Interpretable Off-Policy Evaluation in Reinforcement Learning by Highlighting Influential Transitions

Omer Gottesman[1] **Joseph Futoma**[1] **Yao Liu**[2] **Sonali Parbhoo**[1] **Leo Anthony Celi**[3] **Emma Brunskill**[2] **Finale Doshi-Velez**[1]

Abstract

Off-policy evaluation in reinforcement learning offers the chance of using observational data to improve future outcomes in domains such as healthcare and education, but safe deployment in high stakes settings requires ways of assessing its validity. Traditional measures such as confidence intervals may be insufficient due to noise, limited data and confounding. In this paper we develop a method that could serve as a hybrid human-AI system, to enable human experts to analyze the validity of policy evaluation estimates. This is accomplished by highlighting observations in the data whose removal will have a large effect on the OPE estimate, and formulating a set of rules for choosing which ones to present to domain experts for validation. We develop methods to compute exactly the influence functions for fitted Q-evaluation with two different function classes: kernel-based and linear least squares, as well as importance sampling methods. Experiments on medical simulations and real-world intensive care unit data demonstrate that our method can be used to identify limitations in the evaluation process and make evaluation more robust.

1. Introduction

Within reinforcement learning (RL), off-policy evaluation (OPE) is the task of estimating the value of a given evaluation policy, using data collected by interaction with the environment under a different behavior policy (Sutton & Barto, 2018; Precup, 2000). OPE is particularly valuable when interaction and experimentation with the environment is expensive, risky, or unethical—for example, in healthcare or with self-driving cars. However, despite recent interest and progress, state-of-the-art OPE methods still often fail to differentiate between obviously good and obviously bad policies, e.g. in healthcare (Gottesman et al., 2018).

Most of the OPE literature focuses on sub-problems such as improving asymptotic sample efficiency or bounding the error on OPE estimators for the value of a policy. However, while these bounds are theoretically sound, they are often too conservative to be useful in practice (though see e.g. Thomas et al. (2019) for an exception). This is not surprising, as there is a theoretical limit to the statistical information contained in a given dataset, no matter which estimation technique is used. Furthermore, many of the common assumptions underlying these theoretical guarantees are usually not met in practice: observational healthcare data, for example, often contains many unobserved confounders (Gottesman et al., 2019a).

Given the limitations of OPE, we argue that in high stakes scenarios domain experts should be integrated into the evaluation process in order to provide useful actionable results. For example, senior clinicians may be able to provide insights that reduce our uncertainty of our value estimates. In this light, the explicit integration of expert knowledge into the OPE pipeline is a natural way for researchers to receive feedback and continually update their policies until one can make a responsible decision about whether to pursue gathering prospective data.

The question is then what information can humans provide that might help assess and potentially improve our confidence in an OPE estimate? In this work, we consider how human input could improve our confidence in the recently proposed OPE estimator, fitted Q-evaluation (FQE) (Le et al., 2019), as well as importance sampling (IS) methods. We develop an efficient approach to identify the most influential transitions in a batch of observational data, that is, transitions whose removal would have large effects on the OPE estimate. By presenting these influential transitions to a domain expert and verifying that they are indeed representative of the data, we can increase our confidence that our estimated evaluation policy value is not dependent on outliers, confounded observations, or measurement errors. The main contributions of this work are:

[1] Harvard University [2] Stanford University [3] MIT. Correspondence to: Omer Gottesman <gottesman@fas.harvard.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

- *Conceptual*: We develop a framework for using influence functions to interpret OPE, and discuss the types of questions which can be shared with domain experts to use their expertise in debugging OPE.
- *Technical*: We develop computationally efficient algorithms to compute the exact influence functions for several IS estimators as well as two broad function classes for FQE: kernel-based functions and linear functions.
- *Empirical*: We demonstrate the potential benefits of influence analysis for interpreting OPE on a cancer simulator, and present results of analysis together with practicing clinicians of OPE for management of acute hypotension from a real intensive care unit (ICU) dataset.

2. Related work

The OPE problem in RL has been studied extensively. Works fall into two main categories: importance sampling (e.g. Precup (2000); Jiang & Li (2015)) and model-based (often referred to as the direct method), which can be further subdivided into modeling the environment dynamics (e.g. Hanna et al. (2017); Gottesman et al. (2019b)), and directly modeling the value function (e.g. Le et al. (2019)). Some of these works provide bounds on the estimation errors (e.g. Thomas et al. (2015); Dann et al. (2018)). We emphasize, however, that for most real-world applications these bounds are either too conservative to be useful or rely on assumptions which are usually violated.

While there has been considerable recent progress in interpretable machine learning and machine learning with humans in the loop (e.g. Tamuz et al. (2011); Lage et al. (2018)), to our knowledge, there has been little work that considers human interaction in the context of OPE. Oberst & Sontag (2019) proposed framing the OPE problem as a structural causal model, which enabled them to identify trajectories where the predicted counterfactual trajectories under an evaluation policy differs substantially from the observed data collected under the behavior policy. However, that work does not give guidance on what part of the trajectory might require closer scrutiny, nor can it use human input for additional refinement.

Finally, the notion of influence that we use throughout this work has a long history in statistics as a technique for evaluating the robustness of estimators (Cook & Weisberg, 1980). Recently, an approximate version of influence for complex black-box models was presented in Koh & Liang (2017), and they demonstrated how influence functions can make machine learning methods more interpretable. In the context of optimal control and RL, influence functions were first introduced by Munos & Moore (2002) to aid in online optimization of policies. However, their definition of influence as a change in the value function caused by perturbations of the reward at a specific state is quite different from ours.

3. Background

Notation A Markov Decision Process (MDP) is a tuple $\langle \mathcal{X}, \mathcal{A}, P_T, P_R, P_0, \gamma \rangle$, where $\mathcal{X}$, $\mathcal{A}$ and γ are the state space, action space, and the discount factor, respectively. The next state transition and reward distributions are given by $P_T(\cdot|x,a)$ and $P_R(\cdot|x,a)$ respectively, and $P_0(x)$ is the initial state distribution. The state and action spaces could be either discrete or continuous, and the transition and reward functions may be either stochastic or deterministic.

A dataset is composed of a set of N observed transitions $\mathcal{D} = \{(x^{(n)}, a^{(n)}, r^{(n)}, x'^{(n)})\}_{n=1}^{N}$, and we use $\tau^{(n)}$ to denote a single transition. The subset $\mathcal{D}_0 \subseteq \mathcal{D}$ denotes initial transitions from which P_0 can be estimated. Note that although we treat all data points as observed transitions, in most practical applications data is collected in the form of trajectories rather than individual transitions.

A policy is a function $\pi : (\mathcal{X}, \mathcal{A}) \to [0,1]$ that gives the probability of taking each action at a given state ($\sum_{a \in \mathcal{A}} \pi(a|x) = 1$). The value of a policy is the expected return collected by following the policy, $v^{\pi} := \mathrm{E}[g_T | a_t \sim \pi]$, where expectations are taken with respect to the MDP and $g_T := \sum_{t=0}^{T} \gamma^t r_t$ denotes the total trajectory return (sum of discounted rewards). The state-action value function $q^{\pi}(x,a)$ is the expected return for taking action a at state x, and afterwards following π in selecting future actions. The goal of off-policy evaluation is to estimate the value of an *evaluation* policy, π_e, using data collected under a different *behavior* policy, π_b. In this work, we are only interested in estimating v^{π_e} and q^{π_e}, and will therefore drop the superscript for brevity. We will also limit ourselves to deterministic evaluation policies.

For the purpose of kernel-based value function approximation, we define a distance metric, $d((x^{(i)}, a^{(i)}), (x^{(j)}, a^{(j)}))$ over $\mathcal{X} \times \mathcal{A}$. In this work, for discrete action spaces, we will assume $d((x^{(i)}, a^{(i)}), (x^{(j)}, a^{(j)})) = \infty$ when $a^{(i)} \neq a^{(j)}$, but this is not required for any of the derivations.

Fitted Q-Evaluation Fitted Q-Evaluation (Le et al., 2019) models the q-function of π_e and can be thought of as dynamic programming on an observational dataset to compute the value of a given evaluation policy. It is similar to the more well-known fitted Q-iteration method (FQI) (Ernst et al., 2005), except it is performed offline on observational data, and the target is used for evaluation of a given policy rather than for optimization. FQE performs a sequence of supervised learning steps where the inputs are state-action pairs, and the targets at each iteration are given by $y_i(x,a) = r + \gamma \hat{q}_{i-1}(x', \pi_e(x'))$, where $\hat{q}_{i-1}(x,a)$ is the estimator (from a function class $\mathcal{F}$) that best estimates $y_{i-1}(x, a)$. For more information, see Le et al. (2019).

Importance sampling A popular class of OPE estimators consists of IS methods. These methods estimate the value of a policy by taking a sample average of trajectories returns, properly weighted to account for the difference between π_b and π_e. The standard IS estimator is unbiased but has high variance, and there are many variants of this estimator which trade of bias and variance. For more information see (Precup, 2000; Jiang & Li, 2015; Thomas & Brunskill, 2016).

4. OPE diagnostics using influence functions

4.1. Definition of the influence

We aim to make OPE interpretable and easy to debug by identifying transitions in the data which are highly influential on the estimated policy value. We define the *total influence* of transition $\tau^{(j)}$ as the change in the value estimate if $\tau^{(j)}$ was removed:

$$I_j \equiv \hat{v}_{-j} - \hat{v}, \tag{1}$$

where $\hat{v}_{-j}$ is the value estimate using the same dataset after removal of $\tau^{(j)}$. In general, for any function of the data $f(\mathcal{D})$ we will use $f(\mathcal{D}_{-j}) \equiv f_{-j}$ to denote the value of f computed for the dataset after removal of $\tau^{(j)}$.

Another quantity of interest is the change in the estimated value of $q(x^{(i)}, a^{(i)})$ as a result of removing $\tau^{(j)}$, which we call the *individual influence*:

$$I_{i,j} \equiv \hat{q}_{-j}(x^{(i)}, a^{(i)}) - \hat{q}(x^{(i)}, a^{(i)}). \tag{2}$$

The total influence of $\tau^{(j)}$ can be computed by averaging its individual influences over the set $\mathcal{D}_0^*$ of all initial state-action transitions in which $a = \pi_e(x)$:

$$I_j = \frac{1}{|\mathcal{D}_0^*|} \sum_{i \in \mathcal{D}_0^*} I_{i,j}. \tag{3}$$

As we are interested in the robustness of our evaluation, we can normalize the absolute value of the influence of $\tau^{(j)}$ by the estimated value of the policy to provide a more intuitive notion of overall importance:

$$\tilde{I}_j \equiv \frac{|I_j|}{|\hat{v}|}. \tag{4}$$

4.2. Diagnosing OPE estimation

With the above definitions of influence functions, we now formulate and discuss guidelines for diagnosing the OPE process for potential problems.

No influential transitions: OPE appears reliable. As a first diagnostic, we check that none of the transitions influence the OPE estimate by more than a specified influence threshold $\tilde{I}_C$, i.e. for all j we have $\tilde{I}_j \leq \tilde{I}_C$. In such a case we would output that, to the extent that low influences suggests the OPE is stable, the evaluation appears reliable. That said, we emphasize that our proposed method for evaluating OPE methods is not exhaustive, and there could be many other ways in which OPE could fail.

Influential transitions: a human can help. When there are several influential transitions in the data (defined as transitions whose influence is larger than $\tilde{I}_C$), we present them to domain experts to determine whether they are representative, that is, taking action a in state x is likely to result in transition to x'. If the domain experts can validate all influential transitions, we can still have some confidence in the validity of the OPE. If any influential transitions are flagged as unrepresentative or artefacts, we have several options: (1) Declare the OPE as unreliable; (2) Remove the suspect influential transitions from the data and recompute the OPE; (3) Caveat the OPE results as valid only for a subset of initial states that do not rely on that problematic transition.

In situations where there is a large number of influential transitions, manual review by experts may be infeasible. As such, it is necessary to present as few transitions as possible while still presenting enough to ensure that any potential artefacts in the data and/or the OPE process are accounted for. In practice, we find it is common to observe a sequence of influential transitions where removing any single transition has the same effect as removing the entire sequence. An example of this is shown schematically in Figure 1. An entire sequence marked in blue and red leads to a region of high reward, and so all transitions in that sequence will have high influence. The whole influential sequence appears very different from the rest of the data, and a domain expert might flag it as an outlier to be removed. However, we can present the expert with only the red transition and capture the influence of the blue transitions as well, reducing the number of suspect examples to be manually reviewed.

Influential transitions: policy is unevaluatable. When an influential transition, $\tau^{(j)}$, has no nearest neighbors to $(x'^{(j)}, \pi_e(x'^{(j)}))$, we can determine that the evaluation policy cannot be evaluated, even without review by a domain expert. This claim is a result of the fact that such a situation represents reliance of the OPE on transitions for which there is no overlap between the actions observed in the data and the evaluation policy. However, while the evaluation policy is not evaluatable, the influential "dead-end" transitions may still inform experts of what data is required for evaluation to be feasible.

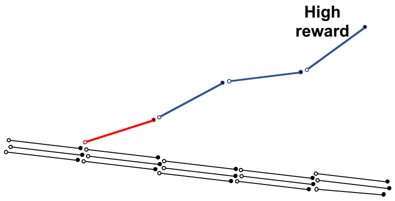


Figure 1. **Schematic of an influential sequence.** All transitions in the sequence leading to a high reward have high influence, but flagging just the red transition for inspection will capture the influence of the blue ones as well.

It should be noted that the applicability of the diagnostics methods discussed above may change depending on whether the FQE function class is parametric or nonparametric. All function classes lend themselves to highlighting of highly influential transitions. However, the notion of stringing together sequences of neighbors, or looking for red flags in the form of influential transitions with no neighbors to their $(x', \pi_e(x'))$ state action pairs only makes sense for nonparametric models. In the case of parametric models, the notion of neighbors is less important as the influence of removing a transition manifests as a change to the learned parameters which affects the value estimates for the entire domain simultaneously. In contrast, for nonparametric methods, removing a transition locally changes the value of neighboring transitions and propagates through the entire domain through the sequential nature of the environment. While we derive efficient ways to compute the influence for both parametric and nonparametric function classes, in the empirical section of this paper we present results for nonparametric kernel-based estimators to demonstrate all diagnostics.

4.3. Influence analysis for importance sampling

The approach of using influence analysis to to asses the validity of the OPE can be naturally extended to IS methods, with a few small changes. Most IS methods use entire trajectories rather than individual transitions as their basic data input, and therefore for IS we would compute the influence of trajectories rather than transitions. This also implies that we cannot identify obvious unevaluateble datasets as described in the previous section. Last, it should be noted that for IS the influence is determined not only by the return of a trajectory, but is also strongly determined by the weights, which may grow exponentially with the horizon.

5. Efficient computation of influence functions

A key technical challenge in performing the proposed influence analysis in OPE is computing the influences efficiently. The brute-force approach of removing a transition and recomputing the OPE estimate is clearly infeasible for all but tiny problems, as it requires refitting N models. The computation of influences in RL is also significantly more challenging than in static one-step prediction tasks, as a change in the value of one state has a ripple effect on all other states that are possible to reach from it. We describe computationally efficient methods to compute the influence functions in two classes of FQE: kernel-based, and linear least squares, as well as several popular IS estimators. Unlike previous works (e.g. (Koh & Liang, 2017)) that approximate the influence function for a broad class of black-box functions, we provide closed-form, analytic solutions for the exact influence function for a broad range of OPE methods.

5.1. Kernel-Based FQE

In kernel based FQE, the function class we choose for estimating the value function of π_e at a point in state-action space is based on similar observations within that space. For simplicity, in the main body of this work we estimate the value function as an average of all its neighbors within a ball of radius R, i.e.

$$\hat{q}(x,a) = \frac{1}{N_{(x,a)}} \sum_i \hat{q}(x^{(i)}, a^{(i)}) \tag{5}$$

where the summation is performed over all $(x^{(i)}, a^{(i)})$ such that $d((x^{(i)}, a^{(i)}), (x,a)) < R$ and $N_{(x,a)}$ is the number of such points. Extension to general kernel functions is straightforward. We introduce a matrix formulation for performing FQE which allows for efficient computation of the influence functions.

Matrix formulation of nearest-neighbors based FQE. We define Δ_{ij} as the event that the starting state-action of $\tau^{(j)}$ is a neighbor of the starting state-action of $\tau^{(i)}$, i.e. $d((x^{(i)}, a^{(i)}), (x^{(j)}, a^{(j)})) < R$. Similarly, we define $\Delta_{i'j}$ as the event that the starting state-action of $\tau^{(j)}$ is a neighbor of the next-state and corresponding π_e action of $\tau^{(i)}$, i.e. $d((x'^{(i)}, \pi(x'^{(i)})), (x^{(j)}, a^{(j)})) < R$. We also define the counts for numbers of neighbors of transitions as $N_i = \sum_{j=1}^{N} \mathbb{I}(\Delta_{ij})$ and $N_{i'} = \sum_{j=1}^{N} \mathbb{I}(\Delta_{i'j})$, where $\mathbb{I}(e)$ is the indicator function.

To perform nearest-neighbors FQE using matrix multiplications, we first construct two nearest-neighbors matrices: one for the neighbors of all state-action pairs, and one for the neighbors of all state-action pairs with pairs of next-states and subsequent actions under π_e. Formally:

$$\mathbf{M}_{ij} = \frac{\mathbb{I}(\Delta_{ij})}{N_i}; \quad \mathbf{M}'_{ij} = \frac{\mathbb{I}(\Delta_{i'j})}{N_{i'}}. \tag{6}$$

The $N \times N$ matrices $\mathbf{M}$ and $\mathbf{M}'$ can be easily computed from the data, and are used to compute the value function for all state-action pairs using the following proposition, the proof of which is given in Appendix 1.1.

Proposition 1. *For all transitions in the dataset, the values for corresponding state-action pairs are given by*

$$\hat{\mathbf{q}}'_t = \left(\sum_{t'=1}^{t} \gamma^{t'-1}\mathbf{M}'^{t'}\right)\mathbf{r} \equiv \mathbf{\Phi}'_t\mathbf{r} \tag{7}$$

$$\hat{\mathbf{q}}_t = \mathbf{M}\left(\sum_{t'=1}^{t} \left(\gamma\mathbf{M}'\right)^{t'-1}\right)\mathbf{r} \equiv \mathbf{\Phi}_t\mathbf{r}. \tag{8}$$

where $\hat{q}'_{t,i}$ and $\hat{q}_{t,i}$ are the estimated policy values at $(x'^{(i)}, \pi_e(x'^{(i)}))$ and $(x^{(i)}, a^{(i)})$, respectively, for $\tau^{(i)}$.

In future derivations, we will drop the time dependence of $\mathbf{\Phi}$ and $\hat{\mathbf{q}}$ on t. This is justified when there are well defined ends of trajectories with no nearest neighbors (or equivalently, trajectories end in an absorbing state), and the number of iterations in the FQE is larger than the longest trajectory.

Influence function computation. Removal of a transition $\tau^{(j)}$ from the dataset can affect $\hat{q}_i$ in two ways. First, $\hat{q}_i$ is a mean over all of its neighbors, indexed by k, of $r^{(k)} + \gamma\hat{q}'_k$. Thus if $(x^{(j)}, a^{(j)})$ is one of the M_{ij}^{-1} neighbors of $(x^{(i)}, a^{(i)})$, removing it from the dataset will change the value of $\hat{q}_i$ by $\frac{\hat{q}_i - \left(r^{(j)} + \gamma\hat{q}'_j\right)}{M_{ij}^{-1} - 1}$. The special case of $M_{ij}^{-1} = 1$ does not pose a problem in the denominator, as given that $i \neq j$ and every transition is a neighbor of itself, if $(x^{(j)}, a^{(j)})$ is a neighbor of $(x^{(i)}, a^{(i)})$, then $M_{ij}^{-1} \geq 2$.

The second way in which removing $\tau^{(j)}$ influences $\hat{q}_i$ is through its effect on intermediary transitions. Removal of $\tau^{(j)}$ changes the estimated value of $\hat{q}'_k$, of all $(x'^{(k)}, \pi_e(x'^{(k)}))$ that $(x^{(j)}, a^{(j)})$ is a neighbor of by $\frac{\hat{q}'_k - \left(r^{(j)} + \gamma\hat{q}'_j\right)}{M'^{-1}_{kj} - 1}$. Multiplying this difference by γ yields the difference in $\hat{q}_k$ due to removal of $\tau^{(j)}$. A change in the value of $\hat{q}_k$ is identical in its effect on the value estimation to changing $r^{(k)}$, a change which is mediated to $\hat{q}_i$ through Φ_{ik}. In the special case that $(x^{(j)}, a^{(j)})$ is the only neighbor of $(x'^{(k)}, \pi_e(x'^{(k)}))$, the value estimate $\hat{q}'_k$ changes from $\hat{q}_j$ to zero.

Combining the two ways in which removal of $\tau^{(j)}$ changes the estimated value $\hat{q}_i$ yields the individual influence:

$$I_{i,j} = \mathbb{I}(\Delta_{ij})\frac{\hat{q}_i - \left(r^{(j)} + \gamma\hat{q}'_j\right)}{M_{ij}^{-1} - 1} + \sum_{k:\Delta_{k'j}} I_{(i,j)}^{(k)}, \tag{9}$$

where we define

$$I_{i,j}^{(k)} = \begin{cases} \gamma\Phi_{ik}\frac{\hat{q}'_k - \left(r^{(j)} + \gamma\hat{q}'_j\right)}{M'^{-1}_{kj} - 1} & M'^{-1}_{kj} > 1 \\ \gamma\Phi_{ik}\hat{q}_j & M'^{-1}_{kj} = 1. \end{cases} \tag{10}$$

Computational complexity. The matrix formulation of kernel based FQE allows us to compute an individual influence in constant time, making influence analysis of the entire dataset possible in $\mathcal{O}(N|\mathcal{D}_0^*|)$ time. Furthermore, the sparsity of $\mathbf{M}$ and $\mathbf{M}'$ allows the FQE itself to be done in $\mathcal{O}(N^2T)$. See Appendix 1.2 for a full discussion.

5.2. Linear Least Squares FQE

In linear least squares FQE, the policy value function $\hat{q}(x, a)$ is approximated by a linear function $\hat{q}(x, a) = \boldsymbol{\psi}(x, a)^\top\mathbf{w}$ where $\boldsymbol{\psi}(x, a)$ is a D-dimensional feature vector for a state-action pair. Let $\mathbf{\Psi} \in \mathbb{R}^{N\times D}$ be the sample matrix of $\boldsymbol{\psi}(x, a)$. Define vector $\boldsymbol{\psi}_\pi(x) = \gamma\boldsymbol{\psi}(x, \pi_e(x))$ and let $\mathbf{\Psi}_p \in \mathbb{R}^{N\times D}$ be the sample matrix of $\boldsymbol{\psi}_\pi(x')$. The least-squares solution of $\mathbf{w}$ is $(\mathbf{\Psi}^\top\mathbf{\Psi} - \gamma\mathbf{\Psi}^\top\mathbf{\Psi}_p)^{-1}\mathbf{\Psi}^\top\mathbf{r}$ (See Appendix 2 for full derivation).

Let $\mathbf{w}_{-j}$ be the solution of linear least squares FQE after removing $\tau^{(j)}$, and $\mathbf{\Psi}_{-j}$, $\mathbf{r}_{-j}$, and $\mathbf{\Psi}_{p,-j}$ be the corresponding matrices and vectors without the $\tau^{(j)}$. Then, $\mathbf{w}_{-j} = (\mathbf{\Psi}_{-j}^\top\mathbf{\Psi}_{-j} - \gamma\mathbf{\Psi}_{-j}^\top\mathbf{\Psi}_{p,-j})^{-1}\mathbf{\Psi}_{-j}^\top\mathbf{r}_{-j}$. The key challenge of computing the influence function is computing $\mathbf{w}_{-j}$ in an efficient manner that avoids recomputing a costly matrix inverse for each j. Let $\mathbf{C}_{-j} = (\mathbf{\Psi}_{-j}^\top\mathbf{\Psi}_{-j} - \gamma\mathbf{\Psi}_{-j}^\top\mathbf{\Psi}_{p,-j})$ and $\mathbf{C} = (\mathbf{\Psi}^\top\mathbf{\Psi} - \gamma\mathbf{\Psi}^\top\mathbf{\Psi}_p)$. We compute $\mathbf{w}_{-j}$ as follows:

$$\mathbf{B}_j \leftarrow \mathbf{C}^{-1} + \frac{\mathbf{C}^{-1}\boldsymbol{\psi}_j\boldsymbol{\psi}_j^\top\mathbf{C}^{-1}}{1 - \boldsymbol{\psi}_j^\top\mathbf{C}^{-1}\boldsymbol{\psi}_j} \tag{11}$$

$$(\mathbf{C}_{-j})^{-1} \leftarrow \mathbf{B}_j - \frac{\gamma\mathbf{B}_j\boldsymbol{\psi}_j\boldsymbol{\psi}_{\pi,j}^\top\mathbf{B}_j}{1 + \gamma\boldsymbol{\psi}_{\pi,j}^\top\mathbf{B}_j\boldsymbol{\psi}_j} \tag{12}$$

$$\mathbf{w}_{-j} \leftarrow (\mathbf{C}_{-j})^{-1}\left(\mathbf{\Psi}^\top\mathbf{r} - r^{(j)}\boldsymbol{\psi}_j\right) \tag{13}$$

The proof of correctness is in Proposition 3 in Appendix 2. The individual influence function is then simply:

$$I_{i,j} = \boldsymbol{\psi}(s^{(i)}, a^{(i)})^\top(\mathbf{w}_{-j} - \mathbf{w}). \tag{14}$$

Computational complexity. The bottleneck of computing $\mathbf{w}_{-j}$ is the matrix multiplication of $D \times D$ matrices which takes at most $\mathcal{O}(D^3)$. All the other matrix multiplications involving size N, e.g. $\mathbf{\Psi}^\top\mathbf{r}$, do not depend on j and could be cached from the original OPE. Thus, the overall complexity for computing $I_{i,j}$ for all i and j is $\mathcal{O}(ND^3)$. Assuming $N > D$, the complexity of the original OPE algorithm is $\mathcal{O}(ND^2)$, where the bottleneck is computing $\mathbf{\Psi}^\top\mathbf{\Psi}$.

5.3. Importance Sampling

IS methods are essentially weighted averages over returns of trajectories, and therefore computing the total influence of a trajectory in a dataset can easily be performed in constant time, as long as certain values a cached. For example, the influence of the j^{th} trajectory for standard IS is

$$I_j = \frac{1}{N-1}\left(\hat{v} - w_{0:T}^{(j)} g_T^{(j)}\right), \tag{15}$$

where N is the number of trajectories, and $w_{0:T}^{(j)}$ and $g_T^{(j)}$ are the IS weight and return of the j^{th} trajectory, respectively. In Appendix 3 we present the derivation of the influence for IS, WIS, PDIS, DR and WDR estimators.

6. Illustration of influence functions in a sequential setting

We now demonstrate and give intuition for how the influence behaves in an RL setting. For the demonstrations and experiments presented throughout the rest of the paper we use the kernel-based FQE method.

Several factors determine the influence of a transition. For transitions to be influential they must have actions which are possible under the evaluation policy and form links in sequences which result in returns different than the expected value. Furthermore, transitions will be more influential the less neighbors they have.

To demonstrate this intuition we present in Figure 2 trajectories from a 2D continuous navigational domain [1]. The agent starts at the origin and takes noisy steps of length 1 at 45° to the axes. The reward for a given transition is a function of the state and has the shape of a Gaussian centered along the approximate path of the agent, represented as the background heat map in Figure 2 (top), where observed transitions are drawn as black line segments. Because distances for the FQE are computed in the state-action space, in this example all actions in the data are the same to allow for distances to be visualized in 2D.

To illustrate how influence is larger for transitions with few neighbors, we removed most of the transitions in two regions (denoted II and III), and compared the distribution of influences in these regions with influences in a data dense region (denoted I). Figure 2 (bottom) shows the distribution over 200 experiments (in each experiment, new data is generated) of the influences of transitions in the different regions. The influence is much higher for transitions in sparse regions with few neighbors, as can be seen by comparing the distributions in regions I and II. This is a desired property, as in analysis of the OPE process, we'd like to be able to present domain experts with transitions that have few neighbors where the sampling variance of a particular transition could have large effect on evaluation.

In region III, despite the fact that the observations examined also have very few neighbors, their influence is extremely low, as they don't lead to any regions where rewards are gained by the agent.

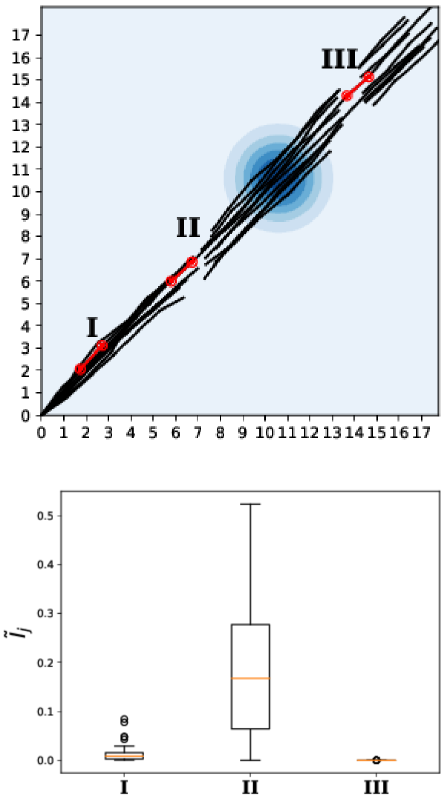


Figure 2. **Conceptual demonstration on a 2D domain.** For transitions in the data to have high influence, they must agree with the evaluation policy and lead to rewarding regions in the state-action space. Additionally, the influence of transitions decreases with the number of their close neighbors.

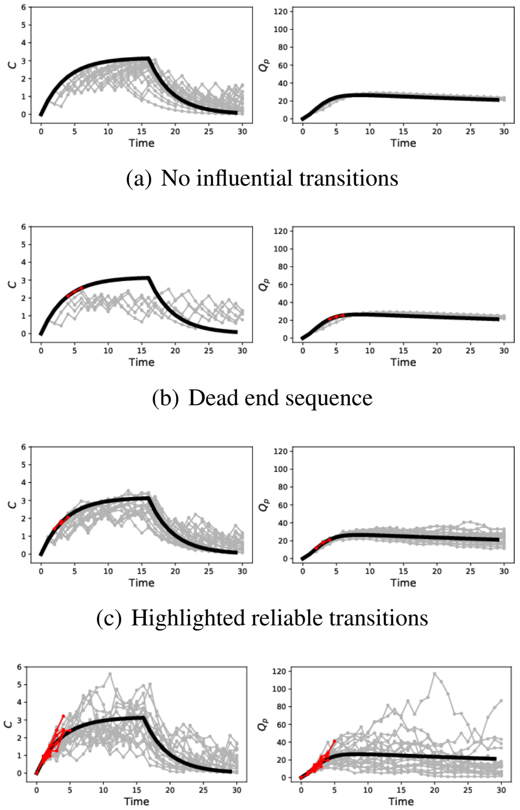
7. Experiments

7.1. Medical cancer simulator

To demonstrate the different ways in which influence analysis can allow domain experts to either increase our confidence in the validity of OPE or identify instances where they are invalid, we first present results on a simulator of cancer dynamics. The 4 dimensional states of the simulator approximate the dynamics of tumor growth, with actions consisting administration of chemotherapy at each timestep representing one month. See Ribba et al. (2012) for details.

In Figure 3 we present four cases in which we attempt to evaluate the policy of treating a patient for 15 months and then discontinuing chemotherapy until the end of treatment at 30 months. Each subplot in Figure 3 shows two of the four state variables as a function of time, under different

[1] Code for reproducing the results in this paper can be found at https://github.com/dtak/interpretable_ope_public.git



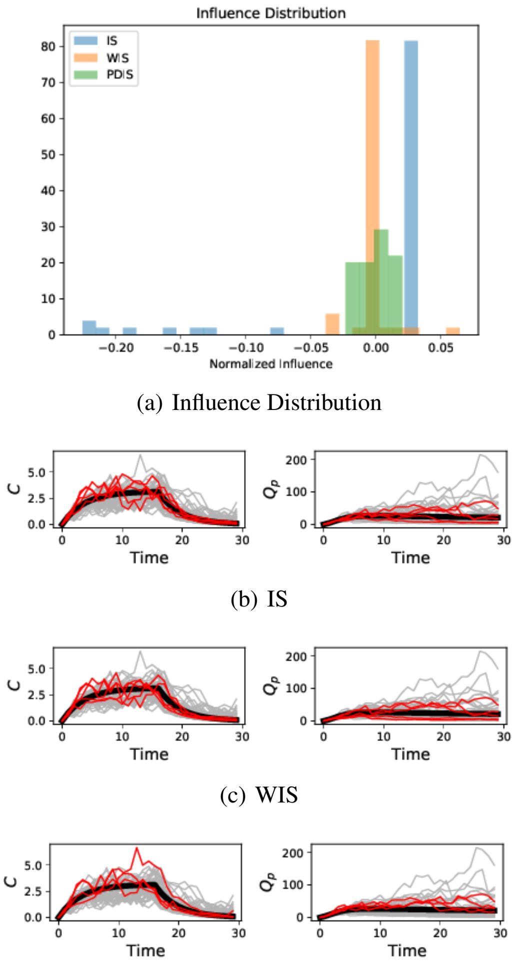
(a) No influential transitions

(b) Dead end sequence

(c) Highlighted reliable transitions

(d) Highlighted problematic transitions

Figure 3. **Influence analysis for simulated cancer data.** Analysis of synthetic cancer simulations demonstrates how influence analysis can differentiate between different diagnostics of the OPE process.



(a) Influence Distribution

(b) IS

(c) WIS

(d) PDIS

Figure 4. **IS influence analysis for simulated cancer data.** For the same dataset, different estimators have different influence distributions, and for each estimator different trajectories have high influence.

conditions which might make evaluation more difficult, such as difference in behavior policy or stochasticity in the environment. The heavy black line represents the expectation of each state dimension at each time-step under the evaluation policy, while the grey lines represent observed transitions under the behavior policy which is ϵ-greedy with respect to the evaluation policy. In all figures, we highlight in red all influential transitions our method would have highlighted for review by domain experts ($\tilde{I}_c = 0.05$).

Case 1: OPE seems reliable. Figure 3(a) represents a typical example where the OPE can easily be trusted. Despite the large difference between the evaluation and behavior policy ($\epsilon = 0.3$), enough trajectories have been observed in the data to allow for proper evaluation, and no transition is flagged as being too influential. The value estimation error in this example is less than 1% and our method correctly labels this dataset as reliable.

Case 2: Unevaluatable. Figure 3(b) is similar in experimental conditions to (a) ($\epsilon = 0.3$ and deterministic transitions), but with less collected data, so that the observations needed to properly estimate the dynamics are not in the data. This can be seen by the lack of overlap between the observed transitions and the expected trajectory, and results in a 38% value estimation error. In real life we will not know what the expected trajectory under the evaluation policy looks like, and therefore will not be able to make the comparison and detect the lack of overlap between transitions under the evaluation and behavior policies. However, our method highlights a very influential sequence which terminates at a dead-end, and thus will correctly flag this dataset as not sufficient for evaluation. Our method in this case is confident enough to dismiss the results of evaluation without need for domain experts, but can still inform experts on what type of data is lacking in order for evaluation to be feasible.

Case 3: Humans might help. In Figures 3(c-d), $\epsilon = 0.3$, but the dynamics have different levels of stochasticity. The less stochastic dynamics in 3(c) allow for relatively accurate evaluation (8% error) but our method identifies several influential transitions which must be presented to a domain expert. These transitions lie on the expected trajectory, and thus a clinician would verify that they represent a typical response of a patient to treatment. This is an example in which our method would allow a domain expert to verify the validity of the evaluation by examining the flagged influential transitions.

Conversely, in 3(d) some extreme outliers lead to a large estimation error (23% error). The influential transitions identified by our method are exactly those which start close to the expected trajectory but deviate significantly from the expected dynamics. A domain expert presented with the these transitions would easily be able to note that the OPE heavily relies on atypical patients and rightly dismiss the validity of evaluation.

To summarize, we demonstrated that analysis of influences can both validate or invalidate the evaluation without need for domain experts, and in intermediate cases present domain experts with the correct queries required to gain confidence in the evaluation results or dismiss them.

Influence analysis for IS - Influence is a method specific quantity. In Figure 4 we present influence analysis results for the cancer environment, with different importance sampling methods. Unlike the FQE experiment where we performed influence analysis of the same estimator for different datasets, here we analyze the same dataset for three different OPE estimaors - IS, WIS and PDIS. In Figure 4 (a) we plot the distribution of the influence of all trajectories in the data, and see that the distributions are qualitatively different for each estimator. Furthermore, in 4 (b-d) we highlight the 5 most influential trajectories for each estimator, and see that they are different for each estimator. The key point we wish to highlight is that influence analysis identifies features of the interaction between a dataset and an estimator, and not of the data alone. This makes sense, as different OPE methods are robust or sensitive to different types of noise or artefacts in the data.

7.2. Analysis of real ICU data - MIMIC III

To show how influence analysis can help debug OPE for a challenging healthcare task, we consider the management of acutely hypotensive patients in the ICU. Hypotension is associated with high morbidity and mortality (Jones et al., 2006), but management of these patients is not standardized as ICU patients are heterogeneous. Within critical care, there is scant high-quality evidence from randomized controlled trials to inform treatment guidelines (de Grooth et al., 2018; Girbes & de Grooth, 2019), which provides an opportunity for RL to help learn better treatment strategies. In collaboration with an intensivist, we use influence analysis to identify potential artefacts when performing OPE on a clinical dataset of acutely hypotensive patients.

Data and evaluation policy. Our data source is a subset of the publicly available MIMIC-III dataset (Johnson et al., 2016). See Appendix 4 for full details of the data preprocessing. Our final dataset consists of 346 patient trajectories (6777 transitions) for learning a policy and another 346 trajectories (6863 transitions) for evaluation of the policy via OPE and influence analysis.

Our state space consists of 29 relevant clinical variables, summarizing current physiological condition and past actions. The two main treatments for hypotension are administration of an intravenous (IV) fluid bolus or initiation of vasopressors. We bin doses of each treatment into 4 categories for ”none”, ”low”, ”medium” and ”high”, so that the full action space consists of 16 discrete actions. Each reward is a function of the next blood pressure (MAP) and takes values in $[-1, 0]$. As an evaluation policy, we use the most common action of a state’s 50 nearest neighbors. This is setup is equivalent to constructing a decision assistance tool for clinicians by recommending the common practice action for patients, and using OPE combined with influence analysis to estimate the efficacy of such a tool. See Appendix 4 for more details on how we setup the RL problem formulation, and for the kernel function used to compute nearest-neighbors.

Presenting queries to a practicing intensivist. Running influence analysis flags 6 influential ($\tilde{I}_C = 0.05$). We show 2 of these transitions in Figure 5 and the rest in Appendix 5. While this analysis highlights individual transitions, our results figures display additional context before and after the suspect transition to help the clinician understand what might be going on.

In Figure 5, each column shows a transition flagged by influence analysis. The top two rows show actions taken (actual treatments in the top row and binned actions in the second row). The remaining three rows show the most important state variables that inform the clinicians’ decisions: blood pressure (MAP), urine output, and level of consciousness (GCS). For these three variables, the abnormal range is shaded in red, where the blood pressure shading is darker highlighting its direct relationship with the reward. Vertical grey lines represent timesteps, and the highlighted influential transition is shaded in grey.

Outcome: Identifying and removing an influential, buggy measurement. The two transitions in Figure 5 highlight potential problems in the dataset that have a large

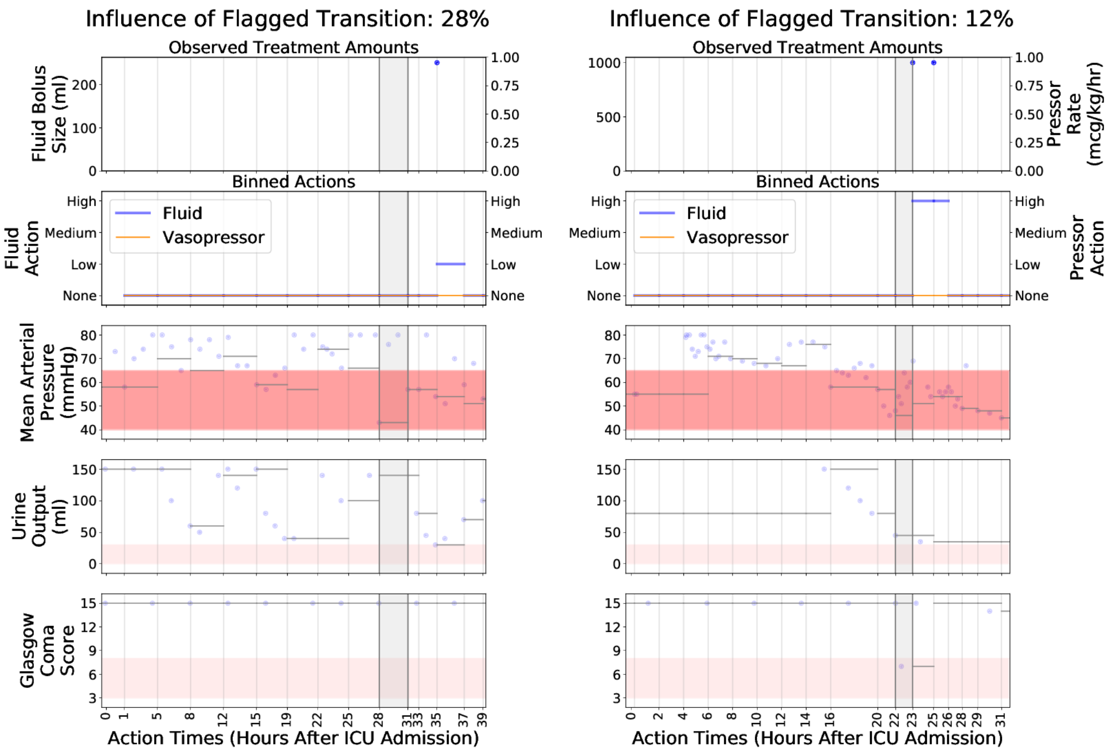


Figure 5. Influence analysis on our real-world dataset discovered six transitions in the evaluation dataset that were especially influential on our OPE. We display two of them in this figure, see Appendix 5 for the remaining four.

influence. In the first transition (left), a large drop in blood pressure is observed at the starting time of this transition, potentially indicating a dangerous hypotensive state. Suprisingly, the patient received no treatment, and this unusual transition has a 29% influence on the OPE estimate. Given additional context just before and after the transition, showing otherwise stable MAP and GCS (patient was conscious and alert) as well as a normal urine output, the intensivist determined the single low MAP value was likely either a measurement error or a clinically insignificant transient episode of hypotension. After correcting the outlier MAP measurement to its most recent normal value (80mmHg) and then rerunning FQE and the influence analysis, the transition no longer has high influence and was not flagged.

Outcome: Identifying and correcting a temporal misalignment. The second highlighted transition (right) features a sudden drop in GCS and worsening MAP values, indicating a sudden deterioration of the patient's state, but treatment is not administered until the next timestep. The intensivist attributed this finding to a time stamp recording error. Again, influence analysis identified an inconsistency in the original data which had undue impact on evaluation. After correcting the inconsistency by shifting the two fluid treatments back by one timestep each, we found that the transition no longer had high influence and was not flagged.

8. Discussion

A key aim of this paper is to formulate a framework for using domain expertise to help in evaluating the trustworthiness of OPE methods for noisy and confounded observational data. The motivation for this research direction is the intersection of two realities: for messy real-world applications, the data itself might never be enough; and domain experts will always need to be involved in the integration of decision support tools, so we should incorporate their expertise into the evaluation process. We showcased influence analysis as one way of performing this task for value-based and IS OPE, but emphasize that such measures can and should be incorporated into other methods as well. For example, when modeling the dynamics in model-based OPE, the results can be tested for their agreement with expert intuition.

We stress that research to integrate human input into OPE methods to increase their reliability complements, and does not replace, the approaches for estimating error bounds and uncertainties over the errors of OPE estimates. The fact that traditional theoretical error bounds rely so heavily on assumptions which are generally impossible to verify from the data alone highlights the need for other techniques for gauging to what extent these assumptions hold.

References

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

Dann, C., Li, L., Wei, W., and Brunskill, E. Policy certificates: Towards accountable reinforcement learning. *arXiv preprint arXiv:1811.03056*, 2018.

de Grooth, H.-J., Postema, J., Loer, S. A., Parienti, J.-J., Oudemans-van Straaten, H. M., and Girbes, A. R. Unexplained mortality differences between septic shock trials: a systematic analysis of population characteristics and control-group mortality rates. *Intensive care medicine*, 44(3):311–322, 2018.

Ernst, D., Geurts, P., and Wehenkel, L. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6(Apr):503–556, 2005.

Girbes, A. R. J. and de Grooth, H.-J. Time to stop randomized and large pragmatic trials for intensive care medicine syndromes: the case of sepsis and acute respiratory distress syndrome. *Journal of Thoracic Disease*, 12(S1), 2019. ISSN 2077-6624. URL http://jtd.amegroups.com/article/view/33636.

Gottesman, O., Johansson, F., Meier, J., Dent, J., Lee, D., Srinivasan, S., Zhang, L., Ding, Y., Wihl, D., Peng, X., et al. Evaluating reinforcement learning algorithms in observational health settings. *arXiv preprint arXiv:1805.12298*, 2018.

Gottesman, O., Johansson, F., Komorowski, M., Faisal, A., Sontag, D., Doshi-Velez, F., and Celi, L. A. Guidelines for reinforcement learning in healthcare. *Nat Med*, 25(1): 16–18, 2019a.

Gottesman, O., Liu, Y., Sussex, S., Brunskill, E., and Doshi-Velez, F. Combining parametric and nonparametric models for off-policy evaluation. In *International Conference on Machine Learning*, pp. 2366–2375, 2019b.

Hanna, J. P., Stone, P., and Niekum, S. Bootstrapping with models: Confidence intervals for off-policy evaluation. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Jiang, N. and Li, L. Doubly robust off-policy value evaluation for reinforcement learning. *arXiv preprint arXiv:1511.03722*, 2015.

Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

Jones, A. E., Yiannibas, V., Johnson, C., and Kline, J. A. Emergency department hypotension predicts sudden unexpected in-hospital mortality: a prospective cohort study. *Chest*, 130(4):941–946, 2006.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1885–1894. JMLR. org, 2017.

Lage, I., Ross, A., Gershman, S. J., Kim, B., and Doshi-Velez, F. Human-in-the-loop interpretability prior. In *Advances in Neural Information Processing Systems*, pp. 10159–10168, 2018.

Le, H. M., Voloshin, C., and Yue, Y. Batch policy learning under constraints. *arXiv preprint arXiv:1903.08738*, 2019.

Munos, R. and Moore, A. Variable resolution discretization in optimal control. *Machine learning*, 49(2-3):291–323, 2002.

Oberst, M. and Sontag, D. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, pp. 4881–4890, 2019.

Precup, D. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, pp. 80, 2000.

Ribba, B., Kaloshi, G., Peyre, M., Ricard, D., Calvez, V., Tod, M., Čajavec-Bernard, B., Idbaih, A., Psimaras, D., Dainese, L., et al. A tumor growth inhibition model for low-grade glioma treated with chemotherapy or radiotherapy. *Clinical Cancer Research*, 18(18):5071–5080, 2012.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tamuz, O., Liu, C., Belongie, S., Shamir, O., and Kalai, A. T. Adaptively learning the crowd kernel. *arXiv preprint arXiv:1105.1033*, 2011.

Thomas, P. and Brunskill, E. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 2139–2148, 2016.

Thomas, P. S., Theocharous, G., and Ghavamzadeh, M. High-confidence off-policy evaluation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Thomas, P. S., da Silva, B. C., Barto, A. G., Giguere, S., Brun, Y., and Brunskill, E. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468):999–1004, 2019.