DATA POISONING ATTACKS ON OFF-POLICY POLICY EVALUATION METHODS

Elita Lobo
University of New Hampshire
eal1063@usnh.edu

Harvineet Singh
New York University
hs3673@nyu.edu

Marek Petrik
University of New Hampshire
mpetrik@cs.unh.edu

Cynthia Rudin
Duke University
cynthia@cs.duke.edu

Hima Lakkaraju
Harvard University
hlakkaraju@hbs.edu

ABSTRACT

Off-policy Evaluation (OPE) methods are crucial for 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 adversarial perturbations to the data. We design a data poisoning attack framework that leverages influence functions to construct perturbations that maximize error in the policy value estimates. Our experimental results show that many OPE methods are highly prone to data poisoning attacks, even for small adversarial perturbations.

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 exploration 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 sufficiently high will the stakeholders 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. Thus, attackers can exploit this property. 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 logging policy. 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.

In this work, we study the effect of data poisoning attacks on OPE methods. More specifically, we ask the following question: *Can we construct small perturbations to the training data that significantly change 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). We solve the optimization problem in a computationally tractable manner, we using influence functions from robust statistics (Koh and Liang, 2017). 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. Our experimental results 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.

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

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 $D{=}((s^i_j, a^i_j, r^i_j)_{j=1}^T)_{i=1}^N$, 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. We will use $r \in \mathbb{R}^{n \times 1}$ to represent the sample reward matrix.

We discuss the OPE methods targeted by our attack framework in detail in Appendix D.

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 $\epsilon \geq 0$ 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 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,

$$\underset{\Delta \in \mathbb{R}^{n \times Q}}{\text{maximize}} \quad \rho(\theta_{\text{pert}}, \Psi + \Delta) - \rho(\theta_{\text{org}}, \Psi) \tag{1a}$$

$$\text{subject to} \quad \theta_{\text{pert}} \in \underset{\theta \in \mathbb{R}^P}{\arg\min}\ L(\theta, \Psi + \Delta) \tag{1b}$$

$$\theta_{\text{org}} \in \underset{\theta \in \mathbb{R}^P}{\arg\min}\ L(\theta, \Psi) \tag{1c}$$

$$\|\delta_i\|_p \leq \epsilon, \quad i = 1, \ldots, N \tag{1d}$$

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

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

The DOPE objective in equation 1a increases the estimated value from the original value, thereby increasing the error. Alternatively, if the attacker wants to decrease the estimated value of the given policy, they may do so by simply changing the sign of the objective. The constraint equation 1b estimates the optimal parameter θ_{pert} from D after perturbing Ψ to $\Psi + \Delta$. The constraint equation 1d 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 equation 1e 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

There are two major challenges with optimizing the DOPE problem in equation 1. First, the constraint equation 1e requires perturbing all possible subsets of data Ψ whose size is at most αn and re-estimating the optimal parameter θ for each perturbation, which is computationally infeasible. Second, equation 1 is a bilevel optimization problem where the inner-level problem equation 1b 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 (1) using the Taylor expansion. We show that the resulting problem is simpler to optimize and has a closed-form solution.

Approximation We define the influence score of the i^{th} 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,

$$\rho(\Psi + \Delta) - \rho(\Psi) \approx \sum_{i=1}^{n} (\nabla_{\Psi_i} \rho(\Psi))^\top \delta_i. \tag{2}$$

Using Eq. equation 2 reduces the optimization in equation 1 to

$$\begin{aligned} &\max_{s \in \{0,1\}^n} \max_{\{\delta_i\}_{i=1}^N \in \mathbb{R}^{n \times Q}} \sum_{i=1}^{n} s_i \cdot I_{\Psi_i}^\top \delta_i \\ &\text{subject to } \sum_{i=1}^{n} s_i = \alpha \cdot n, \qquad \|\delta_i\|_p \le \epsilon \cdot s_i, \ i = 1, \dots n. \end{aligned} \tag{3}$$

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 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 equation 3 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 equation 3.*
2. *For all $i \in [1, \dots n]$, the optimal δ_i^* for $p = 1, 2, \infty$ can be computed in closed-form as*

$$\begin{aligned} &\textit{If } p = \infty, \textit{ then } \delta_i^* = \epsilon \cdot \operatorname{sign}(I_{\Psi_i}) \\ &\textit{If } p = 2, \textit{ then } \delta_i^* = \epsilon \cdot \frac{I_{\Psi_i}}{\|I_{\Psi_i}\|_2}. \\ &\textit{If } p = 1, \textit{ then } \forall j \in [1, Q], \ \delta_{i,j}^* = \begin{cases} \operatorname{sign}(I_{\Psi_i}(j)) \cdot \epsilon \textit{ if } j \in \arg\max_{m \in [1,Q]} I_{\Psi_i}(m) \\ 0 \textit{ otherwise} \end{cases} \end{aligned}$$

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

$$I_{\Psi_i} \approx \left. \frac{\partial \rho(\theta, \Psi)}{\partial \Psi_i} \right|_{\theta_{\text{org}}(\Psi)} + \left. \frac{\partial \rho(\theta, \Psi)}{\partial \theta} \right|_{\theta_{\text{org}}(\Psi)} \frac{\partial \theta(\Psi)}{\partial \Psi_i}. \tag{4}$$

The computation of the partial derivative $\frac{\partial\theta(\Psi)}{\partial\Psi_i}$ is not straightforward. However, we can approximately compute it as $\partial\theta(\Psi)/\partial\Psi_i = H^{-1}_{\theta_{\text{org}}(\Psi)} \partial^2 L(\theta,\Psi_i)/\partial\theta\partial\Psi_i\big|_{\theta_{\text{org}}(\Psi)}$ where $H_{\theta_{\text{org}}(\Psi)} = \partial^2 L(\theta,\Psi)/\partial\theta^2\big|_{\theta_{\text{org}}(\Psi)}$ (Koh and Liang, 2017, Section 2.2). See Appendix C for more details on how to compute the influence score efficiently.
We provide the complete pseudocode and description of our DOPE Attack algorithm in Appendix C.

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 the attack budget. To measure the effectiveness of our attack model, we compute 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 10 runs (10 datasets) (Agarwal et al., 2021). Second, we compare the performance of DOPE with two custom baselines: Random DOPE and Random Attack. We evaluate our DOPE attack on two medical (HIV (Ernst et al., 2006) and Cancer (Gottesman et al., 2020)), two control (Cartpole and MountainCar) (Brockman et al., 2016) and Continuous Gridworld (see Appendix B) domains.

Experimental Results: 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. Figure 1 compares the percentage error in the value estimate of the OPE methods in all domains. Our results show that 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, 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. In our second experiment (Figure 2), we fix frac $= 0.1$ and compare the percentage error in the value estimate of the OPE methods for different percentages of corrupt data points (α) in HIV, Cancer, and Continuous Gridworld domains. We observe that a larger percentage of corrupt data points yields a larger percentage error in the value estimates. This is not surprising since the attacker's budget ϵ is local to each data point and is not impacted by the number of points perturbed. Finally, in our third experiment, we compare the DOPE attack to two custom baselines: Random Attack and Random DOPE Attack on the Continuous Gridworld domain. Here, Random Attack chooses 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 (Calafiore et al., 1998). On the other hand, Random DOPE selects points randomly but updates 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. We fix the value of α to 0.05. 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. Figure 3 demonstrates that 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, as expected, 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). 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.

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 10 runs. We observe that the attack is successful on most methods across domains. CPDIS and WIS are the most resilient OPE methods.

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.

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.

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.

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.

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.

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.

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.

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

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

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 *Advances in Neural Information Processing Systems*, 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, 1966.

Doina Precup. Temporal abstraction in reinforcement learning, 2000.

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

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.

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.

A PROOFS

Proof of Theorem 4.1. Consider the optimization problem in equation 3: Notice that in equation 3, $\forall k \in [1, \dots N]$, I_{Ψ_i} is independent of δ_k and so the optimal perturbation δ_k^* can be independently computed by solving $\delta_k^* \in \arg\max_x I^T_{\Psi_k,\theta,\Psi} x \,\text{s.t.}\, \|x\|_p \leq \epsilon$. 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 (3) boils down to solving

$$\max_{s\in\{0,1\}^N} \sum_{k=1}^{n} \|I_{\Psi_k,\theta,\Psi}\|_q \quad \sum_k s_k = \alpha \cdot n. \tag{5}$$

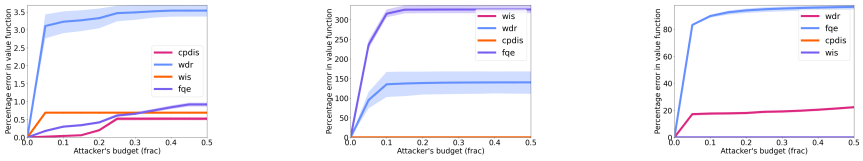
It is now easy to see that the optimal set of transitions for the approximate attack problem in equation 3 is simply the set of αn transitions with the largest value of the q-norm of their influence scores. The closed form solution for δ_k^* at $p = 1, 2, \infty$ follows from standard convex optimization results for $\|x\|_q = \max z^T x \,\text{s.t.}\, \|z\|_p \leq 1$ in (Boyd and Vandenberghe, 2004). □

B EXPERIMENTAL DETAILS

B.1 ADDITIONAL DOMAIN DETAILS

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

B.2 EXPERIMENTAL RESULTS

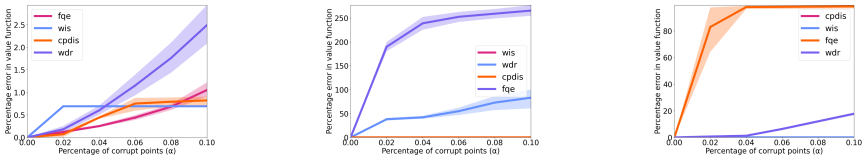


(a) Cancer **(b)** HIV **(c)** Gridworld

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 $\epsilon = \text{frac} \cdot \sigma$ 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.

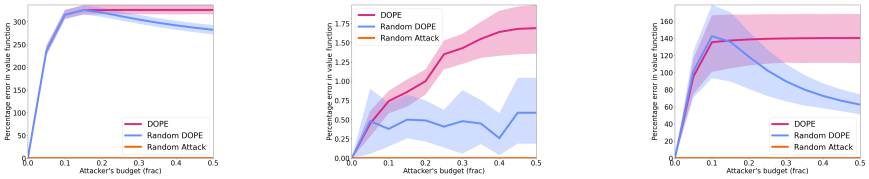
C ALGORITHM

Algorithm outline We outline how to approximately solve the DOPE optimization equation 1 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 $\|I_{\psi_\cdot}\|_q$. 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.



(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 $\epsilon = 1.0\sigma$ 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. DOPE attack outperforms both the Random DOPE and Random attacks at nearly all values of the attacker's budget.

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 $i = 1, \ldots, n$;
$S^*_\alpha \leftarrow \alpha n$ 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 equation 4 ;
  Compute $\delta^*_i \in \arg\max_{\delta \in \mathbb{R}^Q} I^\top_{\Psi_i,\theta,\Psi}\delta$ where $\|\delta\|_p \leq \epsilon$ 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

Efficient Computation of Influence Score: The derivatives in equation 4 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 $c_{\text{prod}} = H^{-1}_{\theta_{\text{org}}(\Psi)} \frac{\partial \rho(\theta,\Psi)}{\partial \theta}\Big|_{\theta_{\text{org}}(\Psi)}$ in $O(nP)$ time using the Pearlmutter's method (Pearlmutter, 1994)

D Additional Preliminaries

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 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:

$$\mathrm{MSBR}(\eta) \;=\; \|q_\eta - \mathcal{T}^\pi q_\eta\|_W^2 \,. \tag{6}$$

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

$$v_{\text{BRM}} \;=\; \sum_{s \in D_0} \sum_{a \in \mathcal{A}} p_0(s) \cdot \pi(s,a) \cdot q_\eta(s,a) \,. \tag{7}$$

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 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) (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

$$v_{\text{WIS}} = \left(\sum_{i=1}^N \rho_{0:T}^i \right)^{-1} \sum_{i=1}^N \rho_{0:L}^i g_T^i, \tag{8}$$

$$v_{\text{PDIS}} = \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \gamma^{t-1} \rho_{0:t}^i r_t^i, \tag{9}$$

$$v_{\text{CPDIS}} = \sum_{t=1}^T \gamma^{t-1} \frac{\sum_{i=1}^N \rho_{0:t}^i r_t^i}{\sum_{i=1}^N \rho_{0:t}^i}. \tag{10}$$

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

$$\begin{aligned} v_{\text{DR}} &= \frac{1}{N} \sum_{i=1}^N \sum_{t=0}^{T-1} \rho_{0:t}^i w_t^i + \frac{1}{N} \sum_{i=1}^N v_\eta(s_0^i). \\ v_{\text{WDR}} &= \sum_{i=1}^N \sum_{t=0}^{T-1} \frac{\rho_{0:t}^i}{\sum_{i=1}^N \rho_{0:t}^i} w_t^i + \frac{1}{N} \sum_{i=1}^N v_\eta(s_0^i). \end{aligned} \tag{11}$$

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.