# Off-policy Evaluation in Infinite-Horizon Reinforcement Learning with Latent Confounders

**Anonymous authors**
Paper under double-blind review

## 1 Introduction

Off-policy evaluation (OPE) in reinforcement learning (RL) considers the problem of estimating the value (i.e., the average long-term reward) of a *target policy* based on data collected by following a *behavior policy*. OPE is particularly important in settings where experimentation is limited, such as healthcare and education. But, in these very same settings, observed actions are often confounded by *unobserved* variables making OPE even more difficult.

In this work, we study OPE in an infinite-horizon, ergodic Markov decision process with unobserved confounders, where states and actions can act as proxies for the unobserved confounders. We show how, given only a latent variable model for states and actions, the policy value can be identified from off-policy data. Our method involves two stages. In the first, we show how to use proxies to estimate stationary distribution ratios, extending recent work on infinite-horizon OPE to the confounded setting. In the second, we show optimal balancing can be combined with such learned ratios to obtain policy value while avoiding direct modeling of reward functions. We establish theoretical guarantees of consistency and demonstrate our method empirically.

## 2 Problem Setting

We consider Markov Decision Processes with Unmeasured Confounding, or MDPUC (Zhang & Bareinboim, 2016). An MDPUC is specified by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{U}, P_T, \mathcal{R}, f_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A} = \{1, \ldots, m\}$ is the space of actions, $\mathcal{U}$ is the space of confounders, $P_T(s' \mid s, a, u)$ gives the probability of transitioning to state $s'$ from state $s$ given action $a$ and confounders $u$, $\mathcal{R}(s, a, u)$ gives the distribution of reward given action $a$ was taken in state $s$ with confounders $u$, and $f_0$ gives the distribution over starting states. Note that in most cases we are only interested in the expected reward in each configuration, so we let $\mu_a(s, u) = \mathbb{E}[\mathcal{R}(s, a, u)]$. An important assumption we make here is that the confounder values $U$ at each time step are iid, which differentiates the MDPUC setting from the more general POMDP setting. Finally, we use $S'$ to refer to the state succeeding state $S$ in a trajectory.

We assume access to $N \geq 1$ trajectories of off-policy data, of lengths $T_1, \ldots, T_N$. At each time period of each trajectory we assume that we observe the state $S$, the action that was taken in that state $A$, and the corresponding reward that was received $R$. Importantly, we do *not* observe the corresponding confounder value $U$. We assume that each trajectory was logged from a common behavior policy $\pi_b$, which depends on the confounders, where $\pi_b(a \mid s, u)$ gives the probability that $\pi_b$ takes action $a$ given state $s$ and confounders $u$. In addition we will frequently index our data by concatenating the trajectories together and using indices $1, \ldots, n$, where $n = \sum_{i=1}^N T_i$.

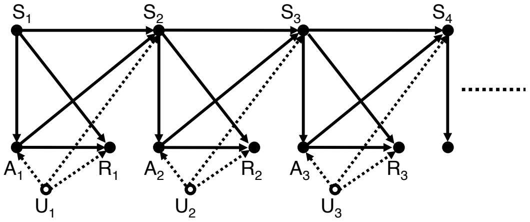
Our task is to estimate the value of some evaluation policy $\pi_e$, which follows the same semantics as $\pi_b$, and whose actions may optionally depend on the confounders $U$ (for simplicity, even in the case that its actions depend on $S$ only, we still use the notation $\pi_e(a \mid s, u)$). Importantly, we assume that Markov chain of tuples $(S, A, U, R, S')$ is ergodic for policies $\pi_b$ and $\pi_e$, meaning that the policies have unique stationary distributions. We refer to these stationary distributions as $f_b$ and $f_e$, and denote expectations with respect to them as $\mathbb{E}_b$ and $\mathbb{E}_e$. Then we define the *value* of policy $\pi_e$ to be

$$V(\pi_e) = \mathbb{E}_e[\mu_A(S, U)]. \tag{1}$$

In addition we will use the notation $d(X)$ to denote the stationary density ratio under $\pi_e$ versus $\pi_b$, for any random variable $X$ that is measurable with respect to $(S, A, U, S')$. That is, $d$ is defined such that for any such variable $X$ and any function $g$ we have $\mathbb{E}_{f_e}[g(X)] = \mathbb{E}_{f_b}[d(X)g(X)]$. Note that this involves slight abuse of notation since the function $d$ depends on the random variable $X$, but the definition of $d$ should be clear in context.

Finally, we assume that we have access to an oracle $\hat{\varphi}$, which provides an approximation of the posterior distribution of $U$ given $S$, $A$, and $S'$. Such an oracle can be implemented using existing inference algorithms for probabilistic models with latent variables. Note that by the MDPUC structure (Figure 1) $U$ is conditionally indpedentant of all other states and actions given this triplet. It is assumed that $\hat{\varphi}$ allows for approximate sampling of $U$ values from the posterior (in the continuous $U$ case), or defines the approximate posterior (in the discrete $U$ case).

Figure 1: Graphical representation of MDPUC in which action selection, state transition, and reward value are confounded.



## 3 Methodology

### 3.1 Weighted Estimator for Policy Value

We can first note that the policy value we are estimating is given by

$$\begin{aligned} V(\pi_e) &= \mathbb{E}_e[\mu_A(S,U)] \\ &= \sum_{a=1}^{m} \mathbb{E}_e[\pi_e(a \mid S,U)\mu_a(S,U)] \\ &= \sum_{a=1}^{m} \mathbb{E}_b[d(S)\pi_e(a \mid S,U)\mu_a(S,U)], \end{aligned}$$

where the last step follows from the observation that $d(S,U) = d(S)$, since the conditional distribution of $U$ given $S$ is the same under each policy. Next, given any weighting random variable $W$ measurable in $S$, $A$, and $S'$, the expected value of the corresponding weighted estimator is given by

$$\begin{aligned} \mathbb{E}_b[WR] &= \mathbb{E}_b[W\mu_A(S,U)] \\ &= \sum_{a=1}^{m} \mathbb{E}_b[W\delta_{Aa}\mu_a(S,U)]. \end{aligned}$$

Now consider the weighted estimator for $V(\pi_e)$ given by $\hat{\tau}_W = \sum_{i=1}^{n} W_i R_i$, where $\{R_1, \dots, R_n\}$ is the set of observed rewards in our $N$ trajectories. Then the above suggests that we can approximate the bias of evaluation by $(1/n)\sum_{i=1}^{n}\sum_{a=1}^{m} f_{ia}\mathbb{E}[\mu_a(S_i,U_i) \mid S_i, A_i, S'_i]$, where $f_{ia} = W_i\delta_{A_i a} - d(S_i)\pi_e(a \mid S_i, U_i)$. Furthermore, given that our data comes from an ergodic Markov Chain, it is intuitive that we should be able to upper bound the variance of this estimator by $C\|W\|^2$ for some constant $C$, where $\|\cdot\|$ denotes the Euclidean norm. These intuitions lead us to the following theorem:

**Theorem 1** (MSE Upper Bound). *For any vector of weights $W$ and vector of functions $g = g_1, \dots, g_m$, define*

$$J(W,g) = \left(\frac{1}{n}\sum_{i=1}^{n}\sum_{a=1}^{m} f_{ia}\mathbb{E}[g_a(S_i,U_i) \mid S_i, A_i, S'_i]\right)^2 + C\|W\|^2. \tag{2}$$

*Then if $C$ is sufficiently large and $J(W, \mu) = O_p(1/n)$, where $\mu = \mu_1, \ldots, \mu_m$ are the true mean reward functions, it follows that $\hat{\tau}_W = V(\pi_e) + O_p(1/\sqrt{n})$.*

The proof shares a similar structure of that of the corresponding theorem in Bennett & Kallus (2019) for confounded contextual bandits. The main difference is that we need to appeal to the Markov Chain central limit theorem (CLT) rather than the regular CLT since by assumption our data comes from an ergodic Markov chain. We refer the reader to the corresponding proof for details.

This suggests finding weights $W$ for weighted evaluation that minimize $\sup_{g \in \mathcal{G}} J(W, g)$ for some vector-valued function class $\mathcal{G}$. It follows easily from the above that if $\mu \in \mathcal{G}$ and we can minimize this upper bound uniformly over $\mathcal{G}$ at an $O(1/n)$ rate, then this estimator gives $O_p(1/\sqrt{n})$-consistency for $V(\pi_e)$. We can also note that if $\mathcal{G}$ is a class of functions with norm at most $\gamma$, then the choice of $C$ is implicit from the choice of $\gamma$ (since scaling up all functions $g_a$ by a factor of $\lambda$ gives an equivalent optimization problem to scaling down $C$ by a factor of $\lambda^2$), so we can arbitrarily fix $C$ or $\gamma$ and perform hyperparameter optimization over the other.

In our experiments, we choose $W$ using this adversarial estimator, with the function class $\mathcal{G}$ defined by the norm $\|g\|^2 = \sum_a \|g_a\|_K$, where $\| \cdot \|_K$ is the norm for the RKHS with kernel $K$. As in Bennett & Kallus (2019), we can show that this estimator is given by

$$\arg\min_W \left( \frac{1}{n^2} \sum_{i,j,a} \mathbb{E}[f_{ia} \tilde{f}_{ja} K((U_i, S_i), (\tilde{U}_j, S_j)) \mid S_i, A_i, S_i', S_j, A_j, S_j'] \right) + C\|W\|^2,$$

where $\tilde{U}_i$ denotes a shadow variable iid to $U_i$ given $S_i, A_i, S_i'$. This objective is approximated using $\hat{\varphi}$, which gives a QP to be solved. The exact QP will be described in the final version of our paper.

### 3.2 Learning Stationary Density Ratio

The above algorithm for learning evaluation weights requires knowing the state stationary density ratio $d(S)$. We describe here a conditional moment formulation for learning this function. First, analogously to Liu et al. (2018), this ratio must satisfy the conditional moment restriction:

$$d(S') = \mathbb{E}_b[d(S)\beta(S, A, S') \mid S'],$$

where $\beta(S, A, S') = \mathbb{E}_{\pi_b}[\pi_e(A \mid S, U)/\pi_b(A \mid S, U) \mid S, A, S']$. This implies that for every measurable function $h$ we must have $\mathbb{E}_b[(d(s)\beta(S, A, S') - d(S'))h(S')] = 0$. In addition the ratio must satisfy the trivial moment restriction $\mathbb{E}_b[d(S)] = 0$. As in Liu et al. (2018) it is easy to argue that $d$ is the unique function satisfying the above moment restrictions.

Now Bennett et al. (2019) has previously provided an efficient GMM-based formulation for solving such conditional moment problems efficiently using machine learning, which we can adapt to this learning problem. Specifically, we let $\mathcal{D}$ and $\mathcal{H}$ both be RKHS function classes, and define $M_i(d, f, c, c') = (\hat{\beta}_i d(S_i) - d(S_i'))f(S_i') + c(d(S_i) - 1) + c'(d(S_i') - 1)$, where $\hat{\beta}_i$ is an estimate of $\beta(S_i, A_i, S_i')$ (obtained by using $\hat{\varphi}$). Then our estimator for $d$ is given by

$$\hat{d} = \arg\min_{d \in \mathcal{D}} \sup_{h \in \mathcal{H}, \|c\| \leq \lambda, \|c'\| \leq \lambda} \frac{1}{n} \sum_{i=1}^{n} \left( M_i(d, f, c, c') - \frac{1}{4} M_i(\tilde{d}, f, c, c')^2 \right),$$

where $\tilde{d}$ is some prior estimate of $d$. In practice we solve the above iteratively until convergence, each time using the previous estimate $\hat{d}$ as $\tilde{d}$, with our initial estimate given by $\hat{d}(s) = 1 \; \forall s$. Given $\mathcal{D}$ and $\mathcal{H}$ are RKHS function classes this optimization problem has a closed form solution, which will be described in the final version of our paper.

## 4 Experimental Results

In this section, we empirically demonstrate our method. We consider the C-ModelWin environment (Fig. 2) that is a confounded variant of the ModelWin environment (Thomas & Brunskill, 2016). C-Modelwin has 3 states and 2 actions. At each step, there is a confounder $U$ that affects action selection, reward value, and state transition. The agent always begins in $S_0$. At step $i$, the agent

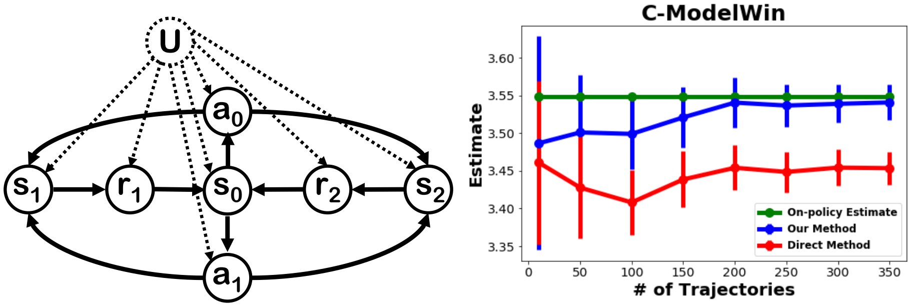


Figure 2: (Left) The C-ModelWin problem. (Right) OPE results with increasing data size.

chooses between two actions $A_0$ and $A_1$ with the probability of $1 - \pi - U_i$ and $\pi + U_i$, respectively. $\pi$ is a parameter that is distinct for behavior and evaluation policies. In our experiments, $\pi = 0.7$ for the behavior policy and and $\pi = 0.1$ for the evaluation policy. In addition, $U_i$s are iid variables that could be $0.1$ and $0.2$ with probabilities of $0.3$ and $0.7$, respectively. At step $i$, if the agent is at $S_0$ and chooses $A_0$, then with the probability of $0.7 + U_i$ and $0.3 - U_i$ it makes a transition to $S_1$ and $S_2$ and receives a zero reward. If it chooses $A_1$, then with the probability of $0.3 - U_i$ and $0.7 + U_i$ it makes a transition to $S_1$ and $S_2$ and receives a zero reward. When the agent makes a transition from $S_1$ to $S_0$, it receives a reward of $10 + 20 \times U_i$ and when it makes a transition from $S_2$ to $S_0$, it receives a reward of $-10 - 20 \times U_i$.

We compare our method with the direct estimation method. In this approach, we impute $U$ values for each data point by sampling from the posterior probability $\hat{\varphi}(.|S, A, S')$. We use this dataset to fit a regression model for the mean reward $\mu(.)$ given $S$, $A$, and $U$. Since C-ModelWin is a discrete environment, we can simply fit this regression model using tabular methods. We can then write the predicted reward as $(1/n) \sum_{i=1}^{n} d(S_i) \sum_u \hat{\varphi}(u|S_i, A_i, S_i') \sum_a \pi_e(a|S_i, u)\mu(S_i, a, u)$.

Figure 2 compares the estimates of our method with the direct estimator and the true on-policy reward. Based on this figure, as we have more trajectories both bias and variance of our estimator converge to zero. On the other hand, although the variance of the direct estimator decreases, its bias does not vanish. Therefore, as we can see the estimate of our method gets close to the true on-policy estimate while this does not hold for the direct estimator. It is worth mentioning that while the true on-policy reward in this problem is 3.55, simply averaging off-policy rewards (i.e., having a naive estimator) gives the estimate of 0.29 which is significantly far from the true value and shows the impact of having confounders.

## References

Andrew Bennett and Nathan Kallus. Policy evaluation with latent confounders via optimal balance. In *Advances in Neural Information Processing Systems*, pp. 4827–4837, 2019.

Andrew Bennett, Nathan Kallus, and Tobias Schnabel. Deep generalized method of moments for instrumental variable analysis. In *Advances in Neural Information Processing Systems*, pp. 3559–3569, 2019.

Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *Advances in Neural Information Processing Systems*, pp. 5356–5366, 2018.

Philip S. Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 2139–2148, 2016.

Junzhe Zhang and Elias Bareinboim. Markov decision processes with unobserved confounders: A causal approach. Technical Report R-23, Columbia CausalAI Laboratory, 2016.