Scalable Safety-Critical Policy Evaluation with Accelerated Rare Event Sampling

Mengdi Xu[1], Peide Huang[1], Fengpei Li[2,3], Jiacheng Zhu[1], Xuewei Qi[4], Kentaro Oguchi[4], Zhiyuan Huang[1,5], Henry Lam[2], Ding Zhao[1]

***Abstract*— Evaluating rare but high-stakes events is one of the main challenges in obtaining reliable reinforcement learning policies, especially in large or infinite state/action spaces where limited scalability dictates a prohibitively large number of testing iterations. On the other hand, a biased or inaccurate policy evaluation in a safety-critical system could potentially cause unexpected catastrophic failures during deployment. This paper proposes the Accelerated Policy Evaluation (APE) method, which simultaneously uncovers rare events and estimates the rare event probability in Markov decision processes. The APE method treats the environment nature as an adversarial agent and learns towards, through adaptive importance sampling, the zero-variance sampling distribution for the policy evaluation. Moreover, APE is scalable to large discrete or continuous spaces by incorporating function approximators. We investigate the convergence property of APE in the tabular setting. Our empirical studies show that APE can estimate the rare event probability with a smaller bias while only using orders of magnitude fewer samples than baselines in multi-agent and single-agent environments.**

I. INTRODUCTION

There has been a growing interest in applying reinforcement learning (RL) to complex high-stakes robotics problems, including controlling autonomous vehicles and healthcare assistant robots [1], [2]. As one can expect, before deployment of any RL policy, such safety-critical applications often require an accurate policy evaluation because inaccurate estimations could lead to false optimism or even catastrophic consequences. However, a main difficulty for policy evaluation in these settings is that we are crucially interested in assessing certain rare but high-stake events in safety-critical systems. The *de facto* standard evaluation method is the vanilla Monte Carlo (MC) which, in a multitude of practical settings of interest, requires a prohibitively large number of testing before the evaluation can be deemed statistically valid [3]. For example, [1] shows that, in order to demonstrate the safety of self-driving cars, the number of miles that needs to be clocked in is technically in the hundreds of billions. Consequently, when such large-volume testing is high-stakes, costly, or time-consuming, a growing number of studies have been developed to accelerate the estimation of rare event probability as the policy evaluation metric [4], [5]. However, existing literature typically either fails to exploit the sequential, interactive nature of tasks by confining to specific failure-causing initial conditions [6], [7] or only focuses on small state/action spaces due to the curse of dimensionality [8], [9].

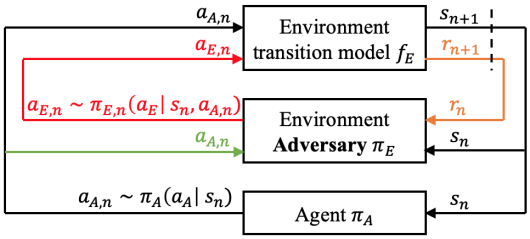


Fig. 1: The interactions among the environment transition model, the agent and the environment adversary introduced by our proposed accelerated policy evaluation method. The reward signal r is for updating the adversary policy π_E. The subscript n denotes the time step.

In this paper, we develop an efficient and scalable policy evaluation method for estimating rare event probabilities in sequential decision-making tasks, which expedites the policy evaluation in large or continuous state/action spaces. In particular, we specify a rare event set and estimate the probability of entering the set under a given RL policy in an environment of interest. We focus on episodic settings with *an adjustable grey-box testing ground* that simulates the environment (i.e., a simulator with adjustable disturbances or vehicle policies), which is the common practice in third-party evaluations or in curriculum design [6], [10]. We summarize our major contributions as follows:

- We propose the *Accelerated Policy Evaluation* (APE) method, suitable for Markov decision processes (MDPs) and motivated by adaptive importance sampling [9]. In a novel way, APE introduces an adversary that controls the environment transition at each step and gradually exploits sequential interactions that cause rare events. We further theoretically establish the convergence property of APE in the tabular setting.
- We propose a scalable implementation of APE and denote it as *APE_scalable*. *APE_scalable* incorporates two function approximators: a Gaussian Process (GP) [11] to estimate rare event probability with high data efficiency, and a conditional normalizing flow model [12] to expressively represent the adversary's policy. We further design a two-timescale gradient-based updating

[1] Mengdi Xu, Peide Huang, Jiacheng Zhu, Zhiyuan Huang, and Ding Zhao are with Carnegie Mellon University. mengdixu, peideh, jzhu4, dingzhao@andrew.cmu.edu

[2] Fengpei Li and Henry Lam are with Columbia University. khl2114@columbia.edu

[3] Fengpei Li is also with Morgan Stanley Machine Learning Research. fl2412@columbia.edu

[4] Xuewei Qi and Kentaro Oguchi are with Toyota Motor North America R&D. tony.qi, kentaro.oguchi@toyota.com

[5] Zhiyuan Huang is also with Tongji University. zhuang2@andrew.cmu.edu

rule for GP to adapt to the rare event setting.
- We empirically show that APE_scalable estimates the rare event probability with a smaller bias and orders of magnitude fewer samples than baselines, in driving scenarios [13] and in a LunarLander control task [14].

II. Related Work

Motivated by the inherent difficulty of assessing rare, safety-critical events in lab or field tests, there has been a growing literature to maximize the sampling efficiency of rare event probability estimation. Two powerhouses are the importance sampling method [4], [15], [16] and the multi-level splitting method [17]. However, little literature focuses on MDP settings. Recent developments in [6], [18] adjust the environment setting at the beginning of each episode while ignoring the intermediate interactive steps. However, our proposed APE method is designed to leverage the sequential nature of tasks and hence is more suitable in the RL settings. The APE is similar to the adaptive stochastic approximation (ASA) for Markov chains [9]. Nonetheless, ASA restricts to discrete state space and is not incorporated with the decision process, thus handicapping its applicability for sophisticated decision-making agents such as continuous-space RL. Our setting is similar to [5] because both utilize a dynamic programming paradigm. However, [5] relies on discretizing and decomposing the simulation scene to make it scalable.

APE is within the regime of safe validation [19] and specifically focuses on finding the distribution of failure-causing disturbances and estimating the failure rate. Adversarial evaluation [6], [20], mainly motivated by attacks on RL agents, also aims to find failure cases but in a distribution-free manner. Instead of searching for worst-case scenarios as in adversarial attacks [21], APE aims to directly find those rare but possible (failure) cases when the agent interacts with the original environment of interest.

APE utilizes updating rules involving dynamic programming and importance weights over changing policies. The rules share a similar form with off-policy evaluation [22] methods including Retrace [23], V-trace [24], and importance resampling [25]. Concretely, the environment's ground truth policy $\pi_{E,gt}$ and the proposed adversary policy π_E in APE are analogous to the evaluation policy and behavior policy in off-policy evaluation [26], respectively.

III. Rare Event Probability as Policy Value

A general MDP consists of a tuple $(\mathcal{S}, \mathcal{A}, p, r)$. $\mathcal{S}$ and $\mathcal{A}$ denote the state and action space. s' denotes the proceeding state of s. The stochastic transition model is $p(s'|s,a)$: $s, s' \in \mathcal{S}, a \in \mathcal{A}$ and the one-step reward is $r(s,s'), s, s' \in \mathcal{S}$. In the rare event setting, we consider the state space $\mathcal{S} = \mathcal{T} \cup \mathcal{I}$ is a partition of terminal states $\mathcal{T}$ and interior states $\mathcal{I}$, $\mathcal{T} \cap \mathcal{I} = \emptyset$. We assume $\mathcal{T}$ is reachable from any states in $\mathcal{I}$ for all policies under consideration and the episode length τ is finite almost surely. The rare event set is denoted by $\mathcal{R}$ and $\mathcal{R} \subset \mathcal{T}$.

We formulate the rare event probability (or the probability of hitting $\mathcal{R}$ before $\mathcal{T} \backslash \mathcal{R}$) starting from an initial state $s_0 \in \mathcal{I}$ in an environment of interest, as the expected undiscounted total reward until termination. With $v^\star$ as the true value, we have the following Bellman equation holds:

$$v^\star(s_0) := \mathbb{E}_{p_{E,gt},\pi_A}\Big[\sum_{n=0}^{\tau-1} r(s_n, s_{n+1})\Big] = \tag{1}$$
$$\sum_{a_A \in \mathcal{A}_A} \pi_A(a_A|s_0) \sum_{s_1 \in \mathcal{S}} p_{E,gt}(s_1|s_0,a_A)\Big[r(s_0,s_1) + v^\star(s_1)\Big],$$

where π_A is the given Markov policy (typically obtained through training). $p_{E,gt}$ is the ground truth transition probability of the environment of interest. n is the timestep. Furthermore, $a_A \sim \pi_A$ is the agent action, and $\mathcal{A}_A$ is the agent action space. The one-step reward is an indicator function with $r(s,s') = 1$ if $s' \in \mathcal{R}$ and 0 otherwise. We set $v^\star(s) = 0$ for $s \in \mathcal{T}$. The goal is to estimate $v^\star(s), s \in \mathcal{I}$.

In settings with large/continuous state space, approximating values in (1) is expensive or infeasible. Moreover, it is hard to guarantee a satisfactory level of estimation error because the considered rare event region $\mathcal{R}$ is not visited frequently enough. In order to frequently visit $\mathcal{R}$ in these high or infinite-dimensional situations, one is motivated to consider the importance sampling [27] to modify the environment transition probability, which achieves variance reduction by approaching a target (zero-variance) distribution and learns to generate rare events efficiently.

IV. Accelerated Policy Evaluation (APE)

In this section, we introduce the proposed APE paradigm. In MDPs, the environment transition probability relates to both agent A's policy π_A and the environment's transition probability $p_E(s'|a_A,s)$, $p_{s,s'} = \sum_{a_A \in \mathcal{A}_A} \pi_A(a_A|s) p_E(s'|a_A,s)$. Therefore, $v^\star(s_0)$ can be reformulated by rolling out trajectories with the newly proposed environment transition probability $p_E^{(n)}$ and agent policy $\pi_A^{(n)}$ at step n as follows:

$$\begin{aligned} v^\star(s_0) &= \mathbb{E}_{p_E^{(n)},\pi_A^{(n)}}\Big[\sum_{n=0} \rho_n(s_n, s_{n+1}, a_A) \cdot r(s_n, s_{n+1})\Big] \\ &= \sum_{a_A \in \mathcal{A}_A} \pi_A^{(0)}(a_A|s_0) \sum_{s_1 \in \mathcal{S}} p_E^{(0)}(s_1|s_0,s_A) \cdot \\ &\quad \rho_0(s_0,s_1,a_A)\Big[r(s_0,s_1) + v^\star(s_1)\Big], \end{aligned} \tag{2}$$

where $\rho_n(s_n, s_{n+1}, a_A)$ is the importance weight at step n,

$$\rho_n(s_n, s_{n+1}, a_A) = \frac{\pi_A(a_A|s_n) \cdot p_{E,gt}(s_{n+1}|a_A, s_n)}{\pi_A^{(n)}(a_A|s_n) \cdot p_E^{(n)}(s_{n+1}|a_A, s_n)}. \tag{3}$$

a) Environment Nature as an Adversary. One key idea in APE is to treat uncertainties in environment transition probabilities $p_E(\cdot|\cdot,\cdot)$ (e.g., the actions of surrounding vehicles on the road or sensor noises in a robotics control task), as stochastic decisions made by the environment nature E with policy $\pi_E(\cdot) : \mathcal{S} \times \mathcal{A}_A \times \mathcal{A}_E \to \mathbb{R}$, where $\mathcal{A}_E$ is the environment nature's action space. Treating the environment nature as an adversary agent is widely used in robust learning and multi-agent RL [28]. It reduces the computation complexity when the selected environment agent E's action space $\mathcal{A}_E$

has a smaller dimension than that of the state space $\mathcal{S}$, and relaxes the assumption of a known environment transition model $f_E : \mathcal{S} \times \mathcal{A}_A \times \mathcal{A}_E \to \mathcal{S}$. Note that we *assume known and adjustable environment policy π_E and unknown f_E in* (5) which lead to a grey-box environment setting.

Formally, at step n, we have the following equations hold

$$p(s_{n+1}|a_{A,n}, s_n) = \pi_E(a_{E,n}|a_{A,n}, s_n) \tag{4}$$

$$s_{n+1} = f_E(s_n, a_{A,n}, a_{E,n}) \tag{5}$$

$$a_{E,n} = f_E^{-1}(s_n, a_{A,n}, s_{n+1}) \tag{6}$$

Moreover, we only focus on importance sampling over $p_E(s_{n+1}|a_A, s_n)$ and leaves $\pi_A^{(n)}(a_A|s_n) = \pi_A(a_A|s_n), \forall n$. Therefore the importance weight in (3) becomes

$$\begin{aligned}\rho_n(s_n, s_{n+1}, a_{A,n}) &= \frac{p_{E,gt}(s_{n+1}|a_{A,n}, s_n)}{p_E^{(n)}(s_{n+1}|a_{A,n}, s_n)} \\ &= \frac{\pi_{E,gt}(a_{E,n}|a_{A,n}, s_n)}{\pi_E^{(n)}(a_{E,n}|a_{A,n}, s_n)}.\end{aligned} \tag{7}$$

b) Updating Rules for $v(s)$ and $\pi_E(a_E|s, a_A)$. Calculating the expectation in (2), which includes the rare event probability, is generally intractable in non-discrete settings. In APE, we use TD method [29] which combines MC simulation and dynamic programming to update the value $v(s)$. More concretely, at the end of step n,

$$\begin{aligned}&v^{(n+1)}(s_n) \leftarrow (1-\alpha_n)v^{(n)}(s_n) + \\ &\sum_{a_A \in \mathcal{A}_A} \rho_n(s_n, s_{n+1}, a_A)\alpha_n[r(s_n, s_{n+1}) + v^{(n)}(s_{n+1})] \quad (8) \\ &\approx (1-\alpha_n)v^{(n)}(s_n) + \alpha_n \cdot \\ &\quad [r(s_n, s_{n+1}) + v^{(n)}(s_{n+1})] \cdot \rho_n(s_n, s_{n+1}, a_{A,n}), \quad (9)\end{aligned}$$

where α_n is a decaying learning rate. To make APE suitable for online learning and get rid of the summation over agent actions, we approximate the importance weight over state transition probability with the one-step importance weight in (7) to get an online stochastic version as in (9). (9) is accurate if the agent policy π_A is deterministic.

Motivated by ASA [9], which accelerates the rare event probability estimation in Markov chains, we design the updating rule for π_E as follows. At the end of step n, the un-normalized policy around the newly sampled action $a_{E,n}$ is derived with the updated value function $v^{(n+1)}$ and the original environment policy $\pi_{E,gt}$

$$\begin{aligned}&\tilde{\pi}_E^{(n+1)}(a_{E,n}|a_{A,n}, s_n) \leftarrow \max\Big(\delta, \\ &\pi_{E,gt}(a_E|a_{A,n}, s_n)\Big(\frac{r(s_n, s_{n+1}) + v^{(n+1)}(s_{n+1})}{v^{(n+1)}(s_n)}\Big)\Big), \quad (10)\end{aligned}$$

where δ is a positive constant to guarantee that the one-step importance weight ρ_n is bounded.

APE accelerates the policy value estimation by learning a properly designed importance policy π_E to reduce the estimation variance. Different from adaptive MC [8] and cross-entropy methods (CEM) [30], APE utilizes a stochastic approximation paradigm and updates state values based on each single transition without waiting for a final outcome (without entering the terminal set).

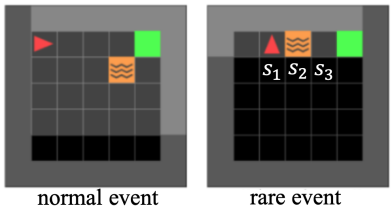


Fig. 2: A minigrid environment with rare events.

TABLE I: Estimation results at convergence in minigrid. We obtain the ground truth rare event probabilities via vanilla MC with a sufficiently large number of episodes.

methods	MC		APE	
metric	mean	std	mean	std
$v(s_1)$ (10^{-3})	3.99	0.38	3.90	0.24
$v(s_2)$ (10^{-3})	3.93	0.47	3.94	0.23
$v(s_3)$ (10^{-3})	4.00	0.50	3.94	0.17
num. of time steps	78300	10600	8884	2595
num. of episodes	19306	2612	552	134

c) Convergence in Tabular Case

Theorem 1: Assume tabular MDP settings with discrete state and action spaces and let $\pi_E^\star$ denote the zero-variance environment policy. If step size α_n in Eq. 8 satisfies $\sum_n \alpha_n = \infty$ and $\sum_n \alpha_n^2 < \infty$, the iterative updating rules (Eq. 8 and Eq. 10) satisfies $v^{(n)} \to v^\star$ and $\pi_E^{(n)} \to \pi_E^\star$.

The proof of Theorem 1 directly follows the structure of Theorem 1 in [9]. By defining the value function only related to state, the difference between the APE value update rule and that in [9] lies in the definition of the importance weight. However, the importance weight-related terms are included in a sequence of random vectors with zero mean and bounded norm, which do not affect the contraction of the remaining part. Based on the Perron-Frobenius theorem, we can show that the remaining part is a contraction operator and thus has a fixed point. Then by applying Theorem 1 and Theorem 3 in [31], we have the convergence of value function v. The convergence of π_E results from the convergence of v and the definition of π_E's updating rule.

A. Accelerated Policy Evaluation with gym-minigrid

We first empirically show that the proposed APE method *without using function approximations* converges to the ground truth rare event probabilities in discrete MDP settings. We experiment in a gym-minigrid [32] environment as shown in Fig. 2. Rare events happen when the agent (the red triangle) fails to reach its destination (the green square), such as getting stuck in non-terminal states or entering the lava (the yellow block). Grids denoted as s_1, s_2 and s_3 are possible initial states of lava. The agent policy π_A is trained with the deep Q learning [33] until convergence. The environment policy π_E controls the policy of the lava.

We compare the efficiency of the proposed APE with vanilla MC for estimating the rare event probability. We define convergence of an estimated value when the estimation deviation in the past 2000 steps is less than 1% of the final estimate. Tab. I shows that when the ground truth rare event

probability is around 4×10^{-3}, APE successfully converges to the true rare event probability. APE also demonstrates high efficiency by requiring one magnitude fewer data and achieves a smaller variance than MC at convergence.

V. Scalable Implementation: APE_scalable

In this section, we introduce how to improve the scalability of APE to handle large discrete or continuous MDP settings. The general idea is to select proper function approximators to represent the value function $v(s)$ and the environment policy π_E. Before doing that, we first establish some notations. The value function approximator $v_\psi(s)$ with parameter ψ takes state s as input and outputs a probability ranging from 0 to 1. The policy approximator $\pi_{E,\theta}$ parameterized with θ takes the state-action concatenation (s, a_A) as input and outputs a probability density on a_E. Therefore, we have $\rho_{n,\theta} = \pi_{E,gt}(a_{E,n}|a_{A,n}, s_n)/\pi^{(n)}_{E,\theta}(a_{E,n}|a_{A,n}, s_n)$. At step n, a data pair $d_n = (s_n, a_{A,n}, a_{E,n}, s_{n+1}, r(s_n, s_{n+1}), \rho_{n,\theta})$ is appended to a history dataset $\mathcal{D}$. Assume $s \in \mathcal{S} \subset \mathbb{R}^{d_s}$, $a_A \in \mathcal{A}_A \subset \mathbb{R}^{d_a}$, and $a_E \in \mathcal{A}_E \subset \mathbb{R}^{d_e}$.

A. Value Function Approximation

We learn $v_\psi(s)$ by minimizing differences between the predicted values and TD targets $v_{\psi,TD}(s_n)$. For data pair d_n, we have

$$\rho_{n,\theta} = \pi_{E,gt}(a_{E,n}|a_{A,n}, s_n)/\pi^{(n)}_{E,\theta}(a_{E,n}|a_{A,n}, s_n), \quad (11)$$

$$v_{\psi,TD}(s_n) = (r(s_n, s_{n+1}) + v_\psi(s_{n+1})) \cdot \rho_{n,\theta}. \quad (12)$$

One approximation approach is representing $v(s)$ with an expressive deep neural network (DNN) with low prediction complexity. However, DNNs are sensitive to imbalanced data and suffer from the catastrophic forgetting problem [34], [35], especially in online learning settings. A competitive model for DNN is the Gaussian Process (GP) regression model. The equivalence between GPs and infinitely wide DNNs is derived in [36]. GPs are data-efficient and flexible in making predictions as non-parametric models but sacrifice the prediction complexity [11]. Considering that the sampled rare events may be drowned out in $\mathcal{D}$, especially in the initial learning phase, we choose to use GP as the function approximator for its data efficiency.

a) Use GP to Represent $v_\psi(\cdot)$. Let the data buffer for training and prediction as $\mathcal{D}_{GP} = \{(s_i, v_{\psi,TD}(s_i))^m\}$. The predicted value given $s^\star \in \mathcal{S}$ is drawn from a distribution

$$p(f|\mathcal{D}, s^\star) = \mathcal{N}(\mathrm{m}(f), \mathrm{cov}(f)). \quad (13)$$

Define $x_i = s_i$ and $y_i = v_{\psi,TD}(s_i)$ and aggregate them into $X \in \mathbb{R}^{d_s \times m}$ and $Y \in \mathbb{R}^{1 \times m}$. The mean function is $\mathrm{m}(f) = K(s^\star, X)[K_i(X, X) + \sigma^2 I]^{-1} Y$ and the covariance matrix is $\mathrm{cov}(f) = K(s^\star, s^\star) - K(s^\star, X)[K(X, X) + \sigma^2 I]^{-1} K_i(X, s^\star)$. σ is the standard deviation of observation noise. The matrix K is fully specified by the kernel function $k(\cdot, \cdot)$, which defines the function smoothness. We use the scaled squared exponential kernel $k(x, x') = w^2 \exp(-\frac{1}{2} \sum_{j=1}^{d_s} w_j (x^j - x'^j)^2)$, where w_j is the reciprocal of the lengthscale of state dimension j, and w is the output scale. $\psi = \{w, w_1, ..., w_{d_s}, \sigma\}$ contents kernel parameters.

We update ψ with gradient descent to minimize the negative log-likelihood for $\mathcal{D}_{GP}$ together with an L_1 regularizer over lengthscale parameters:

$$\psi \leftarrow \psi - \alpha_v \nabla_\psi \Big[\lambda \cdot \sum_j^{d_s} |w_j| + \sum_{(s_i, v_{\psi,TD}(s_i)) \in \mathcal{D}_{GP}} - \log p(v_{\psi,TD}(s_i)|s_i, \mathcal{D}_{GP})\Big], \quad (14)$$

where λ is the regularization penalty, and α_v is the learning rate. Different from standard GP regression with a fixed number of data points and fixed targets, APE learns GP in an online learning manner via streaming collected data pairs [37] as well as in a dynamic programming paradigm with changing TD targets similar to [38].

B. Environment Adversary Policy Approximation

In online setting, the environment's policy updates based on the most recent data pair d_n and newly updated $v_\psi(s)$. The un-normalized target density around the newly sampled action $a_{E,n}$ conditioned on $C_n = [a_{A,n}, s_n]$ is

$$\tilde{\pi}_E(a_{E,n}|C_n) = \pi_E(a_{E,n}|C_n)\Big(\frac{r(s_n, s_{n+1}) + v_\psi(s_{n+1})}{v_\psi(s_n)}\Big). \quad (15)$$

To cope with continuous spaces, we achieve the modification by adding a normal distribution $\mathcal{N}(a_{E,n}, \sigma)$ with σ much smaller than that of the ground truth policy $\pi_E(a_E|x, a_A)$. The target probability density function conditioned on C_n is

$$p_E(a_E|C_n) = (\pi_{E,\theta}(a_E|C_n) + \beta \cdot \mathcal{N}(a_{E,n}, \sigma))/(1 + \beta), \quad (16)$$

where $\beta = \sqrt{2\pi}\sigma(\tilde{\pi}_E(a_{E,n}|C_n) - \pi_{E,\theta}(a_{E,n}|C_n))$.

The function approximation $\pi_{E,\theta}$ should have a low sample complexity to quickly sample actions at each time step. $\pi_{E,\theta}$ is desired to have interpolation/extrapolation ability across conditions to help accelerate the evaluation process by ensuring that two conditional distributions are close if the distance between their conditioned values is close. Considering that the additive modification of the target density function in (15) results in a Gaussian mixture model with an increasing mixture size, $\pi_{E,\theta}$ is also desired to be flexible enough to model multi-mode distributions. In practice, we use the conditional Masked Autoregressive Flow model (cMAF) model [12] to represent $\pi_E(a_E|a_A, s)$ to meet previous desiderata.

a) Use cMAF to Represent $\pi_{E,\theta}$. We obtain a closed-form parametric representation of $\pi_E(a_E|a_A, s)$ using a cMAF. A cMAF is a type conditional Normalizing Flow (cNF) [12], which is a generative model that uses invertible mappings to transform a simple probability distribution into a complex one conditioned on other random variables. Compared with sample-based representation approaches such as MCMC, cNF directly generates one sample by calling one reversible path and has the capacity to model distributions that go beyond single-mode Gaussian distributions.

Algorithm 1 Scalable Accelerated Policy Evaluation

Input: Agent policy to evaluate π_A, ground truth environment policy $\pi_{E,gt}$, warm start policy $\hat{\pi}_E$
Parameter: Total evaluation steps N
Output: $v_\psi(\cdot)$, $\pi_{E,\theta}(\cdot)$

Pretrain $\pi_{E,\theta}(\cdot|a_A, x)$ with target conditional probability $\hat{\pi}_E(\cdot|a_A, s),\ \forall a_A \in \mathcal{A}_A, s \in \mathcal{S}$
for $n = 0$ **to** $N-1$ **do**
  Reset environment
  Initialize empty episode buffer $\mathcal{D}_e$
  repeat
    Sample actions $a_A \sim \pi_A(\cdot|s)$, $a_E \sim \pi_{E,\theta}(\cdot|a_A, s)$
    Execute a_A and a_E, observe s' and $r(s, s')$
    Get one-step importance weight $\rho_{n,\theta}$ with (11)
    Add $d = (s, a_A, a_E, s', r(s, s'), \rho_{n,\theta})$ to $\mathcal{D}$ and $\mathcal{D}_e$
    Update $v_\psi(\cdot)$ with two-timescale based on Algo.2
    Update $\pi_{E,\theta}$ with dense rewards based on Algo.3
    $s \leftarrow s'$
    $n \leftarrow n + 1$
  until episode finish
end for
return $v_\psi(\cdot)$, $\pi_{E,\theta}(\cdot)$

Algorithm 2 Update v_ψ with Two-timescale

Input: new data d, current $\pi_{E,\theta}$
Parameters: v_ψ learning rate α_v, ψ update interval n_ψ
Output: v_ψ

Calculate TD target $v_{\psi,TD}(s)$ of d with (12)
Append $(s, v_{\psi,TD}(s))$ to $\mathcal{D}_{GP}$
if $n \% n_\psi = 0$ **then**
  Train value function v_ψ with gradient descent (14)
end if

Algorithm 3 Update $\pi_{E,\theta}$ with Dense Rewards

Input: new data d, current v_ψ, episode buffer $\mathcal{D}_e$
Parameters: $\pi_{E,\theta}$ learning rate α_π, dense reward discount factor γ_r, momentum coefficient β_θ
Output: $\pi_{E,\theta}$

Append d to $\mathcal{D}_e$
if $r(s, s') = 1$ **then**
  Calculate adversary policy target with (16) based on episode buffer $\mathcal{D}_e$ and dense rewards
  Get updated θ' of $\pi_{E,\theta}(\cdot)$ with gradient descent (18)
  $\theta \leftarrow \beta_\theta \theta' + (1 - \beta_\theta)\theta$
end if

We denote the generative function in cNFs as $g_\theta : z \to a_E$, and the differentiable normalizing function as $f_\theta = g_\theta^{-1}$. The base distribution p_Z of hidden variable Z is easy to sample from, such as a normal distribution or a uniform distribution. At step n, we hope to minimize the KL divergence $L_\theta^{(n)}$ between the target and the cNF distribution.

$$\begin{aligned} L_\theta^{(n)} =& \mathbb{E}_{\pi_{E,\theta}(a_E|C_n)}[\log \pi_{E,\theta}(a_E|C_n) - \log p_E(a_E|C_n)] \\ =& \mathbb{E}_{p_Z(z|C_n)}[\log \pi_{E,\theta}(g_\theta(z)|C_n) - \log p_E(g_\theta(z)|C_n)] \\ =& \mathbb{E}_{p_Z(z|C_n)}[\log p_Z(z|C_n) - \log|\det \partial_z g_\theta(z)| \\ & - \log p_E(g_\theta(z)|C_n)]. \end{aligned} \tag{17}$$

We update θ via gradient descent. We first sample from the simple base distribution $p_Z(z|C_n)$ and pass the sampled $z_i, i = [M]$ to the flow to get samples in target domain $a_{E,i} = g_\theta(z_i), i = [M]$. The KL divergence $L_\theta^{(n)}$ is then approximated with the sample mean. With learning rate α_π,

$$\begin{aligned} \theta \leftarrow& \theta - \alpha_\pi \nabla_\theta \sum_{i=1}^{M} \Big[\log p_Z(z_i|C_n) \\ & - \log|\det \partial_z g_\theta(z_i)| - \log p_E(g_\theta(z_i)|C_n) \Big]. \end{aligned} \tag{18}$$

C. Algorithm Description

We present the APE_scalable algorithm in Algo. 1 and summarize the algorithm highlights as follows:

Warm start. We initialize the environment policy π_E with a warm start policy $\hat{\pi}_E$ when the rare event probability $v^\star(s_0)$ is pretty small. The design of the warm start distribution may leverage domain knowledge. One possible choice is using large variances.

GP two-timescale update rules. We develop a two-timescale updating scheme for GP dynamic programming. We update data pairs more frequently than the GP kernel parameters to improve training stability as in Algo. 2. We store representative data as in sparse GP [39] and early-stop appending data when sufficient representative data points.

cMAF update with dense rewards. In practice, purely updating π_θ in an online manner raises several issues. First, it causes that cMAF easily over-fits to the latest target distribution across conditions (diminishing the effect of conditions). Second, it may induce training instability due to a quite sparse reward signal $r(s_{n-1}, s_n)$. Third, it requires high computational complexity for frequently updating cMAF. Therefore, we propagate the reward signal to the whole episode $\hat{r}(s_{n-1}, s_n) = \gamma_r^{\tau-n} r(s_{\tau-1}, s_\tau), n = 1, \ldots, \tau$, where γ_r is the reward discount factor. τ is the episode termination step. To improve training stability, we further add momentum to the parameter θ. We early-stop training when achieving satisfactory sampled rare event rate.

VI. Experiments

In this section, We empirically show that the proposed APE_scalable method *with function approximations* outperforms baselines by estimating the rare event probability with smaller biases and orders of magnitude fewer samples.

A. Baselines

We compare the proposed APE_scalable with three baselines: *APE_discrete*, *CEM_discrete* and vanilla *MC*. APE_discrete and CEM_discrete discretize each dimension of s and a_E with a small slot interval, and use dictionaries to store the rare event probability estimate and the environment policy π_E at each slot. Note that vanilla MC does not require π_E for estimation or store dictionaries. We get the ground truth rare event probability $v^\star(s_0)$ via MC with

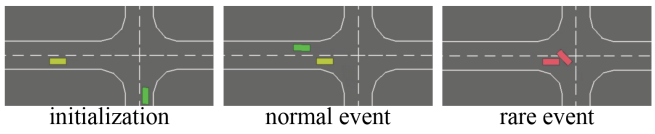


Fig. 3: The **Intersection** environment with rare events.

Fig. 4: The **LunarLander** environment with rare events.

sufficiently large number of episodes. Let $\hat{v}(s_0)$ denote the estimation and $\bar{v}(s_0)$ denote the sampled rare-event rate along estimation process. We evaluate the estimation error based on bias $Bias(s_0) = \mathbb{E}[\hat{v}(s_0)] - v^\star(s_0)$, standard deviation $Std(s_0) = \sqrt{\mathbb{E}[(\mathbb{E}[\hat{v}(s_0)] - \hat{v}(s_0))^2]}$, and risk $MSE = \sqrt{Bias^2(s_0) + Std^2(s_0)}$.

B. Environments

Intersection The agent to evaluate (the green vehicle) aims to perform a collision-free unprotected left turn in a two-lane intersection with one surrounding vehicle (the yellow vehicle). A rare event is a crash, and the adversary policy is the surrounding vehicle's policy. We hope to evaluate a fixed deterministic policy π_A, which follows the lane with constant velocity. Note that APE is also applicable to stochastic policies. We build two intersection scenarios based on highway-env [13]: `Int-v0` with $v^\star(s_0) = 6.34 \times 10^{-2}$, and `Int-v1` with $v^\star(s_0) = 1.70 \times 10^{-6}$. APE can handle more than two agents by treating all the other agents as part of the environment [40]. In `Int-v0`, we initialize all methods with the $\pi_{E,gt}$ and present experiment results in Fig. 5 (a). In `Int-v1`, we initialize APE_scalable, APE_discrete and CEM_discrete with a slightly skewed distribution (as shown in the first column in Fig. 6) with a larger variance than the ground truth $\pi_{E,gt}$ to sample rare events at the initial stage. We present experiment results of intersections in Fig. 5 (b).

Lunarlander We hope to evaluate a heuristic policy by OpenAI which aims to land a lunar lander on a fixed landing pad in `LunarLander` [14] with $v^\star(s_0) = 1.30 \times 10^{-3}$. The rare event is defined as if the lander crashes, flies too far away from the landing pad or the episode exceeds the maximum length. The environment agent is an adversary that controls perturbations over actions to execute to mimic controller errors. This example provides a clue for selecting factors controlled by the environment adversary, and the experimenters may draw inspiration from robust learning and add uncertainty/noise to components of MDPs [41], [42], [43]. In `LunarLander`, we initialize all methods with the $\pi_{E,gt}$ and present experiment results in Fig. 5 (c).

C. Discussion about Evaluation Performance

Although APE_discrete approaches the true rare event probability in `Int-v0`, it has a larger estimate variance than APE_scalable. In `Int-v1` and `LunarLander`, APE_scalable successfully converges to the true value while APE_discrete does not. The smaller variance and faster convergence of APE_scalable compared with APE_discrete across environments indicate that the discretization of continuous tasks drops the environment or action structure information and leads to biased evaluation results.

Interestingly, CEM_discrete achieves higher rare event sample rates than APE_scalable in `Int-v0` and `Int-v1`, but heavily underestimates the true rare event probability. We conjecture that CEM_discrete fails to explore enough and its environment adversary policy π_E only concentrates on a subset of failure modes. While in `LunarLander`, CEM_discrete increases the sampled rare event rate but is less efficient compared with APE_scalable.

Both APE_scalable and vanilla MC approach the true value in `Int-v0` and `LunarLander`, since the rare event probability is relatively large. Despite of that, APE still outperforms MC in terms of the variance of probability estimation in `LunarLander`. But in `Int-v1`, APE_scalable converges within 3,000 steps (around 1,000 episodes) while MC may require $1,762,193 = \log(0.05)/\log(1 - v^\star(s_0))$ episodes until having 95% confidence interval. Note that we omit the evaluation curve using MC in `Int-v1` since MC hardly samples even one rare event within 3,000 steps. It shows that APE_scalable dominates MC in estimation accuracy when the rare event probability is small.

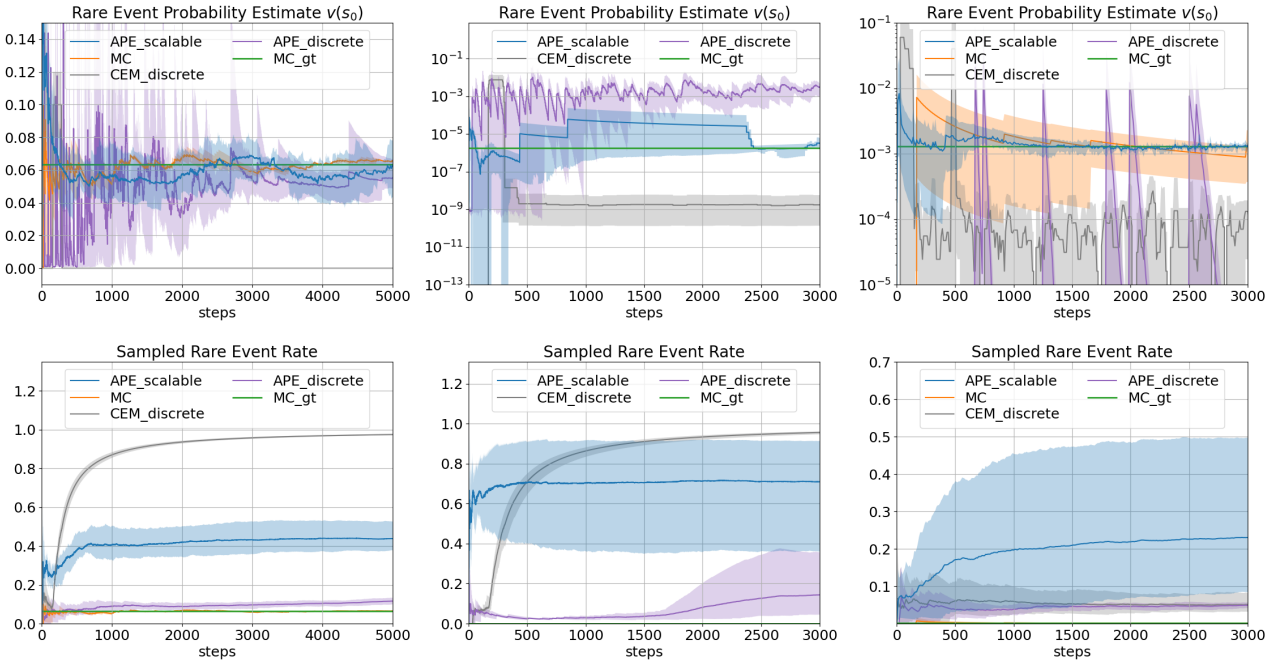
D. Discussion about Different Metrics

From Tab. II, APE_scalable has the smallest bias across three environments compared to baselines but not the smallest variances or MSEs. Note that although in `Int-v0` and `Int-v1` CEM_discrete has near-zero variance, it heavily under-estimates the rare event probability as shown in Fig. 5. Similarly, in `LunarLander`, APE_discrete underestimates the rare event probability by one magnitude but has a smaller variance than other baselines. Such under-estimation phenomena are risky and not desirable in the rare event setting since they may cause catastrophic consequences by deploying overly trusted RL agents.

E. Discussion about Environment Adversary π_E

In `Int-v0`, the sampled rare event rate using APE_scalable (0.44) is 7.5 times greater than MC (0.064), and is 3.3 times greater than APE_discrete (0.12). In `LunarLander`, the final sampled rare event rate using APE_scalable is more than 100 times greater than MC. Interestingly, in the more challenging `Int-v1`, APE_scalable generates much more rare events with sampled rare event rate 0.75, which is 441,176 times greater than MC (1.70×10^{-6}) and 4.7 times greater than APE_discrete (0.16). Together with the smaller estimation biases discussed before, the learnt policy π_E with APE_scalable is closer to the zero-variance distribution, and APE_scalable performs better with smaller $v^\star(s_0)$.

We notice that the environment adversary policies at convergence with different random seeds hold a similar interpretable behavior mode, one example of which is visualized in Fig. 6 for `Int-v1`. We see that the learned adversary policy π_E tends to accelerate (being aggressive) at the beginning of



(a) `Int-v0` : $v^\star(s_0) = 6.34 \times 10^{-2}$ (b) `Int-v1` : $v^\star(s_0) = 1.70 \times 10^{-6}$ (c) `LunarLander` : $v^\star(s_0) = 1.30 \times 10^{-3}$

Fig. 5: Evaluation results in `Int-v0`, `Int-v1` and `LunarLander`. Each line is run with 5 random seeds. The shaded area has lower and upper bound corresponding to 5% and 95% quantile. APE_scalable performs well in moderate high-dimensional tasks with higher estimate accuracies and sampled rare event rates compared with baselines. The performances of all the methods are measured with estimate errors and sampled rare event rates.

TABLE II: Numerical evaluation results in `Int-v0`, `Int-v1` and `LunarLander`. Each number is evaluated with 5 random seeds. The best performance (the smallest number in each column) is highlighted in bold.

Environment	Int-v0 (Unit 10^{-2})			Int-v1 (Unit 10^{-6})			LunarLander (Unit 10^{-3})		
Metrics	*Bias*	*Std*	*MSE*	*Bias*	*Std*	*MSE*	*Bias*	*Std*	*MSE*
APE_scalable	**-0.07**	6.27	6.27	**1.08**	2.88	3.07	**-0.02**	1.28	1.28
APE_discrete	-0.75	0.47	**0.88**	2447.75	149.44	2452.31	-1.30	**0.00**	1.16
CEM_discrete	-6.34	**0.00**	6.34	-1.70	**0.00**	**1.70**	-1.17	0.07	1.27
MC	0.23	6.57	6.57	-1.70	**0.00**	**1.70**	0.08	0.42	0.43

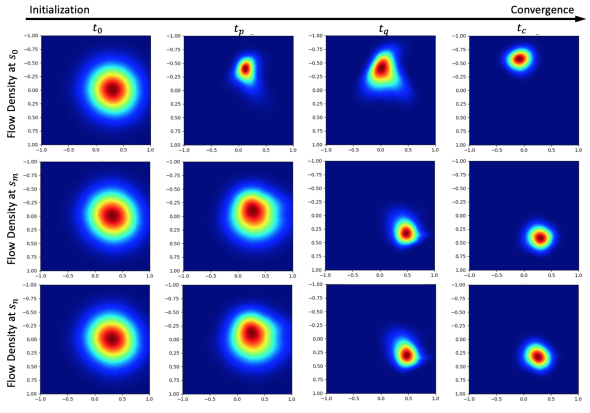


Fig. 6: The evolution of π_E in `Int-v1` conditioned at the initial state s_0, a random intermediate state s_m and a state before collision s_n. The time steps $t_0 < t_p < t_q < t_c$.

each episode and keep the original behavior $\pi_{E,gt}$ (being conservative) in the remaining steps.

VII. Conclusion

Motivated by the limitations of existing policy evaluation methods facing rare events, we propose APE, which scales to complex tasks by drawing from powerful function approximators and explores rare events caused by sequential interactions via learning an adversary with adaptive importance sampling. We demonstrate the effectiveness of APE with its orders of magnitude sample savings to achieve convergence. APE provides a fundamental tool to allow the evaluation, and subsequent deployment, of intelligent agents in safety-critical systems that are otherwise computationally prohibitive. It also increases policy interpretability by uncovering agent behaviors in extreme cases (rare events).

The scalability of APE in evaluating rare events in sequential environments opens up two important future directions. First is the training of rare-event-aware intelligent agents. This requires a minimax training mechanism to optimize the agent's strategies against the avoidance of rare events, the latter captured via the environment adversary learned from adaptive importance sampling as in this paper. Second is more powerful update methods for the value function and importance distribution, such as incorporating (Bayesian) uncertainty estimates in the GP regression model and using multi-step TD.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support from the National Science Foundation (under grants IIS-1849280 and IIS-1849304), and unrestricted research grant from the Toyota Motor North America. The ideas, opinions and conclusions presented in this paper are solely those of the authors.

REFERENCES

[1] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.

[2] B. Chen, Z. Liu, J. Zhu, M. Xu, W. Ding, L. Li, and D. Zhao, "Context-aware safe reinforcement learning for non-stationary environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 10 689–10 695.

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

[4] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa, and C. S. Pan, "Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 595–607, 2017.

[5] A. Corso, R. Lee, and M. J. Kochenderfer, "Scalable autonomous vehicle safety validation through dynamic programming and scene decomposition," *arXiv preprint arXiv:2004.06801*, 2020.

[6] J. Uesato, A. Kumar, C. Szepesvari, T. Erez, A. Ruderman, K. Anderson, N. Heess, P. Kohli, *et al.*, "Rigorous agent evaluation: An adversarial approach to uncover catastrophic failures," *arXiv preprint arXiv:1812.01647*, 2018.

[7] M. O'Kelly, A. Sinha, H. Namkoong, R. Tedrake, and J. C. Duchi, "Scalable end-to-end autonomous vehicle testing via rare-event simulation," in *Advances in Neural Information Processing Systems*, 2018, pp. 9827–9838.

[8] P. Y. Desai and P. W. Glynn, "A markov chain perspective on adaptive monte carlo algorithms," in *Proceeding of the 2001 Winter Simulation Conference (Cat. No. 01CH37304)*, vol. 1. IEEE, 2001, pp. 379–384.

[9] T. I. Ahamed, V. S. Borkar, and S. Juneja, "Adaptive importance sampling technique for markov chains using stochastic approximation," *Operations Research*, vol. 54, no. 3, pp. 489–504, 2006.

[10] M. Dennis, N. Jaques, E. Vinitsky, A. Bayen, S. Russell, A. Critch, and S. Levine, "Emergent complexity and zero-shot transfer via unsupervised environment design," *arXiv preprint arXiv:2012.02096*, 2020.

[11] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.

[12] G. Papamakarios, T. Pavlakou, and I. Murray, "Masked autoregressive flow for density estimation," in *Advances in Neural Information Processing Systems*, 2017, pp. 2338–2347.

[13] E. Leurent, "An environment for autonomous driving decision-making," https://github.com/eleurent/highway-env, 2018.

[14] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[15] D. Zhao, X. Huang, H. Peng, H. Lam, and D. J. LeBlanc, "Accelerated evaluation of automated vehicles in car-following maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 733–744, 2017.

[16] M. Arief, Z. Huang, G. K. S. Kumar, Y. Bai, S. He, W. Ding, H. Lam, and D. Zhao, "Deep probabilistic accelerated evaluation: A certifiable rare-event simulation methodology for black-box autonomy," *arXiv preprint arXiv:2006.15722*, 2020.

[17] F. Cérou and A. Guyader, "Adaptive multilevel splitting for rare event analysis," *Stochastic Analysis and Applications*, vol. 25, no. 2, pp. 417–443, 2007.

[18] W. Ding, B. Chen, B. Li, K. J. Eun, and D. Zhao, "Multimodal safety-critical scenarios generation for decision-making algorithms evaluation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1551–1558, 2021.

[19] A. Corso, R. J. Moss, M. Koren, R. Lee, and M. J. Kochenderfer, "A survey of algorithms for black-box safety validation," *arXiv preprint arXiv:2005.02979*, 2020.

[20] R. Werpachowski, A. György, and C. Szepesvári, "Detecting overfitting via adversarial examples," *arXiv preprint arXiv:1903.02380*, 2019.

[21] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1625–1634.

[22] T. Xie, Y. Ma, and Y.-X. Wang, "Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling," in *Advances in Neural Information Processing Systems*, 2019, pp. 9668–9678.

[23] R. Munos, T. Stepleton, A. Harutyunyan, and M. Bellemare, "Safe and efficient off-policy reinforcement learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 1054–1062.

[24] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, *et al.*, "Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures," *arXiv preprint arXiv:1802.01561*, 2018.

[25] M. Schlegel, W. Chung, D. Graves, J. Qian, and M. White, "Importance resampling for off-policy prediction," in *Advances in Neural Information Processing Systems*, 2019, pp. 1799–1809.

[26] R. Cotta, M. Hu, D. Jiang, and P. Liao, "Off-policy evaluation of probabilistic identity data in lookalike modeling," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019, pp. 483–491.

[27] S. Asmussen and P. W. Glynn, *Stochastic simulation: algorithms and analysis*. Springer Science & Business Media, 2007, vol. 57.

[28] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in *International Conference on Machine Learning*, 2017, pp. 2817–2826.

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

[30] P.-T. de Boer, "Analysis and efficient simulation of queueing models of telecommunication systems," *Doctoral dissertation. (Centre for Telematics and Information Technology University of Twente)*, 2000.

[31] J. N. Tsitsiklis, "Asynchronous stochastic approximation and q-learning," *Machine learning*, vol. 16, no. 3, pp. 185–202, 1994.

[32] M. Chevalier-Boisvert, L. Willems, and S. Pal, "Minimalistic gridworld environment for openai gym," https://github.com/maximecb/gym-minigrid, 2018.

[33] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[34] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.

[35] A. Chrysakis and M.-F. Moens, "Online continual learning from imbalanced data," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1952–1961.

[36] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, "Deep neural networks as gaussian processes," *arXiv preprint arXiv:1711.00165*, 2017.

[37] D. Nguyen-Tuong, J. Peters, and M. Seeger, "Local gaussian process regression for real time online model learning," *Advances in neural information processing systems*, vol. 21, pp. 1193–1200, 2008.

[38] M. P. Deisenroth, J. Peters, and C. E. Rasmussen, "Approximate dynamic programming with gaussian processes," in *2008 American Control Conference*. IEEE, 2008, pp. 4480–4485.

[39] D. Tran, R. Ranganath, and D. M. Blei, "The variational gaussian process," *arXiv preprint arXiv:1511.06499*, 2015.

[40] S. V. Albrecht and P. Stone, "Autonomous agents modelling other agents: A comprehensive survey and open problems," *Artificial Intelligence*, vol. 258, pp. 66–95, 2018.

[41] C. Tessler, Y. Efroni, and S. Mannor, "Action robust reinforcement learning and applications in continuous control," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6215–6224.

[42] H. Zhang, H. Chen, C. Xiao, B. Li, M. Liu, D. Boning, and C.-J. Hsieh, "Robust deep reinforcement learning against adversarial perturbations on state observations," *arXiv preprint arXiv:2003.08938*, 2020.

[43] H. Xu and S. Mannor, "Distributionally robust markov decision processes," *Mathematics of Operations Research*, vol. 37, no. 2, pp. 288–300, 2012.