Data Poisoning Attacks on Off-Policy Policy Evaluation Methods

Abstract

Off-policy Evaluation (OPE) methods are crucial to evaluating policies in high-stakes domains such as healthcare where exploration is often infeasible 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 in 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 of the 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 logged data **????**. These algorithms are instrumental in high-stakes decision problems such as medicine and finance, where exploration is often infeasible or too expensive **??**. In such cases, one must estimate the value of a policy by only using a batch of transition data collected using a fixed behavior policy. In addition, the exact behavior policy is often unknown too **?**. If the value corresponding to a policy as estimated by the OPE methods is sufficiently high, the stakeholders will deploy the policy, and otherwise, they reject it. It is therefore essential that the OPE methods do not severely overestimate values of bad policies **?** or underestimate the values of good policies.

The sensitivity of OPE methods to adversarial contamination is not well understood yet. The complexity of OPE methods can enable an attacker to introduce large errors in OPE estimates with only small perturbations. For example, since the value of a policy at a given state is dependent on its value in all other future states, even small errors in the value-estimates of these later states can accumulate and result in large errors in the value estimates at the initial states, where important strategic decisions are often made. Thus, an attacker can design an effective attack model that exploits this property to make a significant impact.

Importance sampling methods for OPE are also susceptible to errors in the importance sample weights. Some OPE methods **???** use importance-sampling weights to correct for the shift in the data observed when evaluating a given policy with data collected using a different policy. These weights are highly dependent on the behavior policy probabilities. The attacker can thus perturb the data so that the agent wrongly estimates the behavior policy and hence, introduce large errors in the value estimate of a given policy. Therefore, these vulnerabilities warrant a thorough analysis of the effect of data-poisoning attacks on OPE methods. Although several prior works have investigated the effect of adversarial attacks on policy learning **???** in online and batch RL settings, they mainly work focus on teaching an agent to learn an adversarial policy or driving the agent to an adversarial state **???** and does not specifically investigate the effect of these attacks on OPE methods.

In this paper, we answer the following question: Can we add small perturbations to training data that significantly change the estimate of the value of a given policy? We propose a generic data poisoning attack framework for OPE methods. The framework constructs strong adversarial perturbations by leveraging the influence function tool from robust statistics **????**. Influence functions have been popularly in Machine Learning and Robust statistics to estimate

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

the effect of small perturbations in the data on an empirically learned estimator **??**. However, they have not been widely explored in the context of RL. To the best of our knowledge, our work is the first to study data-poisoning attacks on a wide range of OPE methods.

As our main contribution, we formalize the problem of data-poisoning attacks on five OPE methods - Bellman Residual Minimization (BRM) **?**, Weighted Importance Sampling (WIS), Weighted Per-Decision IS (WPDIS) **???**, Consistent Per-Decision IS (CPDIS) **?**, and Weighted Doubly Robust methods (DR) **?**. We propose a generic adversarial attack framework and show how it can be used to attack diverse model-free OPE methods. We empirically evaluate our attack framework on two medical domains, one synthetic domain, and two control domains. Through experiments, we demonstrate that by corrupting only 3%-5% of the observed states, we can achieve more than 900% and 180% error in the estimate of the value function of the optimal policy in the Cartpole and MountainCar domains, respectively. Our experimental results also show that out of the five OPE methods, DR is the least robust, and CPDIS is the most robust to such train-time adversarial attacks. Finally, our results question the reliability of policy values derived using OPE methods and strongly suggest 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). A 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 $v^\pi : \mathcal{S} \to \mathbb{R}$ at state s is the expected discounted returns of the policy starting from state s and is given by $v^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s'_{t+1}) | \pi, s\right]$. 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 $q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ 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 is the unique fixed point of the Bellman operator $\mathcal{T}^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ defined as

$$(\mathcal{T}^\pi q)(s, a) := R(s, a, s') + \gamma \sum_{s', a'} P(s, a, s') \pi(a'|s') q(s', a'). \tag{1}$$

We assume the standard batch RL setting (**?**), where the agent is given a batch of $n = N \times T$ transition tuples $D = ((s^i_j, a^i_j, r^i_j)^T_{j=1})^N_{i=1}$, 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. For all states s, we denote by $\xi(s) \in \mathbb{R}^d$ the features of state s. To ease notation in **??**, we define the state-action feature vector $\phi(s, a) \in \mathbb{R}^{|\mathcal{A}|d}$ as a vector containing state features $\xi(s)$ at the indices corresponding to a and zero elsewhere, i.e. $\phi(s, a)[ad : (a+1)d] \leftarrow \xi(s)$. Then, $\Phi \in \mathbb{R}^{n \times d}$ denotes the sample feature matrix 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 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 **?**.

Direct Methods estimate the value of the evaluation policy by solving for the fixed point of the Bellman Equation (**??**) 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 **??**. This method solves a sequence of supervised learning problems with state-action features $\phi(s, a)$ as 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),

$$\text{MSBR}(\eta) = \|q_\eta - (\mathcal{T}^\pi) q_\eta\|^2_U. \tag{2}$$

where the Q-value function q is parameterized by parameters η. $U = \text{diag}[\mu^\pi]$ where $\mu \in [0, 1]^S$ represents the stationary state distribution of policy π. The value of a policy can then be computed as $v_{BRM} = \sum_{s \in D_0} p_0(s) \cdot \sum_{a \in \mathcal{A}} \sum_{a \in \mathcal{A}} \pi(s, a) \cdot q_\eta(s, a)$.

Importance Sampling Methods (IS) **??** 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 *Weighted Per-Decision, Consistent Per-Decision*, and *Weighted* IS methods (WPDIS, CPDIS, WIS) **???**. Let $g^i_T = \sum_{t=0}^{T} \gamma^t r^i_t$ represent the returns observed for the i^{th} 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)**?**. Further, let $\rho^i_{0:t} = \prod_{t'=0}^{t} {}^{\pi(a^i_{t'}|s^i_{t'})}/_{\pi^{\theta_b}_b(a^i_{t'}|s^i_{t'})}$ where $\forall s \in \mathcal{S}, a \in \mathcal{A}$, $\pi^{\theta_b}_b(a|s) = {}^{e^{\phi(s,a)\theta_b}}/_{\sum_{a' \in \mathcal{A}} e^{\phi(s,a)\theta_b}}$, represent the importance sampling weights for time step t,

then the WIS, PDIS and CPDIS value-function estimates are defined as,

$$v_{is} = \frac{1}{\sum_{i=1}^{N} \rho^i_{0:T}} \sum_{i=1}^{N} \rho^i_{0:L} g^i_T, \tag{3}$$

$$v_{pdis} = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \gamma^{t-1} \rho^i_{0:t} r^i_t, \tag{4}$$

$$v_{cpdis} = \frac{1}{N} \sum_{t=1}^{T} \gamma^{t-1} \frac{\sum_{i=1}^{N} \rho^i_{0:t} r^i_t}{\sum_{i=1}^{N} \rho^i_{0:t}}. \tag{5}$$

Doubly Robust Estimator *Hybrid Methods* combine both Direct and IS methods to generate value estimates with low bias and variance. The *Doubly Robust* (DR) estimator **?**, for example, decreases the variance in the IS estimate by using the estimate from a Direct method like BRM as a control variate. Further, the DR estimator is guaranteed to be consistent under relaxed assumptions. The DR estimator given by

$$v_{dr} = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T-1} (\rho^i_{0:t} r^i_t - \rho^i_{0:t} q_{\theta_q}(s^i_t, a^i_t) + \rho^i_{0:t} v_{\theta_q}(s^i_t)). \tag{6}$$

where $v_{\theta_q}(s^i_t) = \sum_{a \in \mathcal{A}} \pi(s,a) \cdot q_{\theta_q}(s,a)$. Here the parameters of the value function q is estimated using Direct OPE methods like BRM.

Based on empirical studies, there are no clear winners among the three methods (**?**). Therefore, we illustrate our attack 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 **??**. 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 . For the attack to be unnoticeable, the attack can only perturb α fraction of the transitions in D while conforming to some perturbation budget $\epsilon \geq 0$ 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 for the policy being evaluated the attacker may choose to decrease or increase its estimated value 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: (a) **Features (Ψ) :** the part of the transitions targeted by the attack. (b) **Value estimation function (ρ) :** function used by OPE method for computing the value of the policy. (c) **Estimator (θ) :** model parameters learned by the OPE method from the data. We formally define each component in detail in _methods@cref_methods@cref_methods@cref**??**. (d) **Loss function (L) :** loss optimized by the OPE method for model-fitting.

We can now formulate our attack model as

$$\begin{aligned}
\underset{\Delta \in \mathbb{R}^{n \times Q}}{\text{maximize}} \quad & \rho(\theta, \Psi + \Delta) - \rho(\hat{\theta}, \Psi) \\
\text{subject to} \quad & \theta \in \underset{\theta' \in \mathbb{R}^P}{\arg\min}\ L(\theta', \Psi + \Delta) && \text{(7a)} \\
& \|\delta_i\|_p \leq \epsilon, \quad i = 1, \ldots, N && \text{(7b)} \\
& \sum_{i=1}^{n} \mathbf{1}_{\|\delta_i\| \neq 0} \leq \alpha \cdot n. && \text{(7c)}
\end{aligned}$$

The DOPE objective in (**??**) increases the value estimate of the policy π to increase the error in the same. Alternatively, if the attacker desires to decrease the value function estimate of the given policy, he may do so by simply changing the sign of the objective. The constraint (**??**) estimates the optimal parameter θ from D after replacing Ψ with $\Psi + \Delta$.

The constraint (**??**) ensures that the perturbation added to the Ψ is limited to the user-defined perturbation budget ϵ. This prevents the attack framework from generating adversarial transitions that can be easily detected as anomalous. And, finally, the constraint (**??**) limits the number of transitions that the attacker can perturb. Finally, note that $\rho(\hat{\theta}, \Psi)$ in (**??**) is a constant and can be ignored while solving the optimization problem.

Next, we discuss how to adapt this framework to attack a variety of off-policy policy evaluation methods.

3.1 ATTACKING OPE METHODS USING THE DOPE FRAMEWORK

In this section, we first formally define the four components of the DOPE framework. Then, we show how our DOPE framework can be adapted to attack five distinct OPE methods discussed in **??**.

We denote by $\psi(s, a, r, s') \in \mathbb{R}^Q$, an arbitrary component of the transition tuple $\langle s, a, r, s' \rangle$ in D, that is selected for perturbation by the attacker. We note that $\psi(s, a, r, s')$ could either be the state features $\xi(s)$ or the reward r vector. We use $\Psi \in \mathbb{R}^{n \times Q}$ represent the corresponding sample matrix constructed from D. We denote by $\hat{\theta}(\Psi) \in \mathbb{R}^P$ the parameter of interest for a given OPE method, that is empirically estimated from the data D. It is important to note that the parameter $\hat{\theta}(\Psi)$ could represent different components in different OPE methods, for example, θ represents the parameters of the value-function θ_q in BRM and the parameters of the estimated behavior policy θ_b in IS. We use $L(\theta, \Psi)$ with

$L(\dot{;}) : \mathbb{R}^P \times \mathbb{R}^{n\times Q} \to \mathbb{R}$ to represent the empirical loss function used by the OPE method to derive the optimal parameter $\hat{\theta}(\Psi)$, i.e., $\hat{\theta}(\Psi) \in \arg\min_{\theta' \in \mathbb{R}^P} L(\theta', \Psi)$. L in BRM and DR is the MSBR whereas in IS methods, L represents the MLE loss optimized for finding the behavior policy parameters. We use $\rho(\hat{\theta}(\Psi), \Psi)$ with $\rho : \mathbb{R}^P \times \mathbb{R}^{n\times Q} \to \mathbb{R}$ to represent the function used by the OPE method to compute the mean value of the evaluation policy π at the initial states. For example, ρ in BRM represents v_{brm}.

We will interchangeably use the shorthand $\rho(\Psi) := \rho(\hat{\theta}(\Psi), \Psi)$. Further, it is worth noting that $L(\theta, \Psi)$ and $\rho(\theta, \Psi)$ may also depend on other components of D which remain fixed throughout the attack and hence, they are excluded from the definitions of functions ρ and L. It is important to note that $L(\theta, \Psi)$ is required to be twice continuously differentiable and linearly separable with respect to the transitions in D and $\rho(\theta, \Psi)$ is required to be continuously differentiable with respect to θ and ψ. These assumptions, as we see in **??**, are important for computation of influence-functions **?**.

Using the afforementioned definitions, we summarize our attack on the five methods discussed in **??** in **??**.

4 OPTIMIZING THE DOPE OBJECTIVE

There are two major challenges that make the optimization problem in (**??**) difficult. First, the third constraint in (**??**) is non-differentiable and requires the attacker to select a set of at most αn transitions which we denote by S_α, such that perturbing these transitions results in maximum change in the value of the policy, in the desired direction. We denote this set of transitions by S_α. It is important to realize that finding this set requires perturbing all possible subsets of data Ψ whose size is not greater than αn and computing the optimal parameter θ for each perturbation. Thus, the number of such subsets is much larger than $\binom{n}{\alpha n}$ and therefore computing this set is computationally expensive and practically infeasible **?**. Second, the inner-level optimization problem in **??** is often non-linear for OPE methods which makes the bilevel-attack formulation an NP-Hard problem **?**. We address these two problems by deriving an approximation of the bilevel optimization problem in **??** using the Taylor expansion **??** and show that the resultant problem is simpler to optimize and has a closed form solution. In **??**, we empirically demonstrate the effectiveness of our approximate solution on several domains.

We being by defining the influence score of the i^{th} data point $I_{\Psi_i,\theta,\Psi} = \nabla_{\delta_i}\rho(\Psi)$ as the approximate influence of perturbing the i-th data point by δ_i on $\rho(\Psi)$. Then, using the Taylor expansion of $\rho(\Psi + \Delta)$, we can approximate the net error in the value-function estimate as the weighted sum of the influence-scores of individual data points. $\rho(\Psi + \Delta) - \rho(\Psi)$ as

$$\rho(\Psi + \Delta) - \rho(\Psi) \approx \sum_{i=1}^{n} (\nabla_{\delta_i}\rho(\Psi))^\top \delta_i. \tag{8}$$

Substituting **??** in **??**, our problem boils down to

$$\begin{aligned} &\max_{s\in\{0,1\}^n} \max_{\{\delta_k\}_{k=1}^N \in \mathbb{R}^{n\times Q}} \sum_{k=1}^{n} s_i \cdot I_{\Psi_i,\theta,\Psi}^\top \delta_k \\ &\text{subject to } \sum_{k=1}^{n} s_k = \alpha \cdot n, \\ &\qquad \|\delta_k\|_p \le \varepsilon \cdot s_k,\ k = 1, \dots n. \end{aligned} \tag{9}$$

Here, $s \in \{0,1\}^N$ is a vector of binary indicators such that $s_i = 1$ indicates that the i^{th} transition is amongst the αn transitions selected for perturbation. We can now compute an approximately optimal set of perturbations in polynomial time, as we show in **??**.

Proposition 4.1. *Let (s^*, Δ^*) be the optimal solution to the problem in* (**??**)*. Define the* Approximate Influential Set S_α *as $S_\alpha = \{i : s_i^* = 1, \forall i \in [1, \dots, n]\}$. Then,*

1. *S_α can be constructed by choosing the set of αn transitions with the largest q-norm of their influence scores $I_{\psi,\theta,\Psi}$.*
2. *For all $k \in [1, \dots n]$, the optimal δ_k^* for $p = 1, 2, \infty$ can be computed as*

$$\begin{aligned} &\textit{If } p = 1,\ \forall j \in [1, Q], \\ &\delta_{k,j} = \begin{cases} \epsilon\, if\ j \in \arg\max_{j\in[1,Q]} I_{\Psi_k,\theta,\Psi}(j) \\ 0\ \textit{otherwise} \end{cases} \\ &\textit{If } p = \infty,\ \textit{then}\ \ \delta_k = \epsilon * \text{sign}(I_{\Psi_k,\theta,\Psi}) \\ &\textit{If } p = 2,\ \textit{then}\ \ \delta_k = \epsilon \cdot \frac{I_{\Psi_k,\theta,\Psi}}{\|I_{\Psi_k,\theta,\Psi}\|_2}. \end{aligned} \tag{10}$$

We remark that the optimal perturbations Δ^{**} for the approximate problem in **??**, when substituted in the objective of **??** gives us a lower bound on the maximum error that can be achieved in the value-function estimate while constraining to the specified budget, i.e., $\rho(\Psi + \Delta^*) - \rho(\Psi)$, where Δ^* is the optimal solution to **??**. This simply follows from the construction of our original problem (**??**).

$$\begin{aligned} err &= \rho(\Psi + \Delta^{**}) - \rho(\Psi) \\ &= \max_{\Delta\in\mathbb{R}^{n\times Q}} \rho(\Psi + \Delta) - \rho(\Psi) \\ &\ge \rho(\Psi + \Delta^*) - \rho(\Psi) \end{aligned} \tag{11}$$

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

DOPE Attack Templates				
Method	Estimator θ	Features Ψ	Value Estimation Function $\rho(\Psi)$	Loss $L(\theta, \Psi)$
BRM **?**	θ_q	Φ or r	v_{brm}	MSBR
WIS **??**	θ_b	Φ or r	v_{wis}	MLE
WPDIS **?**	θ_b	Φ or r	v_{wpdis}	MLE
CWPDIS **?**	θ_b	Φ or r	v_{cwpdis}	MLE
DR **?**	θ_b, θ_q	Φ or r	v_{DR}	MLE + MSBR / MSBR

Figure 1: DOPE Attack Templates for OPE Methods.

calculus, we get

$$\forall i \in [1 \dots n],\ I_{\Psi_i,\theta,\Psi} \approx \left(\frac{\partial \rho(\theta, \Psi)}{\partial \delta_i}\bigg|_{\hat{\theta}(\Psi)} + \frac{\partial \rho(\theta, \Psi)}{\partial \theta}\bigg|_{\hat{\theta}(\Psi)} \frac{\partial \hat{\theta}(\Psi)}{\partial \delta_i} \right). \tag{12}$$

The partial derivative $\frac{\partial \theta(\Psi)}{\partial \delta_i}$ is the effect of perturbing Ψ_i by a small δ_i on the parameter θ, which can be approximately computed using the influence-function in (**??**) as ${}^{\partial\theta(\Psi)}/_{\partial\delta_i} = H^{-1}_{\hat{\theta}(\Psi)} {}^{\partial L(\theta,\Psi_i)}/_{\partial\theta\partial\Psi_i}\big|_{\hat{\theta}(\Psi)}$ where $H_{\hat{\theta}(\Psi)} = {}^{\partial^2 L(\theta,\Psi)}/_{\partial\theta^2}\big|_{\hat{\theta}(\Psi)}$. For more details on the influence functions, please see Influence functions in **??**.

Thus to compute $I_{\Psi_i,\theta,\Psi}$ in **??**, we require that $L(\theta, \Psi)$ is required to be twice continuously differentiable and linearly separable with respect to the transitions in D and $\rho(\theta, \Psi)$ to be continuously differentiable with respect to θ and ψ Although, these conditions may seem restrictive, they hold true for many of the OPE methods we discuss.

The derivatives in (**??**) can be easily computed using Python automatic-differentiation software like Pytorch, Tensorflow **??**. The hessian-inverse vector product $H^{-1}_{\hat{\theta}(\Psi)}$ can be approximately computed in $O(NP)$ time using Pearlmutter's method **?** for fast Hessian-vector product and Taylor approximation of the inverse of Hessian matrix as shown in **?**.

5 ALGORITHM

Input: Batch of data D, attack budget ϵ, % of corrupt transitions α, norm-type p, threshold μ

Construct Ψ from D
Set $\Psi^{cor} \leftarrow \Psi$
$S_\alpha \leftarrow []$
Compute $\|I_{\Psi_k,\theta,\Psi}\|_q$ for all $k \in [1, \dots, n]$ s. t. $\frac{1}{p} + \frac{1}{q} = 1$
Set $S_\alpha \leftarrow$ Indices of top αn transitions with the largest $\|I_{\Psi_k,\theta,\Psi}\|_q$

repeat
  $\theta \leftarrow \arg\min_{\theta' \in \mathbb{R}^P} L(\theta', \Psi^{cor})$
  for $k \in S_\alpha$ **do**
    Compute $I_{\Psi^{cor}_k,\theta,\Psi^{cor}}$ using (**??**).
    Compute $\delta^*_k \in \arg\max_{\delta \in \mathbb{R}^Q} I^T_{\Psi^{cor}_k,\theta,\Psi^{cor}} \delta$ s. t. $\|\Psi_k - (\Psi^{cor}_k + \delta)\|_p \leq \epsilon$
    Use line-search to find step-size β such that $\rho(\theta, \Psi^{tmp}) - \rho^k(\theta, \Psi^{cor}) < 0$ where Ψ^{tmp} is constructed by replacing Ψ^{cor}_k with $(1-\beta)\Psi^{cor}_k + \beta(\Psi_k + \delta^*_k)$ in Ψ^{cor}
    Set $\Psi^{cor} \leftarrow \Psi^{tmp}$
  end
until $|\rho^{k+1}(\Psi^{cor}) - \rho^k(\Psi^{cor})| \leq \mu$;
return D'

We outline our DOPE framework in **??**. To summarize, our algorithm for approximately solving (**??**) consists of two main steps. In the first step, we compute an approximation of the optimal set of transitions to perturb $\hat{S}_\alpha$ by choosing αn transitions in Ψ with the largest q-norm of their influence scores $\|I_{\psi,\theta,\Psi}\|_q$. In the second step, we compute the steepest feasible descent direction δ for all transitions in $\hat{S}_\alpha$ and use line-search to update the transitions in $\hat{S}_\alpha$. The second step may be repeated until no further perturbation to transitions in $\hat{S}_\alpha$ decreases $\Delta\rho$.

6 ADDITIONAL RESULTS

Next, we show how to avoid expensive influence function computational costs for BRM method by deriving closed form expressions for the influence score of data points in linear BRM method, under two settings a) when the adversary perturbs only the state features b) When the adversary perturbs the reward features.

Proposition 6.1. *If the attacker only perturbs the reward vector r constructed from batch of transition tuples D. Then, the influence score of the i^{th} data point $I_{r_i,\theta,\Psi}$ for the FQE*

method can be computed as

$$I_{r,\theta,\Psi} = 4\|\Phi^T(\Phi - \gamma\Phi^c{}_p)\|^2_{(\Phi^T\Phi)^{-1}} \cdot \Phi\left((\Phi^T\Phi)^{-1}(\Phi^T\Phi - \gamma\Phi^T\Phi_p)\right) \tag{13}$$

Proposition 6.2. *If the attacker only perturbs the state feature matrix Φ. Then, the influence score of the i^{th} data point $I_{\phi(s_i,a_i),\theta,\Psi}$ for the FQE method can be computed as*

7 EXPERIMENTS

In this section we conduct numerous ablation studies to identify the strengths and weaknesses of our attack framework. Specifically, we aim to answer the following key questions through our experiments.

- What should be the range of the attacker's budget for an effective attack?
- Do the perturbed data points have to be outliers to introduce large errors in the value function estimate?
- What percentage of data points need to be perturbed for an effective attack?
- What effect does the discount factor have on the impact of the attack?
- How does the attack impact the importance sampling weights?
- Can deletion of influential data points be an effective defence mechanism against data poisoning attacks?
- How does a random perturbation perturbation attack compare to our data poisoning attack?
- Is there an advantage of perturbing influential data points over perturbing randomly selected data points?

8 RELATED WORK

9 CONCLUSION AND FUTURE WORK

A NOTATION

We will use $\Delta = (\delta_i)_{i=1}^N$ to denote the perturbation matrix where $\delta_i \in \mathbb{R}^Q$ is a vector of perturbations added to Ψ_i.

B PROOFS

C EXPERIMENTAL RESULTS

Cancer: The cancer domain models the growth of tumors in cancer patients subject to chemotheraphy. The domain consists of 4-dimensional states that represent the growth-dynamics of the tumor in the patient ad 2 actions that indicates if a given patient is to be administered chemotheraphy or not at a given time step. Each episode is of length 30. The dataset D comprises of 80 trajectories collected using the behaviour policy.

HIV: The HIV domain has 6-dimensional states representing the state of the patient anf 4 actions that represent 4 different types of treatments. Each episode is of a fixed length $T = 50$. The dataset D comprises of 80 trajectories collected using the behaviour policy.

Mountain Car: In the Mountain Car 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 represent the current position of the car and the current time-step and 3 actions represent - drive forward, drive backward and don't move. The average length of the episodes is 150. We collect 80 trajectories by simulating the behavior policy.

Cartpole: The cartpole domain 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 2- actions represent the force applied to the pole. We construct D with 50 trajectories collected using the behavior policy.

C.1 WHAT SHOULD BE THE RANGE OF ATTACKER'S BUDGET?

In this experiment, we investigate how much budget should the attacker be allotted to observe a significant change in the value of the optimal policies in the 4 domains listed above. For this, we fix the percentage of transitions perturbed to 0.05 across all domains. We vary the perturbation budget from 0.01 to 0.25 in steps of size 0.04. Since the objective optimized by our framework is non-convex, we found that it often gets stuck in local optima which makes it difficult to observe a clear trend in the percentage change in the value estimates with change in the perturbation budget. To resolve this issue, we warm start each optimization with the solution obtained using a lower budget value. Further, in FQE, we allow the attacker to perturb transitions in a way that does not result in large train-errors. We set a threshold of 100 on the mean projected Bellman error. **??** shows how the change in the value-function varies with the attacker's budget.

C.2 DO THE PERTURBED TRANSITIONS LOOK ANOMALOUS?

A natural question that arises is whether the perturbed transitions have to be outliers to make a significant impact on the value-function estimate. **??**, **??**, **??** shows the transitions of a subset of trajectories before and after the data-poisoning attack on FQE, Importance Sampling and Doubly Robust method respectively. The attack budget ϵ is set to 0.05 and

Domain	Most Affected Method	Percentage Error in Value function
Cancer	FQE/DR	22%
Custom	DR	7%
HIV	DR	2500%
Cartpole	DR	900%
MountainCar	DR	180%

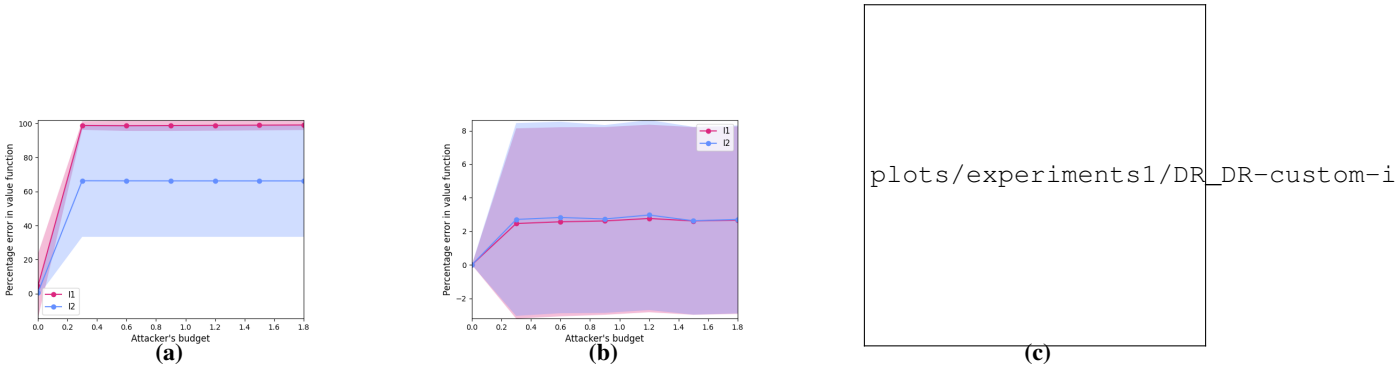

Figure 2: ??, ??, ?? shows the relationship between the errors in the value estimates given by FQE, IS and DR (left to right) and attacker's budget ϵ. On an average, the percentage of error in the value function estimate increases with increase in the budget.

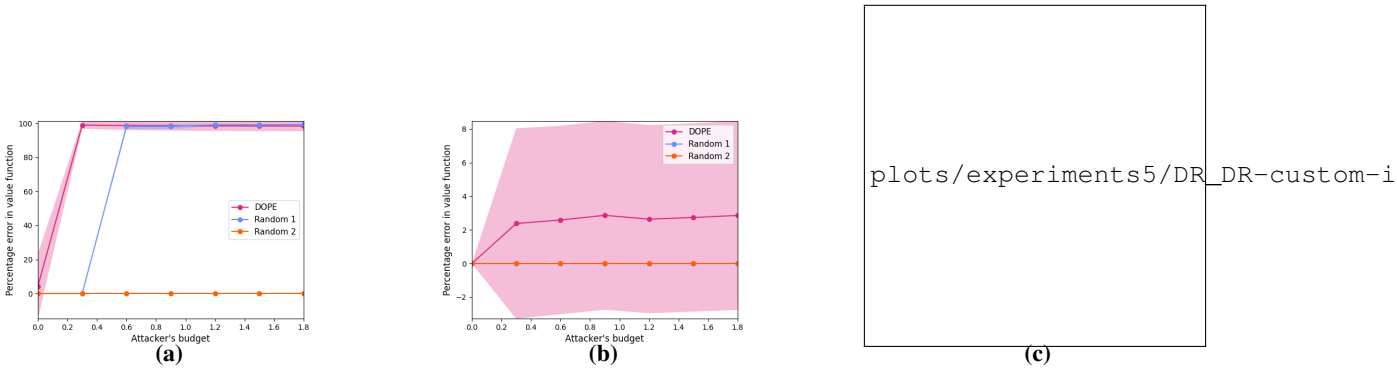

Figure 3: ??, ??, ?? compares the effect of random perturbations to adversarial perturbations constructed using the DOPE framework on the error in the value function estimates of FQE, IS and DR methods (left to right).add description

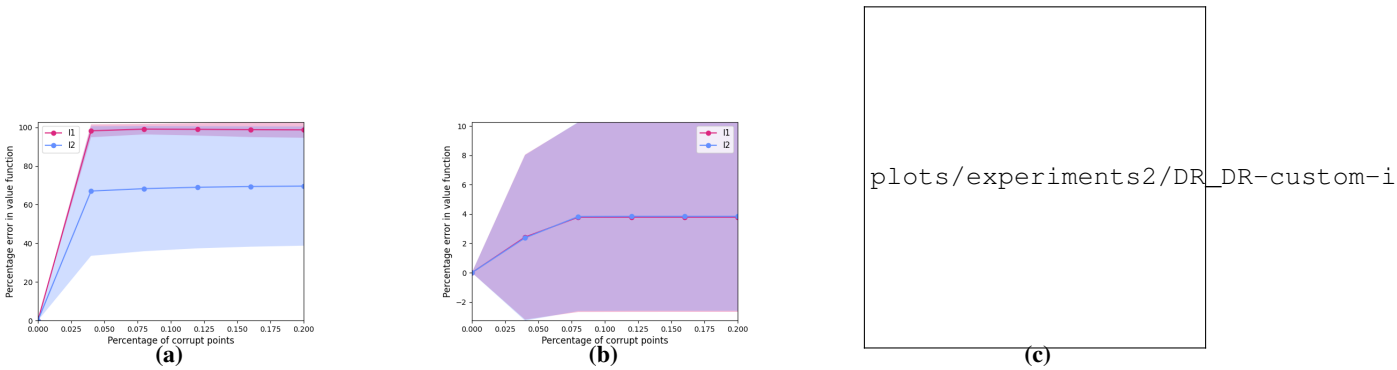

Figure 4: ??, ??, ?? shows the relationship between the errors in the value estimates given by FQE, IS and DR (left to right) and percentage of corrupt data points α.

the percentage of corrupt transitions α is set to 0.05. Notice that across all 4 domains and all 3 OPE methods, most

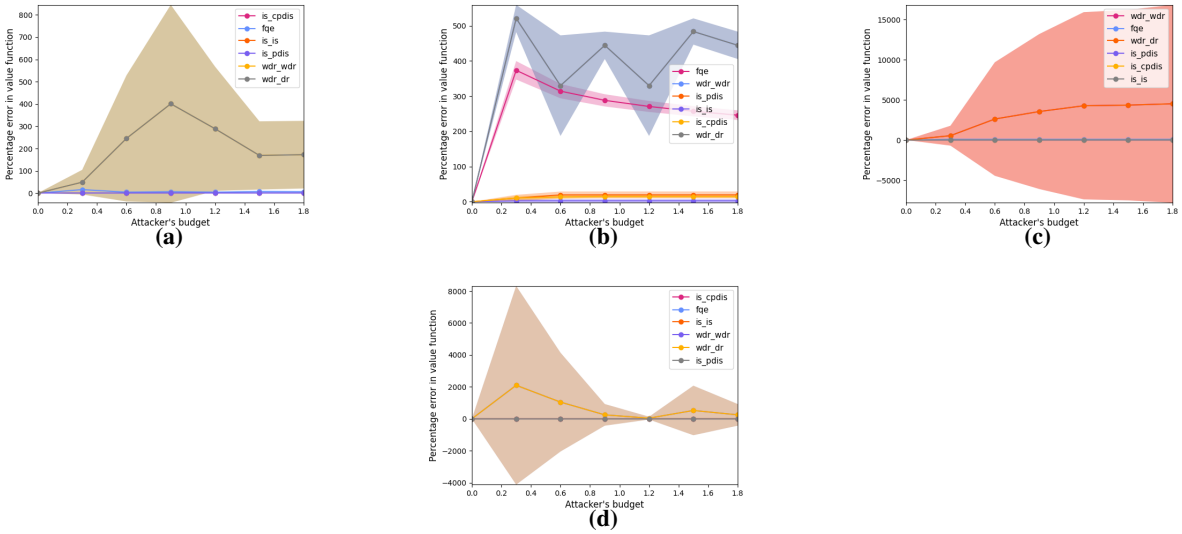

Figure 5: ??, ??, ??, ?? compares the effect of DOPE attack on DR, FQE, IS, PDIS and CPDIS methods in Cancer, HIV, Custom, Cartpole domains (left to right).

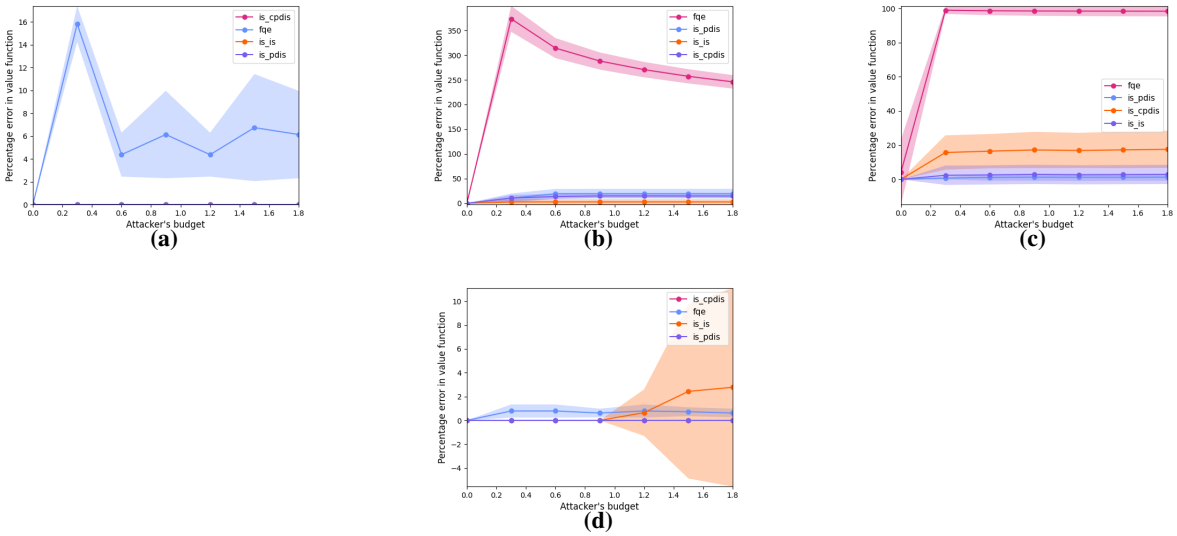

Figure 6: ??, ??, ??, ?? compares the effect of DOPE attack on FQE, IS, PDIS and CPDIS methods in Cancer, HIV, Custom, Cartpole domains (left to right).

of the perturbed transitions do not look resemble outliers. However, these perturbed transitions results in astonishingly large change in the value-function estimate as shown in **??**. Overall, these experimental results reveal that weak outlier detection methods are likely to not detect the adversarial transitions in the dataset D.

C.3 WHAT PERCENTAGE OF TRANSITIONS SHOULD THE ATTACKER PERTURB?

In this experiment, we vary the percentage of transitions that the attacker is allowed to corrupt in D. **??** shows that, overall in FQE and DR method, corrupting x% of the points is sufficient to observe a large change in the value-function estimate. On the other hand, IS methods require much larger the attacker to corrupt much larger percentage of transitions to observe a significant difference in the value estimates.

C.4 EFFECT OF DISCOUNT FACTOR/HORIZON

A large discount-factor value indicates that the value-function at a given state has a larger dependence on future rewards as compared to when the discount-factor has a smaller value. We can thus expect the impact of the attack

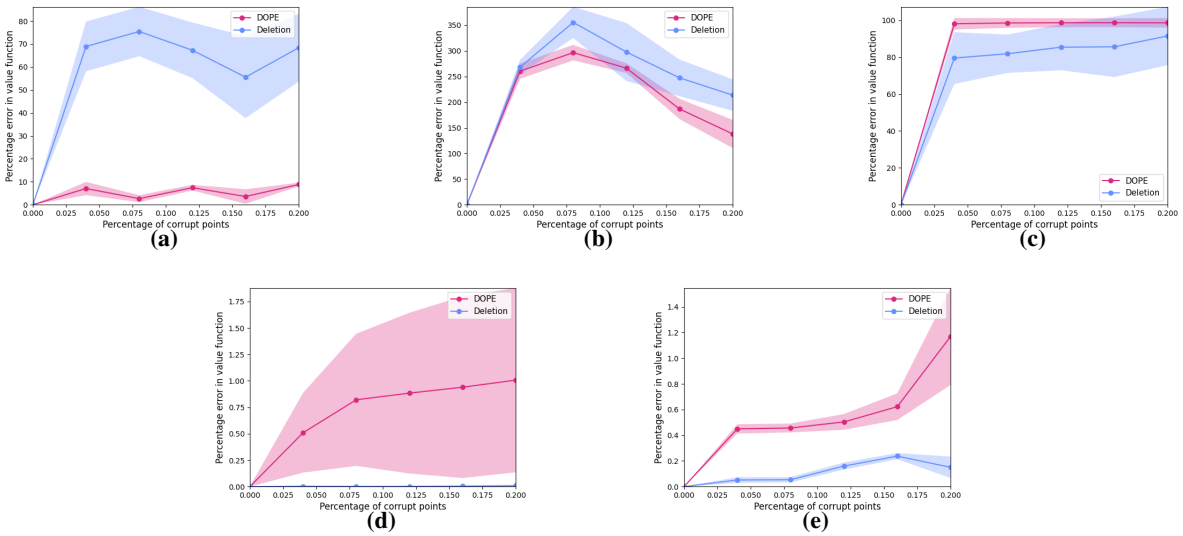

Figure 7: **??**, **??**, **??**, **??**, **??** compares the effect of deletion as a defence mechanism for FQE method in Cancer, HIV, Custom, Cartpole and MountainCar domains (left to right).

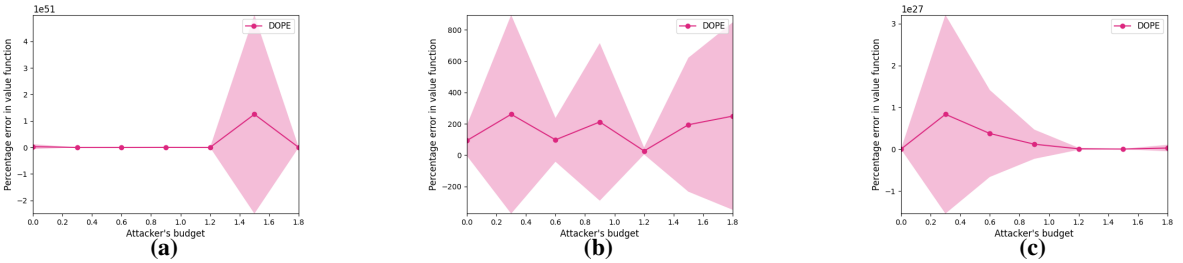

Figure 8: **??**, **??**, **??** shows the percentage change in importance sampling weights with change in budget for IS, PDIS and CPDIS methods (left to right).

to be larger for larger values of discount-factor especially in FQE and Doubly Robust methods. To empirically validate this conjecture, we set α to 0.05 and vary the budget ϵ from 0.0 to 0.25 in steps of size 0.04. **??** shows that, overall, the impact of attack does indeed increase with increase in the value of the discount-factor. This means that the our attack model correctly exploits this vulnerability of RL to have a larger impact on the value-function.

C.5 IMPACT OF ATTACK ON IMPORTANCE SAMPLING WEIGHTS

Next, we evaluate if our attack model correctly corrupts the behavior policies to introduce large errors in the value estimate of the evaluation policy. For this, we plot the importance-sampling weights before and after the attack on IS method in the Cancer and Mountain Car domains. We focus only on these two domains since the rewards in the Cancer domain is always positive and the rewards in the Mountain Car domain is always negative. Hence, it is easier to predict the direction of change in weights in both of these domains. In the Mountain Car domain, we can expect the attack model to achieve a significant decrease in the value-function estimate by lowering the behavior probabilities for the observed samples. We set a threshold equal to 0.01 on the behavior probabilities to avoid any suspicion. We can thus expect the importance-sampling weights to increase after the attack in both the domains. To observe this effect, we set $\alpha = 0.5$ and $\epsilon = 0.25$ and solve for maximizing the value in the case of Cancer domain and minimizing the value in the case of Mountain Car domain. The results in **??** demonstrate that the weights indeed increase significantly after the attack indicating that our attack-framework correctly exploits the behavior policy to achieve its objective.

C.6 DEBUGGING ADVERSARIAL ATTACKS

Recently, leave-one-out error also known as influence of deleting a data point on the value-function estimate was proposed as a diagnostic tool for identifying adversarial transitions **?**. We investigate if deleting $\alpha\%$ of the transitions with the largest leave-one-out error can serve as a

naive defence-mechanism against data-poisoning attacks. In **??**, we provide a histograms of the leave-one-out error computed on datasets with $\alpha\%$ corrupt data points. Note that deleting transitions in IS methods is not feasible as it would make the trajectories containing those transitions redundant for the value-function estimation. Hence, we only consider this defence mechanism for FQE method. **??** displays the value of the policy before the attack, after the attack and after deleting αn transitions with the largest q-norm of their influence.

C.7 INFLUENCE ATTACK VS RANDOM ATTACK

In this experiment, we compare our DOPE attack with a Random Attack, wherein the attacker adds random perturbations to randomly selected αn transitions. 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 value of the budget ϵ, we average the percentage change in the value estimate of the policy observed in a random attack over 100 trials. The experimental results in **??** demonstrates that in contrast to the DOPE attack, the Random Attack fails to make a significant change in the value-function estimate and therefore, can be used as an alternative to our attack model.

C.8 RANKING OPE METHODS BY THEIR SENSITIVITY TO DATA-POISONING ATTACKS

In this experiment we evaluate and compare the robustness of FQE, IS, PDIS, CPDIS and DR methods. We judge the robustness of each method by comparing the average of the percentage change in the value estimates observed after the attack, computed over N trials. **??** shows the average percentage change in the value estimates observed in all 5 methods for different values of the budget. These results demonstrate that, as expected, DR is least robust while CPDIS is most robust to the data-posioning attacks. We conjecture that the robustness of CPDIS is due to the weights at any time-step t being similar across trajectories. And thus, normalizing the weights across trajectories basically neutralises the effect of the data-poisoning attack.

C.9 CHOOSING RANDOM POINTS TO PERTURB

One might conjecture if there is any benefit to computing the influence of all transitions and then greedily select points with the largest influence instead of randomly selecting αn transitions to perturb. To investigate this, we randomly select αn transitions to perturb in each experiment and vary the budget ϵ from 0.0 to 0.25 in step size of 0.4. We set $\alpha = 0.05$ for all the domains. The selected transitions are then perturbed according to the update in **??**. **??** displays the change change in the value-function estimate with change in the budget ϵ.

D APPROXIMATELY OPTIMAL PERTURBATIONS

It thus follows that $\forall k \in [1, \dots N]$, the optimal perturbation δ_k^* can be independently computed by solving $\delta_k^* \in \arg\max_x I^T_{\Psi_k,\theta,\Psi} x \,\text{s.t.}\, \|x\|_p \leq \epsilon$. This is easy to compute for $p = 1, 2, \infty$ as we show in **??**.

Next, we need to solve for the optimal value of s. From the theory of convex optimization, we know that the p-norm of any vector $x \in \mathbb{R}^M$ $\|x\|_q$ can be expressed as $\|x\|_q = \max z^T x \,\text{s.t.}\, \|z\|_p \leq 1$ where $\frac{1}{p} + \frac{1}{q} = 1$. q might be confused with q function? Hence, given the optimal-perturbation $\delta_k^* \forall k \in [1, \dots n]$, the problem in **??** boils down to solving reference to be edited

$$\max_{s\in\{0,1\}^N} \sum_{k=1}^{n} \|I_{\Psi_k,\theta,\Psi}\|_q \quad \sum_k s_k = \alpha \cdot n. \tag{14}$$

lot of white space around equations needs to be fixed It is now easy to see that the optimal set of transitions for the approximate attack problem in **??** is simply the set of αn transitions with the largest value of the q-norm of their influence scores. We formally state these results in **??**.

E IMPLEMENTATION DETAILS

Implementation Details: For each domain, we generate Radial Basis features for each observed state s and next-state s' in the transition tuples $(s, a, r, s') \in D$. For simplicity, we assume that every (s, s') pair in $(s, a, r, s') \in D$ is unique. We note that,in our experiments, the attacker only perturbs original features of observed states s in the transition tuples and these perturbations do not affect the observed next-states s' in $(s, a, r, s') \in D$. Further, we use Regularized Ridge Regression to estimate the parameters of the q-value function in FQE and Regularized Logistic Regression to estimate the behavior policies in IS and DR methods. For simplicity, we assume that the behavior policy in DR is provided by the experts and not learned from D. Inspite of this assumption, the attacker can still exploit the importance-sampling weights to construct We provide the details of the hyperparameter used, in the Appendix.

F RELATED WORK

G ADDITIONAL PRELIMINARIES

H TAKEAWAYS

- With ϵ value as small as $0.5std(x_1 - x_2)$, we can

I 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 $L : \Theta \times \mathcal{X} \to \mathbb{R}$ notation for map L can be omitted 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

$$\begin{aligned} \mathcal{I}_{z_\delta,\theta,D} &= \frac{\theta_{z,\delta} - \theta}{\delta} \approx \frac{\partial \theta}{\partial x} \\ &\approx \left(-H_\theta^{-1} \frac{\partial^2 L(\theta, z)}{\partial \theta \partial x} \right) \text{ where } H_\theta = \frac{\partial^2 L(\theta, D)}{\partial^2 \theta} \end{aligned} \tag{15}$$

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 **?** for more details.