# Data Poisoning Attacks on Off-Policy Policy Evaluation Methods (Supplementary Material)

**Elita Lobo**[1] **Harvineet Singh**[2] **Marek Petrik**[1] **Cynthia Rudin**[3] **Himabindu Lakkaraju**[4]

[1]University of New Hampshire, Durham, NH, USA
[2]New York University, New York, NY, USA
[3]Duke University, Durham, NC, USA
[4]Harvard University, Boston, MA, USA

## 1 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 $\theta$ can be approximated via Taylor expansion as

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

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

## 2 PROOFS:

*Proof of Theorem 4.1.* Recall the optimization problem in (10):

$$\begin{aligned}&\max_{s\in\{0,1\}^n}\max_{\{\delta_i\}_{i=1}^N}\left\{\sum_{i=1}^n s_i I_{\Psi_i}^\top \delta_i \mid \|\delta_i\|_p \le \varepsilon, \forall i\right\}, \\ &\text{subject to } \sum_{i=1}^n s_i = \alpha\cdot n\,.\end{aligned} \tag{2}$$

Notice that in (2), $\forall k \in [1,\dots N]$, $I_{\Psi_i}$ is independent of $\delta_k$ and so the optimal perturbation $\delta_k^*$ can be independently computed by solving $\delta_k^* \in \arg\max_x\{I_{\Psi_k,\theta,\Psi}^T x \mid \|x\|_p \le \varepsilon\}$. The $p$-norm $\|x\|_p$ of any vector $x \in \mathbb{R}^M$ can be expressed using its dual norm as $\|x\|_p = \max\left\{z^T x \mid \|z\|_q \le 1\right\}$ where $1/p + 1/q = 1$ [Boyd and Vandenberghe, 2004]. Thus, given the optimal-perturbation $\delta_k^*$ for each $k \in 1,\dots,n$, the problem in (10) boils down to solving

$$\begin{aligned}&\max_{s\in\{0,1\}^N}\sum_{k=1}^n \|I_{\Psi_k,\theta,\Psi}\|_q \\ &\sum_k s_k = \alpha\cdot n.\end{aligned} \tag{3}$$

It is now easy to see that the optimal set of transitions for the approximate attack problem in (10) is simply the set of $\alpha n$ transitions with the largest value of the $q$-norm of their influence scores. The closed-form solution for $\delta_k^*$ at $p = 1, 2, \infty$ follows from standard convex optimization results for dual norms [Boyd and Vandenberghe, 2004]. □

## 3 EXPERIMENTAL DETAILS:

### 3.1 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 approximators 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.

*Accepted for the 38th Conference on Uncertainty in Artificial Intelligence* (UAI 2022).

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 values.

4. Link to code: https://github.com/elitalobo/DOPE

### 3.2 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 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 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.

## 4 EXAMPLES OF TWICE CONTINUOUSLY DIFFERENTIABLE LOSS FUNCTIONS FOR DOPE FRAMEWORK :

All the loss functions ($L$) that we leverage in this work such as Mean Squared Bellman residual (MSBR) for learning the Q-value function, and the Cross-Entropy Loss (referred to as CEL in the paper) for fitting the multinomial logistic regression model are twice continuously differentiable with respect to the parameters $\theta$. Below, we show that these loss functions are twice continuously differentiable.

In BRM and WDR, $\theta = \eta$ represents the parameters of the q-value function $q_\eta$. The parameters $\eta$ are estimated from the data by minimizing the Mean Squared Bellman Residual (MSBR). We compute the derivative of MSBR below to show that this loss function is twice differentiable and satisfies the assumption of our attack framework.

$$\begin{aligned}\text{MSBR}(\eta) &= \|q_\eta - \mathcal{T}^\pi q_\eta\|_W^2 \\ &= \|\Phi\eta - (r + \gamma\Phi_p\eta)\|_2^2 \\ \frac{\partial \text{MSBR}(\eta)}{\partial \eta} &= 2\cdot(\Phi - \gamma\Phi_p)^T(\Phi\eta - (r+\gamma\Phi_p\eta)) \\ \frac{\partial^2 \text{MSBR}(\eta)}{\partial \eta^2} &= 2\cdot(\Phi-\gamma\Phi_p)^T(\Phi-\gamma\Phi_p)\end{aligned} \tag{4}$$

In the case of Importance Sampling-based OPE methods such as WIS, PDIS, and CPDIS, the behavior policy parameters ($\theta = \theta_b \in \mathbb{R}^{A\cdot d}$ ) are estimated from the data using a multinomial logistic regression model. Hence, we compute below the second-order derivative of the cross-entropy loss of the multinomial logistic regression model and show that this loss function is twice differentiable as well and satisfies the assumption of our attack framework.

The, cross entropy loss for $\theta = \theta_b$ is given by

$$\begin{aligned}\text{CEL}(\theta) &= \log\left(\prod_{l=1}^{n} \frac{\exp(\theta_{a_l}^T \xi(s_l))}{\sum_{i=1}^{A}\exp(\theta_i^T\xi(s_l))}\right) \\ &= \sum_{l=1}^{n}\log\left(\frac{\exp(\theta_{a_l}^T\xi(s_l))}{\sum_{i=1}^{A}\exp(\theta_i^T\xi(s_l))}\right) \\ &= \sum_{l=1}^{n}\left(\theta_{a_l}^T\xi(s_l) - \log\left(\sum_{j=1}^{A}\exp(\theta_j^T\xi(s_l))\right)\right).\end{aligned} \tag{5}$$

We can compute the second order derivative of the cross

| Hyperparameter values for Cancer domain | |
|---|---|
| Hyperparameter | Value |
| Number of trajectories | 500 |
| Policy Network layers | $64 \times 28$ |
| Normalize rewards | No |
| Regularization for $\pi_b$ | 1e-2 |
| Regularization for $q_\eta$ | 1e-2 |
| Discount factor | 0.95 |
| Trajectory Length (T) | 30 |
| Direction of Attack | +1 |
| Num. Epochs for CEL | 5000 |

| Hyperparameter values for HIV domain | |
|---|---|
| Hyperparameter | Value |
| Number of trajectories | 1000 |
| Policy Network layers | $300 \times 50$ |
| Normalize rewards | Yes |
| Regularization for $\pi_b$ | 1e-2 |
| Regularization for $q_\eta$ | 1e-2 |
| Discount factor | 0.98 |
| Trajectory Length (T) | 50 |
| Direction of Attack | -1 |
| Num. Epochs for CEL | 5000 |

| Hyperparameter values for Continuous Gridworld domain | |
|---|---|
| Hyperparameter | Value |
| Number of trajectories | 500 |
| Policy Network layers | 24 |
| Normalize rewards | No |
| Regularization for $\pi_b$ | 1e-2 |
| Regularization for $q_\eta$ | 1e-2 |
| Discount factor | 0.95 |
| Trajectory Length (T) | 50 |
| Direction of Attack | -1 |
| Num. Epochs for CEL | 5000 |

| Hyperparameter values for MountainCar domain | |
|---|---|
| Hyperparameter | Value |
| Number of trajectories | 250 |
| Policy Network layers | 60 |
| Normalize rewards | No |
| Regularization for $\pi_b$ | 1e-2 |
| Regularization for $q_\eta$ | 1e-2 |
| Discount factor | 0.99 |
| Trajectory Length (T) | 150 |
| Direction of Attack | +1 |
| Num. Epochs for CEL | 5000 |

entropy loss as follows:

$$\begin{aligned}
\frac{\partial \text{CEL}(\theta)}{\partial \theta_{a_l}} &= \sum_{l=1}^{n} \left( \xi(s_l) - \frac{\exp(\theta_{a_l}^T \xi(s_l)) \xi(s_l)}{\sum_{j=1}^{A} \exp(\theta_j^T \xi(s_l))} \right) \\
\frac{\partial^2 \text{CEL}(\theta)}{\partial \theta_{a_l} \theta_k (k \neq a_l)} &= \sum_{l=1}^{n} \frac{\exp(\theta_k^T \xi(s_l)) \exp(\theta_{a_l}^T \xi(s_l)) \xi(s_l)^T \xi(s_l)}{(\sum_{j=1}^{A} \exp(\theta_j^T \xi(s_l)))^2} \\
\frac{\partial^2 \text{CEL}(\theta)}{\partial \theta_{a_l}^2} &= \sum_{l=1}^{n} \left( - \frac{\exp(\theta_{a_l}^T \xi(s_l)) \xi(s_l)^T \xi(s_l)}{\sum_{j=1}^{A} \exp(\theta_j^T \xi(s_l))} + \right. \\
&\quad \left. \frac{\exp(\theta_{a_l}^T \xi(s_l))^2 \xi(s_l)^T \xi(s_l)}{(\sum_{j=1}^{A} \exp(\theta_j^T \xi(s_l)))^2} \right).
\end{aligned}$$

## 5 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

| Hyperparameter values for Cancer domain | |
|---|---|
| Hyperparameter | Value |
| Number of trajectories | 1000 |
| Policy Network layers | $100 \times 24$ |
| Normalize rewards | No |
| Regularization for $\pi_b$ | 1e-2 |
| Regularization for $q_\eta$ | 1e-2 |
| Discount factor | 0.98 |
| Trajectory Length (T) | 100 |
| Direction of Attack | +1 |
| Num. Epochs for CEL | 5000 |

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 aims 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 a 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. In [Wu et al., 2022], authors propose methods to certify the robustness of a policy learned from offline data after a poisoning attack. It outputs the least cumulative reward that can be attained by a poisoned 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:** The 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 objectives optimized.

# 6 EXPERIMENTAL RESULTS:

## 6.1 EFFECT OF INCREASING RANDOMNESS OF THE BEHAVIOR POLICY ON DOPE ATTACK:

In all our experiments, we chose small values of $\epsilon$ for the behavior policy to examine the cases where the OPE methods are difficult to attack. A larger value of epsilon would result in a larger state-action distribution mismatch between the datasets collected using the behavior policy and the datasets that would have been collected with the evaluation policy.

This distribution mismatch would result in large importance sampling weights and out-of-distribution estimation errors and increase the variance in the value function estimates. As a result, the OPE methods would become more brittle and thus, more vulnerable to data poisoning attacks.

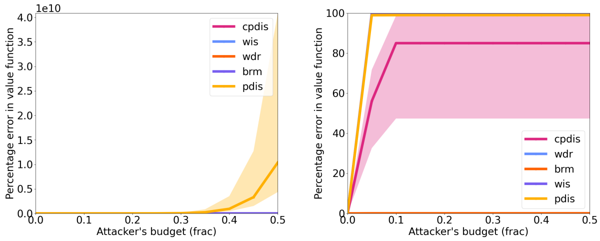
To illustrate this effect, we compare the percentage error in the value function estimates of a near-optimal policy in the HIV domain for two different values of $\epsilon$, 0.05 and 0.25. For this experiment, we set the perturbation budget to $\varepsilon = 0.5\sigma$ and percentage of corrupt points to $\alpha = 0.05$. We report the interquartile mean of the percentage error in the value function estimates observed across 5 trials in Table 1. Our results in Table 1 indicate that OPE methods like BRM, WDR are more vulnerable to the data poisoning attack for larger values of $\epsilon$.

## 6.2 ANOMALY DETECTION METHODS

In this experiment, we investigate if standard anomaly detection methods can identify the poisoned data points from the dataset.

For this purpose, we use two popular state-of-the-art anomaly detection methods [Emmott et al., 2013], namely, the Isolation Forests [Liu et al., 2008] and the Local Outlier Factor [Breunig et al., 2000] method. We set the perturbation budget $\varepsilon$ to be $0.5\sigma$ and the percentage of corrupt points to be $\alpha = 0.05$. We report the True Positive Rate (Fraction of perturbed data points tagged as outliers) and the False Positive Rate (Fraction of original data instances tagged as outliers). Our experimental results with the aforementioned anomaly detection methods, and the WDR OPE method across Cancer, HIV, and Gridworld domains are shown in in Table 2 and Table 3. While the Isolation Forests method has a high true positive rate, it also has a high false-positive rate indicating that several original data instances are being tagged as outliers. On the other hand, the Local Outlier Factor method exhibits low true positive and false-positive rates. The following results suggest that the perturbed data points are not readily distinguishable from the original data instances. These results are not surprising as the budget constraint embedded in our optimization problem Equation (8e) ensures that the original data instances are perturbed in a manner that cannot be easily detected by naive anomaly detection techniques.

## 6.3 EFFECTIVENESS OF DOPE ATTACK



**(a)** Cartpole

**(b)** Mountain Car

**Figure 1:** Figures 1a and 1b 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 $p = 1$.

## 6.4 COMPARISON WITH PROJECTED DOPE ATTACK METHOD

Here we compare the DOPE attack to Projected DOPE Attack. In Projected DOPE Attack, we first compute the set of top $\alpha n$ influential points and their influences. Next, we set the optimal perturbations for the most influential points as the projection of their influences on the constrained space defined by the attack budget constraints. We fix the value of $\alpha$ to $0.05$ and vary the budget $\varepsilon$ from $0.0$ to $0.25$ with step size $0.04$.

Results for all the domains are shown in Figures 8 to 12. These results indicate that there is no clear winner between DOPE and Projected DOPE as they both can perform well depending on the environment and the datasets collected.

| Methods | BRM | WIS | PDIS | CPDIS | WDR |
|---|---|---|---|---|---|
| epsilon=0.05 | 334.2 | 5.83e-3 | 1.61 | 0.22 | 118.35 |
| epsilon=0.25 | 427.15 | 0.0 | 2.59 | 0.06 | 1489.22 |

**Table 1:** Percentage errors in the value function estimates observed for different values of $\varepsilon$ on the HIV domain.

| OPE Method = WDR | True Positive Rate | False Positive Rate |
|---|---|---|
| Cancer | 1.0 | 0.26 |
| HIV | 0.47 | 0.16 |
| Gridworld | 1.0 | 0.9 |

**Table 2:** Results with Isolation Forests anomaly detection method and WDR Method.

| OPE Method = WDR | True Positive Rate | False Positive Rate |
|---|---|---|
| Cancer | 0.02 | 0.05 |
| HIV | 0.08 | 0.07 |
| Gridworld | 0.01 | 0.07 |

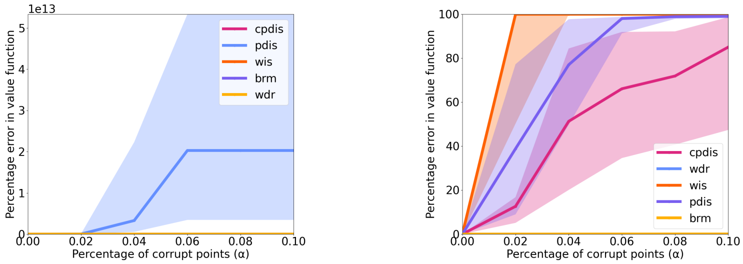
**Table 3:** Results with Local Outlier Factor anomaly detection method and WDR Method.

| OPE Method = PDIS | True Positive Rate | False Positive Rate |
|---|---|---|
| Cancer | 1.0 | 0.31 |
| HIV | 0.17 | 0.17 |
| Gridworld | 1.0 | 0.5 |

**Table 4:** Results with Isolation Forests anomaly detection method and PDIS method.

| OPE Method = PDIS | True Positive Rate | False Positive Rate |
|---|---|---|
| Cancer | 0.0 | 0.05 |
| HIV | 0.32 | 0.07 |
| Gridworld | 0.03 | 0.06 |

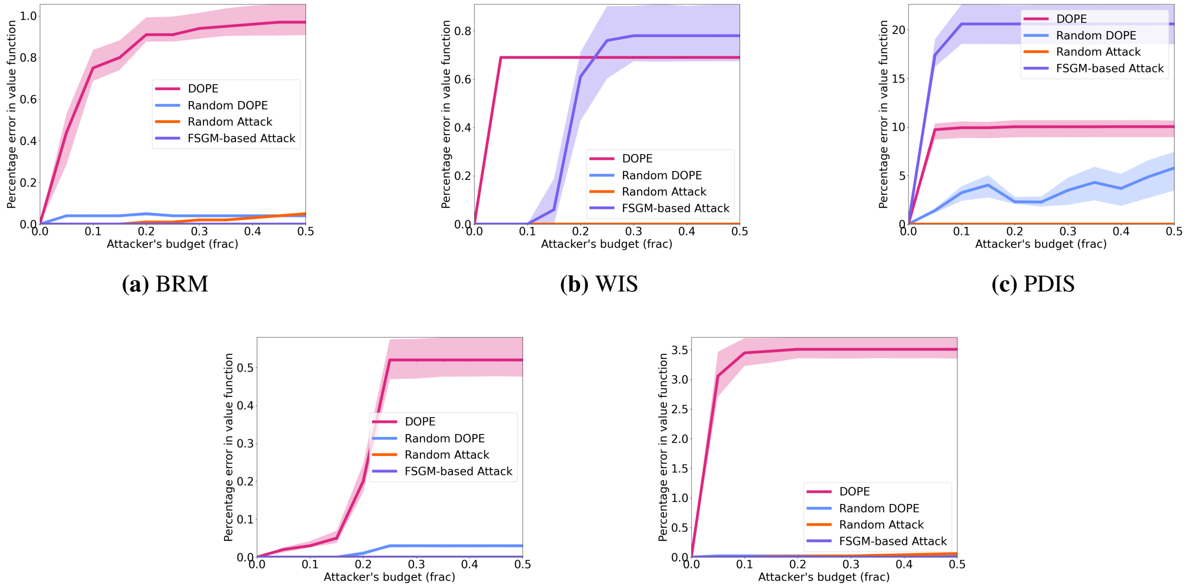
**Table 5:** Results with Local Outlier Factor anomaly detection method and PDIS Method.



**(a)** Cartpole

**(b)** MountainCar

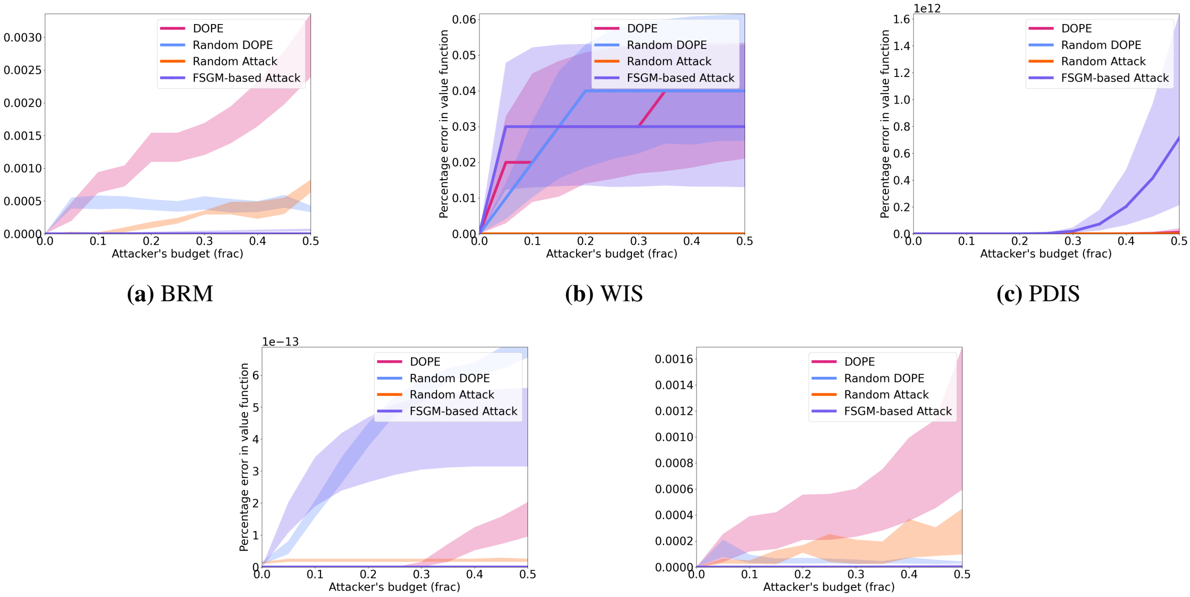
**Figure 2:** Figures 2a and 2b compares the effect of DOPE attack on BRM, WIS, PDIS, CPDIS and WDR methods in in Cartpole and MountainCar domains (left to right) for different percentages of corruption $\alpha$ and $p = 1$.



**(a)** BRM **(b)** WIS **(c)** PDIS

**(d)** CPDIS **(e)** WDR

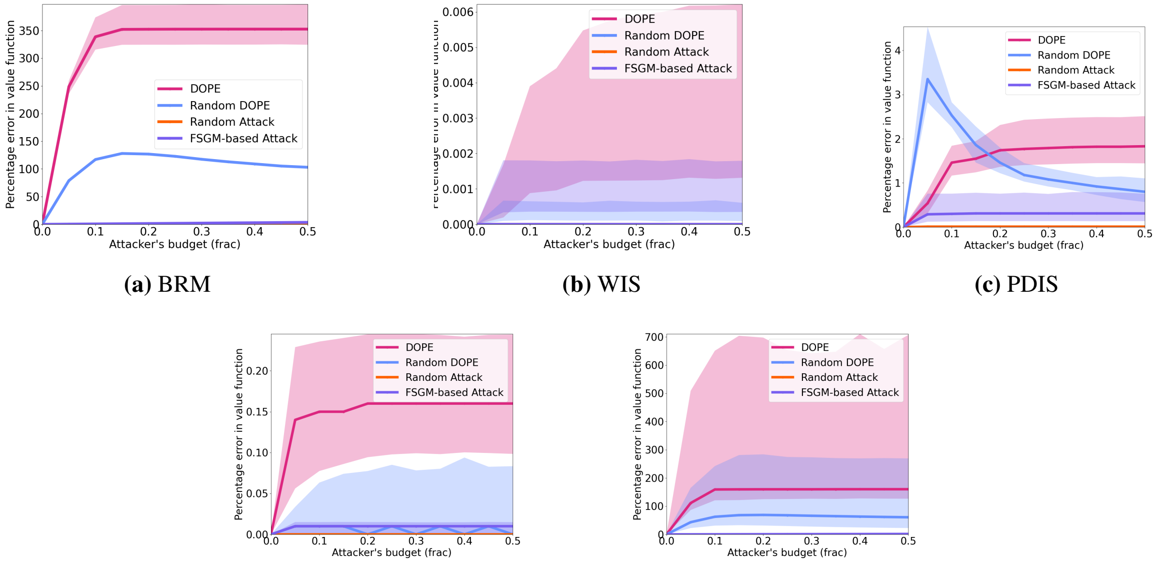
**Figure 3:** Figures 3a to 3e compares the effect of random attack, Random DOPE attack, FSGM-based Attack and DOPE attack on the error in the value function estimates of BRM, WIS, PDIS, CPDIS and WDR methods (left to right) in Cancer domain.



**(a)** BRM **(b)** WIS **(c)** PDIS

**(d)** CPDIS **(e)** WDR

**Figure 4:** Figures 4a to 4e compares the effect of random attack, Random DOPE attack, FSGM-based Attack and DOPE attack on the error in the value function estimates of BRM, WIS, PDIS, CPDIS and WDR methods (left to right) in Cartpole domain.



**(a)** BRM

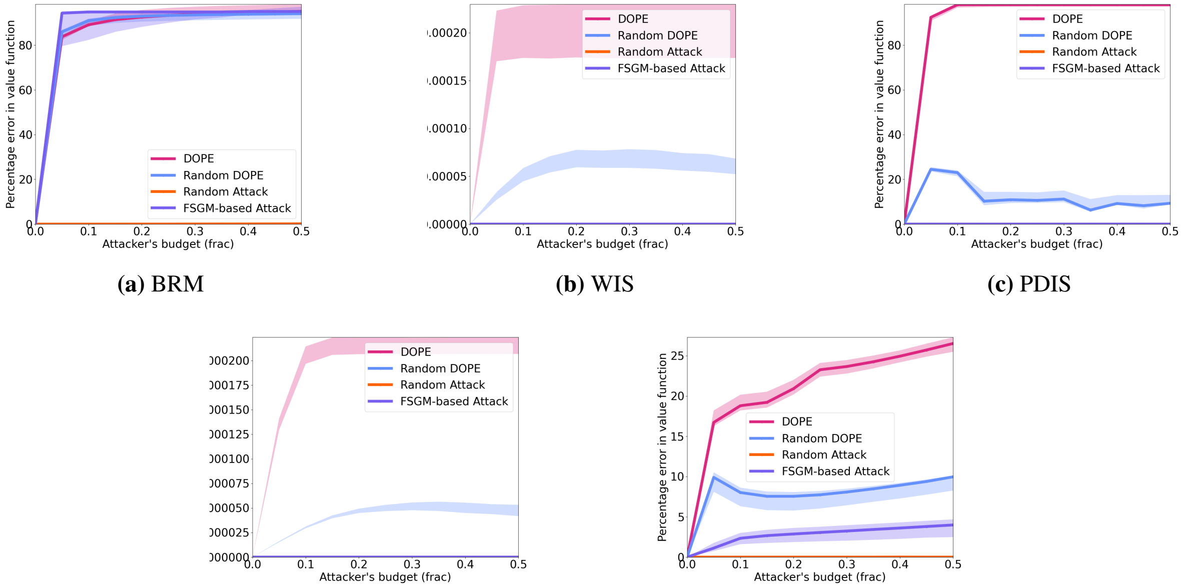
**(b)** WIS

**(c)** PDIS

**(d)** CPDIS

**(e)** WDR

**Figure 5:** Figures 5a to 5e compares the effect of random attack, Random DOPE attack, FSGM-based 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.



**(a)** BRM

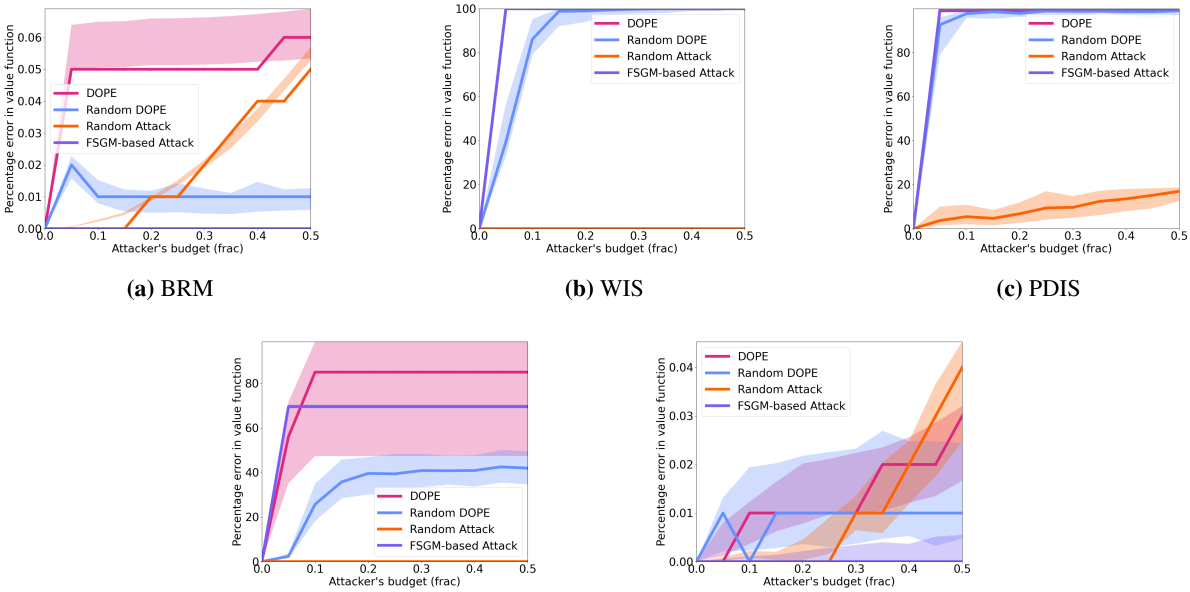
**(b)** WIS

**(c)** PDIS

**(d)** CPDIS

**(e)** WDR

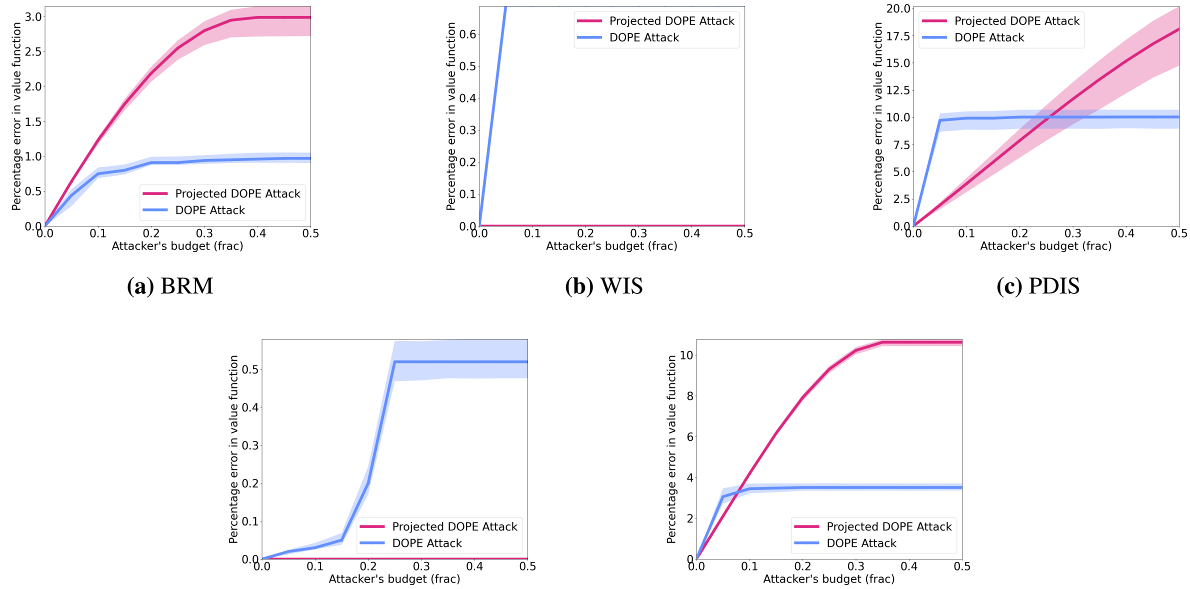
**Figure 6:** Figures 6a to 6e compares the effect of random attack, Random DOPE attack, FSGM-based Attack and DOPE attack on the error in the value function estimates of BRM, WIS, PDIS, CPDIS and WDR methods (left to right) in Continuous Gridworld domain.



**(a)** BRM **(b)** WIS **(c)** PDIS

**(d)** CPDIS **(e)** WDR

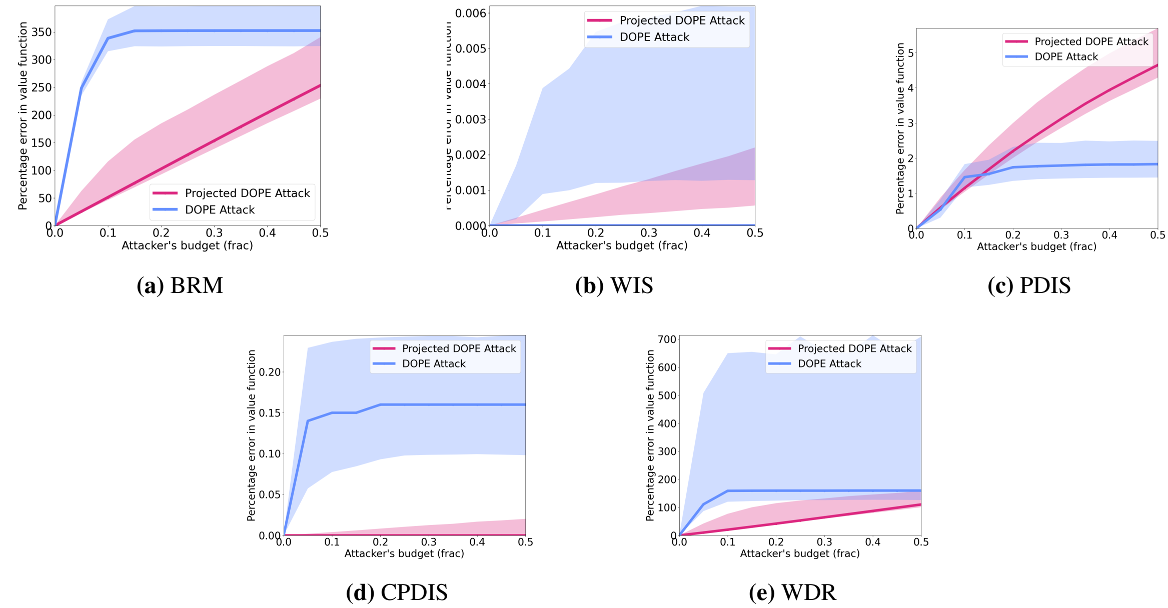
**Figure 7:** Figures 7a to 7e compares the effect of random attack, Random DOPE attack, FSGM-based attack and DOPE attack on the error in the value function estimates of BRM, WIS, PDIS, CPDIS and WDR methods (left to right) in MountainCar domain.



**(a)** BRM **(b)** WIS **(c)** PDIS

**(d)** CPDIS **(e)** WDR

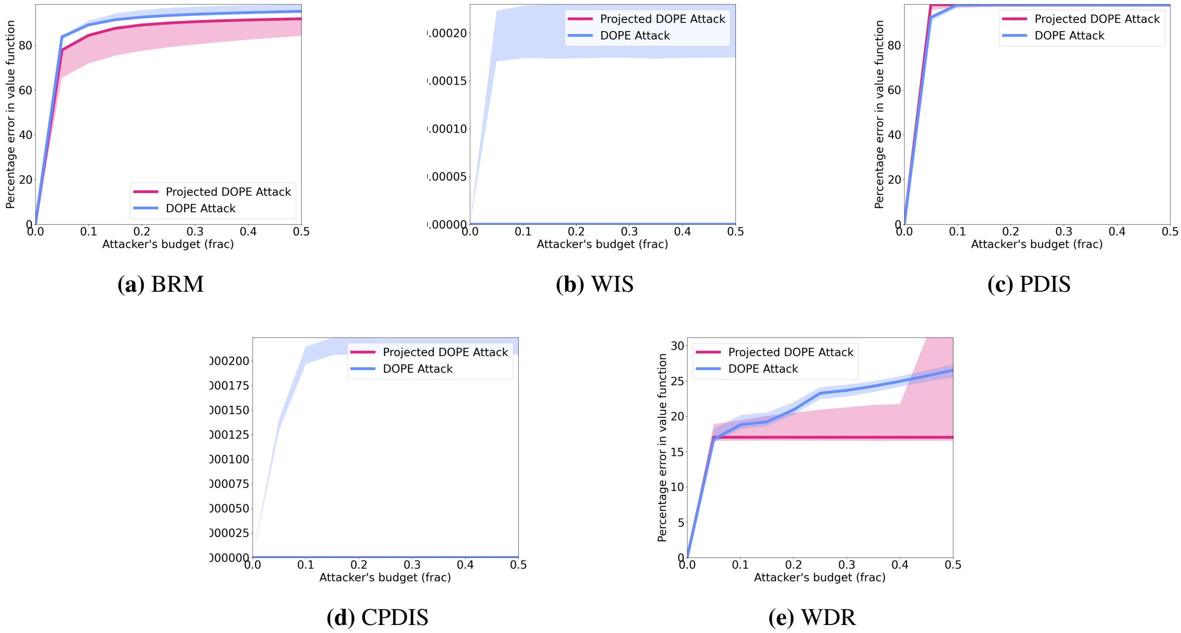
**Figure 8:** Figures 8a to 8e compares the effect of Projected DOPE attack and DOPE attack on the error in the value function estimates of BRM, WIS, PDIS, CPDIS and WDR methods (left to right) in Cancer domain.



**(a)** BRM **(b)** WIS **(c)** PDIS

**(d)** CPDIS **(e)** WDR

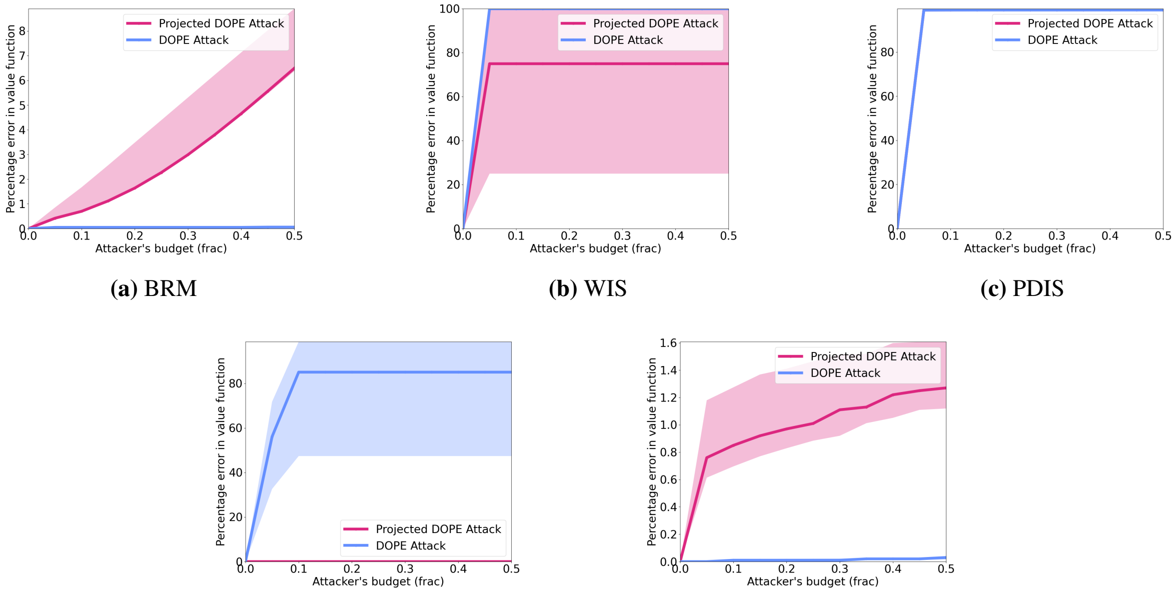
**Figure 9:** Figures 9a to 9e compares the effect of Projected DOPE attack and DOPE attack on the error in the value function estimates of BRM, WIS and PDIS, CPDIS, WDR methods (left to right) in HIV domain.



**(a)** BRM **(b)** WIS **(c)** PDIS

**(d)** CPDIS **(e)** WDR

**Figure 10:** Figures 10a to 10e compares the effect of Projected 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.



(**a**) BRM

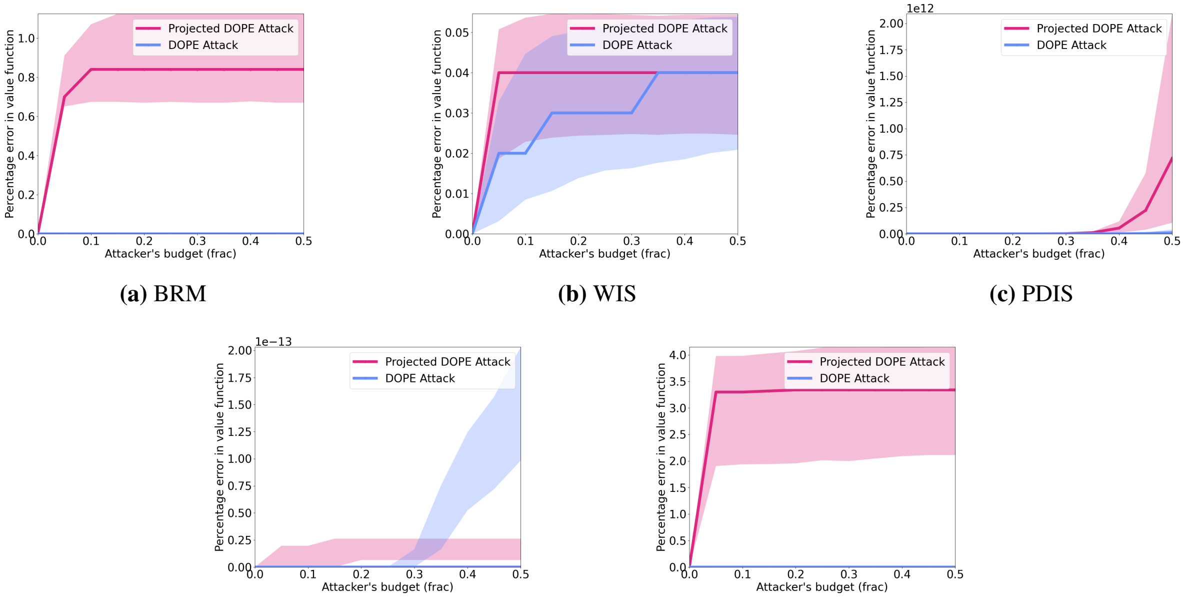
(**b**) WIS

(**c**) PDIS

(**d**) CPDIS

(**e**) WDR

**Figure 11:** Figures 11a to 11e compares the effect of Projected DOPE 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**) BRM

(**b**) WIS

(**c**) PDIS

(**d**) CPDIS

(**e**) WDR

**Figure 12:** Figures 12a to 12e compares the effect of Projected DOPE and DOPE attack on the error in the value function estimates of BRM, WIS, PDIS, CPDIS and WDR methods (left to right) in Cartpole domain.

## References

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.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. SIGMOD '00, page 93–104. Association for Computing Machinery, 2000.

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.

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, 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)*, 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.

Andrew F. Emmott, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong. Systematic construction of anomaly detection benchmarks from real data. ODD '13, page 16–21. Association for Computing Machinery, 2013.

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.

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.

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.

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.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008.

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.

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.

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.

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.

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.

Fan Wu, Linyi Li, Huan Zhang, Bhavya Kailkhura, Krishnaram Kenthapadi, Ding Zhao, and Bo Li. COPA: Certifying robust policies for offline reinforcement learning against poisoning attacks. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=psh0oeMSBiF`.

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.