Efficient Multi-Policy Evaluation for Reinforcement Learning

Shuze Daniel Liu[1], **Claire Chen**[2], **Shangtong Zhang**[1]

[1]Department of Computer Science, University of Virginia
[2]School of Arts and Science, University of Virginia
shuzeliu@virginia.edu, clairechen@email.virginia.edu, shangtong@virginia.edu

Abstract

To unbiasedly evaluate multiple target policies, the dominant approach among RL practitioners is to run and evaluate each target policy separately. However, this evaluation method is far from efficient because samples are not shared across policies, and running target policies to evaluate themselves is actually not optimal. In this paper, we address these two weaknesses by designing a tailored behavior policy to reduce the variance of estimators across all target policies. Theoretically, we prove that executing this behavior policy with manyfold fewer samples outperforms on-policy evaluation on every target policy under characterized conditions. Empirically, we show our estimator has a substantially lower variance compared with previous best methods and achieves state-of-the-art performance in a broad range of environments.

Introduction

We explore the multi-policy evaluation problem, where we aim to estimate the performance of multiple target policies. In reinforcement learning (RL, Sutton and Barto (2018)), multi-policy evaluation is prevalent for model selections (Schulman et al. 2017; Prechelt 2002). A simple method to evaluate multiple policies is to perform online policy evaluation for each target policy separately. However, the number of required online samples scales up quickly with the number of target policies.

In many scenarios, heavily relying on massive online data is not preferable. Firstly, collecting massive online data can be both expensive and slow when interacting with the real world (Li 2019; Zhang 2023; Chen, Liu, and Zhang 2024; Liu, Chen, and Zhang 2024a). Secondly, even if there is a simulator, for complex problems such as data center cooling, each step may still cost 10 seconds (Chervonyi et al. 2022). Thus, building an RL system demanding millions of steps remains expensive.

To address the expensive nature of online data, offline RL is proposed to mitigate the dependency on online data. However, RL systems built only on offline datasets have uncontrolled bias. A policy showing high performance on offline data may actually perform very poorly in real deployment (Levine 2018). Therefore, both online and offline RL practitioners heavily use online methods to evaluate the performance of policies.

To efficiently evaluate multiple policies, previous works try to reuse online samples generated by other target policies. Agarwal et al. (2017) show that naively combining data generated by other policies may actually worsen the estimation. Data from other policies must be carefully reweighed before consideration. However, in multi-step reinforcement learning, those weights require knowing complicated covariance terms between every pair of target policies (Lai, Zou, and Song 2020). Such strong prior knowledge is rarely available and makes these methods impractical. To avoid complex weights, other literature (Dann, Ghavamzadeh, and Marinov 2023) tried to reuse online samples by assuming deterministic policies and a flexible environment that can start from any desired state. These assumptions rarely hold.

In our work, we design a tailored behavior policy to **efficiently** and **unbiasedly** evaluate all target policies. Our method does not require knowing any complex covariance and applies to general RL settings without any restrictive assumptions. Our contribution is two-fold.

Theoretically, our method is always unbiased (Theorem 1) and is proven to achieve lower variance than the on-policy estimator for each target policy under characterized conditions (Theorem 4, Theorem 3). Moreover, we introduce a similarity metric between policies and prove that the number of required samples for our method does not scale with the number of target policies under the similarity condition.

Empirically, compared with previous best methods, we show our estimator has a substantially lower variance. Our method requires much fewer samples to reach the same level of accuracy and achieves state-of-the-art performance in a broad range of environments.

Related Work

Multiple target policies. In multi-policy evaluation, traditional approaches often evaluate each policy separately using on-policy Monte Carlo methods. However, this ordinary method ignores the potential similarity between target policies and is crude for two reasons. **First**, the method does not utilize data sampled by other policies, causing the number of required online samples to scale quickly with the number of target policies. **Second**, even for a single target policy, the on-policy evaluation method is still not the optimal choice.

Through a tailored behavior policy (Liu and Zhang 2024), the variance of the on-policy Monte Carlo evaluation can be reduced while achieving an unbiased estimation.

To address the inefficiency in multi-policy evaluation problem, Dann, Ghavamzadeh, and Marinov (2023) present an algorithm to reuse online samples from target policies. However, their algorithm works only when all target policies are deterministic, which is also highly restricted. *By contrast, our method copes with stochastic policies.* The key difference is that they consider the plain approach by reusing samples from target policies, while we propose a tailored behavior for multiple target policies, which is designed to generate samples that all similar policies can efficiently share. Liu and Zhang (2024) also design a behavior policy for off-policy evaluation. However, they only consider a single target policy, which is narrower than our settings. In the empirical results section, we also show that our method outperforms theirs (Liu and Zhang 2024), in multi-policy evaluation problems under the multi-step RL setting. Using a shared behavior policy tailored for all similar target policies, our method achieves state-of-the-art performance and does better than all existing methods.

Multiple logging policies. Other approaches consider using data from multiple logging policies to perform off-policy evaluation, although only aiming at a single target policy. We call them logging policies because in their works (Agarwal et al. 2017; Lai, Zou, and Song 2020; Kallus, Saito, and Uehara 2021), data are previously logged from certain behavior policies and are fixed. This is different from our setting, in which we design an active data-collecting policy for multiple target policies.

Agarwal et al. (2017) point out that directly combining data from different policies may increases the estimation variance. They then propose two new estimators by reweighting data from different policies. However, their method is restricted to the contextual bandit setting. *By contrast, we work on the multi-step reinforcement learning setting, which is much broader.* Lai, Zou, and Song (2020) extend the method from Agarwal et al. (2017) into multi-step RL. Nevertheless, getting the desired weights for different logging policies requires knowing complicated covariance terms between every pair of logging policies. That is, given K logging policies, their method needs to compute K^2 covariances. Such strong prior knowledge is rarely available and is computationally expensive, making the method impractical. Furthermore, they ignore bias from any off-policy estimator. *By contrast, with the tailored behavior policy, our estimator is inherently and provably unbiased.* Kallus, Saito, and Uehara (2021) also explore off-policy evaluation with multiple target policies in RL setting. They combine the reweighting strategy with the control variate method, leading to a reduced variance estimation. However, getting the weights proposed by their method requires knowledge of state visitation densities, whose approximation is very challenging in MDPs with large stochasticity and function approximation (cf. model-based RL (Sutton 1990; Sutton et al. 2008; Deisenroth and Rasmussen 2011; Chua et al. 2018)). Due to this impracticability, Kallus, Saito, and Uehara (2021) only conduct experiment of their method in the contextual bandit setting, remaining the experiment on the multi-step RL setting untouched. *By contrast, our method avoids reliance on any terms that are impractical to estimate.*

Background

In this work, we focus on a finite horizon Markov Decision Process (MDP), as defined by Puterman (2014), with a finite state space $\mathcal{S}$, a finite action space $\mathcal{A}$, a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, a transition probability function $p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$, an initial distribution $p_0 : \mathcal{S} \to [0, 1]$, and a constant horizon length T. To simplify notations, we consider the undiscounted setting. But our results can be naturally applied to the discounted setting (Puterman 2014) as long as the horizon is fixed and finite. We define $[n] \doteq \{0, 1, \dots, n\}$ for any integer n.

There are K policies to be evaluated. In this paper, any index with parenthesis around it (e.g. $\pi^{(k)}$) is related to the *policy index*. We define abbreviations $\pi^{(k)}_{i:j} \doteq \left\{\pi^{(k)}_i, \pi^{(k)}_{i+1}, \dots, \pi^{(k)}_j\right\}$ and $\pi^{(k)} \doteq \pi^{(k)}_{0:T-1}$, where $\pi^{(k)}_t : \mathcal{A} \times \mathcal{S} \to [0, 1]$ defines the probability of selecting action A_t given the state S_t at time $t \in [T-1]$. An initial state S_0 is sampled from p_0 at time step 0. At each time step t, after the execution of an action, a finite reward $R_{t+1} = r(S_t, A_t)$ is obtained and a successor state S_{t+1} is sampled from $p(\cdot \mid S_t, A_t)$.

We define the return at time step t as $G_t \doteq \sum_{i=t+1}^{T} R_i$. The state- and action-value function is defined as $v_{\pi^{(k)},t}(s) \doteq \mathbb{E}_{\pi^{(k)}}[G_t \mid S_t = s]$ and $q_{\pi^{(k)},t}(s, a) \doteq \mathbb{E}_{\pi^{(k)}}[G_t \mid S_t = s, A_t = a]$. The performance of the policy π is defined as $J(\pi^{(k)}) \doteq \sum_s p_0(s) v_{\pi^{(k)},0}(s)$. We adopt the total rewards performance metric, introduced by Puterman (2014), as a measurement of the performance. In this work, we focus on the Monte Carlo methods, which have been widely adopted since their introduction by Kakutani (1945). We draw samples of $J(\pi^{(k)})$ by executing the policy $\pi^{(k)}$ online. The empirical average of the sampled returns converges to $J(\pi^{(k)})$ as the number of samples increases. Since this method estimates the performance of a policy $\pi^{(k)}$ by running itself, it is called on-policy learning (Sutton 1988).

Henceforth, we study off-policy learning, in which we need to estimate the total rewards $J(\pi^{(k)})$ of a policy $\pi^{(k)}$, called the target policy, by running a different policy μ, known as the behavior policy. Each trajectory $\{S_0, A_0, R_1, S_1, A_1, R_2, \dots, S_{T-1}, A_{T-1}, R_T\}$ is generated by a behavior policy μ with $S_0 \sim p_0, A_t \sim \mu_t(\cdot|S_t)$, $t \in [T-1]$. We use

$$\tau^{\mu_{t:T-1}}_{t:T-1} \doteq \{S_t, A_t, R_{t+1}, \dots, S_{T-1}, A_{T-1}, R_T\}$$

to denote a segment of a random trajectory generated by the behavior policy μ from the time step t to the time step $T-1$ inclusively. The key tool for off-policy learning is importance sampling (IS) (Rubinstein 1981), which is used to reweight rewards collected by μ to give an unbiased estimate of $J(\pi^{(k)})$. For each policy $\pi^{(k)}$, the importance sampling ratio at time step t is defined as $\rho^{\pi^{(k)},\mu}_t \doteq \frac{\pi^{(k)}_t(A_t|S_t)}{\mu_t(A_t|S_t)}$.

Then, the product of importance sampling ratios from time t to $t' \geq t$ is defined as $\rho_{t:t'}^{\pi^{(k)},\mu} \doteq \prod_{i=t}^{t'} \frac{\pi_i^{(k)}(A_i|S_i)}{\mu_i(A_i|S_i)}$. Among the various ways to use the importance sampling ratios in off-policy learning (Geweke 1988; Hesterberg 1995; Koller and Friedman 2009; Thomas 2015), we use the per-decision importance sampling estimator (PDIS, Precup, Sutton, and Singh (2000)) in this paper and leave the study of others for future work. For $\pi^{(k)}$, the PDIS Monte Carlo estimator is defined as $G_k^{\text{PDIS}}\big(\tau_{t:T-1}^{\mu_{t:T-1}}\big) \doteq \sum_{i=t}^{T-1} \rho_{t:i}^{\pi^{(k)},\mu} R_{i+1}$, which is unbiased for any behavior policy μ that covers target policy $\pi^{(k)}$ (Precup, Sutton, and Singh 2000). That is, when $\forall s$, $\forall a,\ \mu_t(a|s) = 0 \implies \pi_t^{(k)}(a|s) = 0$, we have $\forall t,\ \forall s$, $\mathbb{E}[G_k^{\text{PDIS}}\big(\tau_{t:T-1}^{\mu_{t:T-1}}\big) \mid S_t = s] = v_{\pi^{(k)},t}(s)$. We also leverage the recursive form of the PDIS estimator:

$$G_k^{\text{PDIS}}\big(\tau_{t:T-1}^{\mu_{t:T-1}}\big) \tag{1}$$
$$= \begin{cases} \rho_t^{\pi^{(k)},\mu}\left(R_{t+1} + G_k^{\text{PDIS}}\big(\tau_{t+1:T-1}^{\mu_{t+1:T-1}}\big)\right) & t \in [T-2], \\ \rho_t^{\pi^{(k)},\mu} R_{t+1} & t = T-1. \end{cases}$$

Because the PDIS estimator is unbiased, reducing its variance is sufficient for the improvement of its sample efficiency. We achieve this variance reduction goal for multiple policies by designing a tailored behavior policy.

Variance Reduction in Statistics

In this section, we propose the mathematical framework for variance reduction using importance sampling ratios. Let A be a discrete random variable with a finite set of possible values $\mathcal{A}$, and assume it follows a probability mass function $\pi^{(k)} : \mathcal{A} \to [0,1]$, called target policy. Additionally, let $q : \mathcal{A} \to \mathbb{R}$ be a function that maps elements of $\mathcal{A}$ to real numbers. Our objective is to estimate $\mathbb{E}_{A\sim\pi^{(k)}}[q(A)]$ for each $\pi^{(k)}$, where k is an index within a finite set $[K]$. Since in this paper, data can be generated from multiple distributions, we specify their source clearly. We reserve the superscript with brackets $[\cdot,\cdot]$ to denote the source and the index of samples. For example, $A^{[\pi^{(k)},i]}$ is the ith sample generated by running $\pi^{(k)}$. We use n_k to denote the total number of samples sampled by policy $\pi^{(k)}$. The plain Monte Carlo methods then samples $\left\{A^{[\pi^{(k)},1]}, \ldots, A^{[\pi^{(k)},n_k]}\right\}$ from each $\pi^{(k)}$ and use the empirical average $\frac{1}{n_k}\sum_{i=1}^{n_k} q(A^{[\pi^{(k)},i]})$ as the estimate for each $\mathbb{E}_{A\sim\pi^{(k)}}[q(A)]$.

The importance sampling is introduced as a variance reduction technique in statistics, where the main idea is to sample $\left\{q(A^{[\mu,i]})\right\}_{i=1}^{N}$ following a distribution μ and use $\frac{1}{N}\sum_{i=1}^{N} \rho^{\pi^{(k)},\mu}(A^{[\mu,i]}) q(A^{[\mu,i]})$ as the estimate, where $\rho^{\pi^{(k)},\mu}(A) \doteq \frac{\pi^{(k)}(A)}{\mu(A)}$ is the importance sampling ratio. In this statistics section, we propose the optimal behavior policy μ that evaluates all target policies $\pi^{(k)}$ simultaneously by sharing samples. We also define the similarity of policies and prove when target policies satisfy the similarity condition, samples needed to estimate all of them do not scale with the number of policies K. These ideas are later extended into the multi-step reinforcement learning (RL) setting in the following section.

Assuming that $\forall i$, μ covers $\pi^{(k)}$, i.e.,

$$\forall a, \mu(a) = 0 \implies \pi^{(k)}(a) = 0. \tag{2}$$

Then, the importance sampling ratio weighted empirical average is unbiased, i.e., $\forall k$, $\mathbb{E}_{A\sim\pi^{(k)}}[q(A)] = \mathbb{E}_{A\sim\mu}[\rho^{\pi^{(k)},\mu}(A)q(A)]$. If we carefully design the sampling distribution μ, the variance can be reduced. We formulate this problem of searching a variance-reducing sampling distribution for K policies as an optimization problem

$$\min_{\mu\in\Lambda_-} \quad \sum_{k\in[K]} \mathbb{V}_{A\sim\mu}(\rho^{\pi^{(k)},\mu}(A)q(A)),$$

where Λ_- is the classical search space (Rubinstein 1981; Zhang 2022; Liu, Chen, and Zhang 2024b; Qian et al. 2024) defined as

$$\Lambda_- \doteq \left\{\mu \in \Delta(\mathcal{A}) \mid \forall a, \forall k, \mu(a) = 0 \Rightarrow \pi^{(k)}(a) = 0\right\}.$$

Here, $\Delta(\mathcal{A})$ denotes the set of all probability distributions on the set $\mathcal{A}$. In other words, Λ_- includes all distributions that cover $\left\{\pi^{(k)}\right\}_{k=1}^{K}$. In this work, we enlarge Λ_- to Λ, which is defined as

$$\Lambda \doteq \left\{\mu \in \Delta(\mathcal{A}) \mid \forall a, \forall k, \mu(a) = 0 \Rightarrow \pi^{(k)}(a)q(a) = 0\right\}. \tag{3}$$

The space Λ weakens the assumption in (2). We prove that any distribution μ in Λ still gives unbiased estimation, though $\Lambda_- \subseteq \Lambda$.

Lemma 1. *$\forall \mu \in \Lambda, \forall k$,*

$$\mathbb{E}_{A\sim\mu}\left[\rho^{\pi^{(k)},\mu}(A)q(A)\right] = \mathbb{E}_{A\sim\pi^{(k)}}\left[q(A)\right].$$

Its proof is in the appendix. We now consider the variance minimization problem on Λ, i.e.,

$$\min_{\mu\in\Lambda} \quad \sum_{k\in[K]} \mathbb{V}_{A\sim\mu}(\rho^{\pi^{(k)},\mu}(A)q(A)). \tag{4}$$

The following lemma gives an optimal solution μ^* to the optimization problem (4).

Lemma 2. *Define $\mu^*(a) \propto \sqrt{\sum_{k\in[K]} \pi^{(k)}(a)^2 q(a)^2}$. Then μ^* is an optimal solution to* (4)*.*

Its proof is in the appendix. Here, $\mu(a) \propto f(a)$ with some non-negative $f(a)$ means

$$\mu(a) \doteq f(a)/\textstyle\sum_b f(b).$$

If $f(a) = 0$ for all a, the above "reweighted" distribution is not well defined. We then use the convention to interpret $\mu(a)$ as a uniform distribution, i.e., $\mu(a) = 1/|\mathcal{A}|$. *This convention in using $\propto$ is adopted in the rest of the paper for simplicity.*

When estimating $\mathbb{E}_{A\sim\pi^{(k)}}[q(A)]$, $\pi^{(k)}(a)q(a)$ shows how much an action contributes to the expectation and is heavily used (Owen 2013; Liu and Zhang 2024). Denote

$$w^{(k)}(a) \doteq \left(\pi^{(k)}(a)q(a)\right)^2, \tag{5}$$

$$\bar{w}(a) \doteq \sum_{j\in[K]} w^{(j)}(a)/K. \tag{6}$$

We use $\eta^{(k)}(a)$ to denote the similarity between $\pi^{(k)}$ and the average $\bar{w}(a)$,

$$\eta^{(k)}(a) \doteq w^{(k)}(a)/\bar{w}(a). \tag{7}$$

Naturally, $\eta^{(k)}(a) = 1$ when all policies are the same on a. Define $\underline{\eta} \doteq \min_{k,a} \eta^{(k)}(a)$ and $\overline{\eta} \doteq \max_{k,a} \eta^{(k)}(a)$, we have $\forall k, a$,

$$\underline{\eta} \leq \eta^{(k)}(a) \leq \overline{\eta}. \tag{8}$$

In the following theorem, we compare the variance of estimation methods. For off-policy evaluation, our designed μ^* generates n samples. For on-policy evaluation, when evaluating multiple policies, it is common for different policies to generate different numbers of samples. Thus, to achieve a fair and general enough comparison, each target policy $\pi^{(k)}$ generates n_k samples. There is no constraint on n_k, as long as $\sum_{k=1}^{K} n_k = n$. Using $A^{[\pi^{(k)},i]}$ to denote the ith sample generated following $\{\pi^{(k)}\}$, we define the empirical average for all $\pi^{(k)}$ as

$$E^{\text{on},\pi^{(k)}} \doteq \frac{\sum_{i=1}^{n_k} q(A^{[\pi^{(k)},i]})}{n_k}. \tag{9}$$

Similarly, using $A^{[\mu^*,i]}$ to denote the ith sample generated by μ^*, We define the empirical average for all $\pi^{(k)}$ as

$$E^{\text{off},\pi^{(k)}} \doteq \frac{\sum_{i=1}^{n} \rho^{\pi^{(k)},\mu^*}(A^{[\mu^*,i]}) q(A^{[\mu^*,i]})}{n}. \tag{10}$$

Then, we characterize sufficient conditions on policy similarity such that with the same total samples, off-policy evaluation with our tailored behavior policy μ^* achieves a lower variance than on-policy Monte Carlo on each $\pi^{(k)}$.

Lemma 3. *$\forall k \in [K]$,*

$$\mathbb{V}_{A\sim\mu^*}\left(E^{\textit{off},\pi^{(k)}}\right) \leq \mathbb{V}_{A\sim\pi^{(k)}}\left(E^{\textit{on},\pi^{(k)}}\right),$$

if the similarity $\eta(\cdot)$ has $\forall k$,

$$\begin{aligned}&\sqrt{\frac{\overline{\eta}}{\underline{\eta}}}\left(\sum_a \pi^{(k)}(a)q(a)\right)^2 - \left(\frac{n}{n_k} - 1\right)\Delta^{(k)} \\ \leq &\sum_a \pi^{(k)}(a)q(a)^2,\end{aligned} \tag{11}$$

where

$$\Delta^{(k)} \doteq \left[\sum_a \pi^{(k)}(a)q(a)^2 - \left(\sum_a \pi^{(k)}(a)q(a)\right)^2\right].$$

Its proof is in the appendix. In Lemma 3, we show under characterized conditions, using only the same total samples n generated by μ^*, the off-policy estimator already achieves a lower variance than on-policy estimator for each target policy $\pi^{(k)}$. Now, we present a stronger lemma by allowing each target policy to also generate n samples, resulting in a total of nK samples, which is K times larger than n. Using the empirical average for on-policy estimator as defined in (9), we now have, for all $\pi^{(k)}$,

$$E^{\text{on},\pi^{(k)}} = \sum_{i=1}^{n} q(A^{[\pi^{(k)},i]})/n. \tag{12}$$

Then, we simplify the variance of the on-policy estimator for $\pi^{(k)}$ as

$$\begin{aligned}&\mathbb{V}_{A\sim\pi^{(k)}}(E^{\text{on},\pi^{(k)}}) \\ =&\mathbb{V}_{A\sim\pi^{(k)}}\left(\frac{\sum_{i=1}^{n} q(A^{[\pi^{(k)},i]})}{n}\right) \qquad \text{(By (12))} \\ =&\frac{1}{n}\mathbb{V}_{A\sim\pi^{(k)}}\left(\sum_{i=1}^{n} q(A^{[\pi^{(k)},i]})\right) \\ =&\mathbb{V}_{A\sim\pi^{(k)}}(q(A)).\end{aligned}$$

In the last step, we leverage the independence of samples. Similarly, using the definition of empirical average for off-policy estimator as defined in (10), we have

$$\mathbb{V}_{A\sim\pi^{(k)}}(E^{\text{off},\pi^{(k)}}) = \mathbb{V}_{A\sim\mu^*}\left(\rho^{\pi^{(k)},\mu^*}(A)q(A)\right).$$

Then, we formalize the superiority for the "n-to-Kn" comparison in the following theorem.

Lemma 4. *$\forall k \in [K]$,*

$$\mathbb{V}_{A\sim\mu^*}\left(\rho^{\pi^{(k)},\mu^*}(A)q(A)\right) \leq \mathbb{V}_{A\sim\pi^{(k)}}(q(A)),$$

if the similarity $\eta(\cdot)$ has $\forall k$,

$$\sqrt{\frac{\overline{\eta}}{\underline{\eta}}}\left(\sum_{a\in\mathcal{A}} \pi^{(k)}(a)q(a)\right)^2 \leq \sum_{a\in\mathcal{A}} \pi^{(k)}(a)q(a)^2. \tag{13}$$

Its proof is in the appendix. The superiority of using our designed behavior policy μ^* comes from two sources. First, μ^* generates samples that all similar policies can efficiently share. Second, it is designed to generate low-variance and unbiased samples compared with the on-policy evaluation.

Variance Reduction in Reinforcement Learning

We extend the techniques discussed in the statistics section into multi-step reinforcement learning (RL). In this section, Theorem 1 is the RL version of Lemma 1 for unbiasedness. Theorem 2 is the RL version of Lemma 2 for behavior policy design. Theorem 3 and 4 are the RL version of Lemma 3 and 4, respectively, for variance reduction.

As discussed in the related work section, the major caveat in multi-policy evaluation problems is data sharing. Without efficient data sharing, the total number of samples required for evaluating all policies increases rapidly with the number of target policies. Previous works try to reuse collected data across multiple target policies. However, their method rely on either (1) **restrictive assumptions**, namely, deterministic policies and flexible environment starting at any desired state (Dann, Ghavamzadeh, and Marinov 2023), or (2) **impractical knowledge**, namely, complicated covariances (Lai, Zou, and Song 2020) and state visitation densities at very step (Kallus, Saito, and Uehara 2021). Thus, none of the existing methods (Dann, Ghavamzadeh, and Marinov 2023; Lai, Zou, and Song 2020; Kallus, Saito, and Uehara 2021; Agarwal et al. 2017) is implementable in the multi-step RL setting.

In this work, we tackle this notorious problem of efficient multi-policy evaluation in RL without any impracticability. We seek to reduce the variance $\sum_{k\in[K]} \mathbb{V}\left(G_k^{\text{PDIS}}\left(\tau_{0:T-1}^{\mu_{0:T-1}}\right)\right)$

by designing a proper behavior policy μ. Certainly, we need to ensure that the PDIS estimator with this behavior policy is unbiased.

In the off-policy evaluation problem, classic reinforcement learning (Sutton and Barto 2018) requires coverage assumption to ensure unbiased estimation. This means they only consider a set of policies that cover $\left\{\pi^{(k)}\right\}_{k=1}^{K}$, i.e.,

$$\Lambda_{-} \doteq \{\mu \mid \forall k, t, s, a, \mu_t(a|s) = 0 \implies \pi_t^{(k)}(a|s) = 0\}.$$

Similar to (3), we enlarge Λ_{-} to

$$\begin{aligned}\Lambda \doteq &\{\mu \mid \forall k, t, s, a, \mu_t(a|s) = 0 \\ &\implies \pi_t^{(k)}(a|s) q_{\pi^{(k)},t}(s,a) = 0\}.\end{aligned}$$

We prove every policy $\mu \in \Lambda$ still achieves unbiased estimation in the following theorem.

Theorem 1 (Unbiasedness)**.** *$\forall \mu \in \Lambda$, $\forall k$, $\forall t$, $\forall s$,*

$$\mathbb{E}\left[G_k^{PDIS}\left(\tau_{t:T-1}^{\mu_{t:T-1}}\right) \mid S_t = s\right] = v_{\pi^{(k)},t}(s).$$

Its proof is in the appendix. One immediate consequence of Theorem 1 is that $\forall \mu \in \Lambda, \forall k, \mathbb{E}\left[G_k^{\text{PDIS}}\left(\tau_{0:T-1}^{\mu_{0:T-1}}\right)\right] = J(\pi^{(k)})$. In this paper, we consider a set $\hat{\Lambda}$ such that $\Lambda_{-} \subseteq \hat{\Lambda} \subseteq \Lambda$. $\hat{\Lambda}$ inherits the unbiasedness property of Λ and is less restrictive than Λ_{-}, the classical search space of behavior policies. This $\hat{\Lambda}$ will be defined shortly. We now formulate our problem as

$$\min_{\mu \in \hat{\Lambda}} \quad \sum_{k \in [K]} \mathbb{V}\left(G_k^{\text{PDIS}}\left(\tau_{0:T-1}^{\mu_{0:T-1}}\right)\right). \tag{14}$$

By the law of total variance, for any $\mu \in \hat{\Lambda}$, we decompose the variance of the PDIS estimator as

$$\begin{aligned}&\sum_{k \in [K]} \mathbb{V}\left(G_k^{\text{PDIS}}\left(\tau_{0:T-1}^{\mu_{0:T-1}}\right)\right) \\ =&\sum_{k \in [K]} \mathbb{E}_{S_0}\left[\mathbb{V}\left(G_k^{\text{PDIS}}\left(\tau_{0:T-1}^{\mu_{0:T-1}}\right) \mid S_0\right)\right] \\ &+ \mathbb{V}_{S_0}\left(\mathbb{E}\left[G_k^{\text{PDIS}}\left(\tau_{0:T-1}^{\mu_{0:T-1}}\right) \mid S_0\right]\right) \\ =&\sum_{k \in [K]} \mathbb{E}_{S_0}\left[\mathbb{V}\left(G_k^{\text{PDIS}}\left(\tau_{0:T-1}^{\mu_{0:T-1}}\right) \mid S_0\right)\right] \\ &+ \mathbb{V}_{S_0}\left(v_{\pi^{(k)},0}(S_0)\right). \qquad \text{(by Theorem 1)}\end{aligned} \tag{15}$$

The second term in (15) is a constant given a target policy $\pi^{(k)}$ and is unrelated to the choice of μ. In the first term, the expectation is taken over S_0 that is determined by the initial probability distribution p_0. Consequently, to solve the problem (14), it is sufficient to solve for each s,

$$\min_{\mu \in \hat{\Lambda}} \quad \sum_{k \in [K]} \mathbb{V}\left(G_k^{\text{PDIS}}\left(\tau_{0:T-1}^{\mu_{0:T-1}}\right) \mid S_0 = s\right).$$

Denote the variance of the state value for the next state given the current state-action pair (s,a) as $\nu_{\pi^{(k)},t}(s,a)$. We have $\nu_{\pi^{(k)},t}(s,a) = 0$ for $t = T-1$ and otherwise

$$\nu_{\pi^{(k)},t}(s,a) \doteq \mathbb{V}_{S_{t+1}}\left(v_{\pi^{(k)},t+1}(S_{t+1}) \mid S_t = s, A_t = a\right). \tag{16}$$

To achieve variance reduction compared with on-policy evaluation, we aim to design $\hat{\mu}_t$ as an optimal solution to the following problem

$$\min_{\mu_t \in \hat{\Lambda}} \quad \sum_k \mathbb{V}\left(G_k^{\text{PDIS}}\left(\tau_{t:T-1}^{\left\{\mu_t, \pi_{t+1}^{(k)}:\pi_{T-1}^{(k)}\right\}}\right) \mid S_t = s\right), \tag{17}$$

The high-level intuition is that we aim to find the optimal behavior policy μ_t for the current step, assuming that in the future we perform the on-policy evaluation. To define optimality, we first specify the set of policies we are concerned about. To this end, we define that $\forall k$, $\hat{q}_{\pi^{(k)},t}(s,a) \doteq q_{\pi^{(k)},t}(s,a)^2$ for $t = T-1$ and otherwise

$$\begin{aligned}\hat{q}_{\pi^{(k)},t}(s,a) \doteq\ & q_{\pi^{(k)},t}(s,a)^2 + \nu_{\pi^{(k)},t}(s,a) \\ &+ \textstyle\sum_{s'} p(s'|s,a) \mathbb{V}\left(G_k^{\text{PDIS}}\left(\tau_{t+1:T-1}^{\pi_{t+1:T-1}^{(k)}}\right) \mid S_{t+1} = s'\right).\end{aligned} \tag{18}$$

Notably, $\hat{q}_{\pi^{(k)},t}(s,a)$ is always *non-negative* since all the summands are non-negative. Accordingly, we define $\hat{\Lambda} \doteq \{\mu \mid \forall k, t, s, a, \mu_t(a|s) = 0 \Rightarrow \pi_t^{(k)}(a|s)\hat{q}_{\pi^{(k)},t}(s,a) = 0\}$. From (18), we observe for any k, t, s, a, $\hat{q}_{\pi^{(k)},t}(s,a) \geq q_{\pi^{(k)},t}(s,a) \geq 0$. As a result, if $\mu_t \in \hat{\Lambda}$, we have $\mu_t(a|s) = 0 \Rightarrow \pi_t^{(k)}(a|s)\hat{q}_{\pi^{(k)},t}(s,a) = 0 \Rightarrow \pi_t^{(k)}(a|s) q_{\pi^{(k)},t}(s,a) = 0$. Thus, $\hat{\Lambda} \subseteq \Lambda$. To summarize, we have $\Lambda_{-} \subseteq \hat{\Lambda} \subseteq \Lambda$. $\hat{\Lambda}$ inherits the unbiased property of Λ (Theorem 1) and is larger than the classic space Λ_{-} considered in previous works (Precup, Sutton, and Singh 2000; Maei 2011; Sutton, Mahmood, and White 2016; Sutton and Barto 2018).

Now, we define the optimal behavior policy as

$$\hat{\mu}_t(a|s) \propto \sqrt{\textstyle\sum_{k=1}^{K} \pi_t^{(k)}(a|s)^2 \hat{q}_{\pi^{(k)},t}(s,a)}. \tag{19}$$

$\hat{q}$ defined in (18) is different from q, and is always non-negative. We confirm the optimality of $\hat{\mu}_t$ in the following theorem.

Theorem 2 (Behavior Policy Design)**.** *For any k, t and s, the behavior policy $\hat{\mu}_t(a|s)$ defined in* (19) *is an optimal solution to the following problem*

$$\min_{\mu_t \in \hat{\Lambda}} \quad \sum_k \mathbb{V}\left(G_k^{PDIS}\left(\tau_{t:T-1}^{\left\{\mu_t, \pi_{t+1}^{(k)}:\pi_{T-1}^{(k)}\right\}}\right) \mid S_t = s\right).$$

Its proof is in the appendix. Next, we formalize the similarity between target policies. Similar to (5), (6) in the statistics setting, $\forall k$, $\forall t$, $\forall s$, we denote

$$w_t^{(k)}(s,a) \doteq \pi_t^{(k)}(a|s)^2 \hat{q}_{\pi^{(k)},t}(s,a), \tag{20}$$

$$\bar{w}_t(s,a) \doteq \left(\textstyle\sum_{j \in [K]} w_t^{(j)}(s,a)\right)/K. \tag{21}$$

Then, adopting the notation from (7) and (8), we denote the similarity between $\pi_t^{(k)}$ and the average $\bar{w}_t$ as

$$\eta_t^{(k)}(s,a) \doteq w_t^{(k)}(s,a)/\bar{w}_t(s,a). \tag{22}$$

When policies are the same, $\forall k, t, s$, $\eta_t^{(k)}(s,a) = 1$. Define $\underline{\eta}_t \doteq \min_{k,s,a} \eta_t^{(k)}(s,a)$ and $\overline{\eta} \doteq \max_{k,a} \eta_t^{(k)}(s,a)$, we have $\forall t, k, s, a$,

$$\underline{\eta}_t \leq \eta_t^{(k)}(s,a) \leq \overline{\eta}_t. \tag{23}$$

Next, to extend the variance reduction property from statistics (Lemma 3) into reinforcement learning, we also allow

each target policy to generate n_k samples. With a similar notation, we have the empirical average for all $\pi^{(k)}$ as

$$E_{t:T-1}^{\text{on},\pi^{(k)}} \doteq \frac{\sum_{i=1}^{n_k} G_k^{\text{PDIS}}\left(\tau_{t:T-1}^{[\pi_{t:T-1}^{(k)},i]}\right)}{n_k}, \tag{24}$$

where $\tau^{[\pi^{(k)},i]}$ is the ith trajectory obtained by running $\pi^{(k)}$. To achieve a fair comparison, when doing off-policy estimation by following $\hat{\mu}$, we generate $n = \sum_{k=1}^{K} n_k$ samples. Likewise, define

$$E_{t:T-1}^{\text{off},\pi^{(k)}} \doteq \frac{\sum_{i=1}^{n} G_k^{\text{PDIS}}\left(\tau_{t:T-1}^{[\hat{\mu}_{t:T-1},i]}\right)}{n}. \tag{25}$$

We have the following theorem.

Theorem 3 (Variance Reduction with Same Sample Sizes)**.** *$\forall k$, $\forall t$, $\forall s$,*

$$\mathbb{V}\left(E_{t:T-1}^{off,\pi^{(k)}} \mid S_t = s\right) \leq \mathbb{V}\left(E_{t:T-1}^{on,\pi^{(k)}} \mid S_t = s\right).$$

if the similarity η has $\forall k, \forall t, \forall s$,

$$\sqrt{\frac{\overline{\eta_t}}{\underline{\eta_t}}}\left(\sum_a \pi_t^{(k)}(a|s)\sqrt{\hat{q}_{\pi^{(k)},t}(a|s)}\right)^2 - \left(1 - \tfrac{n_k}{n}\right)\Delta_t^{(k)}(s)$$

$$\leq \sum_a \pi_t^{(k)}(a|s)\hat{q}_{\pi^{(k)},t}(s,a), \tag{26}$$

where

$$\Delta_t^{(k)}(s) \doteq \mathbb{E}_{A_t\sim\hat{\mu}_t}\left[\rho^{\pi^{(k)},\hat{\mu}^2}\nu_{\pi^{(k)},t}(S_t,A_t) \mid S_t = s\right]$$
$$+\mathbb{V}_{A_t\sim\hat{\mu}_t}\left(\rho^{\pi^{(k)},\hat{\mu}}q_{\pi^{(k)},t}(S_t,A_t) \mid S_t = s\right).$$

Its proof is in the appendix. We then compare the datasets when the behavior policy $\hat{\mu}$ and each target policy $\pi^{(k)}$ both generate n samples, resulting in a "n-to-nK" comparison, similar to Lemma 4.

Theorem 4 (Variance Reduction)**.** *$\forall k$, $\forall t$, $\forall s$,*

$$\mathbb{V}\left(G_k^{PDIS}\left(\tau_{t:T-1}^{\hat{\mu}_{t:T-1}}\right) \mid S_t = s\right)$$
$$\leq \mathbb{V}\left(G_k^{PDIS}\left(\tau_{t:T-1}^{\pi_{t:T-1}^{(k)}}\right) \mid S_t = s\right),$$

if the similarity η has $\forall k, \forall t, \forall s$,

$$\sqrt{\frac{\overline{\eta_t}}{\underline{\eta_t}}}\left(\sum_a \pi_t^{(k)}(a|s)\sqrt{\hat{q}_{\pi^{(k)},t}(s,a)}\right)^2 \tag{27}$$
$$\leq \sum_a \pi_t^{(k)}(a|s)\hat{q}_{\pi^{(k)},t}(s,a).$$

Its proof is in the appendix. This theorem implies that in the multi-step RL setting, running our tailored behavior policy $\hat{\mu}$ also ensures that the number of required samples does not scale with the number of target policies under similarity conditions. The reduced variance of our method depends on the similarity between target policies, which can be easily checked through learning $\hat{q}$ with offline data. Thus, if RL practitioners are not confident in the similarity between target policies, they can verify it before actual deployment without consuming any online data.

Algorithm 1: Multi-Policy Evaluation (MPE) algorithm

1: **Input:** K target policies $\pi^{(k)}$, an offline dataset $\mathcal{D} = \{(t_i, s_i, a_i, r_i, s_i')\}_{i=1}^m$
2: **Output:** a behavior policy $\hat{\mu}$
3: Approximate $q_{\pi^{(k)},t}$ from $\mathcal{D}$ using any offline RL method (e.g. Fitted Q-Evaluation)
4: Compute $\hat{r}_{\pi^{(k)},i}$ for data pairs in $\mathcal{D}$ by (48)
5: Construct $\mathcal{D}^{(k)} \doteq \left\{(t_i, s_i, a_i, \hat{r}_{\pi^{(k)},i}, s_i')\right\}_{i=1}^m$
6: Approximate $\hat{q}_{\pi^{(k)},t}$ from $\mathcal{D}^{(k)}$ by (49) using any offline method (e.g. Fitted Q-Evaluation)
7: **Return:** $\hat{\mu}_t(a|s) \propto \sqrt{\sum_{k=1}^K \pi_t^{(k)}(a|s)^2 \hat{q}_{\pi^{(k)},t}(s,a)}$

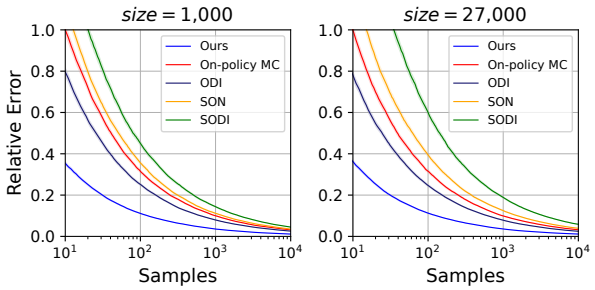

Figure 1: Results on Gridworld. Each curve is averaged over 900 runs (30 groups of policies, each having 30 independent runs). Shaded regions denote standard errors and are invisible for some curves because they are too small.

Empirical Results

We evaluate $K = 10$ target policies simultaneously by executing the tailored behavior policy $\hat{\mu}$ with n total samples. We name our method multiple policy evaluation (MPE) estimator. We present our empirical comparisons with the following baselines: **(1)** The canonical on-policy Monte Carlo estimator with n_k samples for each target policy $\pi^{(k)}$, summing to a total of $n = \sum_{k=1}^K n_k$ samples. **(2)** The offline data informed estimator (ODI, Liu and Zhang (2024)) that runs each behavior policy (designed for each target policy $\pi^{(k)}$) for n_k samples, summing to a total of $n = \sum_{k=1}^K n_k$ samples. **(3)** The shared-sample on-policy Monte Carlo estimator (SON), where we evaluate each target policy with shared data collected by canonical on-policy Monte Carlo estimators of all K policies, resulting in $n = \sum_{k=1}^K n_k$ samples used to evaluate every target policy. **(4)** The shared-sample ODI estimator (SODI), where we evaluate each target policy with shared data collected by ODI estimators of all K policies. Since each single behavior policy from the ODI estimator collects n_k samples, each target policy in SODI leverages $n = \sum_{k=1}^K n_k$ samples.

As a demonstration of concept, we set $K = 10$ and $n_k = \frac{n}{K}$ for each of the 10 target policies. Target policies are drawn from the training process of proximal policy optimization (PPO) algorithm (Schulman et al. 2017). We learn

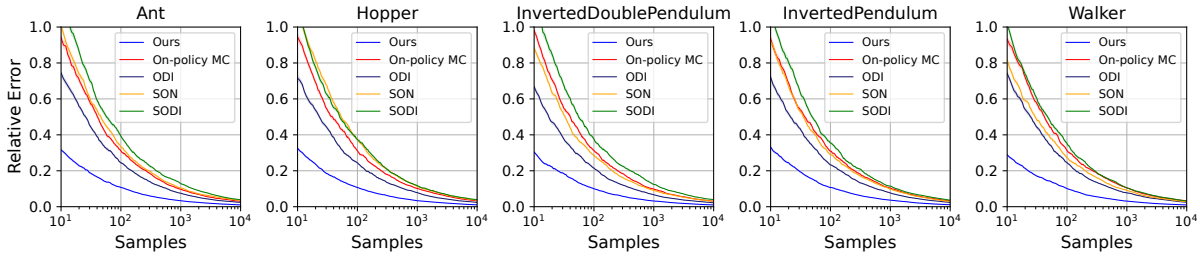

Figure 2: Results on MuJoCo. Each curve is averaged over 900 runs (30 groups of target policies, each having 30 independent runs). Shaded regions denote standard errors and are invisible for some curves because they are too small.

our behavior policy $\hat{\mu}$ using Algorithm 1. Hyperparameters are the same across all MuJoCo and Gridworld experiments. Experimental details are in the appendix.

Gridworld: We use Gridworld with $m^3 = 1,000$ and $m^3 = 27,000$ states, where each Gridworld has a width m and height m with a time horizon $T = m$.

Env Size	Ours	On-policy MC	ODI	SON	SODI
1,000	**0.125**	1.000	0.637	1.289	2.073
27,000	**0.129**	1.000	0.601	1.561	3.532

Table 1: Relative variance of estimators on Gridworld. The relative variance is defined as the variance of each estimator divided by the variance of the on-policy Monte Carlo estimator. Numbers are averaged over 900 independent runs (30 groups of target policies, each having 30 independent runs).

Env Size	Ours	On-policy MC	ODI	SON	SODI
1,000	**126**	1000	632	1264	2046
27,000	**131**	1000	629	1568	3501

Table 2: Episodes needed to achieve the same of estimation accuracy that on-policy Monte Carlo achieves with 1000 episodes. Numbers are averaged over 900 independent runs (30 groups of target policies, each having 30 independent runs) and their standard errors are shown in Figure 1.

Figure 1 shows our method outperforms all baselines by a large margin. The *relative error* is defined as the estimation error divided by the estimation error of the on-policy MC at the beginning of x-axis. The *samples* on the x-axis represents the total online episodes for multi-policy evaluation. The blue line in the graph is below other lines, indicating that our method requires fewer samples to achieve the same accuracy. To quantify the variance reduction, Table 1 shows our method reduces variance to about 12.5% compared with the on-policy Monte Carlo estimator. Table 2 shows that to achieve the same estimation error that the on-policy Monte Carlo estimator achieves with 1000 samples, our estimator only needs about 130 samples saving about 87% of online interactions, achieving state-of-the-art performance.

MuJoCo: Next, we conduct experiments in MuJoCo robot simulation tasks (Todorov, Erez, and Tassa 2012). MuJoCo is a physics engine containing various stochastic environments, where the goal is to control a robot to achieve different behaviors such as walking, jumping, and balancing. Figure 2 shows our method is consistently better than all baselines. The tables in the appendix show similar patterns as in the Gridworld experiment. In particular, our estimator reduces the variance to about 10% compared with the on-policy Monte Carlo estimator and saves about 90% of online interactions.

An interesting observation to demonstrate the discrepancy among target policies is that SODI and SON generally perform worse than On-policy MC and ODI. This result suggests that when target policies lack sufficient similarity, reusing data without a carefully designed joint behavior policy leads to high-variance estimation. Additionally, while ODI outperforms On-policy MC, SODI performs worse than SON. This may be because each behavior policy in SODI is specially tailored for its own target policy, making it vulnerable to target policy change. *These observations confirm the notorious difficulty of data sharing across multiple policies, highlighting the need for a tailored and shared behavior policy to efficiently facilitate data sharing.*

Conclusion

In this paper, we introduce a novel approach for multi-policy evaluation by designing a tailored behavior policy that efficiently and unbiasedly evaluates multiple target policies.

Theoretically, our method eliminates the need for restrictive assumptions or infeasible knowledge required by previous methods. Our method achieves lower variance compared to on-policy evaluation for each target policy under similarity conditions (Theorem 3, Theorem 4) and ensures the number of required samples does not scale with the number of target policies when similarity conditions hold.

Empirically, our method outperforms previously best-performing methods, achieving state-of-the-art performance across various environments. One promising future direction is to extend our variance reduction method to policy improvement and achieve efficient policy learning.

Acknowledgements

This work is supported in part by the US National Science Foundation (NSF) under grants III-2128019 and SLES-2331904. Claire Chen is supported in part by an Ingrassia Family Echols Scholars Research Grant.

References

Agarwal, A.; Basu, S.; Schnabel, T.; and Joachims, T. 2017. Effective evaluation using logged bandit feedback from multiple loggers. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Chen, C.; Liu, S.; and Zhang, S. 2024. Efficient Policy Evaluation with Safety Constraint for Reinforcement Learning. *arXiv preprint arXiv:2410.05655*.

Chervonyi, Y.; Dutta, P.; Trochim, P.; Voicu, O.; Paduraru, C.; Qian, C.; Karagozler, E.; Davis, J. Q.; Chippendale, R.; Bajaj, G.; et al. 2022. Semi-analytical industrial cooling system model for reinforcement learning. *arXiv preprint arXiv:2207.13131*.

Chua, K.; Calandra, R.; McAllister, R.; and Levine, S. 2018. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. In *Advances in Neural Information Processing Systems*.

Dann, C.; Ghavamzadeh, M.; and Marinov, T. V. 2023. Multiple-policy High-confidence Policy Evaluation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*.

Deisenroth, M. P.; and Rasmussen, C. E. 2011. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *Proceedings of the International Conference on Machine Learning*.

Geweke, J. 1988. Antithetic acceleration of Monte Carlo integration in Bayesian inference. *Journal of Econometrics*.

Hesterberg, T. 1995. Weighted Average Importance Sampling and Defensive Mixture Distributions. *Technometrics*.

Huang, S.; Dossa, R. F. J.; Ye, C.; Braga, J.; Chakraborty, D.; Mehta, K.; and Araújo, J. G. 2022. CleanRL: High-quality Single-file Implementations of Deep Reinforcement Learning Algorithms. *Journal of Machine Learning Research*.

Kakutani, S. 1945. Markoff process and the Dirichlet problem. *Proceedings of the Japan Academy*.

Kallus, N.; Saito, Y.; and Uehara, M. 2021. Optimal off-policy evaluation from multiple logging policies. In *International Conference on Machine Learning*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations*.

Koller, D.; and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Lai, J.; Zou, L.; and Song, J. 2020. Optimal Mixture Weights for Off-Policy Evaluation with Multiple Behavior Policies. *arXiv preprint arXiv:2011.14359*.

Le, H. M.; Voloshin, C.; and Yue, Y. 2019. Batch Policy Learning under Constraints. In *Proceedings of the International Conference on Machine Learning*.

Levine, S. 2018. Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review. *arXiv preprint arXiv:1805.00909*.

Li, L. 2019. A perspective on off-policy evaluation in reinforcement learning. *Frontiers of Computer Science*.

Liu, S.; Chen, C.; and Zhang, S. 2024a. Doubly Optimal Policy Evaluation for Reinforcement Learning. *arXiv preprint arXiv:2410.02226*.

Liu, S.; Chen, S.; and Zhang, S. 2024b. The ODE Method for Stochastic Approximation and Reinforcement Learning with Markovian Noise. *arXiv preprint arXiv:2401.07844*.

Liu, S.; and Zhang, S. 2024. Efficient Policy Evaluation with Offline Data Informed Behavior Policy Design. In *Proceedings of the International Conference on Machine Learning*.

Maei, H. R. 2011. *Gradient temporal-difference learning algorithms*. Ph.D. thesis, University of Alberta.

Nachum, O.; Chow, Y.; Dai, B.; and Li, L. 2019. DualDICE: Behavior-Agnostic Estimation of Discounted Stationary Distribution Corrections. In *Advances in Neural Information Processing Systems*.

O'Donoghue, B.; Osband, I.; Munos, R.; and Mnih, V. 2018. The Uncertainty Bellman Equation and Exploration. In *Proceedings of the International Conference on Machine Learning*.

Owen, A. B. 2013. *Monte Carlo theory, methods and examples*. Stanford.

Prechelt, L. 2002. Early stopping-but when? In *Neural Networks: Tricks of the trade*, 55–69. Springer.

Precup, D.; Sutton, R. S.; and Singh, S. P. 2000. Eligibility Traces for Off-Policy Policy Evaluation. In *Proceedings of the International Conference on Machine Learning*.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Qian, X.; Xie, Z.; Liu, X.; and Zhang, S. 2024. Almost Sure Convergence Rates and Concentration of Stochastic Approximation and Reinforcement Learning with Markovian Noise. *arXiv preprint arXiv:2411.13711*.

Rubinstein, R. Y. 1981. *Simulation and the Monte Carlo Method*. Wiley.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*.

Sherstan, C.; Bennett, B.; Young, K.; Ashley, D. R.; White, A.; White, M.; and Sutton, R. S. 2018. Directly estimating the variance of the λ-return using temporal-difference methods. *arXiv preprint arXiv:1801.08287*.

Sutton, R. S. 1988. Learning to Predict by the Methods of Temporal Differences. *Machine Learning*.

Sutton, R. S. 1990. Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming. In *Proceedings of the International Conference on Machine Learning*.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction (2nd Edition)*. MIT press.

Sutton, R. S.; Mahmood, A. R.; and White, M. 2016. An Emphatic Approach to the Problem of Off-policy Temporal-Difference Learning. *Journal of Machine Learning Research*.

Sutton, R. S.; Szepesvári, C.; Geramifard, A.; and Bowling, M. H. 2008. Dyna-Style Planning with Linear Function Approximation and Prioritized Sweeping. In *Proceedings of the Conference in Uncertainty in Artificial Intelligence*.

Tamar, A.; Castro, D. D.; and Mannor, S. 2016. Learning the Variance of the Reward-To-Go. *Journal of Machine Learning Research*.

Thomas, P. S. 2015. *Safe reinforcement learning*. Ph.D. thesis, University of Massachusetts Amherst.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. MuJoCo: A physics engine for model-based control. In *Proceedings of the International Conference on Intelligent Robots and Systems*.

Zhang, S. 2022. *Breaking the deadly triad in reinforcement learning*. Ph.D. thesis, University of Oxford.

Zhang, S. 2023. A New Challenge in Policy Evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.