities of as many points as possible to close to the clipping threshold.

In all our experiments, we perturb only state features. Finally, the values of the hyperparameters used in our experiments are discussed in Appendix C.

We evaluate the effectiveness of our attack model by computing the percentage error in the value function estimate relative to the initial value estimate. We report the 95% bootstrap confidence intervals of the interquartile mean (IQM) of percentage error using our results from the 5 runs (5 datasets) since the IQM confidence intervals are found to be more reliable in practice [Agarwal et al., 2021]. In this setting, a large percentage error indicates that the OPE method is less robust to adversarial contamination.

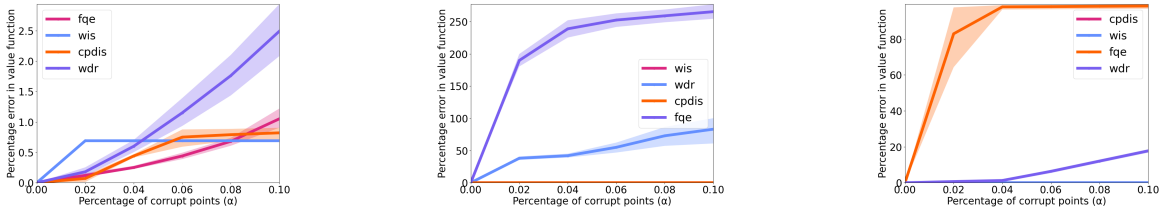
Domain	BRM		WIS		PDIS		CPDIS		WDR	
	lb	ub	lb	ub	lb	ub	lb	ub	lb	ub
Cancer	0.9	1.2	0.7	0.7	13.7	16.2	0.7	0.9	**89.0**	**101.9**
HIV	**254.1**	**276.9**	0.0	0.1	1.2	**1120.0**	0.0	0.1	**54.4**	**101.2**
Gridworld	**96.8**	**99.3**	0.0	0.0	**97.9**	**98.2**	0.0	0.0	16.7	17.7
Cartpole	0.0	0.0	0.0	0.1	**3.44e12**	**5.33e13**	0.0	0.0	0.0	0.0
MountainCar	0.1	0.1	**100.0**	**100.0**	**98.2**	**99.6**	**47.4**	**98.7**	0.0	0.0

Table 2: Summary of the errors achieved by data poisoning across domains and OPE algorithms. Here lb and ub denote the lower limit and upper limit of 95% bootstrap confidence intervals of interquartile mean of percentage error in the value estimates, over 5 runs. We observe that the attack is successful on most of the methods across domains. CPDIS and WIS are the most resilient OPE methods.



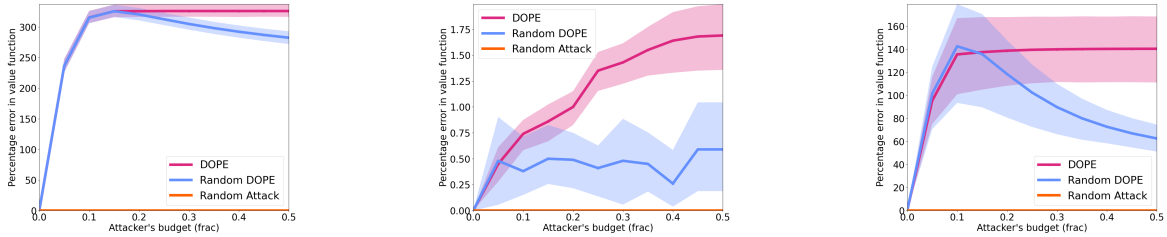
(a) Cancer **(b)** HIV **(c)** Gridworld

Figure 1: Figures 1a to 1c compares the effect of DOPE attack on BRM, WIS, PDIS and CPDIS and WDR methods in the Cancer, HIV and Continuous Gridworld domains (left to right) for different values of attacker's budget $\varepsilon = \text{frac} \cdot \sigma$ and $p = 1$ (ℓ_1 norm). Larger the value of frac, the larger are the perturbations added by the DOPE attack, and accordingly we observe larger errors in the value estimates.



(a) Cancer **(b)** HIV **(c)** Gridworld

Figure 2: Figures 2b to 2c compares the effect of DOPE attack on BRM, WIS, PDIS and CPDIS and WDR methods in Cancer, HIV and Continuous Gridworld domains (left to right) for different percentages of corruption α at $\varepsilon = 1.0\sigma$ and $p = 1$ (l_1 norm). Larger the value of α, the larger is the number of points perturbed by the DOPE attack, and accordingly we observe larger errors in the value estimates.



(a) BRM **(b)** IS **(c)** WDR

Figure 3: Figures 3a to 3c compare the effects of Random attack, Random DOPE attack (an ablated version of DOPE), and DOPE attack on the error in the value function estimates of BRM, IS and DR methods (left to right) in Continuous Gridworld domain. The percentage error in the random attack is small relative to the percentage error due to DOPE and Random DOPE attack, and hence its curve lies close to the x-axis. DOPE attack outperforms both the Random DOPE and Random attacks at nearly all values of the attacker's budget.

5.2 EFFECTIVENESS OF DOPE ATTACK

Here we evaluate the effectiveness of the DOPE attack on five OPE methods for a range of attack budgets. In our first experiment, we fix the percentage of corrupt data points $\alpha = 0.05$ and vary the budget ε as frac $\cdot\, \sigma$, where frac varies from 0.0 to 0.51 in step-sizes of 0.05 and $\sigma^2 = \frac{2}{N\cdot(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \|\xi(s_i) - \xi(s_j)\|_p^2$ is the standard deviation of all pairwise distances between the state-features in the dataset. Figures 1 and 4 compare the percentage error in the value estimate of the OPE methods in all domains. Our results show that even when corrupting only 5% of the data points, the attacker need not perturb the state features significantly to achieve large errors in the value estimate. In fact, with a perturbation budget as small as $\epsilon = 0.5\sigma$, DOPE can result in a substantial error in the policy's value in HIV, Cancer, Mountain Car, and Continuous Gridworld domains. Further, a larger attacker's budget means the DOPE model has more leeway on the perturbations that it can add to the dataset, and hence, we observe larger errors for larger budget values. Note that the percentage errors of CPDIS and WIS in Figures 1b and 1c are too small to be visible in the plots.

In the second experiment, we vary the percentage of corrupt data points between 0.0 and 0.10 with a step size of 0.02 for all the domains (Figure 2). We fix the perturbation budget ϵ to 1.0σ. Our experimental results in Figures 1, 2 and 4 demonstrate that corrupting only 0.05% of the data points using DOPE is sufficient to observe a significant error in the value estimate of a given policy. It is important to realize that the attacker's budget ε is local to each data point and is not impacted by the number of points perturbed. Hence, we see that a larger percentage of corrupt data points yields a larger percentage error in the value estimates. Note that the percentage errors of CPDIS and WIS in Figures 2b and 2c are too small relative to BRM and WDR and therefore are not clearly visible in the plots.

Finally, we summarize the impact of DOPE attack ($\epsilon = 0.5\sigma$ and $\alpha = 1.0$, $p = 1$) on all OPE methods and domains in Table 2. It can be seen that the DOPE attack has a very high impact on BRM, PDIS, and WDR methods and almost negligible impact on CPDIS and WIS methods. We hypothesize that CPDIS and WIS methods may be more robust because the weight normalization that they employ potentially minimizes the importance of any individual data point, especially when the rewards are uniformly distributed throughout the trajectory. On the other hand, the weights in PDIS are not normalized, and therefore, it appears that in Cartpole and HIV domains, the DOPE attack model is able to significantly impact the importance sampling weights and result in significant errors in the value estimates. In WDR, the attacker can introduce errors through both, the Q-value function learned from the data as well as the importance sampling weights, and therefore, we observe significant errors in the value estimates of WDR method.

5.3 COMPARISON WITH BASELINES

Here we compare the DOPE attack to two custom baselines: Random Attack and Random DOPE Attack (ablation of DOPE Attack). In Random Attack, we choose αn random points to perturb and sample perturbations for these points from a uniform l_1 norm ball with a radius equal to the perturbation budget ϵ. For more details on the sampling algorithm, see Algorithm 4.1 in [Calafiore et al., 1998]. In Random DOPE Attack, we select points randomly and update them using Theorem 4.1. The purpose of using this ablation is to investigate the benefit of selecting data points to perturb based on their influence scores as suggested in Theorem 4.1. Given these baselines, we fix the value of α to 0.05 and vary the budget ϵ from 0.0 to 0.25 with step size 0.04. For each dataset and each value of the budget ϵ, we average the percentage change in the value estimate for Random DOPE attack and Random attack over 50 trials. Results with the gridworld domain are shown in Figure 3. See Figures 6 to 9 in Appendix for results with other datasets.

The experimental results in Figures 3 and 6 to 9 demonstrate that in contrast to the DOPE attack, the Random Attack fails to introduce any significant error in the value-function estimate and, therefore, cannot be used as an alternative to the DOPE attack model. Further, it can be seen that when the points to perturb are randomly selected (Random DOPE), it is likely to result in a smaller adversarial impact than when influential data points are chosen for perturbations (DOPE). These results are not surprising as we would expect the value function to be highly dependent on the influential data points. In some domains like Cancer and HIV, there is very little difference between the performance of DOPE and Random DOPE attacks. We hypothesize that this is due to all data points having similar influence scores.

6 CONCLUSION

We proposed a novel data poisoning framework to analyze the sensitivity of OPE methods to adversarial contamination at train-time. We formulated the data poisoning problem as a bilevel-optimization problem and proposed a computationally tractable solution that leverages the notion of influence functions from robust statistics literature. Using the proposed framework, we analyzed the sensitivity of five popular OPE methods on multiple datasets from medical and control domains. Our experimental results on various medical and control domains demonstrated that existing OPE methods are highly vulnerable to adversarial contamination thus highlighting the need for developing OPE methods that are statistically robust to train-time data poisoning attacks.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 2021.

Vahid Behzadan and Arslan Munir. Vulnerability of deep reinforcement learning to policy induction attacks. In *International Conference on Machine Learning and Data Mining*, 2017.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Tamara Broderick, Ryan Giordano, and Rachael Meager. An automatic finite-sample robustness metric: Can dropping a little data change conclusions?, 2021.

G. Calafiore, F. Dabbene, and R. Tempo. Uniform sample generation in l/sub p/ balls for probabilistic robustness analysis. In *IEEE Conference on Decision and Control*, volume 3, 1998.

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, 12 2019.

Gilad Cohen, Guillermo Sapiro, and Raja Giryes. Detecting adversarial samples using influence functions and nearest neighbors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

R. Dennis Cook and Sanford Weisberg. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 22(4):495–508, 1980.

Ilias Diakonikolas and Daniel M. Kane. Recent advances in algorithmic high-dimensional robust statistics, 2019.

Damien Ernst, Guy-Bart Stan, Jorge Goncalves, and Louis Wehenkel. Clinical data based optimal sti strategies for hiv: a reinforcement learning approach. In *IEEE Conference on Decision and Control*, pages 667–672, 2006.

Minghong Fang, Neil Zhenqiang Gong, and Jia Liu. *Influence Function Based Data Poisoning Attacks to Top-N Recommender Systems*, page 3019–3025. Association for Computing Machinery, New York, NY, USA, 2020.

Amir-massoud Farahmand, Mohammad Ghavamzadeh, Csaba Szepesvári, and Shie Mannor. Regularized policy iteration. In *International Conference on Neural Information Processing Systems*, page 441–448, 2008.

Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. In *International Conference on Learning Representations*, 2020.

Omer Gottesman, Joseph Futoma, Yao Liu, Sonali Parbhoo, Leo Celi, Emma Brunskill, and Finale Doshi-Velez. Interpretable off-policy evaluation in reinforcement learning by highlighting influential transitions. In *International Conference on Machine Learning*, pages 3658–3667, 2020.

Frank R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974.

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

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661, 2016.

H. Kahn and A. W. Marshall. Methods of Reducing Sample Size in Monte Carlo Computations. *Operations Research*, 1(5), November 1953.

Panagiota Kiourti, Kacper Wardega, Susmit Jha, and Wenchao Li. Trojdrl: Evaluation of backdoor attacks on deep reinforcement learning. 2020.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, page 1885–1894, 2017.

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

Jernej Kos and Dawn Song. Delving into adversarial attacks on deep policies, 2017.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020.

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 *International Joint Conference on Artificial Intelligence*, pages 3756–3762, 2017.

Yuzhe Ma, Xuezhou Zhang, Wen Sun, and Xiaojin Zhu. Policy poisoning in batch reinforcement learning and control. In *Thirty-third Conference on Neural Information Processing Systems*, 2019.

Rémi Munos and Andrew Moore. Variable resolution discretization in optimal control. In *Machine Learning*, pages 291–323, 2002.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. 2019.

Barak A. Pearlmutter. Fast exact multiplication by the Hessian. *Neural Computing*, 6(1), 1994. ISSN 0899-7667.

M. J. D. Powell and J. Swann. Weighted uniform sampling: A Monte Carlo technique for reducing variance. *IMA Journal of Applied Mathematics*, 2(3):228–236, 09 1966.

Doina Precup. Temporal abstraction in reinforcement learning, 2000.

Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla. Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning. In *International Conference on Machine Learning*, pages 7974–7984, 2020.

Reuven Y. Rubinstein. *Simulation and the Monte Carlo Method*. John Wiley & Sons, Inc., USA, 1st edition, 1981.

Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Philip Thomas. Safe reinforcement learning, 2015.

Philip S. Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence off-policy evaluation. In *AAAI Conference on Artificial Intelligence*, page 3000–3006, 2015.

Daniel Ting and Eric Brochu. Optimal subsampling with influence functions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, 2018.

A. W. van der Vaart. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998.

Cameron Voloshin, Hoang M. Le, Nan Jiang, and Yisong Yue. Empirical study of off-policy policy evaluation for reinforcement learning, 2020.

Wolfram Wiesemann, Angelos Tsoukalas, Polyxeni-Margarita Kleniati, and Berç Rustem. Pessimistic bilevel optimization. *SIAM Journal on Optimization*, 23(1):353–380, 2013.

Chenwang Wu, Defu Lian, Yong Ge, Zhihao Zhu, and Enhong Chen. *Triple Adversarial Learning for Influence Based Poisoning Attack in Recommender Systems*, page 1830–1840. Association for Computing Machinery, 2021a.

Xian Wu, Wenbo Guo, Hua Wei, and Xinyu Xing. Adversarial policy training against deep reinforcement learning. In *USENIX Security Symposium*, 2021b.

Haoqi Zhang and David Parkes. Value-based policy teaching with active indirect elicitation. In *National Conference on Artificial Intelligence*, 2008.

Haoqi Zhang, David C. Parkes, and Yiling Chen. Policy teaching through reward function learning. In *ACM Conference on Electronic Commerce*, 2009.

Xuezhou Zhang, Yuzhe Ma, Adish Singla, and Xiaojin Zhu. Adaptive reward-poisoning attacks against reinforcement learning. In *International Conference on Machine Learning*, pages 11225–11234, 2020a.

Xuezhou Zhang, Xiaojin Zhu, and Laurent Lessard. Online data poisoning attacks. In *Conference on Learning for Dynamics and Control*, pages 201–210, 2020b.

Xuezhou Zhang, Yiding Chen, Jerry Zhu, and Wen Sun. Corruption-robust offline reinforcement learning, 2021.

A ADDITIONAL PRELIMINARIES

Influence functions Influence function is a popular tool used to quantify the change in an empirically learned estimator with small changes in data. Consider a supervised learning problem with input space $\mathcal{X}$ and output space $\mathcal{Y}$, a batch of data $(z)_{i=1}^n$ where $z_i = (x_i, y_i) \in (X \times Y)$ and an unknown prediction function $f : \mathcal{X} \to \mathcal{Y}$ where f is parameterized by $\theta \in \Theta$. Given a convex and doubly differentiable loss function $L(\theta, z)$ such that $L : \Theta \times \mathcal{X} \to \mathbb{R}$ and $\theta \in \arg\min_{\theta' \in \Theta} \frac{1}{n} \sum_{i=1}^n L(\theta', z_i)$ is the empirical risk minimizer, then, the effect $I_{z,\theta,D}$ of perturbing a data point $z \to z_\delta = (x+\delta, y)$ on the parameter θ can be approximated via Taylor expansion as

$$\begin{aligned} \mathcal{I}_{z_\delta,\theta,D} &= \frac{\theta_{z,\delta} - \theta}{\delta} \approx \frac{\partial \theta}{\partial x} \\ &\approx \left(-H_\theta^{-1} \frac{\partial^2 L(\theta, z)}{\partial \theta \partial x} \right) \text{ where } H_\theta = \frac{\partial^2 L(\theta, D)}{\partial^2 \theta} \end{aligned} \tag{14}$$

where $\theta_{z,\delta}$ are the new optimal parameters learned from the training data point after replacing z by z_δ. We refer the readers to [Koh et al., 2018] for more details.

B PROOFS:

Proof of Theorem 4.1. Consider the optimization problem in (15):

$$\begin{aligned} &\max_{s \in \{0,1\}^n} \max_{\{\delta_k\}_{k=1}^N \in \mathbb{R}^{n \times Q}} \sum_{k=1}^n s_k \cdot I_{\Psi_i}^\top \delta_k \\ &\text{subject to } \sum_{k=1}^n s_k = \alpha \cdot n, \\ &\qquad \|\delta_k\|_p \leq \varepsilon \cdot s_k, \ k = 1, \dots n. \end{aligned} \tag{15}$$

Notice that in (15), $\forall k \in [1, \dots N]$, I_{Ψ_i} is independent of δ_k and so the optimal perturbation δ_k^* can be independently computed by solving $\delta_k^* \in \arg\max_x I_{\Psi_k,\theta,\Psi}^T x \text{ s. t. } \|x\|_p \leq \varepsilon$. Further, from the theory of convex optimization [Boyd and Vandenberghe, 2004], we know that the p-norm $\|x\|_p$ of any vector $x \in \mathbb{R}^M$ can be expressed using its dual norm as $\|x\|_p = \max z^T x \text{ s. t. } \|z\|_q \leq 1$ where $1/p + 1/q = 1$. Thus, given the optimal-perturbation $\delta_k^* \, \forall k \in [1, \dots n]$, the problem in Equation (10) boils down to solving

$$\begin{aligned} &\max_{s \in \{0,1\}^N} \sum_{k=1}^n \|I_{\Psi_k,\theta,\Psi}\|_q \\ &\qquad \sum_k s_k = \alpha \cdot n. \end{aligned} \tag{16}$$

It is now easy to see that the optimal set of transitions for the approximate attack problem in (10) is simply the set of αn transitions with the largest value of the q-norm of their influence scores. The closed form solution for δ_k^* at $p = 1, 2, \infty$ follows from standard convex optimization results for $\|x\|_q = \max z^T x \text{ s. t. } \|z\|_p \leq 1$ in [Boyd and Vandenberghe, 2004]. □

C EXPERIMENTAL DETAILS:

C.1 ADDITIONAL OPTIMIZATION TRICKS USED IN EXPERIMENTS:

1. Recall that we use the DQN algorithm to learn the optimal Q-value function using a neural network, from which we derive the evaluation policy. In the case of the Cartpole and Mountain Car domains, we use this Q-value network to transform the state features into features $\phi(s, a)$. Specifically, we use the output of the second last layer of the Q-value network as the transformed state features. We do this to get a more accurate feature representation for linear function appproximators which in turn would result in a more accurate initial value function estimate.
2. In all our experiments, we use line-search to find the optimal step size to update the state-features with the perturbations derived using Theorem 4.1. If for a given attacker's budget, we have access to the error in the value-function estimate for a lower value of the attacker's budget, then, we use it as the minimum threshold error to achieve while applying the line search. Applying this method enables us to achieve a monotonic trend in the percentage error in the value estimate with respect to the perturbation budget. The monotonic trend is otherwise difficult to achieve especially when the Loss function is non-convex.
3. To optimize the DOPE objective for any given OPE method, we need have differentiable evaluation policy action probabilities. In the case, where the evaluation policy is a deterministic Q-learning policy, we obtain differentiable action probabilities by applying softmax to the q-values with very small temperature value.

C.2 ADDITIONAL DOMAIN DETAILS:

Cancer: This domain [Gottesman et al., 2020] models the growth of tumors in cancer patients. It consists of 4-dimensional states which represent the growth dynamics of the tumor in the patient, and two actions that indicate if a given patient is to be administered chemotherapy or not at a given time step.
HIV: The HIV domain has 6-dimensional states representing the state of the patient, and four actions that represent four different types of treatments.
MountainCar: In the Mountain Car [Brockman et al., 2016] domain, the task is to drive a car positioned between two mountains to the top of the mountain on the right in the

Hyperparameter values for Cancer domain	
Hyperparameter	Value
Number of trajectories	500
Policy Network layers	64×28
Normalize rewards	No
Regularization for π_b	1e-2
Regularization for q_η	1e-2
Discount factor	0.95
Trajectory Length (T)	30
Direction of Attack	+1
Num. Epochs for MLE	5000

Hyperparameter values for HIV domain	
Hyperparameter	Value
Number of trajectories	1000
Policy Network layers	300×50
Normalize rewards	Yes
Regularization for π_b	1e-2
Regularization for q_η	1e-2
Discount factor	0.98
Trajectory Length (T)	50
Direction of Attack	-1
Num. Epochs for MLE	5000

Hyperparameter values for Continuous Gridworld domain	
Hyperparameter	Value
Number of trajectories	500
Policy Network layers	24
Normalize rewards	No
Regularization for π_b	1e-2
Regularization for q_η	1e-2
Discount factor	0.95
Trajectory Length (T)	50
Direction of Attack	-1
Num. Epochs for MLE	5000

Hyperparameter values for MountainCar domain	
Hyperparameter	Value
Number of trajectories	250
Policy Network layers	60
Normalize rewards	No
Regularization for π_b	1e-2
Regularization for q_η	1e-2
Discount factor	0.99
Trajectory Length (T)	150
Direction of Attack	+1
Num. Epochs for MLE	5000

shortest time possible. The 2-dimensional state represents the car's current position and the current time-step, and the three actions represent: drive forward, drive backward, and do not move.

Cartpole: The Cartpole domain [Brockman et al., 2016] models a simple control problem where the goal is to apply +1/-1 force to keep a pole attached to a moving cart from falling. The 2-dimensional state represents the cartpole dynamics, and the two actions represent the force applied to the pole.

Continuous Gridworld: The gridworld domain consists of a 2-dimensional state space that represent the coordinates of

Hyperparameter values for Cancer domain	
Hyperparameter	Value
Number of trajectories	1000
Policy Network layers	100×24
Normalize rewards	No
Regularization for π_b	1e-2
Regularization for q_η	1e-2
Discount factor	0.98
Trajectory Length (T)	100
Direction of Attack	+1
Num. Epochs for MLE	5000

the agent and 2 actions (a_0, a_1) that determines the direction and step size of the agent. The task is to begin at coordinate $(1, 1)$ and move towards coordinates $(50, 50)$. Taking action a_0 at (x, y) transitions the agent to $(x + 0.2, y + 0.45)$ with probability 1.0. On the other hand, taking action a_1 transitions the agent to $(x + 0.3, y + 0.5)$ with probability 0.95 and to $(1, 1)$ with probability 0.05. If the agent transitions to (x', y'), the agent receives a reward of $(x + 0.5y)$. We set the maximum length of the episode to 50 and collected 500 trajectories using the behavior policy.

D RELATED WORK

Adversarial attacks have been extensively studied in Reinforcement Learning [Gleave et al., 2020, Wu et al., 2021a, Lin et al., 2017, Zhang et al., 2020a,b, Lin et al., 2017, Kiourti et al., 2020, Chen et al., 2019]. These attacks can be broadly classified into two main categories - train-time attacks (data-poisoning attacks) and test-time attacks.

Test-time attacks: In test-time attacks in RL [Lin et al., 2017, Gleave et al., 2020, Behzadan and Munir, 2017, Kos and Song, 2017, Wu et al., 2021b, Chen et al., 2019, Huang et al., 2017], the attacker manipulates test-time observations to fool the agent to take target malicious actions, without directly changing the agent's policy. In this setting, the noise added to the test-time observations at any time step does not directly impact the agent's future decisions. A large majority of the work that focuses on test-time attacks aim to either minimize the agent's rewards [Huang et al., 2017, Behzadan and Munir, 2017] or lead the agent to adversarial states [Lin et al., 2017], which differs from our goal of perturbing train-time observations to maximize error in the value estimate of a given policy for a given OPE method.

Train-time attacks: In train-time or data-poisoning attacks, the adversary perturbs the training data by a small margin to facilitate erroneous learning of decision models. Prior work on data-poisoning have mainly targeted supervised learning models in Machine Learning [Koh et al., 2018, Koh and Liang, 2017, Fang et al., 2020, Wu et al., 2021a, Steinhardt et al., 2017]. However, recently there has been emerging interests in data-poisoning attacks on Batch RL agents [Zhang et al., 2021, Ma et al., 2019, Rakhsha et al., 2020] and Online RL Agents [Zhang et al., 2020b,a, Rakhsha et al., 2020, Zhang and Parkes, 2008, Zhang et al., 2009]. In pioneering research work, [Zhang et al., 2020b] proposed a framework that perturbs rewards such that a Batch RL agent learns an adversarial target policy. In the following work, [Rakhsha et al., 2020] proposed a framework for poisoning rewards and transition dynamics to force a Batch agent to learn an adversarial target policy. [Zhang et al., 2020b] develops fast adaptive data-poisoning attacks on online RL agents where rewards must be perturbed in real-time. Nonetheless, these data-poisoning works differ from our work in two main aspects: a)They target learning of optimal adversarial policies, whereas our work targets learning erroneous value-function estimates for any given policy and OPE method b) our main goal is to analyze the sensitivity of different OPE algorithms to train-time attacks which has not been explored in any of these previous work.

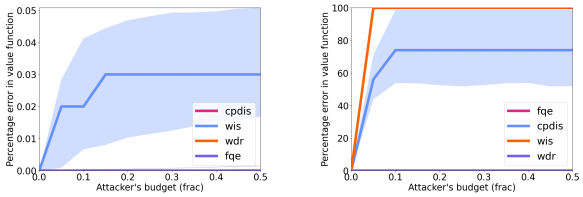
Finally, our work is similar in vein to the bilevel-optimization framework proposed by [Koh et al., 2018] for data-poisoning attacks on supervised learning algorithms with data sanitization defense mechanisms. However, in contrast to this work, we exploit specific properties of OPE algorithms to construct stronger data-poisoning attacks as well as compare the sensitivity of different OPE algorithms in RL.

Influence functions: Influence function was originally introduced in robust statistics [Cook and Weisberg, 1980, Hampel, 1974] to understand the effect of perturbing of removing a train data point on small linear models estimated from the data. In more recent work, influence functions have been used as an diagnostic tool in deep learning and reinforcement learning algorithms to detect adversarial training data points [Broderick et al., 2021, Koh et al., 2018, Koh and Liang, 2017, Gottesman et al., 2020, Cohen et al., 2020], optimal sub-sampling [Ting and Brochu, 2018] and to aide decision-policy optimization [Munos and Moore, 2002]. A few work have also proposed influence-functions based data-poisoning attacks on supervised learning algorithms [Koh et al., 2018, Koh and Liang, 2017, Wu et al., 2021a, Fang et al., 2020]. However, our work differs from theirs in terms of context (reinforcement learning) and ob-

jectives optimized.

D.1 EXPERIMENTAL RESULTS:

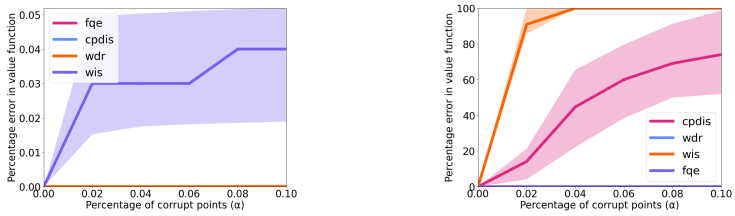
D.1.1 Effectiveness of DOPE Attack



(a) Cartpole **(b)** Mountain Car

Figure 4: Figures 4a and 4b compares the effect of DOPE attack on BRM, WIS, PDIS, CPDIS and WDR methods in Cartpole and Mountain Car domains for different values of attacker's budget $\varepsilon = frac \cdot \sigma$ and $p = 1$.

D.1.2 Comparison with Baselines



(a) Cartpole **(b)** MountainCar

Figure 5: Figures 5a and 5b compares the effect of DOPE attack on BRM, WIS, PDIS, CPDIS and WDR methods in in Cartpole and MountainCar domains (left to right) for different percentages of corruption α and $p = 1$.

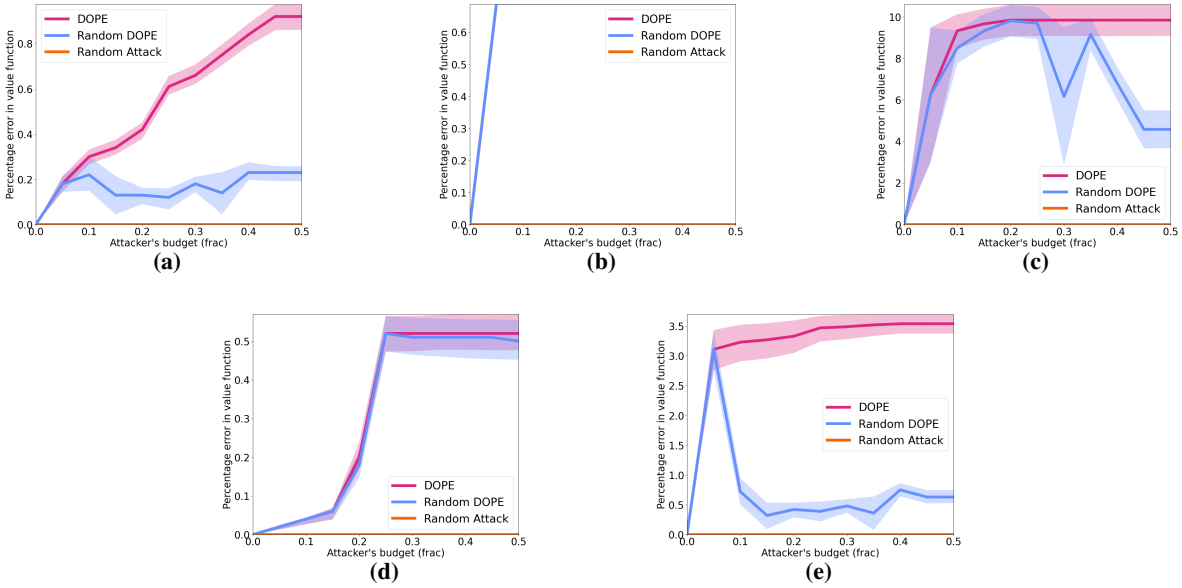


Figure 6: Figures 6a to 6e compares the effect of random attack, Random DOPE attack and DOPE attack on the error in the value function estimates of BRM, WIS, PDIS, CPDIS and WDR methods (left to right) in Cancer domain.

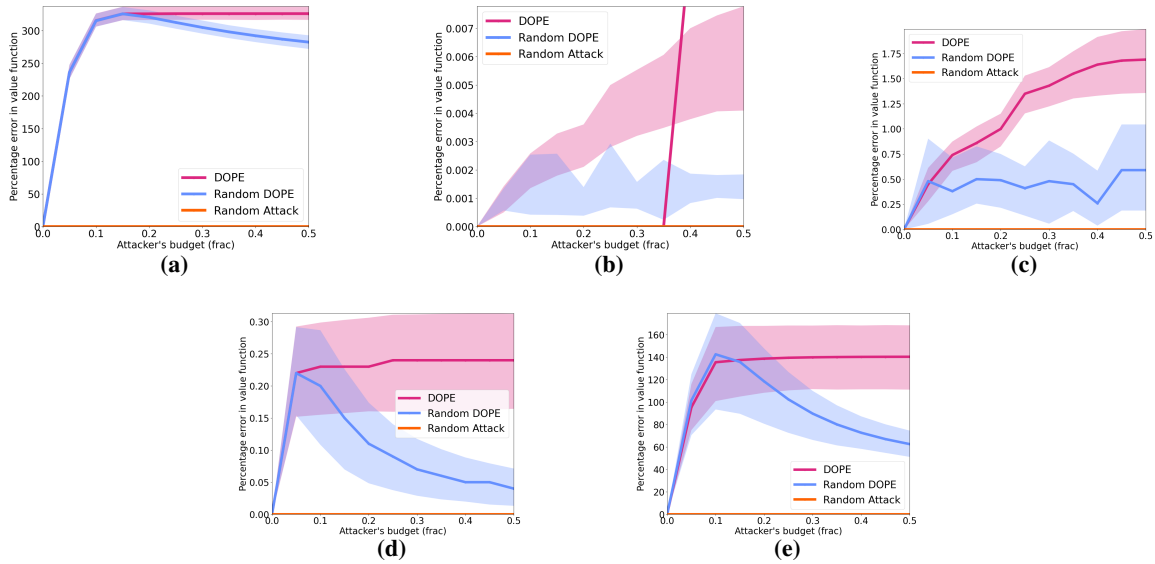


Figure 7: Figures 7a to 7e compares the effect of random attack, Random DOPE attack and DOPE attack on the error in the value function estimates of BRM, WIS and PDIS, CPDIS, WDR methods (left to right) in HIV domain.

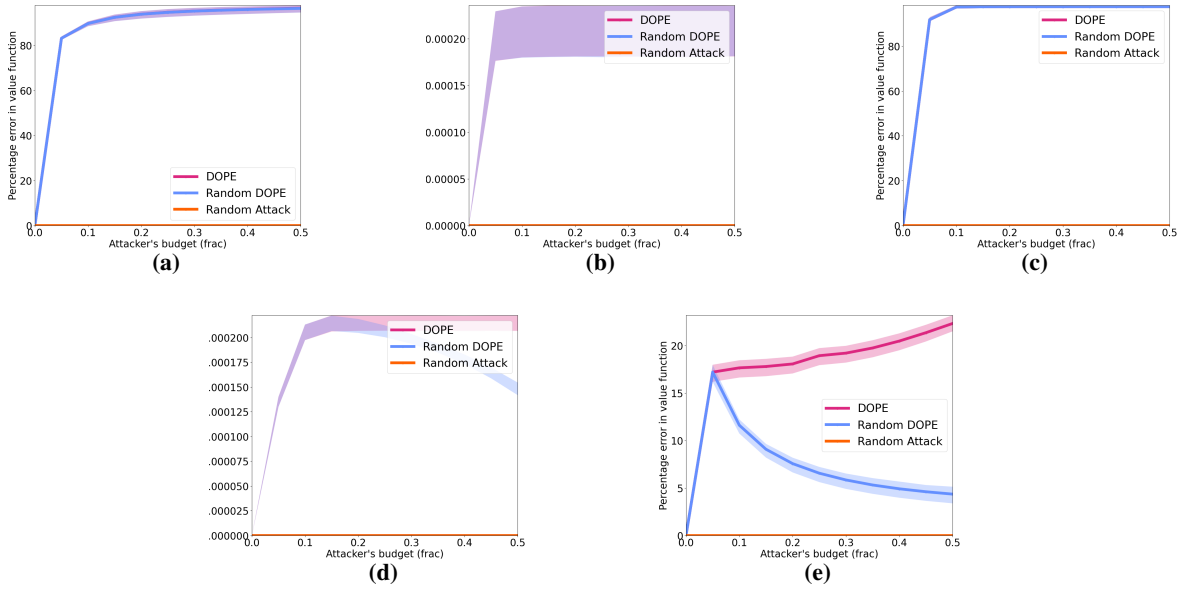


Figure 8: Figures 8a to 8e compares the effect of random attack, Random DOPE attack and DOPE attack on the error in the value function estimates of BRM, WIS, PDIS, CPDIS and WDR methods (left to right) in Continuous Gridworld domain.

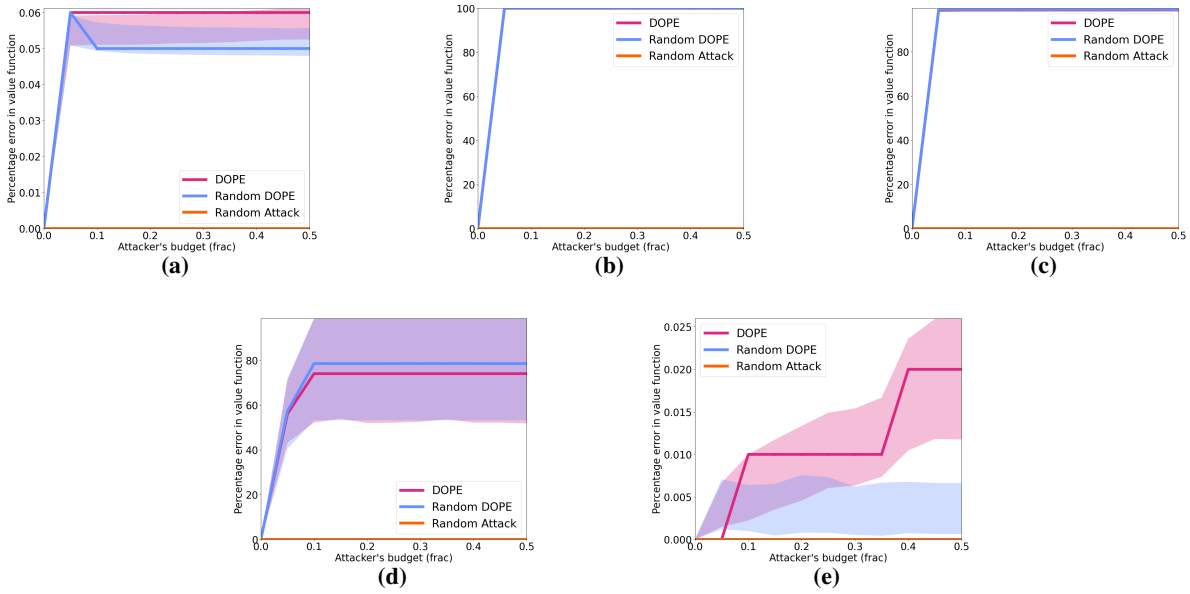
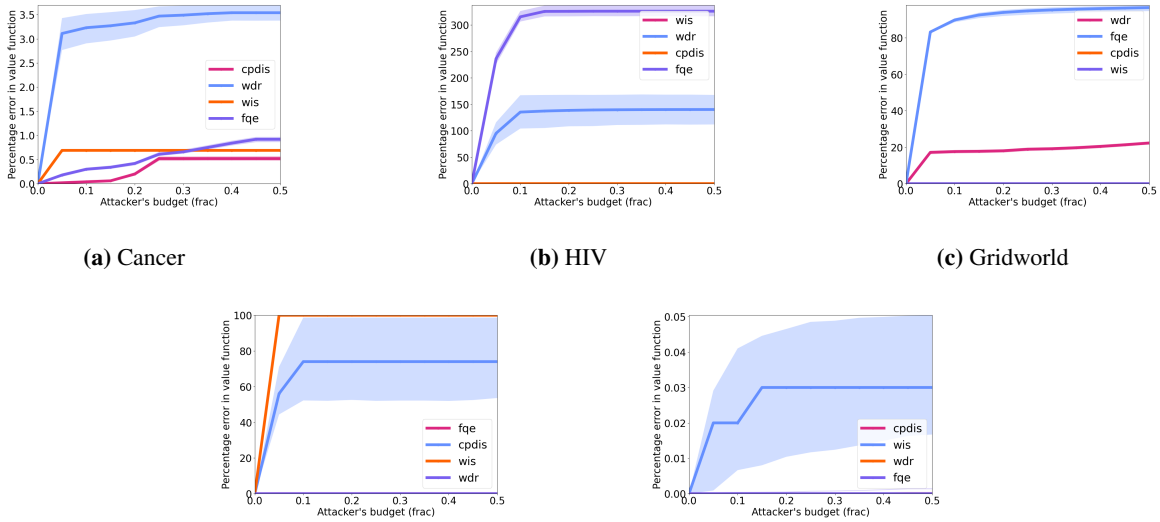


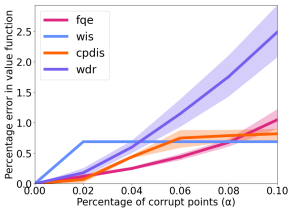
Figure 9: Figures 9a to 9e compares the effect of random attack, Random DOPE attack and DOPE attack on the error in the value function estimates of BRM, WIS, PDIS, CPDIS and WDR methods (left to right) in MountainCar domain.



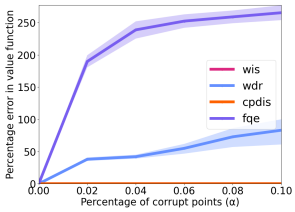
(a) Cancer **(b)** HIV **(c)** Gridworld

(d) MountainCar **(e)** Cartpole

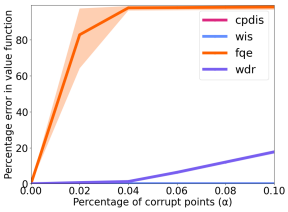
Figure 10: Figures 10a to 10e compares the effect of DOPE attack on BRM, WIS, PDIS and CPDIS and WDR methods in the Cancer, HIV and Continuous Gridworld, Mountain Car and Cartpole domains (left to right) for different values of attacker's budget $\varepsilon = \text{frac} \cdot \sigma$ and $p = 2$ (ℓ_2 norm). Larger the value of frac, the larger are the perturbations added by the DOPE attack, and accordingly we observe larger errors in the value estimates.



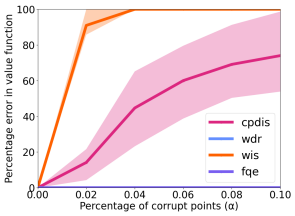
(a) Cancer



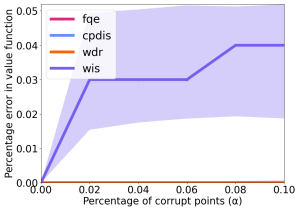
(b) HIV



(c) Gridworld



(d) MountainCar



(e) Cartpole

Figure 11: Figures 11b to 11e compares the effect of DOPE attack on BRM, WIS, PDIS and CPDIS and WDR methods in Cancer, HIV and Continuous Gridworld, Mountain Car and Cartpole domains (left to right) for different percentages of corruption α at $\varepsilon = 1.0\sigma$ and $p = 2$ (l_2 norm). Larger the value of α, the larger is the number of points perturbed by the DOPE attack, and accordingly we observe larger errors in the value estimates.