Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

GREEN SIMULATION BASED POLICY OPTIMIZATION WITH PARTIAL HISTORICAL TRAJECTORY REUSE

Hua Zheng
Wei Xie

Department of Mechanical and Industrial Engineering
Northeastern University
360 Huntington Ave
Boston, MA 02115, USA

ABSTRACT

Built on our previous study on green simulation assisted policy gradient (GS-PG), in this paper, we consider infinite-horizon Markov decision processes and create a new importance sampling based policy gradient optimization approach to support dynamic decision making. The existing GS-PG method was designed to learn from complete episodes or process trajectories, which limits its applicability to low-data situations and flexible online process control. To overcome this limitation, the proposed approach utilizes a mixture likelihood ratio (MLR) based policy gradient and intelligently select and reuse the most related historical transition samples to improve the policy gradient estimation and accelerate the learning of optimal policy. Our empirical study demonstrates that it can improve optimization convergence and enhance the performance of state-of-the-art policy optimization approaches such as actor-critic method and proximal policy optimizations.

1 INTRODUCTION

In recent years, various policy optimization approaches are developed to solve reinforcement learning (RL) and process control problems. They often consider parametric policies and search for optimal solution through stochastic policy gradient approach. Historical observations or samples can be reused to improve optimization convergence, especially in low-data situations. According to the base unit of observations to reuse, policy gradient algorithms can be classified into episode-based and step-based approaches (Metelli et al. 2020). Episode-based approaches perform importance sampling (IS) on full trajectories accounting for the distributional difference induced by target and behavior policies. The importance weight is built on the cumulative product of likelihood ratios (LR) of state-action transitions occurring within each trajectory. This can lead to extremely high variance, especially for those problems with long planning horizon (Andradóttir et al. 1995; Schlegel et al. 2019). Thus, the trajectory-based reuse strategy is not applicable to many applications, such as personalized bio-drug manufacturing, with: (1) small amount of data; (2) complex state-action transition model and long planning horizon; and (3) requiring real-time flexible process control.

Instead of reusing entire historical trajectory samples, we need to create an intelligent and flexible strategy that can select and reuse the most related parts of trajectory which can change for different random scenarios. For example, during cell therapy manufacturing, the metabolic state can evolve with time during the cell life cycle and also therapeutic cells can have metabolic shift under heterogeneous culture conditions. To improve prediction and guide real-time process control, we can reuse historical step-based observations that have cell metabolic state and bioprocessing dynamics similar to the target distribution. *Therefore, step-based policy gradient algorithms can take state-action transitions as the base unit of observations to reuse (Metelli et al. 2020) and overcome the limitations of the trajectory-based reuse strategy.*

978-1-6654-7661-4/22/$31.00 ©2022 IEEE

In this paper, we focus on step-based policy gradient for infinite-horizon Markov Decision Processes (MDPs). That means we update policy parameters per step (or mini-batch of steps), which requires a single state-action transition LR for each historical sample to account for the difference in the state-occupancy measure or the stationary state distribution induced by different target and behavior policies. Various step-based policy gradient algorithms have been proposed during recent years. The studies in distribution correction (DICE) (Nachum et al. 2019; Yang et al. 2020) provide ways to estimate these state occupancy ratios in RL. Proximal policy optimization (PPO), as one of the most popular step-based policy gradient approaches (Schulman et al. 2017), uses a clipped surrogate objective to control incentives for the new candidate policy to get far from the old policy and thus avoid too much policy parameter updates at one step. Actor-Critic algorithm, as a classic and theoretically solid policy optimization framework, jointly optimizes the value function (critic) and the policy (actor); see for example Bhatnagar et al. (2009).

In our previous study (Zheng et al. 2020), we created a new experience replay approach called green simulation assisted reinforcement learning (GS-RL) for episode-based policy optimization. This approach can automatically select the most relevant historical trajectory episodes based on a comparison of gradient variance between historical episodes and current episodes, i.e., episodes collected by following the candidate policy. Then the selected historical trajectories are used to improve policy gradient estimation through multiple importance sampling techniques. Our theoretical and empirical studies have showed that the VRER based policy gradient estimator can improve sample efficiency and lead to a superior performance in convergence.

Build on Zheng et al. (2020), in this paper, we extend the GS-RL from a episode/trajectory-based algorithm to a step-based algorithm and create a variance reduction based epxerience replay (VRER) approach for infinite horizon MDPs. This approach can select and reuse the most relevant historical observations on state-action transitions to improve policy gradient estimation. The proposed VRER approach is general and it can be integrated into various stochastic policy gradient approaches to improve optimization convergence. In the paper, we provide an algorithm to utilize it to enhance two state-of-the-art policy optimization algorithms, including Actor-Critic algorithm and PPO.

Therefore, the key contributions of this study include: (1) create the VRER based policy optimization that can selectively reuse the most relevant historical transitions or partial trajectory observations; (2) develop multiple importance sampling (MIS) based off-policy actor-critic method; and (3) analytically and empirically show that the proposed VRER based policy gradient approach can improve the policy gradient estimation, speed up the optimal convergence for RL problems, and support real-time process control.

The paper is organized as follows. We start in Section 2 by introducing the notation and basics about RL policy optimization, importance sampling (IS), and multiple importance sampling. Then we propose the mixture likelihood ratio (MLR) based policy gradient estimation in Section 3. We create a selection rule that allows us to automatically select the most relevant historical transitions to improve the policy gradient estimation accuracy in Section 4. We further show how this selection strategy can be customized into the policy gradient optimization and introduce the variance reduction experience replay with the resulting algorithms. We conclude this paper with the comprehensive empirical study on the proposed framework in Section 5. The implementation of our algorithm can be found at https://github.com/zhenghuazx/vrer_policy_optimization.

2 PROBLEM DESCRIPTION

We study reinforcement learning and process control problems in which an agent acts on a complex stochastic system by sequentially choosing actions over a sequence of time steps in order to maximise a cumulative reward. We formulate the problem of interest as an infinite-horizon Markov decision process (MDP) specified by $(\mathscr{S},\mathscr{A},r,\mathbb{P},\boldsymbol{s}_1)$, where $\mathscr{S}$ is the state space, $\mathscr{A}$ is the action space, and a reward function is denoted by $r:\mathscr{S}\times\mathscr{A}\to\mathbb{R}$. An initial state distribution is specified by the density $p_1(\boldsymbol{s}_1)$. The stationary state transition model $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t)$ satisfies the Markov property $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t,\ldots,\boldsymbol{s}_1,\boldsymbol{a}_1)=p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t)$. The system starts at an initial state $\boldsymbol{s}_1$ at time $t=1$ drawn from $p_1(\boldsymbol{s}_1)$. At time t, the agent observes the state $\boldsymbol{s}_t\in\mathscr{S}$, takes an action $\boldsymbol{a}_t\in\mathscr{A}$ by following a parametric policy distribution, denoted by $\pi(\boldsymbol{s}_t|\boldsymbol{a}_t;\boldsymbol{\theta})$

specified with parameters $\boldsymbol{\theta}$, and receives a reward $r_t(\boldsymbol{s}_t,\boldsymbol{a}_t) \in \mathbb{R}$. The future return is the total discounted reward, denoted by $r_t^{\gamma} = \sum_{t'=t}^{\infty} \gamma^{t'-t} r(\boldsymbol{s}_{t'},\boldsymbol{a}_{t'})$, where $\gamma \in (0,1)$ denotes the discount factor.

Suppose that the policy $\pi_{\boldsymbol{\theta}}$ is continuous and differentiable with respect to its parameters $\boldsymbol{\theta}$. For each candidate policy specified by $\boldsymbol{\theta}$, the state value function $V^{\pi}(\boldsymbol{s})$ and the Q-function $Q^{\pi}(\boldsymbol{s},\boldsymbol{a})$ are defined to be the expected total discounted reward-to-go, i.e.,

$$\begin{aligned} V^{\pi}(\boldsymbol{s}) &= \mathbb{E}[r_1^{\gamma}|\boldsymbol{s}_1=\boldsymbol{s};\pi] = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(\boldsymbol{s}_t,\boldsymbol{a}_t) \middle| \boldsymbol{s}_1=\boldsymbol{s};\pi\right], \\ Q^{\pi}(\boldsymbol{s},\boldsymbol{a}) &= \mathbb{E}[r_1^{\gamma}|\boldsymbol{s}_1=\boldsymbol{s},\boldsymbol{a}_1=\boldsymbol{a};\pi] = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(\boldsymbol{s}_t,\boldsymbol{a}_t) \middle| \boldsymbol{s}_1=\boldsymbol{s},\boldsymbol{a}_1=\boldsymbol{a};\pi\right]. \end{aligned}$$

The agent's goal is to find an optimal policy that maximises the cumulative discounted reward, denoted by $J(\pi) = \mathbb{E}[\sum_{t=1}^{\infty} \gamma^{t-1} r(\boldsymbol{s}_t,\boldsymbol{a}_t)|\pi]$. For any feasible policy $\boldsymbol{\theta}$, we assume that the Markov chains, i.e., $\{\boldsymbol{s}_t\}_{t\geq\infty}$ and $\{\boldsymbol{s}_t,\boldsymbol{a}_t\}_{t\geq\infty}$, are irreducible and aperiodic. We denote the improper discounted state distribution as

$$d^{\pi}(\boldsymbol{s}) = \int_{\mathscr{S}} \sum_{t=1}^{\infty} \gamma^{t-1} p(\boldsymbol{s}_1) p(\boldsymbol{s}_t=\boldsymbol{s}|\boldsymbol{s}_1;\pi) d\boldsymbol{s}_1.$$

Then we can write the policy optimization problem with the performance objective as an expectation,

$$\max_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(\boldsymbol{s}_t,\boldsymbol{a}_t) \middle| \pi\right] = \int d^{\pi}(\boldsymbol{s}) \int \pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s}) r(\boldsymbol{s},\boldsymbol{a}) d\boldsymbol{s} d\boldsymbol{a} = \mathbb{E}_{\boldsymbol{s}\sim d^{\pi}(\boldsymbol{s}),\boldsymbol{a}\sim\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s})}[r(\boldsymbol{s},\boldsymbol{a})], \quad (1)$$

where $\mathbb{E}_{\boldsymbol{s}\sim d^{\pi}(\boldsymbol{s}),\boldsymbol{a}\sim\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s})}[\cdot]$ denotes the expected value with respect to the discounted state distribution $d^{\pi}(\boldsymbol{s})$ and the policy distribution $\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s})$. We denote the stationary probability function of state-action pair by

$$\rho_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a}) = \pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s}) d^{\pi}(\boldsymbol{s}).$$

To simplify notation, we superscript the value function $V^{\pi}(\boldsymbol{s})$ and the advantage function, denoted by $A^{\pi}(\boldsymbol{s},\boldsymbol{a})$ that will be defined in Section 2.1, by π rather than $\pi_{\boldsymbol{\theta}}$.

2.1 Stochastic Policy Gradient Estimation

Policy gradient optimization is perhaps the most popular class of RL algorithms designed to solve the optimization problem (1). At each k-th iteration, we can iteratively update the policy parameters,

$$\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \eta_k \widehat{\nabla J}(\boldsymbol{\theta}_k),$$

where η_k is learning rate or step size and $\widehat{\nabla J}(\boldsymbol{\theta}_k)$ is an estimator of policy gradient $\nabla J(\boldsymbol{\theta}_k)$. For notational convenience, ∇ denotes the gradient with respect to policy parameters $\boldsymbol{\theta}$ unless specified otherwise. Under regularity conditions, *Policy Gradient Theorem* (Sutton et al. 1999) reformulates the policy gradient as

$$\nabla J(\boldsymbol{\theta}) = \int d^{\pi}(\boldsymbol{s}) \int \nabla \pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s}) Q^{\pi}(\boldsymbol{s},\boldsymbol{a}) d\boldsymbol{s} d\boldsymbol{a} = \mathbb{E}_{\boldsymbol{s}\sim d^{\pi}(\boldsymbol{s}),\boldsymbol{a}\sim\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s})}[\nabla \log \pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s}) Q^{\pi}(\boldsymbol{s},\boldsymbol{a})]. \quad (2)$$

This theorem has an important practical value because it reduces the computation of the performance gradient to a simple expectation (Silver et al. 2014). By applying sample average approximation (SAA) on the expectation in (2), we have the *naive policy gradient (PG) estimator*

$$\widehat{\nabla J}_k^{PG} \equiv \widehat{\nabla J}^{PG}(\boldsymbol{\theta}_k) = \frac{1}{n} \sum_{j=1}^{n} g_k\left(\boldsymbol{s}^{(k,j)}, \boldsymbol{a}^{(k,j)}\right),$$

where n is the number of replications. The *scenario-based policy gradient estimate* at $\boldsymbol{\theta}_k$ is represented as,

$$g_k(\boldsymbol{s},\boldsymbol{a}) \equiv g(\boldsymbol{s},\boldsymbol{a}|\boldsymbol{\theta}_k) = Q^{\pi}(\boldsymbol{s},\boldsymbol{a}) \nabla \log \pi_{\boldsymbol{\theta}_k}(\boldsymbol{a}|\boldsymbol{s}).$$

A widely used variation of policy gradient (2) is to subtract a baseline value from the return $Q^{\pi}(\boldsymbol{s},\boldsymbol{a})$ to reduce the variance of gradient estimation while keeping the unbiased property. A common baseline is to subtract a value function $V^{\pi}(\boldsymbol{s})$; see Bhatnagar et al. (2009), Lemma 2. Then we have a new unbiased policy gradient estimator with lower variance,

$$\nabla J(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}\sim d^{\pi}(\boldsymbol{s}),\boldsymbol{a}\sim\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s})}\left[\nabla\log\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s})\left(Q^{\pi}(\boldsymbol{s},\boldsymbol{a}) - V^{\pi}(\boldsymbol{s})\right)\right]. \tag{3}$$

The difference $A^{\pi}(\boldsymbol{s},\boldsymbol{a}) \equiv Q^{\pi}(\boldsymbol{s},\boldsymbol{a}) - V^{\pi}(\boldsymbol{s})$ is called *advantage*. It intuitively measures the extra reward that an agent can obtain by taking that a particular action $\boldsymbol{a}$. This leads to the "vanilla" policy gradient estimator,

$$g_k(\boldsymbol{s},\boldsymbol{a}) \equiv g(\boldsymbol{s},\boldsymbol{a}|\boldsymbol{\theta}_k) = A^{\pi_{\boldsymbol{\theta}_k}}(\boldsymbol{s},\boldsymbol{a})\nabla\log\pi_{\boldsymbol{\theta}_k}(\boldsymbol{a}|\boldsymbol{s}).$$

According to the Bellman equation, i.e., $Q^{\pi}(\boldsymbol{s},\boldsymbol{a}) = r(\boldsymbol{s},\boldsymbol{a}) + \gamma\mathbb{E}_{\boldsymbol{s}'\sim p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})}[V^{\pi}(\boldsymbol{s}')]$, the advantage function can be expressed as

$$A^{\pi}(\boldsymbol{s},\boldsymbol{a}) = r(\boldsymbol{s},\boldsymbol{a}) + \gamma\mathbb{E}_{\boldsymbol{s}'\sim p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})}[V^{\pi}(\boldsymbol{s}')] - V^{\pi}(\boldsymbol{s}). \tag{4}$$

Let $\hat{V}(\boldsymbol{s})$ denote an unbiased estimator of the value function at state $\boldsymbol{s}$. Then, for any given *state-action transition sample*, denoted by $(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')$, the temporal difference (TD) error, i.e.,

$$\delta(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}') = r(\boldsymbol{s},\boldsymbol{a}) + \gamma\hat{V}(\boldsymbol{s}') - \hat{V}(\boldsymbol{s})$$

is an unbiased estimator of the advantage (4); see Bhatnagar et al. (2009), Lemma 3.

In a nutshell, estimating the advantage function requires a set of observations $\{(\boldsymbol{s}_t,\boldsymbol{a}_t,\boldsymbol{s}_{t+1},r_t)\}$ while the policy gradient estimate involves the estimated advantage function $A^{\pi}(\boldsymbol{s},\boldsymbol{a})$ and the estimated score function $\nabla\log\pi_{\boldsymbol{\theta}_k}(\boldsymbol{a}|\boldsymbol{s})$ at state-action pairs $\{(\boldsymbol{s}_t,\boldsymbol{a}_t)\}$. We will discuss the policy gradient estimation and advantage estimation separately. From Section 2.2 to Section 3.1, we will focus on the multiple importance sampling based policy optimization and policy gradient estimation, while supposing that an unbiased estimator of advantage function is given. Then in Section 3.2, we will show how to estimate the advantage function through temporal difference learning. *In this paper, we will focus on creating a selective historical transition replay approach in order to improve the policy gradient estimation.*

2.2 Importance Sampling and Multiple Likelihood Ratio

In this section, we describe how to utilize important sampling (IS) and multiple likelihood ratio (MLR) to improve the policy gradient estimation. Denote the state-action input pair as $\boldsymbol{x} \equiv (\boldsymbol{s},\boldsymbol{a})$. Let $\rho_i(\boldsymbol{x}) = \rho_{\boldsymbol{\theta}_i}(\boldsymbol{s},\boldsymbol{a})$ represent the stationary sampling generative distribution at the i-th episode obtained under a policy specified by $\boldsymbol{\theta}_i$. For any candidate policy specified by $\boldsymbol{\theta}$, let $\rho(\boldsymbol{x}) = \rho_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a})$ denote the target distribution or likelihood. We are interested in estimating the expected gradient $\nabla J(\boldsymbol{\theta}) = \mathbb{E}_{\rho(\boldsymbol{x})}[g(\boldsymbol{x})] = \mathbb{E}_{\rho_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a})}[g(\boldsymbol{s},\boldsymbol{a}|\boldsymbol{\theta})]$.

When the historical samples generated from the *sampling distribution* ρ_i are selected and reused to estimate the candidate policy gradient $\nabla J(\boldsymbol{\theta}_k)$ under the *target distribution* ρ_k, the importance sampling estimator (Andradóttir et al. 1995; Rubinstein and Kroese 2016) corrects the sampling distribution with the importance weight or likelihood ratio defined as $f(\boldsymbol{x}) = \rho_k(\boldsymbol{x})/\rho_i(\boldsymbol{x})$, i.e.,

$$\widehat{\nabla J}(\boldsymbol{\theta}_k) = \frac{1}{n}\sum_{j=1}^{n} f\left(\boldsymbol{x}^{(i,j)}\right) g_k\left(\boldsymbol{x}^{(i,j)}\right),$$

where $\boldsymbol{x}^{(i,j)} \overset{i.i.d.}{\sim} \rho_i(\boldsymbol{x})$ with $j = 1,2,\ldots,n$. For simplification, we allocate a constant number of replications (i,e., n) for each visit at $\boldsymbol{\theta}$. We assume $\rho_i(\boldsymbol{x}) > 0$ whenever $\rho_k(\boldsymbol{x})g_k(\boldsymbol{x}) \neq 0$. This estimator is unbiased,

$$\mathbb{E}_{\rho_i}\left[\widehat{\nabla J}(\boldsymbol{\theta}_k)\right] = \int \frac{\rho_k(\boldsymbol{x})}{\rho_i(\boldsymbol{x})}\rho_i(\boldsymbol{x})g_k(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \int \rho_k(\boldsymbol{x})\,g_k(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \nabla J(\boldsymbol{\theta}_k).$$

However, the likelihood ratio $\rho_k(\boldsymbol{x})/\rho_i(\boldsymbol{x})$ can be extremely large or small at sample $\boldsymbol{x}$. Without any bound on the likelihood ratio $\rho_k(\boldsymbol{x})/\rho_i(\boldsymbol{x})$, the importance sampling estimator $\widehat{\nabla J}$ can have inflated variance, which is typically induced by large difference between target and proposal distributions. Inspired by the BLR-M metamodel (Feng and Staum 2017) and multiple importance sampling (Veach and Guibas 1995), we utilize the mixture likelihood ratio (MLR) method to address this issue. It has a mixture sampling distribution, denoted by $\ell_M(\boldsymbol{x}) \equiv \sum_{i\in U}\alpha_i\rho_i(\boldsymbol{x})$ with $\sum_{i\in U}\alpha_i = 1$, composed of multiple distribution components, denoted by $\{\rho_i(\boldsymbol{x}) : i \in U\}$, where U represents the reuse set. Thus, the MLR can avoid the limitations induced by using a single proposal distribution $\rho_i(\boldsymbol{x})$ and reduce the variance inflation issue.

Given the samples generated from those sampling distributions, denoted by $\{\boldsymbol{x}^{(i,j)} : i \in U \text{ and } j = 1,2,\ldots,n\}$, the MLR estimator, as stratified sampling from the mixture distribution $\ell_M(\boldsymbol{x})$, becomes

$$\widehat{\nabla J}_k^{MLR} = \frac{1}{|U|}\sum_{i\in U}\frac{1}{n}\sum_{j=1}^{n} f_M\left(\boldsymbol{x}^{(i,j)}\right) g_k\left(\boldsymbol{x}^{(i,j)}\right) \quad \text{with} \quad f_M(\boldsymbol{x}) = \frac{\rho_k(\boldsymbol{x})}{\ell_M(\boldsymbol{x})} = \frac{\rho_k(\boldsymbol{x})}{\sum_{i\in U}\alpha_i\rho_i(\boldsymbol{x})}. \tag{5}$$

To have a unbiased MLR estimator, the weight is selected to be the proportion of historical sample size generated from each proposal distribution component $\rho_i(\boldsymbol{x})$, i.e., $\alpha_i = \frac{n}{\sum_{i\in U} n}$. Suppose there are n historical samples generated from each distribution. Then, we allocate equal weight on $\rho_i(\boldsymbol{x})$, i.e., $\alpha_i = 1/|U|$ for $i \in U$, where $|\cdot|$ denotes set cardinality. This MLR estimator is unbiased (Veach and Guibas 1995),

$$\mathbb{E}\left[\widehat{\nabla J}_k^{MLR}\right] = \mathbb{E}\left[\frac{1}{|U|}\sum_{i\in U}\frac{1}{n}\sum_{j=1}^{n}\frac{\rho_k\left(\boldsymbol{x}^{(i,j)}\right)}{\sum_{i\in U}\alpha_i\rho_i\left(\boldsymbol{x}^{(i,j)}\right)} g_k\left(\boldsymbol{x}^{(i,j)}\right)\right] = \frac{1}{|U|}\sum_{i\in U}\int \frac{\rho_k(\boldsymbol{x})}{\frac{1}{|U|}\sum_{i\in U}\rho_i(\boldsymbol{x})} g_k(\boldsymbol{x})\,\rho_i(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \nabla J(\boldsymbol{\theta}_k).$$

The major advantage of the MLR estimator, compared with the standard IS, is higher sample-efficiency and lower gradient estimation variance. Since we always include the transitions generated in the current iteration, the mixture likelihood ratio $f_M(\boldsymbol{x})$ in (5) reaches its max value when the likelihood $\rho_i(\boldsymbol{x}) = 0$ for all remaining sampling distributions with $i \in U$. Thus, this mixture likelihood ratio is bounded, i.e., $f_M(\boldsymbol{x}) \le |U|$, which can control the policy gradient estimation variance inflation issue. *In this way, the mixture likelihood ratio puts higher weight on the samples that are more likely to be generated by the target distribution $\rho(\boldsymbol{x})$ without assigning extremely large weights on the others.*

3 MIXTURE LIKELIHOOD RATIO ASSISTED POLICY OPTIMIZATION

Given a set of historical samples collected under different stationary distributions and behavior policies, the off-policy strategy is used to find the optimal policy maximizing the expected return. One can reuse the past samples to improve the policy gradient estimation through MLR. Let M_k denote the set of all policies (i.e., actor) and value functions (i.e., critic) that have been visited until the beginning of the k-th iteration. Let U_k be *a reuse set* including the indices of model candidates whose transitions are selected and reused for estimating the policy gradient $\nabla J(\boldsymbol{\theta}_k)$. Denote its cardinality as $|U_k|$. For the discussions in this section, suppose that U_k is given. We will present how to determine the reuse set U_k to improve the policy gradient estimation accuracy in Section 4.

3.1 Off-policy Policy Gradient Estimation

Compared to on-policy alternatives, off-policy approaches do not require full trajectories and they can reuse the selected historical transition samples ("experience replay") to improve the sample efficiency. Specifically, we modify the policy gradient (3) such that the mismatch between the sampling distribution $\rho_{\boldsymbol{\theta}_i}(\boldsymbol{s},\boldsymbol{a})$ and the target distribution $\rho_{\boldsymbol{\theta}_k}(\boldsymbol{s},\boldsymbol{a})$ is compensated by importance sampling estimator (5), i.e.,

$$\nabla J(\boldsymbol{\theta}_k) = \mathbb{E}_{\rho_{\boldsymbol{\theta}_i}}\left[f(\boldsymbol{s},\boldsymbol{a})g(\boldsymbol{s},\boldsymbol{a})\right] = \mathbb{E}_{\rho_{\boldsymbol{\theta}_i}}\left[\frac{\rho_{\boldsymbol{\theta}_k}(\boldsymbol{s},\boldsymbol{a})}{\rho_{\boldsymbol{\theta}_i}(\boldsymbol{s},\boldsymbol{a})} A^{\pi_{\boldsymbol{\theta}_k}}(\boldsymbol{s},\boldsymbol{a})\nabla\log\pi_{\boldsymbol{\theta}_k}(\boldsymbol{a}|\boldsymbol{s})\right]. \tag{6}$$

Let $g_k(\boldsymbol{s},\boldsymbol{a}) = \nabla\log\pi_{\boldsymbol{\theta}_k}(\boldsymbol{a}|\boldsymbol{s})A^{\pi_{\boldsymbol{\theta}_k}}(\boldsymbol{s},\boldsymbol{a})$. We can obtain an unbiased estimator of policy gradient in (6) by using sample average approximation (SAA),

$$\widehat{\nabla J}^{ILR}_{i,k} = \frac{1}{n}\sum_{j=1}^{n}\left[\frac{\rho_{\boldsymbol{\theta}_k}\left(\boldsymbol{s}_t^{(i,j)},\boldsymbol{a}_t^{(i,j)}\right)}{\rho_{\boldsymbol{\theta}_i}\left(\boldsymbol{s}_t^{(i,j)},\boldsymbol{a}_t^{(i,j)}\right)} g_k\left(\boldsymbol{a}_t^{(i,j)},\boldsymbol{s}_t^{(i,j)}\right)\right] \quad \text{and} \quad \widehat{\nabla J}^{ILR}_{k} = \frac{1}{|U_k|}\sum_{i\in U_k}\widehat{\nabla J}^{ILR}_{i,k}, \tag{7}$$

where the historical transitions are generated by $\boldsymbol{s}_t^{(i,j)} \sim d^{\pi_{\boldsymbol{\theta}_i}}(\boldsymbol{s})$ and $\boldsymbol{a}_t^{(i,j)} \sim \pi_{\boldsymbol{\theta}_i}(\boldsymbol{a}|\boldsymbol{s}_t^{(i,j)})$ for $j = 1,2,\ldots,n$. In this paper, we use ILR to represent (individual) likelihood ratio. The MLR policy gradient estimator is

$$\widehat{\nabla J}^{MLR}_{k} = \frac{1}{|U_k|}\sum_{i\in U_k}\frac{1}{n}\sum_{j=1}^{n} f_k\left(\boldsymbol{a}_t^{(i,j)},\boldsymbol{s}_t^{(i,j)}\right) g_k\left(\boldsymbol{a}_t^{(i,j)},\boldsymbol{s}_t^{(i,j)}\right) \quad \text{with} \quad f_k(\boldsymbol{a}_t,\boldsymbol{s}_t) = \frac{\rho_{\boldsymbol{\theta}_k}(\boldsymbol{s}_t,\boldsymbol{a}_t)}{\frac{1}{|U_k|}\sum_{i\in U_k}\rho_{\boldsymbol{\theta}_i}(\boldsymbol{s}_t,\boldsymbol{a}_t)}. \tag{8}$$

The key challenge of utilizing the ILR and MLR policy gradient estimators is computing the stationary distributions $d^{\pi}(\boldsymbol{s})$ in $\rho_{\boldsymbol{\theta}_k}(\boldsymbol{s},\boldsymbol{a}) = \pi_{\boldsymbol{\theta}_k}(\boldsymbol{a}|\boldsymbol{s})d^{\pi}(\boldsymbol{s})$. This problem is also known as distribution corrections (DICE) in RL. Fortunately, a list of approaches has been recently proposed to address the challenge; see for example Nachum et al. (2019), Yang et al. (2020).The off-policy gradient estimator can be simplified by introducing bias. Degris et al. (2012) proposed an off-policy (actor-critic) gradient approximate,

$$\nabla J(\boldsymbol{\theta}_k) \approx \mathbb{E}_{\rho_{\boldsymbol{\theta}_i}}\left[\frac{\pi_{\boldsymbol{\theta}_k}}{\pi_{\boldsymbol{\theta}_i}}\nabla \log \pi_{\boldsymbol{\theta}_k}(\boldsymbol{a}|\boldsymbol{s})A^{\pi_{\boldsymbol{\theta}_k}}(\boldsymbol{s},\boldsymbol{a})\right]. \tag{9}$$

It can preserve the set of local optima to which gradient ascent converges. Although biased, this estimator has been widely used in many state-of-the-art off-policy algorithms (Schulman et al. 2015; Schulman et al. 2017) due to its simplicity and computational efficiency.

The ILR and MLR policy gradient estimators for the approximation (9) can be obtained by replacing stationary state-action distribution $\rho_{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a})$ with policy $\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s})$. We conclude this section by pointing out that we proceed the theoretical analysis with the unbiased off-policy policy gradient (6) and its SAA estimators (7)-(8) in the following sections while the estimator (9) is used in the algorithm and the empirical study.

3.2 Actor-Critic

The actor-critic is a widely used architecture for policy optimization. It includes two main components: actor and critic. The actor corresponds to an action-selection policy, mapping state to action in a probabilistic manner. The critic corresponds to a value function, mapping state to the expected cumulative future reward. The actor searches the optimal policy parameters by using stochastic gradient ascent (SGA) while the critic estimates the action-value function $Q^{\pi}(\boldsymbol{s},\boldsymbol{a})$ by an appropriate policy evaluation algorithm such as temporal-difference learning or Q-learning. Usually, the critic $V^{\pi}(\boldsymbol{s})$ is approximated by a state-value function $V_{\boldsymbol{w}}(\boldsymbol{s})$ specified by parameters $\boldsymbol{w}$ and the actor is represented by a policy function $\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s})$ specified by $\boldsymbol{\theta}$. Such functional approximation of critic can be used in estimating the state-value function $\hat{V}(\boldsymbol{s}) = V_{\boldsymbol{w}}(\boldsymbol{s})$ and thus the TD error (5) becomes $\delta(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}') = r(\boldsymbol{s},\boldsymbol{a}) + \gamma V_{\boldsymbol{w}}(\boldsymbol{s}') - V_{\boldsymbol{w}}(\boldsymbol{s})$. Following the studies in Bhatnagar et al. (2009), a typical actor-critic update can be written as

$$\textbf{TD Error}: \quad \delta_k = r_t + \gamma V_{\boldsymbol{w}_k}(\boldsymbol{s}') - V_{\boldsymbol{w}_k}(\boldsymbol{s}) \tag{10}$$

$$\textbf{Critic}: \quad \boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \eta_w \delta_k \nabla_w V_{\boldsymbol{w}_k}(\boldsymbol{s}) \tag{11}$$

$$\textbf{Actor}: \quad \boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \eta_\theta \nabla J(\boldsymbol{\theta}) \tag{12}$$

where η_w and η_θ represent learning rates for critic and actor respectively. *The step in* (10) *is also referred as to temporal difference learning used to estimate the advantage function.* The policy gradient $\nabla J(\boldsymbol{\theta})$ in (12) is estimated by the MLR policy gradient estimator (8).

4 VARIANCE REDUCTION EXPERIENCE REPLAY FOR POLICY GRADIENT ESTIMATION

In this section, we derive the variance reduced experience replay method and present a general VRER based actor-critic algorithm. We provide a selection criteria in Section 4.1 that can automatically find the most relevant historical transition observations for constructing the reuse set U_k at each k-th iteration and improving the policy gradient estimation accuracy. The dependencies between historical samples collected under selected behavior policies in the previous iterations lead to a general obstacle for most historical sample reusing mechanisms. The MLR, used to leverage the information from previous transition observations, requires sampling distributions to be independent. Thus, in the proposed algorithm, we reduce this interdependence through randomly sampling (Mnih et al. 2015). Specifically, we separate the optimal policy learning algorithm into two steps. In the online step, we collect new samples by following the target policy specified by $\boldsymbol{\theta}_k$. In the offline step, we select historical samples and train the actor critic model by stochastic gradient ascent. In this way, we can view the offline step as a normal offline optimization problem where samples are assumed to be randomly generated from a set of independent stationary state-action distributions ρ_i with $i \leq k$. Therefore, for the theoretical study in Section 4.1, we assume that the transitions are drawn from a set of independent stationary state-action distributions in the offline optimization step.

4.1 Selection Rule for MLR based Policy Gradient Estimator

We first introduce some properties of MLR based policy gradient estimator. Similar results can be found in Veach and Guibas (1995) and Feng and Staum (2017).

Lemma 1 Conditional on the reuse set U_k, the MLR policy gradient estimator is unbiased, i.e.,

$$\mathbb{E}\left[\widehat{\nabla J}_k^{MLR}\middle| U_k\right] = \mathbb{E}\left[g_k(\boldsymbol{\tau})|\boldsymbol{\theta}_k\right] = \nabla J(\boldsymbol{\theta}_k).$$

Proposition 1 Conditional on the reuse set U_k, the total variance of the MLR policy gradient estimator is smaller and equal to that of the average ILR policy gradient estimator,

$$\mathrm{Tr}\left(\mathrm{Var}\left[\widehat{\nabla J}_k^{MLR}\middle| U_k\right]\right) \leq \mathrm{Tr}\left(\mathrm{Var}\left[\widehat{\nabla J}_k^{ILR}\middle| U_k\right]\right).$$

Proof. Similar proofs can be found in Martino et al. (2015), Theorem A.2. and Feng and Staum (2017), Proposition 2.5. □

The perspective of multiple importance sampling and variance reduction experience replay is to select and reuse historical transition samples generated from those behavioral policies and sampling distributions that are close to the target one. We propose the selection criteria in Theorem 1, which measures the distance between the behavioral and target distributions based on the variance of policy gradient estimators obtained by using historical samples versus new samples generated in the current k-th iteration.

Theorem 1 (Selection Rule) At each k-th iteration with the target distribution ρ_k, the reuse set U_k is created to include the stationary distributions, i.e., ρ_i specified by $(\boldsymbol{\theta}_i, \boldsymbol{w}_i)$ with $i \leq k$, whose ILR policy gradient estimator variance is no greater than c times the total variance of the vanilla PG estimator for some constant $c > 1$. Mathematically,

$$\mathrm{Tr}\left(\mathrm{Var}\left[\widehat{\nabla J}_{i,k}^{ILR}\middle| M_k\right]\right) \leq c\,\mathrm{Tr}\left(\mathrm{Var}\left[\widehat{\nabla J}_k^{PG}\middle| M_k\right]\right). \tag{13}$$

Then, based on such reuse set U_k, the total variance of the MLR policy gradient estimator (8) is no greater than $c/|U_k|$ times the total variance of the vanilla PG estimator,

$$\mathrm{Tr}\left(\mathrm{Var}\left[\widehat{\nabla J}_k^{MLR}\middle| M_k\right]\right) \leq \frac{c}{|U_k|}\mathrm{Tr}\left(\mathrm{Var}\left[\widehat{\nabla J}_k^{PG}\middle| M_k\right]\right). \tag{14}$$

Proof. We screen all historical models and select historical samples and visited models that satisfy the rule (13). Let U_k represent the index set of models that are reused or equivalently experience to be replayed. Conditional on all visited models M_k, the historical samples, i.e., $\{(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}')^{(i,j)}$ with $(\boldsymbol{\theta}_i, \boldsymbol{w}_i) \in M_k$ and $j = 1, 2, \ldots, n\}$, are independent. Thus, we have

$$\begin{aligned}
\mathrm{Tr}\left(\mathrm{Var}\left[\widehat{\nabla J}_k^{MLR}\middle| M_k\right]\right) &\overset{(\star)}{\leq} \mathrm{Tr}\left(\mathrm{Var}\left[\widehat{\nabla J}_k^{ILR}\middle| M_k\right]\right) = \frac{1}{|U_k|^2}\sum_{i\in U_k}\mathrm{Tr}\left(\mathrm{Var}\left[\widehat{\nabla J}_{i,k}^{ILR}\middle| M_k\right]\right) \\
&\overset{(\star\star)}{\leq} \frac{c}{|U_k|^2}\sum_{i\in U_k}\mathrm{Tr}\left(\mathrm{Var}\left[\widehat{\nabla J}_k^{PG}\middle| M_k\right]\right) = \frac{c}{|U_k|}\mathrm{Tr}\left(\mathrm{Var}\left[\widehat{\nabla J}_k^{PG}\middle| M_k\right]\right)
\end{aligned}$$

where $(\star)$ follows by applying Proposition 1 and $(\star\star)$ holds by applying the selection rule (13). □

Theorem 1 provides the selection criteria for dynamically and automatically determining the reuse set U_k. It shows that the MLR can greatly reduce the policy gradient estimation variance compared to the vanilla policy gradient estimator through reusing the historical transition samples in U_k. During the optimal policy search, the number of reuse transitions (or $|U_k|$) increases as the iteration k increases. The total variance of the MLR based policy gradient estimator can be significantly reduced.

4.2 Green Simulation Assisted Policy Gradient Algorithm for Partial Trajectory Reuse

We summarize the proposed VRER based policy optimization algorithm in an actor-critic framework in Algorithm 1. At each k-th iteration, we generate n transitions by running experiments following the target policy specified by parameters $\boldsymbol{\theta}_k$ and update the observation set $\mathscr{D}_k$ in Step 1. We select the historical samples that satisfy the selection rule (13) and use the associated policies to create the reuse set U_k in Step 2. For computational reasons, we use the policy gradient estimator (9) in the algorithm. This approximation simplifies the calculation of the likelihood ratio ρ_k/ρ_i to the likelihood ratio of policies $\pi_{\boldsymbol{\theta}_k}/\pi_{\boldsymbol{\theta}_i}$ and thus avoid the substantial computation involved in estimating the stationary distribution $d^{\pi}(\boldsymbol{s})$.

```
Algorithm 1: Actor Critic Method with Variance Reduced Experience Replay.
  Input: the selection threshold constant $c$; the maximum number of iterations $K$; the number of iterations in
    offline optimization $K_{off}$; the number of replications per iteration $n$; the set of historical trajectories from
    the real system $\mathscr{D}_0$; the set of policy parameters visited so far $U_0$; the set of stored likelihoods $\mathscr{L}_0$.
  Initialize actor parameter $\boldsymbol{\theta}_1$ and critic parameter $\boldsymbol{w}_1$. Store them in $M_1 = M_0 \cup \{(\boldsymbol{\theta}_1, \boldsymbol{w}_1)\}$;
  for $k = 1, 2, \ldots, K$ do
      1. Collect a set of transitions $\mathscr{T}_k = \{(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}, r_t)\}_{t=1}^{n}$ from real system by running policy $\boldsymbol{\pi}_{\boldsymbol{\theta}_k}(\boldsymbol{a}_t|\boldsymbol{s}_t)$;
        Update the sets $\mathscr{D}_k \leftarrow \mathscr{D}_{k-1} \cup \mathscr{T}_k$;
      2. Initialize $U_k = \emptyset$, screen all historical transitions and policies in $U_k$, and construct the reuse set $U_k$;
      for $(\boldsymbol{\theta}_i, \boldsymbol{w}_i) \in M_k$ do
          (a) Compute and store the new likelihoods:
              $\mathscr{L}_k \leftarrow \mathscr{L}_{k-1} \cup \pi_{\boldsymbol{\theta}_k}(\mathscr{D}_k) \cup \pi_{\boldsymbol{\theta}_{[1:k]}}(\mathscr{T}_k)$
          (b) if $\operatorname{Tr}\left(\operatorname{Var}\left[\widehat{\nabla J}_{i,k}^{ILR} \middle| M_k\right]\right) \leq c\operatorname{Tr}\left(\operatorname{Var}\left[\widehat{\nabla J}_{k}^{PG} \middle| M_k\right]\right)$ then $U_k \leftarrow U_k \cup \{i\}$.
      end
      3. Reuse the historical samples associated with $U_k$ and stored likelihoods $\mathscr{L}_k$ to update actor and critic:
      (a) Let $\boldsymbol{\theta}_k^0 = \boldsymbol{\theta}_k$ and $\boldsymbol{w}_k^0 = \boldsymbol{w}_k$;
      for $h = 0, 1, ..., K_{off}$ do
          (b) TD Error: $\delta_k^h = r_t + \gamma V_{\boldsymbol{w}_k^h}(\boldsymbol{s}') - Q_{\boldsymbol{w}_k^h}(\boldsymbol{s})$;
          (c) Actor Update: $\boldsymbol{\theta}_k^{h+1} \leftarrow \boldsymbol{\theta}_k^h + \eta_k \widehat{\nabla J}_k^{MLR}$;
          (d) Critic Update: $\boldsymbol{w}_k^{h+1} = \boldsymbol{w}_k^h + \eta_k \delta_k \nabla_w V_{\boldsymbol{w}_k^h}(\boldsymbol{s})$;
      end
      4. Update the actor and critic: $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k^{K_{off}}$ and $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k^{K_{off}}$;
      5. Store them to the set $M_{k+1} = M_k \cup \{(\boldsymbol{\theta}_{k+1}, \boldsymbol{w}_{k+1})\}$;
  end
```

The likelihoods are stored and reused in Step 2(a) to reduce the computation cost. Specifically, as the iteration k increases, the number of historical transitions increases. It can be computationally expensive to repeatedly calculate all likelihood ratios required for historical observation selection and policy gradient estimation. Thus, we save and reuse the previous calculated likelihoods. Let $\mathscr{T}_k = \{(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}, r_t)\}_{t=1}^{n}$ represent the set of new transitions generated in the k-th iteration. Update the set of all transition observations, i.e., $\mathscr{D}_k \leftarrow \mathscr{D}_{k-1} \cup \mathscr{T}_k$. Then, the likelihoods of $\mathscr{T}_k$ under any previous visited policy $\boldsymbol{\theta}_i$ are $\pi_{\boldsymbol{\theta}_i}(\mathscr{T}_k) = \{\pi_{\boldsymbol{\theta}_i}(\boldsymbol{a}_t|\boldsymbol{s}_t) : (\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}, r_t) \in \mathscr{T}_k\}$. Let $\pi_{\boldsymbol{\theta}_{[1:k]}}(\mathscr{T}_k) = \{\pi_{\boldsymbol{\theta}_i}(\mathscr{T}_k) : i = 1, \ldots, k\}$. Therefore, at the

k-th iteration, all newly generated likelihoods are the joint set of the values of historical samples at new policy $\boldsymbol{\theta}_k$ and new samples at historical policies $\boldsymbol{\theta}_i$, i.e., $\pi_{\boldsymbol{\theta}_k}(\mathscr{D}_k)\cup\pi_{\boldsymbol{\theta}_{[1:k]}}(\mathscr{T}_k)$. Then we get the MLR policy gradient estimate by applying (8) and train the actor and critic by following (10)-(12) in Step 3. In this offline step, we use the stochastic gradient ascent to iteratively optimize the performance objective with the historical observations in U_k. At the end, we update actor and critic with latest parameters $\boldsymbol{\theta}_k^{K_{off}}$ and $\boldsymbol{w}_k^{K_{off}}$ from the offline optimization step and store them into memory buffer of models M_{k+1}. After that, we repeat the procedure until reaching to the budget limit specified by K iterations.

We conclude this section by pointing out that the choice of optimizers for actor/critic training, the hyperparamter K_{off} and the size of minibatch in SGA are task-specific. In some actor-critic algorithms, the termination of offline optimization or the number of iterations K_{off} are often not fixed. For example, PPO uses early stopping method to determine K_{off}: terminate the offline training if the KL divergence between behavioral and target policies is smaller than some threshold. The interested reader is referred to the literature of stochastic gradient methods (Goodfellow et al. 2016) for details of hyperparamter tuning.

5 EMPIRICAL STUDY

In this section, we present the empirical study assessing the performance VRER in combination with actor critic algorithm (Bhatnagar et al. 2009) and PPO algorithm (Schulman et al. 2017). We study the optimization convergence behavior by using control tasks in Section 5.1, present the sensitivity analysis on the reuse set selection threshold constant c in Section 5.2, and investigate the effects of employing VRER to the gradient variance reduction in Section 5.3. For the implementation, we use two open-sourced libraries, Keras and TensorFlow for policy modeling and automatic differentiation. In addition, we use OpenAI gym (Brockman et al. 2016) to provide the simulation environment of Cartpole and Acrobot problems.

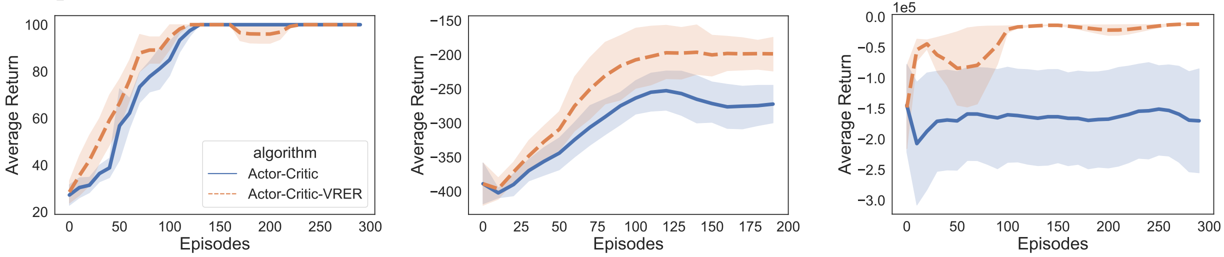
We adopt the yeast cell fermentation simulator from Zheng et al. (2022). To provide the prediction on the process dynamics, we add 4 additional state variables, including time t, the growth rate $\dot{X}_f$, the production rate of citrate acid $\dot{C}$, and the consumption rate of substrate $\dot{S}$. Therefore, the state vector is $\boldsymbol{s}=(X_f,C,S,N,V,t,\dot{X}_f,\dot{C},\dot{S})$, where X_f represents lipid-free cell mass; C measures the citrate concentration, i.e., the actual "product" to be harvested at the end of the fermentation process, generated by the cells' metabolism; S and N are the amounts of substrate (a type of oil) and nitrogen, used for cell growth and production; and V is the working volume of the entire batch. In this paper, we consider a setpoint control problem that aims to maintain the substrate concentration S_t around a fixed value. The reward function is defined as $r(\boldsymbol{s}_t)=-(S_t-S_t^0)^2$, where $S_t^0=20$ g/L is the setpoint of substrate concentration. We consider the feed rate as the action representing the amount of substrate added in each time interval.

The actor and critic models for Actor-Critic and PPO are adopted from the Keras implementation (Chollet et al. 2015). The Actor-Critic model is composed of a shared initial layer with 128 neurons and separate outputs for the actor and critic. The PPO algorithm has separate actor and critic neural network models, both of which have two layers with 64 neurons. For the problems with discrete action, we use softmax activation function on top of the actor network, which calculates the probability of optimal actions. For the fermentation problem with a continuous action (feeding rate of substrate), we use the Gaussian policy for actor model (Sutton and Barto 2018). As the feed rate is strictly regulated and it should stay within a regulation required acceptance range, we truncate the action sampled from Gassuain policy. At each k-th iteration, based on the results obtained from 30 macro-replications, we represent the estimation uncertainty of outputs (i.e., the expected discounted rewards and the total variance of policy gradient) by using the 95% confidence bands based on asymptotic normality assumption.

5.1 Comparison of Algorithm Performance with and without Proposed VRER

In this section, we compare the optimal policy learning performance of VRER using Actor-Critic and PPO algorithms on some classical continuous control benchmarks. We set the same initial learning rate for both actor and critic in Actor-Critic algorithm (i.e., Cartpole: 0.005; Acrobot: 0.001; and Fermentation: 0.001).

For PPO, the learning rates of actor and critic were set to be 0.001 and 0.005 respectively for all three examples. The selection threshold constant was set to be $c = 1.5$ for all experiment in this section.

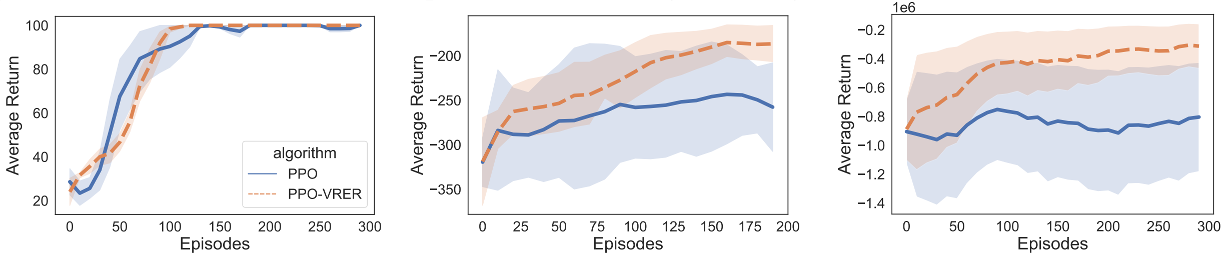


(a) CartPole (b) Acrobot (c) Fermentation

Figure 1: Convergence results for the Actor-Critic algorithm with and without using the proposed VRER.

We plot the mean performance curves and 95% confidence intervals of the actor-critic with and without VRER in Figure 1. For the Cartpole problem, although the Actor-Critic algorithms converge to the optimum with and without using VRER, the Actor-Critic-VRER shows significantly faster convergence than the Actor-Critic. This indicates that the use of VRER gives significant performance improvement. Similar performance improvement can be also seen in Acrobot example (Figure 1b), where Actor-Critic-ARER shows not only the convergence to the optimum but also faster convergence. For the fermentation problem, Actor-Critic-VRER shows performance improvement while Actor-Critic method even fails to converge.

We plot the mean performance curves and 95% confidence intervals of PPO in Figure 2. In Cartpole, the average return of PPO-VRER converges about 25 iterations earlier than PPO. In Acrobot and Fermentation problems, PPO-VRER shows better performance with higher average return and lower variance.



(a) CartPole (b) Acrobot (c) Fermentation

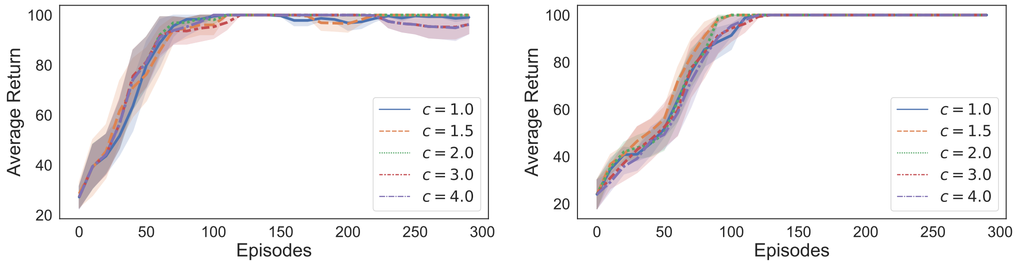
Figure 2: Convergence results for the PPO algorithm with and without using the proposed VRER.

5.2 Sensitivity Analysis on the Selection of Reuse Threshold c

We study the sensitivity of convergence performance of actor critic and PPO algorithms with VRER to the choice of constant c used in the selection criteria (13). Figure 3 shows the convergence behaviors when we solve the Cartpole problem with different values of c. All the performance curves stay close, which indicates that the convergence of the proposed VRER based policy optimization approach is robust to the choice of c.

5.3 Variance Reduction

In this section, we present empirical results to assess the performance of the proposed VRER in terms of reducing the policy gradient estimation variance. We first test the proposed VRER method in conjunction with actor-critic algorithm (Actor-Critic-VRER) in three distinct control examples (Figure 4). The original actor-critic method is an on-policy reinforcement learning algorithm that thus suffers from the high variability of gradient estimators. By selectively reusing historical transition observations through the VRER-based MLR approach, the Actor-Critic algorithm shows a significant reduction in the estimation variance of policy gradient in all three examples, compared to the original policy gradient without any experience replay.

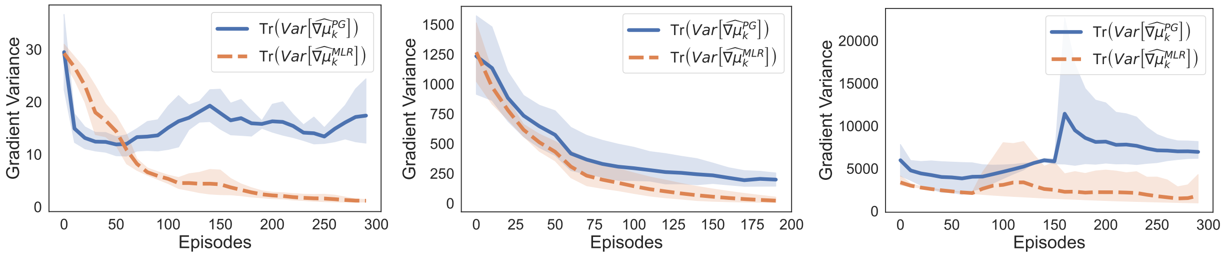


(a) Actor Critic

(b) PPO

Figure 3: Sensitivity analysis of the reuse selection threshold constant c in Cartpole example.

Similar results are also observed for the PPO algorithm. The PPO, instead of using multiple importance sampling, clips/truncates likelihood ratio for policy regularization and therefore eliminates the inflated gradient variance caused by extreme samples (Schulman et al. 2017). The chipping technique, as an alternative to MLR method, provides a simple and computation efficient method to regularize the policy gradient and adjust distributional difference. However it introduces extra bias and thus may cause the algorithm stuck at suboptimum. The results in Figure 5 show that the use of VRER can still reduce the estimation variance of PPO policy gradient estimator even if MLR is replaced by chipped likelihood ratio.

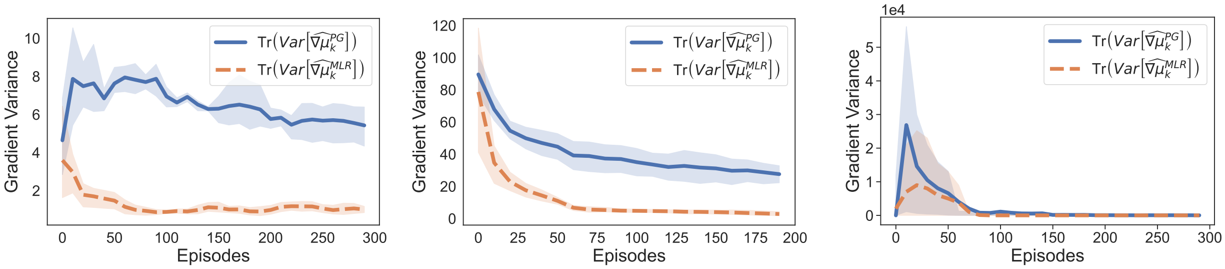


(a) CartPole

(b) Acrobot

(c) Fermentation

Figure 4: Policy gradient estimation variance results of Actor-Critic algorithm with and without VRER.



(a) CartPole

(b) Acrobot

(c) Fermentation

Figure 5: Policy gradient estimation variance results of PPO algorithm with and without VRER.

6 CONCLUSION

To guide real-time process control in low-data situations, we create a variance reduction experience replay approach to accelerate policy gradient optimization. The proposed selection rule guarantees the variance reduction in the policy gradient estimation through selectively reusing the most relevant historical transition observations and automatically allocating more weights to those observations or partial trajectories that are more likely generated by the target stochastic decision process model. The empirical studies show that the incorporation of proposed VRER and MLR with the state-of-the-art policy optimization approaches can substantially improve their optimization convergence especially under the situations with a tight budget.

REFERENCES

Andradóttir, S., D. P. Heyman, and T. J. Ott. 1995. "On the Choice of Alternative Measures in Importance Sampling with Markov Chains". *Operations Research* 43(3):509–519.

Bhatnagar, S., R. S. Sutton, M. Ghavamzadeh, and M. Lee. 2009. "Natural Actor–Critic Algorithms". *Automatica* 45(11):2471–2482.

Brockman, G., V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. 2016. "OpenAI Gym". https://www.gymlibrary.dev/. Accessed 15th October 2022.

Chollet, F. et al. 2015. "Keras". https://keras.io. Accessed 15th October 2022.

Degris, T., M. White, and R. S. Sutton. 2012. "Off-Policy Actor-Critic". In *Proceedings of the 29th International Coference on International Conference on Machine Learning*. June 1st-26th, Edinburgh Scotland, 179–186.

Feng, M., and J. Staum. 2017, October. "Green Simulation: Reusing the Output of Repeated Experiments". *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 27(4):23:1–23:28.

Goodfellow, I., Y. Bengio, and A. Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Martino, L., V. Elvira, D. Luengo, and J. Corander. 2015. "An Adaptive Population Importance Sampler: Learning From Uncertainty". *IEEE Transactions on Signal Processing* 63(16):4422–4437.

Metelli, A. M., M. Papini, N. Montali, and M. Restelli. 2020. "Importance Sampling Techniques for Policy Optimization". *Journal of Machine Learning Research* 21(141):1–75.

Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al. 2015. "Human-Level Control Through Deep Reinforcement Learning". *Nature* 518(7540):529–533.

Nachum, O., Y. Chow, B. Dai, and L. Li. 2019. "DualDICE: Behavior-Agnostic Estimation of Discounted Stationary Distribution Corrections". In *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Volume 32: Curran Associates, Inc.

Rubinstein, R. Y., and D. P. Kroese. 2016. *Simulation and the Monte Carlo method*, Volume 10. John Wiley & Sons.

Schlegel, M., W. Chung, D. Graves, J. Qian, and M. White. 2019. "Importance Resampling for Off-Policy Prediction". In *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Volume 32: Curran Associates, Inc.

Schulman, J., S. Levine, P. Abbeel, M. Jordan, and P. Moritz. 2015. "Trust Region Policy Optimization". In *Proceedings of the 32nd International Conference on Machine Learning*, Volume 37. July 4th-9th, Lille, France, 1889–1897.

Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. 2017. "Proximal Policy Optimization Algorithms". *arXiv preprint arXiv:1707.06347*. https://arxiv.org/pdf/1707.06347.pdf. Accessed 15th October 2022.

Silver, D., G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. 2014. "Deterministic Policy Gradient Algorithms". In *Proceedings of the 31st International Conference on Machine Learning*. June 1st-26th, Beijing, China, 387–395: JMLR.org.

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

Sutton, R. S., D. McAllester, S. Singh, and Y. Mansour. 1999. "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In *Advances in Neural Information Processing Systems*, edited by S. Solla, T. Leen, and K. Müller, Volume 12. Cambridge, Massachusetts: MIT Press.

Veach, E., and L. J. Guibas. 1995. "Optimally Combining Sampling Techniques for Monte Carlo Rendering". In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, 419–428. New York, NY, USA: Association for Computing Machinery.

Yang, M., O. Nachum, B. Dai, L. Li, and D. Schuurmans. 2020. "Off-Policy Evaluation via the Regularized Lagrangian". In *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Volume 33, 6551–6561. Red Hook, NY, USA: Curran Associates Inc.

Zheng, H., W. Xie, and M. B. Feng. 2020. "Green Simulation Assisted Reinforcement Learning with Model Risk for Biomanufacturing Learning and Control". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 337–348. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zheng, H., W. Xie, I. O. Ryzhov, and D. Xie. 2022. "Policy Optimization in Bayesian Network Hybrid Models of Biomanufacturing Processes". *INFORMS Journal on Computing*. In Press.

AUTHOR BIOGRAPHIES

HUA ZHENG is Ph.D. candidate of the Department of Mechanical and Industrial Engineering (MIE) at Northeastern University. His research interests include machine learning, reinforcement learning, computer simulation and stochastic optimization. His email address is zheng.hua1@northeastern.edu. His website is https://zhenghuazx.github.io/hua.zheng/

WEI XIE is an assistant professor in MIE at Northeastern University. Her research interests include interpretable Artificial Intelligence (AI), machine learning, computer simulation, data analytics, and stochastic optimization for cyber-physical system risk management, learning, and automation. Her email address is w.xie@northeastern.edu. Her website is http://www1.coe.neu.edu/∼wxie/