Accelerated Policy Evaluation: Learning Adversarial Environments with Adaptive Importance Sampling

Mengdi Xu\textsuperscript{1}, Peide Huang\textsuperscript{1}, Fengpei Li\textsuperscript{2,3}, Jiacheng Zhu\textsuperscript{1}, Xuewei Qi\textsuperscript{4}, Kentaro Oguchi\textsuperscript{4}, Zhiyuan Huang\textsuperscript{5}, Henry Lam\textsuperscript{2}, Ding Zhao\textsuperscript{1}

1. Carnegie Mellon University 2. Columbia University 3. Morgan Stanley AI CoE
4. Toyota Motor North America R&amp;D 5. Tongji University

Abstract

The evaluation of rare but high-stakes events remains one of the main difficulties in obtaining reliable policies from intelligent agents, especially in large or continuous state/action spaces where limited scalability enforces the use of 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. In this paper, we propose 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 properties of proposed algorithms under suitable regularity conditions. Our empirical studies show that APE estimates rare event probability with a smaller variance while only using orders of magnitude fewer samples compared to baseline methods in both multi-agent and single-agent environments.

1 Introduction

Obtaining an accurate policy evaluation of an intelligent agent is crucial to generate reliable policies before deployment. However, many rare but high-stake events exist in safety-critical systems, which has long been a difficult issue for safety evaluation in autonomous driving [33], finance [27], medicine [58], and healthcare [38]. Unsurprisingly, inaccurate estimations of rare event probability under these contexts could result in false optimism or even catastrophic consequences.

The de facto standard evaluation method uses the vanilla Monte Carlo that, in a multitude of practical settings of interest, requires a prohibitively large number of testing before the evaluation is deemed statistically valid [56, 7, 5]. For example, it is shown in [33] that self-driving cars technically need to clock in hundreds of billions of miles to demonstrate their safety. Motivated by situations when testing is high-stakes and large-volume testing is expensive or time-consuming, a rich literature is developed to accelerate the estimation of rare event probability in the policy evaluation metric [77, 14]. However, existing literature typically either (a) fails to exploit the sequential, interactive nature of tasks and only finds specific initial conditions that cause failures [66, 48, 21] or (b) focuses on finite state/action spaces [30, 20, 2] and suffers from the curse of dimensionality. As intelligent agents are gradually deployed in real-world applications, it is of great interest to develop a rigorous, scalable, and affordable policy evaluation method.

In this paper, we propose the Accelerated Policy Evaluation (APE) method to expedite the policy evaluation, especially for rare event probabilities in complex sequential decision-making tasks. APE accelerates the policy evaluation by learning to reduce the estimation variance through adaptive
importance sampling. Our proposed method addresses the aforementioned limitations of ignoring the sequential nature of MDP and poor scalability as explained below.

First, to exploit the sequential and interactive nature of tasks, APE finds rare events directly from sequential interactions with the agent by treating the environment nature as an adversarial agent. The stochastic environment adversary policy $\pi_E$ is analogous to the uncertainty in the environment transition model, e.g., the actions of surrounding vehicles on the road or sensor noises in a robotics control task. Instead of searching for worst-case scenarios as in adversarial attacks [25], APE aims to directly find those rare but possible (failure) cases when the agent (policy $\pi_A$) interacts with the environment (policy $\pi_{E,gt}$). Notably, APE utilize importance weights for unbiased estimation similarly to adversarial testing [68]. APE updates the environment policy $\pi_E$ towards a zero-variance importance distribution guided by updated rare event probability estimates, based on an adaptive importance sampling scheme [2]. Moreover, different from adaptive Monte Carlo [20] and cross-entropy methods [18, 17], APE iteratively estimates the rare event probability and updates $\pi_E$ in a stochastic approximation paradigm without needing to wait till episode termination.

Second, to improve scalability in large discrete or continuous settings, APE incorporates two function approximators: A Gaussian process (GP) regression model [53], which is data-efficient and flexible, to encode rare event probabilities, and a conditional normalizing flow model [49] to represent the environment policy conditioned on both state $s$ and agent action $a_A$. Considering that the GP regression model makes predictions based on stored input-target pairs and processes a small number of model parameters, we develop a two-timescale updating scheme for GP dynamic programming, with data pairs updated more frequently than the GP kernel parameters. The conditional normalizing flow model [49], which is capable of modelling multi-modal distributions, is suitable for the incremental and iterative update required in APE. We update model parameters of both function approximators with gradient descent. Fig. 1 summarizes the framework of APE. We further establish the convergence properties of APE under suitable regularity conditions. Our empirical studies in a minigrid navigation problem [10], driving intersection scenarios [37] and a lunar lander task [6] show that APE can estimate the rare event probability with a smaller variance using orders of magnitude fewer samples than baselines.

2 Related work

Motivated by the rare nature of safety-critical events in lab and field tests [76, 57], there is a rich literature on rare event probability estimation [7, 14, 66, 54] devoting to maximize the sampling efficiency, among which importance sampling [56, 48, 55, 4] and multi-level splitting [29, 67, 8] stand as two powerhouses. However, few literature addresses the issue under the setting of Markov decision process (MDP). The methodology in [66] proposes an adversarial evaluation which adjusts the environment setting at the beginning of each episode while ignoring the intermediate interactive steps. In contrast, APE is designed to leverage the sequential nature of these tasks and hence is more suitable for reinforcement learning (RL) settings. The closest method to APE is adaptive stochastic approximation (ASA) [2] for Markov chains. 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 [14] in that both utilize a dynamic programming paradigm. However, [14] relies on discretizing and decomposing the simulation scene to make it scalable.

A natural extension of policy evaluation in the presence of rare events uses RL algorithms. REASA [26] models rare events as the intrinsic uncertainty of the transition function independent of the agent’s policy, and reduces variance in value estimation with importance sampling. OFFER [13] proposes to interleave the policy optimization with the environment variable optimization gradient step to achieve variance reduction in policy gradient estimation. With rare events defined as constraint violations,
estimated rare event probability (constraint violation rate) serves as a criterion in constrained/safe RL
related problems \cite{60,63,40,9,73}.

Our work is within the regime of safe validation \cite{15} and specifically focuses on finding the
distribution of failure-causing disturbances and estimating the failure rate. Adversarial evaluation
\cite{60,68}, mainly motivated by attacks to RL agents \cite{31,39}, also aims to find failure cases but in a
distribution-free manner. Our updating rule involving dynamic programming and importance weights
over changing policies shares a similar form with off-policy evaluation \cite{71} methods including
Retrace \cite{44}, V-trace \cite{24}, and importance resampling \cite{59}. Concretely, the environment’s ground
truth policy \( \pi_{E,gt} \) and the proposed importance policy \( \pi_E \) in APE is analogous to the evaluation
policy and behavior policy in off-policy evaluation \cite{32,45,16}, respectively.

3 Accelerated policy evaluation

3.1 Background

A general MDP consists of a tuple \((S, A, p, r)\), which denotes the state space, action space, the
(Markovian) transition probability matrix or kernel density (w.r.t say, Lebesgue measure \( \lambda \) on \( S \))
\((p(y|x,a) : x, y \in S, a \in A)\) and the one-step, non-negative reward function associated with each
state transition \((r(x,y) \geq 0, x, y \in S)\). Beyond the general setting, we specifically consider a MDP
where the agent interacts with a stochastic environment.

In particular, the state space \( S \) is a partition of terminal states \( T \) and interior states \( I \). We assume \( T \)
is reachable from any states in \( I \) for all policies under consideration and \( \tau := \inf\{ n \geq 0 : s_n \in T \} \)
is finite almost surely. The rare event set is denoted by \( R \) and \( R \subset T \). The goal is to evaluate the
rare event probability according to a given performance measure in a sequential decision-making
task. Specifically, the rare event probability (or the probability of hitting \( R \)) before \( \tau \) starting
from an initial state \( s_0 \in S \cap R^c \) can be expressed as the expected undiscounted total reward until
termination:

\[
v^*(s_0) := \mathbb{E}_{p_{E,gt}, \pi_A} \left[ \sum_{n=1}^{\tau} r(s_{n-1}, s_n) \right] \tag{1}
\]

\[
= \sum_{a_A \in A_A} \pi_A(a_A|s_0) \sum_{s_1 \in S} p_{E,gt}(s_1|s_0, a_A) \left[ r(s_0, s_1) + v^*(s_1) \right] \tag{2}
\]

where \( \pi_A \) (the subscript \( A \) denotes the agent) is the given Markov policy (typically obtained through
training). \( p_{E,gt} \) is the original transition probability (we treat it as the environment later, hence
the subscript \( E \) with \( gt \) meaning "ground truth"). Furthermore, \( a_A \sim \pi_A \) is the agent action and
\( A_A \) is the agent action space. The one-step reward \( r(\cdot, \cdot) \) could be a general non-negative function.
Specifically, for the rare event probability estimation, it is an indicator function with \( r(x,y) = 1 \) if
\( y \in R \) and \( 0 \) otherwise. We set \( v^*(s) = 0 \) for \( s \in T \). The goal is to estimate \( (v^*(s), s \in I) \). Note
that \( (2) \) looks different than any other policy evaluation problem. However, in settings with
large/continuous state space, (approximate) value iterations are not only expensive or infeasible, but
also cannot guarantee a satisfactory level of estimation error because the considered rare event region
\( R \) is not visited frequently enough. In order to frequently visit \( R \) in these high or infinite-dimensional
situations, one is motivated to consider importance sampling which achieves variance reduction by
approaching a target (zero-variance) distribution and learns to generate rare events efficiently.

3.2 Methodology

We briefly introduce importance sampling \cite{5} and adaptive importance sampling for Markov chain
\cite{2}. Suppose we want to estimate \( \nu = \mathbb{E}_P[X] \) for some non-negative random variable \( X \) under
\((\Omega, P, F)\), and we are free to choose any sampling measure \( Q \) that is absolute continuous with \( P \) with
Radon–Nikodym derivative \( dP/dQ \). It can be seen that, under any such \( Q \), we have
\( \nu = \mathbb{E}_Q[X \frac{dP}{dQ}] \) and one can leverage this degree of freedom to minimize the variance, or equivalently, the second
moment \( \mathbb{E}_Q[X^2(\frac{dP}{dQ})^2] \). It follows from the Cauchy–Schwarz inequality that

\[
\mathbb{E}_Q[X^2(\frac{dP}{dQ})^2] \geq (\mathbb{E}_Q[X \frac{dP}{dQ}])^2 = \nu^2.
\]
and the minimal-variance (or zero-variance in this scenario) sampling $Q^*$ satisfies $dQ^* \propto XdP$, or equivalently, $dQ^* = \frac{1}{\nu} dP$. As noted in [5], even though the optimal $Q^*$ involves the unknown quantity of interest $\nu$ and is hence unattainable, it still offers valuable insight into good practical choices of $Q$. Now, in (1), since $\pi_A$ is fixed, we can view the process as a Markov chain instead of a MDP. Consequently, using Bellman equation, we can rewrite (1) as

$$v^*(s_0) = \mathbb{E}[r(s_0, S_1) + v^*(S_1)].$$

(3)

Let $p(x, y)$ be the transition probability for the Markov chain. Then, the form of (3), combined with the previous discussion, would lead us to the zero-variance sampling distribution for $v^*(s_0)$:

$$p^*(s_0, y) = \frac{p(s_0, y)(r(s_0, y) + v^*(y))}{v^*(s_0)}.$$  

(4)

The adaptive importance sampling method in [2] thus continues to update the sampling distribution $p^{(n)}$ towards (4) with each update on $v^{(n)}$, the $n$-th step approximation for $v^*$ ($\alpha$ is the learning rate):

$$v^{(n)}(s_{n-1}) = (1-\alpha)v^{(n-1)}(s_{n-1}) + \alpha(r(s_{n-1}, s_n) + v^{(n-1)}(s_n))\frac{p(s_{n-1}, s_n)}{p^{(n-1)}(s_{n-1}, s_n)}.$$  

while the update for $p^{(n)}$ tries to mimic the form of (4) by letting

$$p^{(n)}(s_{n-1}, s_n) = \frac{p^{(n)}(s_{n-1}, s_n)(r(s_{n-1}, s_n) + v^n(s_n))}{v^n(s_{n-1})},$$

with additional clipping and normalization scaling. Finally, in our MDP setting, $v^*(s_0)$ can be reformulated by rolling out trajectories with the newly proposed environment transition probability $p^{(n)}_E$ and agent policy $\pi^{(n)}_A$ at step $n$ as follows:

$$v^*(s_0) = \mathbb{E}_{p^{(n)}_E, \pi^{(n)}_A}\left[ \sum_{n=1}^{\infty} \rho_n(s_n, s_{n+1}, a_A) \cdot r(s_{n-1}, s_n) \right]$$

(5)

$$= \sum_{a_A \in A_A} \pi^{(n)}_A(a_A|s_0) \sum_{s_1 \in S} p^{(n)}_E(s_1|s_0, a_A) \cdot \rho_n(s_n, s_{n+1}, a_A) \left[ r(s_0, s_1) + v^*(s_1) \right],$$

(6)

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

$$\rho_n(s_n, s_{n+1}, a_A) = \frac{\pi_A(a_A|s_n) \cdot p_{E,gt}(s_{n+1}|a_A, s_n)}{\pi^{(n)}_A(a_A|s_n) \cdot p^{(n)}_E(s_{n+1}|a_A, s_n)}.$$  

(7)

In this work, we only focus on importance sampling over environment transition probability $p_{E}(s_{n+1}|a_A, s_n)$ and thus $\pi^{(n)}_A(a_A|s_n) = \pi_A(a_A|s_n), \forall n$ in (7). In practice, we assume a modifiable testing environment/platform (such as a simulator with adjustable disturbance/noise) and a black-box agent policy, which are realistic in third-party evaluation cases.

We first introduce APE in Sec. 3.3 which simultaneously estimates the rare event probability and learns towards a zero-variance importance distribution of environment transition probability for sampling rare events. We then discuss how to improve scalability by incorporating function approximators to represent the probability values and importance distributions in Sec. 3.4.

### 3.3 Accelerated policy evaluation

**Environment nature as an adversarial agent** One key idea in APE is to treat uncertainties in environment transition probabilities as stochastic decisions made by the environment nature. Treating the environment nature as an agent is widely used in robust learning, multi-agent RL and game theoretic approaches [24][31][50]. It reduces the computation complexity when the selected environment agent $E'$s action space $A_E$ has a smaller dimension than that of the state space $X$, and relaxes the assumption of a known environment transition model. The uncertainty in the environment transition probability transfers to the stochasticity of policy $\pi_E$. The environment then evolves based on a deterministic mapping $f_E$. Formally, at step $n$,

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

(8)

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

(9)

$$a_{E,n} = f_{E}^{-1}(s_n, a_{A,n}, s_{n+1})$$  

(10)
Therefore the importance weight for step $n$ becomes
\[
\rho_n(s_n, s_{n+1}, a_{A,n}) = \frac{p_{E,gt}(s_{n+1}|a_{A,n}, s_n)}{p_E^n(s_{n+1}|a_{A,n}, s_n)} = \frac{\pi_{E,gt}(a_{E,n}|a_{A,n}, s_n)}{\pi_E^n(a_{E,n}|a_{A,n}, s_n)}. \tag{11}
\]

**Updating rules for $v(s)$ and $\pi_E(a_E|s, a_A)$** Calculating the expectation in (2) or (9), which includes the rare event probability, is generally intractable in non-discrete settings. In APE, we use TD method [61] which combines Monte Carlo simulation and dynamic programming for updating the value $v(s)$. More concretely, at the end of step $n$,
\[
v^{(n+1)}(s_n) \leftarrow (1 - \alpha)v^{(n)}(s_n) + \alpha[r(s_n, s_{n+1}) + v^{(n)}(s_{n+1})] \sum_{a_{A}\in A_{A}} \rho_n(s_n, s_{n+1}, a_A) \tag{12}
\approx (1 - \alpha)v^{(n)}(s_n) + \alpha[r(s_n, s_{n+1}) + v^{(n)}(s_{n+1})] \cdot \rho_n(s_n, s_{n+1}, a_{A,n}), \tag{13}
\]
where $\alpha$ 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 (11) to get an online stochastic version as in (13). Note that the approximation (13) becomes exact if the agent policy $\pi_A$ is deterministic.

The updating rule for environment policy $\pi_E$ is determined as follows. At the end of step $n$, the un-normalized importance policy that targets around the newly sampled action $a_{E,n}$ is derived with the updated value function $v^{(n+1)}$ and the original environment policy $\pi_{E,gt}$
\[
\pi_E^{(n+1)}(a_{E,n}|a_{A,n}, s_n) \leftarrow \max(\delta, \pi_{E,gt}(a_E|a_{A,n}, s_n)\left(\frac{r(s_n, s_{n+1}) + v^{(n+1)}(s_{n+1})}{v^{(n+1)}(s_n)}\right)), \tag{14}
\]
where $\delta$ is a positive value smaller than the minimum density in the original environment policy $\pi_{E,gt}$ to guarantee that the one-step importance weight $\rho_n$ is bounded.

APE is suitable for online settings where the value $v(s)$ and environment policy $\pi_E$ are iteratively updated based on each single transition without waiting for episodes to terminate. Thus the value estimate has lower variance than that of vanilla Monte Carlo when the importance distribution (the environment policy $\pi_E$) is properly designed. To increase training stability, one natural extension is using multi-step TD target, which we leave for future work.

### 3.4 Scalable implementation

In this section, we introduce how to improve scalability of APE to handle high-dimensional continuous MDP settings. The general idea is to select proper function approximators to represent the value function $v(s)$ and the environment agent importance policy $\pi_{E}$. Before doing that, we first establish some notations. The value function approximator $v_{\psi}(s)$ with parameter $\psi$ takes state $s$ as input and outputs a probability ranging from 0 to 1. The policy approximator $\pi_{E,\theta}$ parameterized with $\theta$ takes the state-action concatenations $(s, a_A)$ as input and outputs a probability density for $a_E$. Therefore, $\rho_{n,\theta} = \pi_{E,\theta}(a_{E,n}|a_{A,n}, s_n)/\pi_{E,\theta}^{(n)}(a_{E,n}|a_{A,n}, s_n)$. 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 $D$. Assume $s \in S \subset \mathbb{R}^{d_s}$, $a_A \in A_A \subset \mathbb{R}^{d_a}$, and $a_E \in A_E \subset \mathbb{R}^{d_e}$. We present the general algorithm in Algol.1

**Function approximation** We learn $v_{\psi}(s)$ via supervised learning by minimizing a specified loss function between predicted value and TD target $v_{\psi,TD}(s_n)$. For data pair $d_n$,
\[
\rho_{n,\theta} = \pi_{E,\theta}(a_{E,n}|a_{A,n}, s_n)/\pi_{E,\theta}^{(n)}(a_{E,n}|a_{A,n}, s_n) \tag{15}
\]
\[
v_{\psi,TD}(s_n) = r(s_n, s_{n+1}) + v_{\psi}(s_{n+1}) \cdot \rho_{n,\theta}. \tag{16}
\]
One approximation approach is to represent $v(s)$ with an expressive deep neural network (DNN), which are parametric universal approximators with low prediction complexity. However, DNNs are sensitive to imbalanced data and suffers from the catastrophic forgetting problem [35, 12], especially in online learning settings. A competitive model to DNN is Gaussian Process (GP) regression model. The equivalence between GPs and infinitely wide DNNs is derived in [50]. GPs are data-efficient and flexible in making predictions as a non-parametric model, but sacrifice the prediction complexity [55].
Algorithm 1: Scalable Accelerated Policy Evaluation

**Input:** evaluation policy $\pi_A$, testing platform or simulation environment $E$, original environment adversary policy $\pi_{E,gt}(a_E|a_A, s)$, evaluation horizon $N$, value function learning rate $\alpha_f$, NF learning rate $\alpha_\pi$.

**Initialization:** $n = 0$, policy model parameters $\theta$ and value function parameters $\psi$.

Pretrain $\pi_{E,\theta}(a_A|x)$ with target conditional probability $\tilde{\pi}_E(a_A|s), \forall a_A \in A_A, s \in S$.

repeat
  Sample agent action $a_A \sim \pi_A(\cdot|s)$ and environment adversary action $a_E \sim \pi_{E,\theta}(\cdot|a_A, s)$.
  Execute $a_A$ and $a_E$ and observe next state $s'$ and cost $r(s, s')$.
  Calculate one-step importance weight $\rho_n, \theta$.
  Add data pair $d = (s, a_A, a_E, s', r(s, s'), \rho_n, \theta)$ to buffer $D$.
  /* Update rare event probability estimation */
  Calculate TD target $v_{\psi, TD}(s)$ of newly collected $d$ and append $(s, v_{\psi, TD}(s))$ to $D_{GP}$.
  Train value function $v_{\psi}$ with gradient descent.
  /* Update environment adversary policy */
  Calculate adversary policy target based on $d$ and updated $v_{\psi}$.
  Train adversary policy $\pi_{E, \theta}(\cdot|a_A, s)$ with gradient descent.
  $s \leftarrow s'$
  $n \leftarrow n + 1$
until episode finish
end for

Considering that the sampled rare events may be drawn out in $D$ especially in the initial learning phase, we choose to use GP as the function approximator for its data efficiency.

Denote the data buffer for GP training and prediction as $D_{GP}$ with size $m$. $D_{GP}$ only stores data pairs consisting of inputs $x_i = s_i$ and TD targets $y_i = v_{\psi, TD}(s_i)$. The predicted value given a new data point $s^* \in S$ using a GP is drawn from a distribution

$$p(f|D, s^*) = \mathcal{N}(m(f), \text{cov}(f)).$$

(17)

The inputs $x_i = s_i$ and targets $y_i = v_{\psi, TD}(s_i)$ in dataset $D_{GP}$ are aggregated into $X \in \mathbb{R}^{d_s \times m}$ and $Y \in \mathbb{R}^{1 \times m}$. The mean function is $m(f) = K(s^*, X)[K(X, X) + \sigma^2 I]^{-1} Y$ and the covariance matrix is $\text{cov}(f) = K(s^*, s^*) - K(s^*, X)[K(X, X) + \sigma^2 I]^{-1}K_i(X, s^*)$. $\sigma$ 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 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. $\psi$ is thus kernel parameters $\{w, w_1, \ldots, w_{d_s}, \sigma\}$.

$\psi$ is updated to minimize the negative log-likelihood for the dataset $D_{GP}$, with an $L_1$ regularizer over lengthscale parameters, using gradient descent. Denoting $\alpha_\psi$ as the learning rate, we use

$$\psi \leftarrow \psi - \alpha_\psi \nabla_\psi \left\lbrack \sum_{(s_i, v_{\psi, TD}(s_i)) \in D_{GP}} -\log p(v_{\psi, TD}(s_i)|s_i, D_{GP}) + \lambda \sum_j |w_j| \right\rbrack,$$

(18)

where $\lambda$ is the regularization penalty. 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 [46] as well as in a dynamic programming paradigm with changing TD targets [19] [22] [23].

Environment adversary policy approximation In online setting, the environment’s policy are updated based on the most recent data pair $d_n$, and newly updated $v_{\psi}(s)$. The unnormalized target density around the newly sampled action $a_{E,n}$ conditioned on $C_n = [a_{A,n}, s_n]$ is defined in [12]. To cope with continuous spaces, we achieve the modification by adding a normal distribution $\mathcal{N}(a_{E,n}, \sigma)$ with $\sigma$ 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 defined in [20], where $\beta = \sqrt{2\pi} \sigma (\tilde{\pi}_E(a_{E,n}|C_n) - \pi_{E,\theta}(a_{E,n}|C_n))$. The variance matrix is $cov(\cdot)$. Considering that the sampled rare events may be drawn out in $D$ especially in the initial learning phase, we choose to use GP as the function approximator for its data efficiency.
We denote the generative function in cNFs as \( g \). The KL divergence in (21) is then approximated with the sample mean. With learning rate \( \alpha \), the incremental modification about the target density function in (19) results in a Gaussian mixture model with an increasing mixture size. Motivated from these, we place no assumptions on the form of the conditional distribution and employ an expressive enough approximation model.

Specifically, we obtain a closed-form parametric representation of \( \pi_E(a_E|a_A, s) \) using a conditional Normalizing Flow (cNF). cNFs are recently proposed in [49, 70, 47], which are generative models that use 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 forward path (or inverse path based on implementation), which will dramatically accelerate the evaluation process. cNF has the capacity to model distributions that go beyond single-mode Gaussian distributions.

We denote the generative function in cNFs as \( g_\theta : z \rightarrow a_E \), 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 between the target \( p_E(a_E|C_n) \) in (20) and cNF \( \pi_{E,\theta}(a_E|C_n) \). Formally,

\[
L_\theta^{(n)} = \mathbb{E}_{\pi_{E,\theta}(a_E|C_n)} \left[ \log \pi_{E,\theta}(a_E|C_n) - \log p_E(a_E|C_n) \right] = \mathbb{E}_{p_Z(z|C_n)} \left[ \log \pi_{E,\theta}(g_\theta(z)|C_n) - \log p_E(g_\theta(z)|C_n) \right] = \mathbb{E}_{p_Z(z|C_n)} \left[ \log p_Z(z|C_n) - \log | \det \partial_z g_\theta(z) | - \log p_E(g_\theta(z)|C_n) \right]
\]

We update \( \theta \) via gradient descent. We first sample from the simple base distribution \( p_Z(z|C_n) \) and pass the sampled \( z_i, i = 1, \ldots, M \) to the flow to get samples in target domain \( a_{E,i} = g_\theta(z_i), i = 1, \ldots, M \). The KL divergence in (21) is then approximated with the sample mean. With learning rate \( \alpha_\pi \),

\[
\theta \leftarrow \theta - \alpha_\pi \nabla_\theta \sum_{i=1}^{M} \left[ \log p_Z(z_i|C_n) - \log | \det \partial_z g_\theta(z_i) | - \log p_E(g_\theta(z_i)|C_n) \right]
\]

**Convergence analysis** We conduct convergence analysis of our approach that provides guidance and justification on our practical implementation. The analysis idea is to look at the discretized state space (note that this discretization is only for theoretical analysis but not the actual algorithm) and cast our update rule as an approximate value iteration. The derived error bound depends on regularity conditions of the probability kernel and the level of approximation accuracy from the parameterization. The details are left in Appendix Sec. [9].

4 Experiments

4.1 Accelerated policy evaluation with gym-minigrid

We first validate the performance of APE in discrete settings without function approximations in a gym-minigrid [10] environment. 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 into the lava (the yellow block). In \( p_{E,gt} \), the lava has a small probability of moving up at each step. The normal and rare events are shown in Fig. [2]. The agent policy \( \pi_A \) is trained with the deep Q learning [43] until convergence. The environment adversary policy \( \pi_E \) controls the policy of the lava, which is the probability of moving left, right or up.

and \( \gamma = 1/(1 + \beta) \) is the normalizing constant. We set

\[
\tilde{\pi}_E(a_{E,n}|C_n) = \pi_E(a_{E,n}|C_n) \left( r(s_n, s_{n+1}) + v_\psi(s_{n+1}) \right) / v_\psi(s_n)
\]

\[
p_E(a_E|C_n) = \gamma (\pi_{E,\theta}(a_E|C_n) + \beta \cdot \mathcal{N}(a_E, \sigma))
\]
We compare the proposed APE with vanilla Monte Carlo (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. The number of transitions and episodes required till convergence are summarized in Table 1. \( s_1, s_2 \) and \( s_3 \) denote initial states with the lava lying in different positions in the second row. We can see that when the ground truth rare event probability is around \( 4 \times 10^{-3} \), APE successfully converges to the true rare event probability. APE requires one magnitude fewer data than the MC and achieves a smaller variance than MC at convergence.

### 4.2 Scalable accelerated policy evaluation with a multi-agent environment: highway-env

We provide one empirical study showing the strength of scalable APE using function approximators with a multi-agent continuous environment, highway-env [37]. The agent to evaluate aims to perform a collision-free unprotected left turn in a two-lane intersection with one challenger vehicle as shown in Fig. 3. The rare event is defined as the crash. We use a fixed deterministic policy \( \pi_A \) with \( a_A \in \mathbb{R} \) which follows the lane with constant velocity. The environment adversary policy \( \pi_E \) can be treated as the challenger vehicle’s policy. Note that APE may be extended to handle more than two agents by adapting the group modeling method and treating all the other agents as part of the environment, which we leave for future work.

We build two intersection scenarios, Intersection-v0 (shorthanded as v0) and Intersection-v1 (shorthanded as v1) with rare event probability \( 6.34 \times 10^{-2} \) and \( 1.70 \times 10^{-6} \), respectively. We compare APE with function approximators (denoted as APE_scalable) with three baselines, APE using discretization (APE_discrete), cross-entropy method with discretization (CEM_discrete) and vanilla MC. More details of baselines are in Appendix Sec. C.

**Learning process highlights** In v0, all methods are initialized with the \( \pi_{E,gt} \). While in v1, considering that \( v^*(s_0) \) is quite small, for methods except for MC, we initialize with a slightly skewed distribution (the first column in Fig. 5) with a larger variance than \( \pi_{E,gt} \) to sample more rare events at the initial stage. We use a two-timescale updating scheme for GP dynamic programming which appends data pairs more frequently to \( D_{GP} \) than updating the GP parameters \( \psi \). More details of algorithm are in Appendix Sec. B.

**Discussion about evaluation performance** In v0 (Fig. 4(a)), although APE_discrete approaches the true rare event probability, it has a larger estimate variance than APE_scalable. When \( v^*(s_0) \) gets smaller in v1 (Fig. 4(b)), 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 indicate that the discretization of continuous tasks drops the environment or action structure information leading to biased evaluation results. APE_scalable does not outperform MC in v0, but in v1, APE_scalable converges within 3k steps (around 1k episodes) while MC may require 1,762.193 = \( \log(0.05)/\log(1 - v^*(s_0)) \) episodes until having 95% confidence interval. It shows that APE_scalable dominants when small rare event probabilities in terms of estimation accuracy. Interestingly, CEM_discrete achieves higher rare event sample rate than APE_scalable, but heavily underestimate the true rare event probability as in Fig. 4(a) and (b). We conjecture that CEM_discrete

---

**Table 1: Estimation results.**

| methods | MC                | APE                |
|---------|-------------------|--------------------|
| metric \((\times 10^3)\) | mean | std   | mean | std  |
| \( v(s_1) \) | 3.99 | 0.38 | 3.90 | 0.24 |
| \( v(s_2) \) | 3.93 | 0.47 | 3.94 | 0.23 |
| \( v(s_3) \) | 4.00 | 0.50 | 3.94 | 0.17 |
| #transitions | 78,300 | 10,600 | 8884 | 2595 |
| #episodes    | 19,306 | 2612 | 552 | 134 |
| 95% CI       | \( n = 15 \) | \( n = 6 \) |
Intersection-v0
\[ v^*(s_0) = 6.4 \times 10^{-2} \]

Intersection-v1
\[ v^*(s_0) = 1.7 \times 10^{-6} \]

LunarLander
\[ v^*(s_0) = 1.3 \times 10^{-5} \]

Figure 4: Evaluation results in Intersection-v0, Intersection-v1 and LunarLander. Each line is run with 5 random seeds. APE_scalable performs well in high-dimensional tasks with higher estimate accuracy and sampled rare event rate compared with baselines. The performances of all the methods are measured with estimate errors and sampled rare event rates.

Figure 5: The evolution of adversary policy \( \pi_E \) in Intersection-v1 conditioned at the initial state \( s_0 \), a random intermediate state \( s_m \) and a state before collision \( s_n \). The time steps \( t_0 < t_p < t_q < t_c \).

Discussion about environment adversary policy \( \pi_E \)  Fig. 4(a) shows that the learned \( \pi_E \) with APE_scalable (0.44) samples 7.5 and 3.3 times more rare events than MC (0.064) and APE_discrete (0.12) in Intersection-v0. Interestingly, in the more challenging Intersection-v1 (Fig. 4(b)), APE_scalable generates much more rare events with sampled rare event rate 0.75, which is 441,176 times and 4.7 times the results using MC (1.7 \times 10^{-6}) and APE_discrete (0.16). Both results indicate that the learnt policy \( \pi_E \) with APE_scalable is closer to the zero-variance distribution, and fails to explore enough and its environment adversary policy \( \pi_E \) only concentrates on one specific failure mode. Note that we omit the evaluation curve using MC in v1 since its estimate within 3k steps does not converge.
APE_scalable excels more with smaller true rare event probability. 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. 5. We see that the learned adversary policy $\pi_E$ tends to accelerate (being aggressive) at the beginning of each episode and then keep the behavior in original $\pi_{E_{gt}}$ (being conservative) in the remaining steps.

4.3 Scalable accelerated policy evaluation with a single-agent environment: lunarlander

We provide another empirical study in the single-agent environment lunarlander. In LunarLander, the agent aims to land a lunar lander on a fixed landing pad. The policy we evaluate is a heuristic policy by OpenAI. 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 can be thought of as the action adversary in a robust reinforcement learning setting [62] and controls perturbations over actions to execute to mimic controller errors. This example provides a clue for selecting factors controlled by the environment adversary policy $\pi_E$. The experimenters may draw inspirations from robust learning and add uncertainty/noise to components of MDPs [69, 74, 72, 75].

For LunarLander in Fig. 4(c), the performance of MC is acceptable since the rare event probability is not very small ($\sim 1.3 \times 10^{-3}$). Despite of that, APE still outperforms MC in terms of the variance of probability estimation. APE estimation converges faster than MC in about 1.5k steps while MC still has a large variance and the mean is oscillating up and down. Other baselines do not perform well possibly due to the high dimensionality. For the sampled rare event rate, APE shows great efficacy in uncovering the failure modes and sampling more and more rare events. The final sampled rare event rate of APE is more than 100 times larger than that of the original environment. CEM_discrete and APE_discrete can increase the sampled rare event rate as well but not as effectively as APE_scalable.

5 Conclusion and broader impacts

Motivated by 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 sequential rare events by 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.

Acknowledgments and Disclosure of Funding

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] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. arXiv preprint arXiv:1705.10528, 2017.
[2] TP Imthias Ahamed, Vivek S Borkar, and S Juneja. Adaptive importance sampling technique for markov chains using stochastic approximation. Operations Research, 54(3):489–504, 2006.
[3] Stefano V Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. Artificial Intelligence, 258:66–95, 2018.
[4] Mansur Arief, Zhiyuan Huang, Guru Koushik Senthil Kumar, Yuanlu Bai, Shengyi He, Wenhao Ding, Henry Lam, and Ding Zhao. Deep probabilistic accelerated evaluation: A certifiable
rare-event simulation methodology for black-box autonomy. arXiv preprint arXiv:2006.15722, 2020.

[5] Søren Asmussen and Peter W Glynn. Stochastic simulation: algorithms and analysis, volume 57. Springer Science & Business Media, 2007.

[6] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. arXiv preprint arXiv:1606.01540, 2016.

[7] James Bucklew. Introduction to rare event simulation. Springer Science & Business Media, 2013.

[8] Frédéric Cérou and Arnaud Guyader. Adaptive multilevel splitting for rare event analysis. Stochastic Analysis and Applications, 25(2):417–443, 2007.

[9] Baiming Chen, Zuxin Liu, Jiacheng Zhu, Mengdi Xu, Wenhao Ding, and Ding Zhao. Context-aware safe reinforcement learning for non-stationary environments. arXiv preprint arXiv:2101.00531, 2021.

[10] Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018.

[11] CHEE-S Chow and John N Tsitsiklis. An optimal one-way multigrid algorithm for discrete-time stochastic control. IEEE transactions on automatic control, 36(8):898–914, 1991.

[12] Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. In International Conference on Machine Learning, pages 1952–1961. PMLR, 2020.

[13] Kamil Ciosek and Shimon Whiteson. Offer: Off-environment reinforcement learning. AAAI Press, page 1819–1825, 2017.

[14] Anthony Corso, Ritchie Lee, and Mykel J Kochenderfer. Scalable autonomous vehicle safety validation through dynamic programming and scene decomposition. arXiv preprint arXiv:2004.06801, 2020.

[15] Anthony Corso, Robert J Moss, Mark Koren, Ritchie Lee, and Mykel J Kochenderfer. A survey of algorithms for black-box safety validation. arXiv preprint arXiv:2005.02979, 2020.

[16] Randell Cotta, Mingyang Hu, Dan Jiang, and Peizhou 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, pages 483–491, 2019.

[17] P-T De Boer, Victor F Nicola, and Reuven Y Rubinstein. Adaptive importance sampling simulation of queueing networks. In 2000 Winter Simulation Conference Proceedings (Cat. No. 00CH37165), volume 1, pages 646–655. IEEE, 2000.

[18] Pieter-Tjerk de Boer. Analysis and efficient simulation of queueing models of telecommunication systems. Doctoral dissertation. (Centre for Telematics and Information Technology University of Twente), 2000.

[19] Marc P Deisenroth, Jan Peters, and Carl E Rasmussen. Approximate dynamic programming with gaussian processes. In 2008 American Control Conference, pages 4480–4485. IEEE, 2008.

[20] Paritosh Y Desai and Peter W Glynn. A markov chain perspective on adaptive monte carlo algorithms. In Proceeding of the 2001 Winter Simulation Conference (Cat. No. 01CH37304), volume 1, pages 379–384. IEEE, 2001.

[21] Wenhao Ding, Baiming Chen, Bo Li, Kim Ji Eun, and Ding Zhao. Multimodal safety-critical scenarios generation for decision-making algorithms evaluation. IEEE Robotics and Automation Letters, 6(2):1551–1558, 2021.

[22] Yaakov Engel, Shie Mannor, and Ron Meir. Bayes meets bellman: The gaussian process approach to temporal difference learning. In Proceedings of the 20th International Conference on Machine Learning (ICML-03), pages 154–161, 2003.

[23] Yaakov Engel, Shie Mannor, and Ron Meir. Reinforcement learning with gaussian processes. In Proceedings of the 22nd international conference on Machine learning, pages 201–208, 2005.

[24] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firouzi, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. arXiv preprint arXiv:1802.01561, 2018.
[25] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1625–1634, 2018.

[26] Jordan Frank, Shie Mannor, and Doina Precup. Reinforcement learning in the presence of rare events. In Proceedings of the 25th international conference on Machine learning, pages 336–343, 2008.

[27] Xavier Gabaix. Variable rare disasters: An exactly solved framework for ten puzzles in macrofinance. The Quarterly journal of economics, 127(2):645–700, 2012.

[28] Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. arXiv preprint arXiv:1809.11165, 2018.

[29] Paul Glasserman, Philip Heidelberger, Perwez Shahabuddin, and Tim Zajic. Multilevel splitting for estimating rare event probabilities. Operations Research, 47(4):585–600, 1999.

[30] Philip Heidelberger. Fast simulation of rare events in queueing and reliability models. ACM Transactions on Modeling and Computer Simulation (TOMACS), 5(1):43–85, 1995.

[31] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. arXiv preprint arXiv:1702.02284, 2017.

[32] Nathan Kallus and Masatoshi Uehara. Double reinforcement learning for efficient off-policy evaluation in markov decision processes. Journal of Machine Learning Research, 21(167):1–63, 2020.

[33] Nidhi Kalra and Susan 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, 94:182–193, 2016.

[34] kamenbliznashki. normalizing-flows. https://github.com/kamenbliznashki/normalizing_flows, 2020.

[35] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence, 5(4):221–232, 2016.

[36] Jae-hoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. arXiv preprint arXiv:1711.00165, 2017.

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

[38] Jinyan Li, Lian-sheng Liu, Simon Fong, Raymond K Wong, Sabah Mohammed, Jinan Fiaidhi, Yunsick Sung, and Kelvin KL Wong. Adaptive swarm balancing algorithms for rare-event prediction in imbalanced healthcare data. PLoS one, 12(7):e0180830, 2017.

[39] Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. Tactics of adversarial attack on deep reinforcement learning agents. arXiv preprint arXiv:1703.06748, 2017.

[40] Zuxin Liu, Hongyi Zhou, Baiming Chen, Sicheng Zhong, Martial Hebert, and Ding Zhao. Safe model-based reinforcement learning with robust cross-entropy method. arXiv preprint arXiv:2010.07968, 2020.

[41] Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher J Pal, and Liam Paull. Active domain randomization. In Proceedings of the Conference on Robot Learning, volume 100 of Proceedings of Machine Learning Research, pages 1162–1176. PMLR, 2020.

[42] Francisco S Melo and M Isabel Ribeiro. Q-learning with linear function approximation. In International Conference on Computational Learning Theory, pages 308–322. Springer, 2007.

[43] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013.

[44] Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In Advances in Neural Information Processing Systems, pages 1054–1062, 2016.
[45] Hongseok Namkoong, Ramtin Keramati, Steve Yadlowsky, and Emma Brunskill. Off-policy policy evaluation for sequential decisions under unobserved confounding. *arXiv preprint arXiv:2003.05623*, 2020.

[46] Duy Nguyen-Tuong, Jan Peters, and Matthias Seeger. Local gaussian process regression for real time online model learning. *Advances in neural information processing systems*, 21:1193–1200, 2008.

[47] Geunseob Oh and Jean-Sebastien Valois. Hcnaf: Hyper-conditioned neural autoregressive flow and its application for probabilistic occupancy map forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14550–14559, 2020.

[48] Matthew O’Kelly, Aman Sinha, Hongseok Namkoong, Russ Tedrake, and John C Duchi. Scalable end-to-end autonomous vehicle testing via rare-event simulation. In *Advances in Neural Information Processing Systems*, pages 9827–9838, 2018.

[49] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347, 2017.

[50] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, pages 2817–2826, 2017.

[51] Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

[52] Arunselvan Ramaswamy and Shalabh Bhatnagar. Analyzing approximate value iteration algorithms. *arXiv preprint arXiv:1709.04673*, 2017.

[53] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[54] Gerardo Rubino and Bruno Tuffin. *Rare event simulation using Monte Carlo methods*. John Wiley & Sons, 2009.

[55] Reuven Y Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997.

[56] Reuven Y Rubinstein and Dirk P Kroese. *Simulation and the Monte Carlo method*, volume 10. John Wiley & Sons, 2016.

[57] John Rushby. Bus architectures for safety-critical embedded systems. In *International Workshop on Embedded Software*, pages 306–323. Springer, 2001.

[58] Arrigo Schieppati, Jan-Inge Henter, Erica Daina, and Anita Aperia. Why rare diseases are an important medical and social issue. *The Lancet*, 371(9629):2039–2041, 2008.

[59] Matthew Schlegel, Wesley Chung, Daniel Graves, Jian Qian, and Martha White. Importance resampling for off-policy prediction. In *Advances in Neural Information Processing Systems*, pages 1799–1809, 2019.

[60] Krishnan Srinivasan, Benjamin Eysenbach, Sehoon Ha, Jie Tan, and Chelsea Finn. Learning to be safe: Deep rl with a safety critic. *arXiv preprint arXiv:2010.14603*, 2020.

[61] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[62] Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*, pages 6215–6224. PMLR, 2019.

[63] Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery rl: Safe reinforcement learning with learned recovery zones. *arXiv preprint arXiv:2010.15920*, 2020.

[64] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[65] Dustin Tran, Rajesh Ranganath, and David M Blei. The variational gaussian process. *arXiv preprint arXiv:1511.06499*, 2015.
[66] Jonathan Uesato, Ananya Kumar, Csaba Szepesvari, Tom Erez, Avraham Ruderman, Keith Anderson, Nicolas Heess, Pushmeet Kohli, et al. Rigorous agent evaluation: An adversarial approach to uncover catastrophic failures. *arXiv preprint arXiv:1812.01647*, 2018.

[67] Manuel Villén-Altamirano and José Villén-Altamirano. Restart: a straightforward method for fast simulation of rare events. In *Proceedings of Winter Simulation Conference*, pages 282–289. IEEE, 1994.

[68] Roman Werbachowski, András György, and Csaba Szepesvári. Detecting overfitting via adversarial examples. *arXiv preprint arXiv:1903.02380*, 2019.

[69] Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

[70] Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

[71] Tengyang Xie, Yifei Ma, and Yu-Xiang Wang. Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. In *Advances in Neural Information Processing Systems*, pages 9668–9678, 2019.

[72] Huan Xu and Shie Mannor. Distributionally robust markov decision processes. *Mathematics of Operations Research*, 37(2):288–300, 2012.

[73] Mengdi Xu, Wenhao Ding, Jiacheng Zhu, Zuxin Liu, Baiming Chen, and Ding Zhao. Task-agnostic online reinforcement learning with an infinite mixture of gaussian processes. *arXiv preprint arXiv:2006.11441*, 2020.

[74] Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. *arXiv preprint arXiv:2003.08938*, 2020.

[75] Kaiqing Zhang, Tao Sun, Yunzhe Tao, Sahika Genc, Sunil Mallya, and Tamer Basar. Robust multi-agent reinforcement learning with model uncertainty. *Advances in Neural Information Processing Systems*, 33, 2020.

[76] Ding Zhao, Xianan Huang, Huei Peng, Henry Lam, and David J LeBlanc. Accelerated evaluation of automated vehicles in car-following maneuvers. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):733–744, 2017.

[77] Ding Zhao, Henry Lam, Huei Peng, Shan Bao, David J LeBlanc, Kazutoshi Nobukawa, and Christopher S Pan. Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques. *IEEE transactions on intelligent transportation systems*, 18(3):595–607, 2016.
A Environments

We study three practical examples to show the strength and generality of APE including the discrete gym-minigrid [10], highway-env [37] which is a multi-agent continuous environment and lunar-lander [6] which aims to solve a single-agent control task. We assume that the evaluation policy $\pi_A$ is mature as in most rare event literature so that rare events are naturally related to failure cases similar to most existing literature. There are other environments used in rare-event related literature, such as MuJoCo humanoid [64] (rare events as falling down situations [66]) and mountain-car [6] (rare events as small perturbations to the initial velocity [13]). The proposed Accelerated Policy Evaluation (APE) method relies on experimenters to choose factors controlled by the environment agent.

A.1 Single-agent environment: lunar-lander

The meanings of 8-dimensional observation and 2-dimensional action in LunarLander are shown in Tab. 2. The agent controls the main engine and orientation engine. The environment adversary adds perturbations into the agent action with frequency 1 FPS. The agent controls the lander at 50 FPS. During every 50-step period, a constant action noise is injected into the commended action $a_A$. In the original environment, the action noise is sampled i.i.d. from a 2-D Gaussian distribution centered at 0 with a standard deviation 0.1.

| Num | Observation | min, max |
|-----|-------------|----------|
| 0   | horizontal coordinate | $-1.5, 1.5$ |
| 1   | vertical coordinate | $-0.5, 2.0$ |
| 2   | horizontal speed | $-1.5, 1.5$ |
| 3   | vertical speed | $-1.5, 1.0$ |
| 4   | angle | $-\pi, \pi$ |
| 5   | angular speed | $-3.0, 3.0$ |
| 6   | if first leg has contact | $\{0, 1\}$ |
| 7   | if second leg has contact | $\{0, 1\}$ |

| Num | Action | min, max |
|-----|--------|----------|
| 0   | main engine power | $-1.0, 1.0$ |
| 1   | orientation engine power | $-1.0, 1.0$ |

Table 2: Observation and action space of LunarLander

A.2 Multi-agent environment: highway-gym

The Intersection-v0 and Intersection-v1 environments are modified based on highway-env [37]. We consider a two-agent example where the environment agent controls the challenger vehicle’s acceleration. The environment action $a_E$ is the coefficient of a linear controller for controlling the challenger vehicle’s acceleration. $acc_E = a_E x_E$ where $x_E \in \mathbb{R}^{d_E}$ is the feature vector of interest. More concretely, the action $a_E \sim \pi_E$ is a multi-dimensional vector with each element corresponding to the coefficient of a feature. Denote the positions and heading of a vehicle as $(x, y, h)$. In Intersection-v0, the state $s = (x_A, y_A, \dot{x}_A, \dot{y}_A, \cos h_A, \sin h_A, x_{rel}, y_{rel}, \dot{x}_{rel}, \dot{y}_{rel}, \cos h_{rel}, \sin h_{rel}) \in \mathbb{R}^{12}$ including the kinematic information of the agent and the challenger vehicle evaluated in the world frame and in the body fixed frame attached to the agent. The state $s$ is within $[-1, 1]$. The environment adversary has a action $a_E \in \mathbb{R}^3$ with value in range $[-2, 2]$. In Intersection-v1, we simply the environment with states $s = (x_{rel}, y_{rel}) \in \mathbb{R}^2$ and adversary actions $a_E \in \mathbb{R}^2$.

B Algorithm

In this section, we describe algorithm highlights of APE. A concrete algorithm with highlights is shown in Algo. 2 with highlights. The code and model hyper-parameters will be public soon.

1. Warm start. We initialize the environment policy $\pi_E$ with a warm start $\hat{\pi}_E$ when the rare event probability $\nu^*(s_0)$ is pretty small. For example, in Intersection-v1 with
Algorithm 2: Scalable Accelerated Policy Evaluation with algorithm highlights

**Input:** evaluation policy $\pi_A$, testing platform or simulation environment $E$, original environment adversary policy $\pi_{E,\theta}(a_E|a_A, s)$, evaluation horizon $N$, value function learning rate $\alpha_j$, NF learning rate $\alpha_\pi$, $\psi$ update interval $n_s$, dense reward decay rate $\gamma_r$, momentum coefficient $\beta_0$

**Initialization:** $n = 0$, policy model parameters $\theta$ and value function parameters $\psi$

Pretrain $\pi_{E,\theta}(a_E|x)$ with target conditional probability $\pi_E(\cdot | a_A, s)$, $\forall a_A \in A_A, s \in S$

for $n = 0$ to $N - 1$ do

  Reset testing platform or simulation environment

  Initialize empty episode buffer $D_e$

  repeat

  Sample agent action $a_A \sim \pi_A(\cdot | s)$ and environment adversary action $a_E \sim \pi_{E,\theta}(\cdot | a_A, s)$

  Execute $a_A$ and $a_E$ and observe next state $s'$ and cost $r(s, s')$

  Calculate one-step importance weight $\rho_{n, 0}$ (15)

  Add data pair $d = (s, a_A, a_E, s', r(s, s'), \rho_{n, 0})$ to buffer $D$

  /* Update rare event probability estimation with two-timescale */

  if $n \% n_\psi = 0$ then

    Train value function $v_\psi$ with gradient descent (18)

  end if

  /* Update environment adversary policy with dense rewards */

  Append $d$ to $D_e$

  if $r(s, s') = 1$ then

    Calculate adversary policy target based on $D_e$ with dense rewards (23) and updated $v_\psi$ (20)

    Get $\theta'$ by training adversary policy $\pi_{E,\theta}(| a_A, s)$ with gradient descent (22)

    $\theta \leftarrow \beta_0 \theta' + (1 - \beta_0) \theta$

  end if

  $s \leftarrow s'$

  $n \leftarrow n + 1$

  until episode finish

end for

$v^*(s_0) = 1.7 \times 10^{-6}$, we initialize APE and CEM with a slightly skewed distribution with a larger variance than the ground truth one to sample rare events at the initial stage. The design of warm start distribution may leverage domain knowledge.

2. **GP two-timescale update.** We use a two-timescale updating scheme for GP dynamic programming which appends data pairs more frequently to $D_{GP}$ than updating the GP parameters $\psi$. Note that both data in $D_{GP}$ and the parameters $\psi$ affects the rare-event prediction probability. In our online setting with data streaming coming in, frequently updating GP kernel parameters $\psi$ leads to training instability based on experiments. Therefore we choose to update $\psi$ with a relatively low frequency. Considering that the prediction complexity of GP is $O(N_{GP}^3)$ where $N_{GP}$ is the number of data points in $D_{GP}$, we need to design a mechanism to only store representative data as in sparse GP literature [65]. In this work, we early stop collecting data when there are sufficient data points.

3. **cMAF update with dense rewards.** We use cMAF [49] model to represent the environment policy $\pi_E$. In a pure online setting, at step $n$, we hope that cMAF is updated towards the target distribution conditioned on $(s_n, a_n, A)$ with (22) and leaves distributions conditioned on other values unaffected. However, in practice the pure online update raises several issues: (1) cMAF over-fitting to the latest target distribution across conditions (diminishing the effect of conditions), (2) training instability due to the quite sparse reward signal $r(s_{n-1}, s_n)$ and (3) high computational complexity for frequently updating cMAF. To address the aforementioned issues, we propagate the reward signal to the whole episode as follows:

$$r(s_{n-1}, s_n) = \gamma_r^{n-n} r(s_{\tau-1}, s_\tau), n = 1, \ldots, \tau,$$

where $\gamma_r$ is the dense reward decay rate. $\tau$ is the episode termination step. To improve training stability, we further add momentum to the parameter $\theta$ of cMAF. We early stop the cMAF training when the sampled rare event rate is satisfactory.
C Baselines

We use a desktop PC with twelve 64-bit CPU (model: Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz) and a GPU (model: NVIDIA GeForce RTX 2080 Ti) as the computing infrastructure. We implement cMAF based on the paper [49] as well as the public package [34] and GP based on GPytorch [28]. When selecting model hyperparameters for APE and baselines, we first randomly search in a coarse range and then do a grid search in a smaller hyperparameter space. APE_discrete and CEM_discrete both divide each dimension of $s$ and $a_E$ with small enough slot interval, and use dictionaries to store the rare event probability estimate and the environment policy $\pi_E$.

Cross-entropy method (CEM) with discretization

We adopt another baseline, the cross-entropy method for Markov Chains [18, 17] to our setting by treating the environment agent’s policy $\pi_E$ as the importance distribution. CEM requires to sample $k$ episodes before updating the importance distribution $\pi_E^{(m)}$. At update iteration $m$, the rare event probability estimate and the updating rule for $\pi_E$ in discrete settings are as follows:

$$v^{(m+1)} = \frac{1}{k} \sum_{j=1}^{k} L_j \cdot I_j(\tau)$$

$$\tilde{\pi}_E^{(m+1)}(a_E|a_A, s) \leftarrow \max \left( \delta_{CEM}, \frac{\sum_{j=1}^{k} L_j \cdot N_j(s, a_A, a_E) \cdot I_j(\tau)}{\sum_{j=1}^{k} L_j \cdot I_j(\tau)} \right),$$

$$L_j = \prod_{n=1}^{\tau} \rho_n(s_n, s_{n+1}, a_{A,n}) = \prod_{n=1}^{\tau} \pi_{E,gt}(a_{E,n}|a_{A,n}, s_n)$$

where $\tilde{\pi}_E^{(m+1)}(a_E|a_A, s)$ is the unnormalized target. $L_j$ is the likelihood of the episode $j$. $\delta_{CEM}$ ensures the policy has positive values. $N_j(s, a_A, a_E)$ is the count of the environment selecting $a_E$ given condition $(s, a_A)$ along episode $j$. $N_j(s, a_A)$ is the count of the environment encountering condition $(s, a_A)$ along episode $j$.

![Figure 6: Visualization of $\pi_E$ in the middle of the evaluation process. The distributions in each row have the same condition $(s, a_A)$. From left to right, the figures in each row present the ground truth distribution, the target distribution, the cMAF distribution after training and cMAF samples.](image)

C.1 Multi-modality of adversary policy along evaluation

In Fig. 6 we visualize the environment agent policy $\pi_E$ at two steps in Intersection-v0 to show that (1) the target density at each update step is composed of an increasing number of Gaussian distributions and (2) cMAF has the capacity to represent multi-modal distributions.

D Convergence analysis

In this section, we examine the convergence properties/issues associated with APE. Since APE focuses on evaluate (1) under a fixed policy $\pi_A$, we can view it as an expected cost evaluation.
problem for a Markov chain with a fixed transition probability. The shifting policy $\pi_E$ corresponds to changing the transition probability to accelerate the evaluation, particularly for rare events under the original transition probability. To be specific, we assume a Markov chain with state $S$, transition kernel $(p(x,y), x, y \in S)$ and definitions on $I, T, R, \tau, (r(x,y), x, y \in S)$ as in the case for MDP in Section 3. We can view the problem as evaluating

$$v^*(x) = \mathbb{E}_{x,p} \left[ \sum_{n=1}^{\tau} r(s_{n-1}, s_n) \right]$$

for $s_0 \in I$, while $v^*$ is set to 0 on $T$. Formally, in the case that $|S| < \infty$, the update rule associated with (11) and (13), in the Markov chain setting, is analogous to updating rules in [2]:

$$v^{(n+1)}(s_n) = (1 - \alpha) v^{(n)}(s_n) + \alpha \left[ r(s_n, s_{n+1}) + v^{(n)}(s_{n+1}) \right] \cdot \frac{p(x_n, x_{n+1})}{p^{(n)}(x_n, x_{n+1})} \quad \text{(value iteration)}$$

$$p^{(n+1)}(x_n, x_{n+1}) = \max \left( p(x_n, x_{n+1}) \left( \frac{r(s_n, s_{n+1}) + v^{(n+1)}(x_{n+1})}{v^{(n+1)}(s_n)} \right), \delta \right) \quad \text{(weight update)}$$

$$p^{(n+1)}(x_n, x_{n+1}) = \frac{p^{(n+1)}(s_n, y)}{\sum_{z \in S} p^{(n+1)}(s_n, z)} \quad \text{(normalization scaling)}$$

where $\delta, \alpha > 0$, $v^{(n)}$, $p^{(n)}$ are the $n$-th iteration of value evaluation and adaptive importance sampling used for simulation, respectively. The convergence (with probability 1) property of the above iterative algorithm is established in [2]. However, we want to investigate the convergence behavior of (11) and (13) in a Markov chain setting where the state space $S$ is continuous/large, and we are updating parametric/non-parametric $v_{\psi^{(n)}}$ and $p_{\theta^{(n)}}$. For this analysis, we focus on the parameterized case (i.e. function approximator) which has better analytical properties, for insights into our practical implementation.

Before we proceed, we briefly outline the main steps and some key observations in our analysis.

First, for the adaptive importance sampling $p_{\theta^{(n)}}$, the weight update and normalization scaling for $p^{(n)}$ in the above equations are replaced by an update on $\theta$, which is based on approximate loss minimization between distributions, e.g. KL divergence. However, even though the rate (and practical performance) of convergence towards $v^*$ depends on how well we could update $p_{\theta^{(n)}}$ towards $p^*$, the so-called "zero-variance" sampling distribution, whether the convergence to a fixed point (if there is one) happens with probability 1 is not affected by the evolution of $p_{\theta^{(n)}}$, as in principle one can always not use adaptive sampling to accelerate. Thus, the convergence of $p_{\theta^{(n)}} \to p^*$ is not required for the convergence of $v_{\psi^{(n)}}$. Second, the analysis on convergence is generally not straightforward for the case $|S| = \infty$. To handle the difficulties arising from the infinite cardinality, we discretize $S$ to allow for a more compact representation. Consequently, we require certain regularity conditions on $(p(x,y), x, y \in S)$ and $\{p_{\theta^{(n)}} : S \times S \to \mathbb{R}^+\}_\theta$. Third, the analysis relies on techniques from Approximate Value Iteration (AVI) (see [52]), which incorporate the error/bias in the stochastic iterative algorithms. In our analysis, such error can arise from the discretization of $S$ and the expressiveness/accuracy of $\{v_{\psi^{(\cdot)}} : S \to \mathbb{R}\}_\psi$ in approximating $(v^*(x), x \in S)$.

D.1 Definitions and assumptions

Let $F_n$ be the filtration generated by first n observation of the chain $\{s_0, s_1, ..., s_n\}$. Let $\{p_{\theta^{(n)}}\}_n$ be the sequence of sampling distribution (derived from $\pi_E$ updates in Algorithm 1, ideally $p_{\theta^{(n)}} \to p^*$, the zero-variance sampling distribution for estimating $v^*$) where $\theta^{(n)}$ is $F_n$-measurable.

**Definition 1.** For a transition kernel $(q(x,y), x, y \in S)$ and $\epsilon$, define

$$M(\epsilon, q) := \sup_{\|x-y\|_\infty \leq \epsilon} \left| \frac{q(x,z) - q(y,z)}{q(y,z)} \right|,$$

with the convention $0/0 = 0$.

**Assumption 1.** We assume (A1.) The state space is a compact subset of $\mathbb{R}^d$; (A2.) The Markov chain under the original transition kernel $p(\cdot, \cdot)$ and $\{p_{\theta^{(n)}}\}_n$ are bounded above by some constant $K$ and guarantee $\mathbb{E} \tau < \infty$ (which implies $\tau < \infty$ w.p.1); (A3.) There exists $\epsilon_0 > 0$ such that

$$\sup_{\epsilon \leq \epsilon_0, q \in \{p_{\theta^{(n)}}\}_n \cup \{p\}} \frac{M(\epsilon, q)}{\epsilon} < M$$

18
for some $M < \infty$; (A4.) There exist some constant $C > 1$ such that

$$|r(x, z) - r(y, z)| \leq C\|x - y\|_{\infty},$$

for all $x, y, z \in S$.

Notice the main goal of adaptive importance sampling is to approach the "zero-variance" sampling distribution for $\nu^*$ which would ideally make the iteration towards "rare events" much faster. On the other hand, from a technical standpoint, the condition $E[\tau] < \infty$ also alleviates the analytical difficulty arising from the fact that our MDP is undiscounted. We start with the following lemma.

**Lemma 1.** Let $\{s_n\}_{n \geq 0}$ be the Markov chain and let the non-negative reward function $r(\cdot, \cdot)$ be uniformly bounded as $\|r\|_{\infty} < R$. Let $\gamma \in (0, 1)$ be a discount factor. Then, as $\gamma \to 1$

$$E[\sum_{t=1}^{\tau} \gamma^{t-1} r(s_{t-1}, s_t)] \to E[\sum_{t=1}^{\tau} r(s_{t-1}, s_t)],$$

when $E[\tau] < \infty$ (the convergence is uniform for all $s_0 \in S$).

**Proof.** Since $t \geq 0$ and it discrete, it follows from Fubini’s theorem that $E[\tau] = \sum_{t=0}^{\infty} P(\tau = t)$. Thus, $\lim_{t \to \infty} \sum_{t > t'} P(\tau = t') = 0$. Given any $\epsilon > 0$, we can find $T > 1$ large enough such that $\sum_{t>T-1} P(\tau \geq t-1) < \frac{\epsilon}{2R}$ and $\gamma$ close enough to 1 such that $\sum_{t=1}^{T-1} (1 - \gamma^{t-1}) < \frac{\epsilon}{2R}$. Then,

$$E[\sum_{t=1}^{\tau} \gamma^{t-1} r(s_{t-1}, s_t)] - E[\sum_{t=1}^{\tau} r(s_{t-1}, s_t)]$$

$$= E[\sum_{t=1}^{\tau} (1 - \gamma^{t-1}) r(s_{t-1}, s_t) \mathbb{1}_{\{\tau \geq t-1\}}]$$

$$\leq R \sum_{t=1}^{T-1} (1 - \gamma^{t-1}) + R \sum_{t>T-1} P(\tau \geq t-1) < \epsilon.$$

As the proof indicates, the rate of convergence for the discounted expected towards the undiscounted one relies on how fast the tail of $\tau$ diminishes, which suggests how a consistent and efficient adaptive sampling distribution that can quickly realize the hitting time $\tau$ can give nice analytical properties.

### D.2 Chain discretization

For the analysis of the continuous Markov chain derived from the MDP update (11) and (13), we first discretize the continuous state $S$. Since $S$ is compact, for any $h > 0$, we can find a finite collection of sets $\{S_h^i\}_{i=1}^{N_h} \subseteq S$ such that $\bigcup_{i \in [N_h]} S_h^i = S$, $S_h^i \cap S_h^j = \emptyset$ for $i \neq j$ and

$$\max_{i \in [N_h]} \sup_{x,y \in S_h^i} \|x - y\|_{\infty} \leq h.$$

To simplify analysis, we further assume that there are no overlaps between different types of states among $\{S_h^i\}_{i=1}^{N_h}$, meaning that for any $i \in [N_h]$, either $S_h^i \subseteq \mathcal{I}$ or $S_h^i \subseteq \mathcal{R}$ or $S_h^i \subseteq \mathcal{T} \cap \mathcal{R}^c$, thus avoiding possible confusion in classifying the states after discretization. Now, for any $i \in [N_h]$, let $s_h^i$ be a representative element of $S_h^i$ and define $\hat{S}_h := \{s_h^i\}_{i=1}^{N_h}$ be the set of all such representatives. Moreover, we arrange $\hat{S}_h \equiv \{s_h^i\}_{i=1}^{N_h}$ to be the representatives contained in $\mathcal{I}$.

Consequently, for any $x \in S$, there exists a unique $x_h \in [N_h]$ such that $x \in S_h^x$ and a corresponding representative element $s_h^x$. Let $\sigma(\cdot) : S \to \hat{S}_h$ be the function that maps $x$ to $s_h^x$.

Now we are in a position to "discretize" the reward $r : S \times S \to \mathbb{R}$ and transition probability $p : S \times S \to \mathbb{R}$. Let $\lambda$ be the Lebesgue measure on $S$. Define:

$$\tilde{r}_h(x,y) = r(\sigma(x), \sigma(y))$$

$$\tilde{p}_h(x,y) = \frac{p(\sigma(x), \sigma(y))}{\int_{S} p(\sigma(x), \sigma(z)) \lambda(dz)}$$
Let $\gamma \in (0, 1)$ be a discount factor (typically close to 1). Define the following operators (Bellman operator from a MDP standpoint) from $\mathbb{R}^S$ to $\mathbb{R}^S$:

$$
Tv(x) = \int_S (r(x, y) + v(y))p(x, y)\lambda(dy),
$$

$$
T\gamma v(x) = \int_S (r(x, y) + \gamma v(y))p(x, y)\lambda(dy),
$$

$$
\tilde{T}^h v(x) = \int_S (\tilde{r}_h(x, y) + v(y))\tilde{p}_h(x, y)\lambda(dy),
$$

$$
\tilde{T}^{\gamma h} v(x) = \int_S (\tilde{r}_h(x, y) + \gamma v(y))\tilde{p}_h(x, y)\lambda(dy).
$$

For $\gamma \in (0, 1)$, the operator is known to be a contraction and a unique fixed point exist (i.e. $Tv = v$). For the operator without discount factor, a unique fixed point exists under regularity conditions [2, 51].

We assume such conditions are met and let $v^*$, $v^*_\gamma$, $\tilde{v}^h, *$ and $\tilde{v}^{h, \gamma}$ be their respective unique fixed points.

Let $\mathcal{L}_h$ be the space of functions that are constant on each $S^i_h$. It can be shown that $\tilde{r}_h, \tilde{p}_h, \tilde{v}^{h, \gamma} \in \mathcal{L}_h$. Consequently, we can view the chain on $S$ with $\tilde{r}_h, \tilde{p}_h$ as a chain on discrete state space $\tilde{S}^h$ with a straightforward discretized version of $r^i_h, p^i_h$ and the value target $\tilde{v}^{h, \gamma}$.

Now, suppose we can obtain $\tilde{v}^{h, \gamma}$ for this discretized chain but we are interested in estimating $v^*$, we would need to analyze $\|v^* - \tilde{v}^{h, \gamma}\|_\infty$.

**Lemma 2.** For any given $\epsilon > 0$, there exists some $h > 0$ such that

$$
\|v^* - \tilde{v}^{h, \gamma}\|_\infty < \epsilon.
$$

**Proof.** It follows from Lemma [1] we can find $\gamma$ close enough to 1 such that

$$
\|v^* - v^*_\gamma\|_\infty < \frac{\epsilon}{3},
$$

$$
\|\tilde{v}^{h, \gamma} - \tilde{v}^{h, \gamma}_\gamma\|_\infty < \frac{\epsilon}{3}
$$

for any $h > 0$. It then follows from our discretization of the chain and [11] (the proof is for MDP but it follows for Markov chain as well, as we can always conceptualize Markov chain as a MDP with only one action at each state) that there is some constant $C > 1$ such that

$$
\|v^*_\gamma - \tilde{v}^{h, \gamma}_\gamma\|_\infty \leq \frac{C}{1 - \gamma}h,
$$

where $C$ does not depend on $\gamma$ or $h$. Thus, if we pick $h$ small enough such that $h < \frac{1 - \gamma}{\epsilonC} \epsilon$, then

$$
\|v^* - \tilde{v}^{h, \gamma}\|_\infty \leq \|v^* - v^*_\gamma\|_\infty + \|v^*_\gamma - \tilde{v}^{h, \gamma}_\gamma\|_\infty + \|\tilde{v}^{h, \gamma}_\gamma - \tilde{v}^{h, \gamma}\|_\infty < \epsilon.
$$

We define

$$
\epsilon_h := \|v^* - \tilde{v}^{h, \gamma}\|_\infty
$$

as the discretization. We discuss in the final section on how this parameter affect the convergence of APE. After we can control $\epsilon_h = \|v^* - \tilde{v}^{h, \gamma}\|_\infty$, we wanted to analyse the error bound between $\tilde{v}^{h, \gamma}$ and our updated $v_{\psi(n)}$ derived from [11] and [13].

### D.3 Approximate value iteration

As mentioned before, we denote the sequence of adaptive sampling distribution (derived from $\pi_{E^i}$ updates) to be $\{p_{\theta(n)}\}_{n \geq 0} \subseteq \{p_{\theta}\}$ where $\theta(n) \in \mathcal{F}_n$. If we have picked the discretization parameter $h$ sufficiently small, and our parametric or non-parametric function approximators can guarantee to be uniformly Lipschitz:

$$
\sup_{\psi \in \Psi} \frac{|v_{\psi(n)}(x) - v_{\psi}(y)|}{h} < L
$$

(27)
for some \( L < \infty \). Under such assumption, we want to show the update \( \{ v^{(n)} \}_{n \geq 0} \) from (19) in the continuous space \( \mathcal{S} \) can be viewed approximately as update towards \( \tilde{v}^h \) on \( \tilde{S}^h \). To be specific, we again invoke the discrete Markov chain view with \((\tilde{S}^h, \tilde{p}_h, \tilde{r}_h)\) (technically the discretized version of \( \tilde{p}_h, \tilde{r}_h \) only defined on \( \tilde{S}^h \times \tilde{S}^h \)). Suppose, on this discrete chain, we are using the following asynchronous stochastic approximation updates with given adaptive importance sampling \( \{ \tilde{p}^{(n)}_h \}_{n \geq 0} \) and learning rate \( \alpha_n \):

\[
\tilde{v}^h_{(i)}(n+1) = \tilde{v}^h_{(i)}(n) + \alpha_n I_{s_{n}=s_i} \left( \tilde{h}^i(n) - \tilde{v}^h_{(i)}(n) + M_i(n+1) \right)
\]

for all \( i \in [N_h] \). Then, the additional operator \( \tilde{A} \) for Bellman operator exists, as we have to use approximation of objective function in high dimensions for Bellman operator. So, if we let \( \tilde{T}^h \) be the discrete version of the previously defined \( \tilde{T}^h \) (this operator was previously \( \mathbb{R}^S \rightarrow \mathbb{R}^S \) but now is restricted to \( \mathbb{R}^{N_h} \)):

\[
\tilde{T}^h = [\tilde{T}^h_1, ..., \tilde{T}^h_{N_h}]^T
\]

\[
\tilde{T}^h_i(x_1, ..., x_{N_h}) = \sum_{i \in [N_h]} \tilde{p}_h(s_i, s_j) x_j + \sum_{i \in [N_h]} \tilde{p}_h(s_i, s_j) \tilde{r}_h(s_i, s_j)
\]

for \( i \in [N_h] \) and we let

\[
M_i(n+1) = [M_i(n+1), ..., M_i(n+1)]
\]

\[
M_i(n+1) = (\tilde{r}_h(s_n, s_{n+1}) + \tilde{v}^h_{(n)}(s_{n+1})) - \tilde{h}_i(n)
\]

for \( i \in [N_h] \). Then it follows from (20) that an operator \( \tilde{T}^h \) is a contraction w.r.t the norm induced from the eigenvector \( w \in \mathbb{R}^{N_h} \) of the transition matrix \( \tilde{p}_h \) (restricted on \( \tilde{S}^h \)):

\[
\| \tilde{T}^h(x) - \tilde{T}^h(y) \|_w \leq K \frac{\|x\|_w}{\alpha} \quad (30)
\]

where \( \|x\|_w = \max_{i \in [N_h]} \|x_i\| \) and \( \alpha \in (0, 1) \). It follows from Theorem 1 of (20) that limit will be almost surely on \( \tilde{S}^h \).

Next, to analyse the effect for parameterized value function in (23), we introduce approximations of the Bellman operator from Approximate Value Iteration (AVI) as in (52), which is a stochastic iterative of value iteration that are asymptotically biased/noisy and we denote this approximation by \( \tilde{A} \). The use of \( \tilde{A} \) allows a generic framework for analysis since a variety of approximations for Bellman operator exists, as we have to use approximation of objective function in high dimensions such as DNN, ANN in Deep RL or linear function approximators, or Gaussian Process and so on. Typically, if we rewrite (23) in a general form (not necessarily restricted to asynchronous update anymore)

\[
\tilde{v}^h_{(n+1)} = \tilde{v}^h_{(n)} + \alpha_n (\tilde{T}^h \tilde{v}^h_{(n)} - \tilde{v}^h_{(n)} + M_{n+1}),
\]

then, the additional operator \( \tilde{A} \) in a parameterized setting (in this case also discretized on \( \tilde{S}^h \) but not in general) gives update

\[
\tilde{v}^h_{(n+1)} = \tilde{v}^h_{(n)} + \alpha_n (\tilde{A} \tilde{v}^h_{(n)} - \tilde{v}^h_{(n)} + M_{n+1}),
\]

or more generally

\[
\tilde{v}^h_{(n+1)} = \tilde{v}^h_{(n)} + \alpha_n (\tilde{T}^h \tilde{v}^h_{(n)} - \tilde{v}^h_{(n)} + M_{n+1}),
\]

where \( \epsilon_n \) represents the approximation error \( \tilde{A} \tilde{v}^h_{(n)} - \tilde{v}^h_{(n)} \) but also could include additional noise term. For example, if in our update (29) we were to use \( \tilde{p}^{(n)}_h(s_{n+1}, s_{n+1}) \) similar to (11) instead of \( \tilde{p}^{(n)}_h(s_{n, s_{n+1}}) \), we can still bound their difference by \( O(h) \) term using Assumption (1) and Definition (1) which we may also incorporate into \( \epsilon_n \).

On the other hand, depending on the situation, \( \tilde{A} \) would be different. For example, in linear function approximators, \( \tilde{A} \) would be a projection operator that map the product of Bellman operator into the spaces spanned by the finite-dimensional basis functions. For DNN method in APE, it involves online learning and approximating bellman operator by updating the parameters which also shares a similar flavor as (22). A strong condition which controls the difference between AVI and VI is to require the (AV3) condition in (53).
Assumption 2.

\[
\limsup_{n \to \infty} \| \epsilon_n \|_w \leq \epsilon_p,
\]

for some fixed \( \epsilon_p > 0 \).

Other conditions (AV1)-(AV5) in [52] have all been discussed in the previous sections. It then follows from Theorem 3 in [52] that AVI converges to some fixed point \( \tilde{v}^{\psi}_{\psi} \), satisfying

\[
\| \tilde{T}^h \tilde{v}^{\psi}_{\psi} \|_w \leq \epsilon_p.
\]

(32)

Notice since \( \tilde{T}^h \) is a contraction w.r.t \( || \cdot ||_w \) norm and \( \tilde{v}^{\psi,\star} \) is the fixed point of \( \tilde{T}^h \), i.e., \( \tilde{v}^{\psi,\star} = \tilde{T}^h \tilde{v}^{\psi,\star} \). Thus, (32) implies

\[
\lim_{\epsilon_p \to 0} \| \tilde{v}^{\psi}_{\psi} \|_w \to 0
\]

(33)

Thus, if we take the view that our update is a special situation where \( h \) is taken to be extremely small along with (27), we may view our constructed update \( \tilde{v}^h_{\psi(n)} \) as a good approximation for the \( v^\star_{\psi(n)} \) derived from (13):

\[
v^\star_{\psi(n)} \approx \tilde{v}^h_{\psi(n)}.
\]

(34)

However, for the analysis of \( \tilde{v}^h_{\psi(n)} \), under the assumptions aforementioned, we have Lemma 2, (32), (33) and (34) which suggest that

\[
\lim_{n \to \infty} \tilde{v}^h_{\psi(n)} \approx v^\star.
\]

(35)

D.4 Concluding remarks on convergence and practical implementation

The convergence analysis relies on several strong assumptions and is not always met in practice. For example, it is well-known Assumption 2 may not hold and Q-learning with function approximator is not necessarily convergent. However, the analysis above provides us with valuable guidelines on how one may improve the practical implementation of APE. We summarize them as follows.

- The \( \epsilon_p \) in Assumption 2 depends on how expressive \( \{ v^\psi_{\psi} \} \) in terms of approximating \( v^\star \). For example, for function approximators, it is known in [42] that \( || \Pi \tilde{v}^{\psi,\star} \|_\infty \) is bounded by a constant times \( || \Pi \tilde{v}^{\psi,\star} \|_\infty \) where \( \Pi \) is the projection operator which maps function into the space spanned by \( \{ v^\psi_{\psi} \} \). The more expressive is \( \{ v^\psi_{\psi} \} \), the more accurate we can bound \( || \tilde{v}^{\psi}_{\psi} \|_\infty \). Thus, in practice, when one found APE is having a hard time exhibiting convergence, it is sensible to try a more expressive collection to approximate \( v^\star \).

- In Lemma 2 we saw the approximation error is related to discretization parameter \( h \). Smaller \( h \) leads to higher accuracy, which in some sense justifies our implementation of APE which directly updates the value on a continuous state \( S \). It can be viewed as our artificial update (28) on an infinitely small discretization \( h \). However, to make the numerical update more robust, we sample a buffer \( D \) and aggregate our value update as opposed to the update (28).

- The choice adaptive sampling distribution is not directly mentioned in the analysis above. However, the approximation error in Lemma 2 also has a implicit dependence on a hidden parameter \( \gamma \) which determines how close is \( \mathbb{E}[\sum_{t=1}^{T+1} \gamma^{t-1} r(s_{t-1}, s_t)] \) from \( \mathbb{E}[\sum_{t=1}^{T} r(s_{t-1}, s_t)] \). The difference between these two quantities largely depend on how fast the tail of \( \tau \) diminishes. The almost only effective way to control this is to adaptively choose importance sampling distribution approaching the “zero-variance” sampling distribution \( p^\star \).