Generalized Off-Policy Actor-Critic

Shangtong Zhang, Wendelin Boehmer, Shimon Whiteson
Department of Computer Science
University of Oxford
{shangtong.zhang, wendelin.boehmer, shimon.whiteson}@cs.ox.ac.uk

Abstract

We propose a new objective, the *counterfactual objective*, unifying existing objectives for off-policy policy gradient algorithms in the continuing reinforcement learning (RL) setting. Compared to the commonly used *excursion objective*, which can be misleading about the performance of the target policy when deployed, our new objective better predicts such performance. We prove the Generalized Off-Policy Policy Gradient Theorem to compute the policy gradient of the counterfactual objective and use an *emphatic* approach to get an unbiased sample from this policy gradient, yielding the Generalized Off-Policy Actor-Critic (Geoff-PAC) algorithm. We demonstrate the merits of Geoff-PAC over existing algorithms in Mujoco robot simulation tasks, the first empirical success of emphatic algorithms in prevailing deep RL benchmarks.

1 Introduction

Reinforcement learning (RL) algorithms based on the policy gradient theorem (Sutton et al., 2000; Marbach and Tsitsiklis, 2001) have recently enjoyed great success in various domains, e.g., achieving human-level performance on Atari games (Mnih et al., 2016). The original policy gradient theorem is on-policy and used to optimize the *on-policy objective*. However, in many cases, we would prefer to learn off-policy to improve data efficiency (Lin, 1992) and exploration (Osband et al., 2018). To this end, the Off-Policy Policy Gradient (OPPG) Theorem (Degris et al., 2012; Maei, 2018; Imani et al., 2018) was developed and has been widely used (Silver et al., 2014; Lillicrap et al., 2015; Wang et al., 2016; Gu et al., 2017; Ciosek and Whiteson, 2017; Espeholt et al., 2018).

Ideally, an off-policy algorithm should optimize the off-policy analogue of the on-policy objective. In the continuing RL setting, this analogue would be the performance of the target policy in expectation w.r.t. the stationary distribution of the *target policy*, which is referred to as the *alternative life objective* (Ghiassian et al., 2018). This objective corresponds to the performance of the target policy when deployed. Previously, OPPG optimizes a different objective, the performance of the target policy in expectation w.r.t. the stationary distribution of the *behavior policy*. This objective is referred to as the *excursion objective* (Ghiassian et al., 2018), as it corresponds to the excursion setting (Sutton et al., 2016). Unfortunately, the excursion objective can be misleading about the performance of the target policy when deployed, as we illustrate in Section 3.

It is infeasible to optimize the alternative life objective directly in the off-policy continuing setting. Instead, we propose to optimize the *counterfactual objective*, which approximates the alternative life objective. In the excursion setting, an agent in the stationary distribution of the behavior policy considers a hypothetical excursion that follows the target policy. The return from this hypothetical excursion is an indicator of the performance of the target policy. The excursion objective measures this return w.r.t. the stationary distribution of the behavior policy, using samples generated by executing the behavior policy. By contrast, evaluating the alternative life objective requires samples from the stationary distribution of the target policy, to which the agent does not have access. In the counterfactual objective, we use a new parameter $\hat{\gamma}$ to control *how counterfactual the objective is*,

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

akin to Gelada and Bellemare (2019). With $\hat{\gamma} = 0$, the counterfactual objective uses the stationary distribution of the behavior policy to measure the performance of the target policy, recovering the excursion objective. With $\hat{\gamma} = 1$, the counterfactual objective is fully decoupled from the behavior policy and uses the stationary distribution of the target policy to measure the performance of the target policy, recovering the alternative life objective. As in the excursion objective, the excursion is never actually executed and the agent always follows the behavior policy.

We make two contributions in this paper. First, we prove the Generalized Off-Policy Policy Gradient (GOPPG) Theorem, which gives the policy gradient of the counterfactual objective. Second, using an emphatic approach (Sutton et al., 2016) to compute an unbiased sample for this policy gradient, we develop the Generalized Off-Policy Actor-Critic (Geoff-PAC) algorithm. We evaluate Geoff-PAC empirically in challenging robot simulation tasks with neural network function approximators. Geoff-PAC outperforms the actor-critic algorithms proposed by Degris et al. (2012); Imani et al. (2018), and to our best knowledge, Geoff-PAC is the first empirical success of emphatic algorithms in prevailing deep RL benchmarks.

2 Background

We use a time-indexed capital letter (e.g., X_t) to denote a random variable. We use a bold capital letter (e.g., $\mathbf{X}$) to denote a matrix and a bold lowercase letter (e.g., $\mathbf{x}$) to denote a column vector. If $x : \mathcal{S} \to \mathbb{R}$ is a scalar function defined on a finite set $\mathcal{S}$, we use its corresponding bold lowercase letter to denote its vector form, i.e., $\mathbf{x} \doteq [x(s_1), \dots, x(s_{|\mathcal{S}|})]^\mathrm{T}$. We use $\mathbf{I}$ to denote the identity matrix and $\mathbf{1}$ to denote an all-one column vector.

We consider an infinite horizon MDP (Puterman, 2014) consisting of a finite state space $\mathcal{S}$, a finite action space $\mathcal{A}$, a bounded reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and a transition kernel $p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$. We consider a transition-based discount function (White, 2017) $\gamma : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ for unifying continuing tasks and episodic tasks. At time step t, an agent at state S_t takes an action A_t according to a policy $\pi : \mathcal{A} \times \mathcal{S} \to [0, 1]$. The agent then proceeds to a new state S_{t+1} according to p and gets a reward R_{t+1} satisfying $\mathbb{E}[R_{t+1}] = r(S_t, A_t)$. The return of π at time step t is $G_t \doteq \sum_{i=0}^{\infty} \Gamma_t^{i-1} R_{t+1+i}$, where $\Gamma_t^{i-1} \doteq \Pi_{j=0}^{i-1} \gamma(S_{t+j}, A_{t+j}, S_{t+j+1})$ and $\Gamma_t^{-1} \doteq 1$. We use v_π to denote the value function of π, which is defined as $v_\pi(s) \doteq \mathbb{E}_\pi[G_t | S_t = s]$. Like White (2017), we assume v_π exists for all s. We use $q_\pi(s, a) \doteq \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$ to denote the state-action value function of π. We use $\mathbf{P}_\pi$ to denote the transition matrix induced by π, i.e., $\mathbf{P}_\pi[s, s'] \doteq \sum_a \pi(a|s) p(s'|s, a)$. We assume the chain induced by π is ergodic and use $\mathbf{d}_\pi$ to denote its unique stationary distribution. We define $\mathbf{D}_\pi \doteq diag(\mathbf{d}_\pi)$.

In the off-policy setting, an agent aims to learn a target policy π but follows a behavior policy μ. We use the same assumption of coverage as Sutton and Barto (2018), i.e., $\forall (s, a), \pi(a|s) > 0 \implies \mu(a|s) > 0$. We assume the chain induced by μ is ergodic and use $\mathbf{d}_\mu$ to denote its stationary distribution. Similarly, $\mathbf{D}_\mu \doteq diag(\mathbf{d}_\mu)$. We define $\rho(s, a) \doteq \frac{\pi(a|s)}{\mu(a|s)}$, $\rho_t \doteq \rho(S_t, A_t)$ and $\gamma_t \doteq \gamma(S_{t-1}, A_{t-1}, S_t)$.

Typically, there are two kinds of tasks in RL, prediction and control.

Prediction: In prediction, we are interested in finding the value function v_π of a given policy π. Temporal Difference (TD) learning (Sutton, 1988) is perhaps the most powerful algorithm for prediction. TD enjoys convergence guarantee in both on- and off-policy tabular settings. TD can also be combined with linear function approximation. The update rule for on-policy linear TD is $w \leftarrow w + \alpha \Delta_t$, where α is a step size and $\Delta_t \doteq [R_{t+1} + \gamma V(S_{t+1}) - V(S_t)] \nabla_w V(S_t)$ is an incremental update. Here we use V to denote an estimate of v_π parameterized by w. Tsitsiklis and Van Roy (1997) prove the convergence of on-policy linear TD. In off-policy linear TD, the update Δ_t is weighted by ρ_t. The divergence of off-policy linear TD is well documented (Tsitsiklis and Van Roy, 1997). To approach this issue, Gradient TD methods (Sutton et al. 2009) were proposed. Instead of bootstrapping from the prediction of a successor state like TD, Gradient TD methods compute the gradient of the projected Bellman error directly. Gradient TD methods are true stochastic gradient methods and enjoy convergence guarantees. However, they are usually two-time-scale, involving two sets of parameters and two learning rates, which makes it hard to use in practice (Sutton et al., 2016). To approach this issue, Emphatic TD (ETD, Sutton et al. 2016) was proposed.

ETD introduces an interest function $i : \mathcal{S} \to [0, \infty)$ to specify user's preferences for different states. With function approximation, we typically cannot get accurate predictions for all states and must thus trade off between them. States are usually weighted by $d_\mu(s)$ in the off-policy setting (e.g., Gradient TD methods) but with the interest function, we can explicitly weight them by $d_\mu(s)i(s)$ in our objective. Consequently, we weight the update at time t via M_t, which is the *emphasis* that accumulates previous interests in a certain way. In the simplest form of ETD, we have $M_t \doteq i(S_t) + \gamma_t \rho_{t-1} M_{t-1}$. The update Δ_t is weighted by $\rho_t M_t$. In practice, we usually set $i(s) \equiv 1$.

Inspired by ETD, Hallak and Mannor (2017) propose to weight Δ_t via $\rho_t \bar{c}(S_t)$ in the Consistent Off-Policy TD (COP-TD) algorithm, where $\bar{c}(s) \doteq \frac{d_\pi(s)}{d_\mu(s)}$ is the density ratio, which is also known as the covariate shift (Gelada and Bellemare, 2019). To learn $\bar{c}$ via stochastic approximation, Hallak and Mannor (2017) propose the COP operator. However, the COP operator does not have a unique fixed point. Extra normalization and projection is used to ensure convergence (Hallak and Mannor, 2017) in the tabular setting. To address this limitation, Gelada and Bellemare (2019) further propose the $\hat{\gamma}$-discounted COP operator.

Gelada and Bellemare (2019) define a new transition matrix $\mathbf{P}_{\hat{\gamma}} \doteq \hat{\gamma}\mathbf{P}_\pi + (1-\hat{\gamma})\mathbf{1}\mathbf{d}_\mu^\mathrm{T}$ where $\hat{\gamma} \in [0, 1]$ is a constant. Following this matrix, an agent either proceeds to the next state according to $\mathbf{P}_\pi$ w.p. $\hat{\gamma}$ or gets reset to $\mathbf{d}_\mu$ w.p. $1 - \hat{\gamma}$. Gelada and Bellemare (2019) show the chain under $\mathbf{P}_{\hat{\gamma}}$ is ergodic. With $\mathbf{d}_{\hat{\gamma}}$ denoting its stationary distribution, they prove

$$\mathbf{d}_{\hat{\gamma}} = (1 - \hat{\gamma})(\mathbf{I} - \hat{\gamma}\mathbf{P}_\pi^\mathrm{T})^{-1}\mathbf{d}_\mu \quad (\hat{\gamma} < 1) \tag{1}$$

and $\mathbf{d}_{\hat{\gamma}} = \mathbf{d}_\pi$ $(\hat{\gamma} = 1)$. With $c(s) \doteq \frac{d_{\hat{\gamma}}(s)}{d_\mu(s)}$, Gelada and Bellemare (2019) prove that

$$\mathbf{c} = \hat{\gamma}\mathbf{D}_\mu^{-1}\mathbf{P}_\pi^\mathrm{T}\mathbf{D}_\mu\mathbf{c} + (1 - \hat{\gamma})\mathbf{1}, \tag{2}$$

yielding the following learning rule for estimating c in the tabular setting:

$$C(S_{t+1}) \leftarrow C(S_{t+1}) + \alpha[\hat{\gamma}\rho_t C(S_t) + (1 - \hat{\gamma}) - C(S_{t+1})], \tag{3}$$

where C is an estimate of c and α is a step size. A semi-gradient is used when C is a parameterized function (Gelada and Bellemare, 2019). For a small $\hat{\gamma}$ (depending on the difference between π and μ), Gelada and Bellemare (2019) prove a multi-step contraction under linear function approximation. For a large $\hat{\gamma}$ or nonlinear function approximation, they provide an extra normalization loss for the sake of the constraint $\mathbf{d}_\mu^\mathrm{T}\mathbf{c} = \mathbf{1}^\mathrm{T}\mathbf{d}_{\hat{\gamma}} = 1$. Gelada and Bellemare (2019) use $\rho_t c(S_t)$ to weight the update Δ_t in Discounted COP-TD. They demonstrate empirical success in Atari games (Bellemare et al., 2013) with pixel inputs.

Control: In this paper, we focus on policy-based control. In the on-policy continuing setting, we seek to optimize the average value objective (Silver, 2015)

$$J_\pi \doteq \textstyle\sum_s d_\pi(s) i(s) v_\pi(s). \tag{4}$$

Optimizing the average value objective is equivalent to optimizing the average reward objective (Puterman, 2014) if both γ and i are constant (see White 2017). In general, the average value objective can be interpreted as a generalization of the average reward objective to adopt transition-based discounting and nonconstant interest function.

In the off-policy continuing setting, Imani et al. (2018) propose to optimize the excursion objective

$$J_\mu \doteq \textstyle\sum_s d_\mu(s) i(s) v_\pi(s) \tag{5}$$

instead of the alternative life objective J_π. The key difference between J_π and J_μ is how we trade off different states. With function approximation, it is usually not possible to maximize $v_\pi(s)$ for all states, which is the first trade-off we need to make. Moreover, visiting one state more implies visiting another state less, which is the second trade-off we need to make. J_μ and J_π achieve both kinds of trade-off according to d_μ and d_π respectively. However, it is J_π, not J_μ, that correctly reflects the deploy-time performance of π, as the behavior policy will no longer matter when we deploy the off-policy learned π in a continuing task.

In both objectives, $i(s)$ is usually set to 1. We assume π is parameterized by θ. In the rest of this paper, all gradients are taken w.r.t. θ unless otherwise specified, and we consider the gradient for only

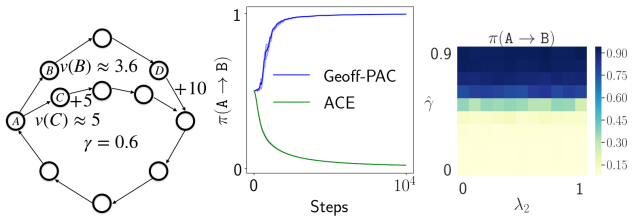


Figure 1: (a) The two-circle MDP. Rewards are 0 unless specified on the edge (b) The probability of transitioning to B from A under target policy π during training (c) The influence of $\hat{\gamma}$ and λ_2 on the final solution found by Geoff-PAC.

one component of θ for the sake of clarity. It is not clear how to compute the policy gradient of J_π in the off-policy continuing setting directly. For J_μ, we can compute the policy gradient as

$$\nabla J_\mu = \sum_s d_\mu(s) i(s) \sum_a \big(q_\pi(s,a)\nabla\pi(a|s) + \pi(a|s)\nabla q_\pi(s,a)\big). \tag{6}$$

Degris et al. (2012) prove in the Off-Policy Policy Gradient (OPPG) theorem that we can ignore the term $\pi(s,a)\nabla q_\pi(s,a)$ without introducing bias for a tabular policy[1] when $i(s) \equiv 1$, yielding the Off-Policy Actor Critic (Off-PAC), which updates θ as

$$\theta_{t+1} = \theta_t + \alpha\rho_t q_\pi(S_t, A_t)\nabla \log \pi(A_t|S_t), \tag{7}$$

where α is a step size, S_t is sampled from $\mathbf{d}_\mu$, and A_t is sampled from $\mu(\cdot|S_t)$. For a policy using a general function approximator, Imani et al. (2018) propose a new OPPG theorem. They define

$$F_t^{(1)} \doteq i(S_t) + \gamma_t\rho_{t-1}F_{t-1}^{(1)}, \quad M_t^{(1)} \doteq (1-\lambda_1)i(S_t) + \lambda_1 F_t^{(1)},$$
$$Z_t^{(1)} \doteq \rho_t M_t^{(1)} q_\pi(S_t, A_t)\nabla \log \pi(A_t|S_t),$$

where $\lambda_1 \in [0,1]$ is a constant used to optimize the bias-variance trade-off and $F_{-1}^{(1)} \doteq 0$. Imani et al. (2018) prove that $Z_t^{(1)}$ is an unbiased sample of ∇J_μ in the limiting sense if $\lambda_1 = 1$ and π is fixed, i.e., $\lim_{t\to\infty} \mathbb{E}_\mu[Z_t^{(1)}] = \nabla J_\mu$. Based on this, Imani et al. (2018) propose the Actor-Critic with Emphatic weightings (ACE) algorithm, which updates θ as $\theta_{t+1} = \theta_t + \alpha Z_t^{(1)}$. ACE is an emphatic approach where $M_t^{(1)}$ is the emphasis to reweigh the update.

3 The Counterfactual Objective

We now introduce the counterfactual objective

$$J_{\hat{\gamma}} \doteq \sum_s d_{\hat{\gamma}}(s)\hat{i}(s)v_\pi(s), \tag{8}$$

where $\hat{i}$ is a user-defined interest function. Similarly, we can set $\hat{i}(s)$ to 1 for the continuing setting but we proceed with a general $\hat{i}$. When $\hat{\gamma} = 1$, $J_{\hat{\gamma}}$ recovers the alternative life objective J_π. When $\hat{\gamma} = 0$, $J_{\hat{\gamma}}$ recovers the excursion objective J_μ. To motivate the counterfactual objective $J_{\hat{\gamma}}$, we first present the two-circle MDP (Figure 1a) to highlight the difference between J_π and J_μ.

In the two-circle MDP, an agent needs to make a decision only in state A. The behavior policy μ proceeds to B or C randomly with equal probability. For this continuing task, the discount factor γ is always 0.6 and the interest is always 1. Under this task specification (White, 2017), the optimal policy under the alternative life objective J_π, which is equivalent to the average reward objective as γ and i are constant, is to stay in the outer circle. However, to maximize J_μ, the agent prefers the inner circle. To see this, first note $v_\pi(\texttt{B})$ and $v_\pi(\texttt{C})$ hardly change w.r.t. π, and we have $v_\pi(\texttt{B}) \approx 3.6$ and $v_\pi(\texttt{C}) \approx 5$. To maximize J_μ, the target policy π would prefer transitioning to state C to maximize $v_\pi(\texttt{A})$. The agent, therefore, remains in the inner circle. The two-circle MDP is tabular, so the policy

[1] See Errata in Degris et al. (2012), also in Imani et al. (2018); Maei (2018).

maximizing $v_\pi(s)$ for all s can be represented accurately. If we consider an episodic task, e.g., we aim to maximize only $v_\pi(\texttt{A})$ and set the discount of the transition back to A to 0, such policy will be optimal under the episodic return criterion. However, when we consider a continuing task and aim to optimize J_π, we have to consider state visitation. The excursion objective J_μ maximizes $v_\pi(\texttt{A})$ regardless of the state visitation under π, yielding a policy π that never visits the state D, a state of the highest value. Such policy is sub-optimal in this continuing task. To maximize J_π, the agent has to sacrifice $v_\pi(\texttt{A})$ and visits state D more. This two-circle MDP is not an artifact due to the small γ. The same effect can also occur with a larger γ if the path is longer. With function approximation, the discrepancy between J_μ and J_π can be magnified as we need to make trade-off in both maximizing $v_\pi(s)$ and state visitation.

One solution to this problem is to set the interest function i in J_μ in a clever way. However, it is not clear how to achieve this without domain knowledge. Imani et al. (2018) simply set i to 1. Another solution might be to optimize J_π directly in off-policy learning, if one could use importance sampling ratios to fully correct $\mathbf{d}_\mu$ to $\mathbf{d}_\pi$ as Precup et al. (2001) propose for value-based methods in the episodic setting. However, this solution is infeasible for the continuing setting (Sutton et al., 2016). One may also use differential value function (Sutton and Barto, 2018) to replace the discounted value function in J_μ. Off-policy policy gradient with differential value function, however, is still an open problem and we are not aware of existing literature on this.

In this paper, we propose to optimize $J_{\hat{\gamma}}$ instead. It is a well-known fact that the policy gradient of the stationary distribution exists under mild conditions (e.g., see Yu (2005)).[2] It follows immediately from the proof of this existence that $\lim_{\hat{\gamma}\to 1}\mathbf{d}_{\hat{\gamma}} = \mathbf{d}_\pi$. Moreover, it is trivial to see that $\lim_{\hat{\gamma}\to 0}\mathbf{d}_{\hat{\gamma}} = \mathbf{d}_\mu$, indicating the counterfactual objective can recover both the excursion objective and the alternative life objective smoothly. Furthermore, we show empirically that a small $\hat{\gamma}$ (e.g., 0.6 in the two-circle MDP and 0.2 in Mujoco tasks) is enough to generate a different solution from maximizing J_μ.

4 Generalized Off-Policy Policy Gradient

In this section, we derive an estimator for $\nabla J_{\hat{\gamma}}$ and show in Proposition 1 that it is unbiased in the limiting sense. Our (standard) assumptions are given in supplementary materials. The OPPG theorem (Imani et al., 2018) leaves us the freedom to choose the interest function i in J_μ. In this paper, we set $i(s) \doteq \hat{i}(s)c(s)$, which, to our best knowledge, is the first time that a non-trivial interest is used. Hence, i depends on π and we cannot invoke OPPG directly as $\nabla J_\mu \neq \sum_d d_\mu(s)i(s)\nabla v_\pi(s)$. However, we can still invoke the remaining parts of OPPG:

$$\sum_s d_\mu(s)i(s)\nabla v_\pi(s) = \sum_s m(s)\sum_a q_\pi(s,a)\nabla\pi(a|s), \tag{9}$$

where $\mathbf{m}^\mathrm{T} \doteq \mathbf{i}^\mathrm{T}\mathbf{D}_\mu(\mathbf{I}-\mathbf{P}_{\pi,\gamma})^{-1}, \mathbf{P}_{\pi,\gamma}[s,s'] \doteq \sum_a \pi(a|s)p(s'|s,a)\gamma(s,a,s')$. We now compute the gradient $\nabla J_{\hat{\gamma}}$.

Theorem 1 (Generalized Off-Policy Policy Gradient Theorem)

$$\nabla J_{\hat{\gamma}} = \underbrace{\sum_s m(s)\sum_a q_\pi(s,a)\nabla\pi(a|s)}_{①} + \underbrace{\sum_s d_\mu(s)\hat{i}(s)v_\pi(s)g(s)}_{②} \quad \big(\hat{\gamma}<1\big)$$

where $\mathbf{g} \doteq \hat{\gamma}\mathbf{D}_\mu^{-1}(\mathbf{I}-\hat{\gamma}\mathbf{P}_\pi^\mathrm{T})^{-1}\mathbf{b}$, $\mathbf{b} \doteq \nabla\mathbf{P}_\pi^\mathrm{T}\mathbf{D}_\mu\mathbf{c}$

Proof. We first use the product rule of calculus and plug in $d_{\hat{\gamma}}(s) = d_\mu(s)c(s)$:

$$\begin{aligned}\nabla J_{\hat{\gamma}} &= \sum_s d_{\hat{\gamma}}(s)\hat{i}(s)\nabla v_\pi(s) + \sum_s \nabla d_{\hat{\gamma}}(s)\hat{i}(s)v_\pi(s)\\ &= \underbrace{\sum_s d_\mu(s)c(s)\hat{i}(s)\nabla v_\pi(s)}_{③} + \underbrace{\sum_s d_\mu(s)\nabla c(s)\hat{i}(s)v_\pi(s).}_{④}\end{aligned}$$

[2] For completeness, we include that proof in supplementary materials.

$\textcircled{1} = \textcircled{3}$ follows directly from (9). To show $\textcircled{2} = \textcircled{4}$, we take gradients on both sides of (2). We have $\nabla \mathbf{c} = \hat{\gamma}\mathbf{D}_\mu^{-1}\mathbf{P}_\pi^\mathrm{T}\mathbf{D}_\mu \nabla \mathbf{c} + \hat{\gamma}\mathbf{D}_\mu^{-1}\nabla \mathbf{P}_\pi^\mathrm{T}\mathbf{D}_\mu \mathbf{c}$. Solving this linear system of $\nabla \mathbf{c}$ leads to

$$\begin{aligned}\nabla \mathbf{c} &= (\mathbf{I} - \hat{\gamma}\mathbf{D}_\mu^{-1}\mathbf{P}_\pi^\mathrm{T}\mathbf{D}_\mu)^{-1}\hat{\gamma}\mathbf{D}_\mu^{-1}\nabla \mathbf{P}_\pi^\mathrm{T}\mathbf{D}_\mu \mathbf{c} = \left(\mathbf{D}_\mu^{-1}(\mathbf{I} - \hat{\gamma}\mathbf{P}_\pi^\mathrm{T})\mathbf{D}_\mu\right)^{-1}\hat{\gamma}\mathbf{D}_\mu^{-1}\nabla \mathbf{P}_\pi^\mathrm{T}\mathbf{D}_\mu \mathbf{c} \\ &= \left(\mathbf{D}_\mu^{-1}(\mathbf{I} - \hat{\gamma}\mathbf{P}_\pi^\mathrm{T})^{-1}\mathbf{D}_\mu\right)\hat{\gamma}\mathbf{D}_\mu^{-1}\nabla \mathbf{P}_\pi^\mathrm{T}\mathbf{D}_\mu \mathbf{c} = \mathbf{g}.\end{aligned}$$

With $\nabla c(s) = g(s)$, $\textcircled{2} = \textcircled{4}$ follows easily. □

Now we use an emphatic approach to provide an unbiased sample of $\nabla J_{\hat{\gamma}}$. We define

$$I_t \doteq c(S_{t-1})\rho_{t-1}\nabla \log \pi(A_{t-1}|S_{t-1}), \quad F_t^{(2)} \doteq I_t + \hat{\gamma}\rho_{t-1}F_{t-1}^{(2)}, \quad M_t^{(2)} \doteq (1-\lambda_2)I_t + \lambda_2 F_t^{(2)}.$$

Here I_t functions as an intrinsic interest (in contrast to the user-defined extrinsic interest $\hat{i}$) and is a sample for $\mathbf{b}$. $F_t^{(2)}$ accumulates previous interests and translates $\mathbf{b}$ into $\mathbf{g}$. λ_2 is for bias-variance trade-off similar to Sutton et al. (2016); Imani et al. (2018). We now define

$$Z_t^{(2)} \doteq \hat{\gamma}\hat{i}(S_t)v_\pi(S_t)M_t^{(2)}, \quad Z_t \doteq Z_t^{(1)} + Z_t^{(2)}$$

and proceed to show that Z_t is an unbiased sample of $\nabla J_{\hat{\gamma}}$ when $t \to \infty$.

Lemma 1 *Assuming the chain induced by μ is ergodic and π is fixed, the limit $f(s) \doteq d_\mu(s)\lim_{t\to\infty}\mathbb{E}_\mu[F_t^{(2)}|S_t = s]$ exists, and $\mathbf{f} = (\mathbf{I} - \hat{\gamma}\mathbf{P}_\pi^\mathrm{T})^{-1}\mathbf{b}$ for $\hat{\gamma} < 1$.*

Proof. Previous works (Sutton et al., 2016; Imani et al., 2018) assume $\lim_{t\to\infty}\mathbb{E}_\mu[F_t^{(1)}|S_t = s]$ exists. Here we prove the existence of $\lim_{t\to\infty}\mathbb{E}_\mu[F_t^{(2)}|S_t = s]$, inspired by the process of computing the value of $\lim_{t\to\infty}\mathbb{E}_\mu[F_t^{(1)}|S_t = s]$ (assuming its existence) in Sutton et al. (2016). The existence of $\lim_{t\to\infty}\mathbb{E}_\mu[F_t^{(1)}|S_t = s]$ with transition-dependent γ can also be established with the same routine.[3] The proof also involves similar techniques as Hallak and Mannor (2017). Details in supplementary materials. □

Proposition 1 *Assuming the chain induced by μ is ergodic, π is fixed, $\lambda_1 = \lambda_2 = 1, \hat{\gamma} < 1, i(s) \doteq \hat{i}(s)c(s)$, then $\lim_{t\to\infty}\mathbb{E}_\mu[Z_t] = \nabla J_{\hat{\gamma}}$*

Proof. The proof involves Proposition 1 in Imani et al. (2018) and Lemma 1. Details are provided in supplementary materials. □

When $\hat{\gamma} = 0$, the Generalized Off-Policy Policy Gradient (GOPPG) Theorem recovers the OPPG theorem in Imani et al. (2018). The main contribution of GOPPG lies in the computation of $\nabla \mathbf{c}$, i.e., the policy gradient of a distribution, which has not been done by previous policy gradient methods. The main contribution of Proposition 1 is the trace $F_t^{(2)}$, which is an effective way to approximate $\nabla \mathbf{c}$. Inspired by Propostion 1, we propose to update θ as $\theta_{t+1} = \theta_t + \alpha Z_t$, where α is a step size.

So far, we discussed the policy gradient for a single dimension of the policy parameter θ, so $F_t^{(1)}, M_t^{(1)}, F_t^{(2)}, M_t^{(2)}$ are all scalars. When we compute policy gradients for the whole θ in parallel, $F_t^{(1)}, M_t^{(1)}$ remain scalars while $F_t^{(2)}, M_t^{(2)}$ become vectors of the same size as θ. This is because our intrinsic interest "function" I_t is a multi-dimensional random variable, instead of a deterministic scalar function like $\hat{i}$. We, therefore, generalize the concept of interest.

So far, we also assumed access to the true density ratio c and the true value function v_π. We can plug in their estimates C and V, yielding the Generalized Off-Policy Actor-Critic (Geoff-PAC) algorithm.[4] The density ratio estimate C can be learned via the learning rule in (3). The value estimate V can be learned by any off-policy prediction algorithm, e.g., one-step off-policy TD (Sutton and Barto, 2018), Gradient TD methods, (Discounted) COP-TD or V-trace (Espeholt et al., 2018). Pseudocode of Geoff-PAC is provided in supplementary materials.

We now discuss two potential practical issues with Geoff-PAC. First, Proposition 1 requires $t \to \infty$. In practice, this means μ has been executed for a long time and can be satisfied by a warm-up before

[3] This existence does not follow directly from the convergence analysis of ETD in Yu (2015).

[4] At this moment, a convergence analysis of Geoff-PAC is an open problem.

training. Second, Proposition 1 provides an unbiased sample for a *fixed* policy π. Once π is updated, $F_t^{(1)}, F_t^{(2)}$ will be invalidated as well as C, V. As their update rule does not have a learning rate, we cannot simply use a larger learning rate for $F_t^{(1)}, F_t^{(2)}$ as we would do for C, V. This issue also appeared in Imani et al. (2018). In principle, we could store previous transitions in a replay buffer (Lin, 1992) and replay them for a certain number of steps after π is updated. In this way, we can satisfy the requirement $t \to \infty$ and get the up-to-date $F_t^{(1)}, F_t^{(2)}$. In practice, we found this unnecessary. When we use a small learning rate for π, we assume π changes slowly and ignore this invalidation effect.

5 Experimental Results

Our experiments aim to answer the following questions. 1) Can Geoff-PAC find the same solution as on-policy policy gradient algorithms in the two-circle MDP as promised? 2) How does the degree of counterfactualness ($\hat{\gamma}$) influence the solution? 3) Can Geoff-PAC scale up to challenging tasks like robot simulation in Mujoco with neural network function approximators? 4) Can the counterfactual objective in Geoff-PAC translate into performance improvement over Off-PAC and ACE? 5) How does Geoff-PAC compare with other downstream applications of OPPG, e.g., DDPG (Lillicrap et al., 2015) and TD3 (Fujimoto et al., 2018)?

5.1 Two-circle MDP

We implemented a tabular version of ACE and Geoff-PAC for the two-circle MDP. The behavior policy μ was random, and we monitored the probability from `A` to `B` under the target policy π. In Figure 1b, we plot $\pi(\texttt{A} \to \texttt{B})$ during training. The curves are averaged over 10 runs and the shaded regions indicate standard errors. We set $\lambda_1 = \lambda_2 = 1$ so that both ACE and Geoff-PAC are unbiased. For Geoff-PAC, $\hat{\gamma}$ was set to 0.9. ACE converges to the correct policy that maximizes J_μ as expected, while Geoff-PAC converges to the policy that maximizes J_π, the policy we want in on-policy training. Figure 1c shows how manipulating $\hat{\gamma}$ and λ_2 influences the final solution. In this two-circle MDP, λ_2 has little influence on the final solution, while manipulating $\hat{\gamma}$ significantly changes the final solution.

5.2 Robot Simulation

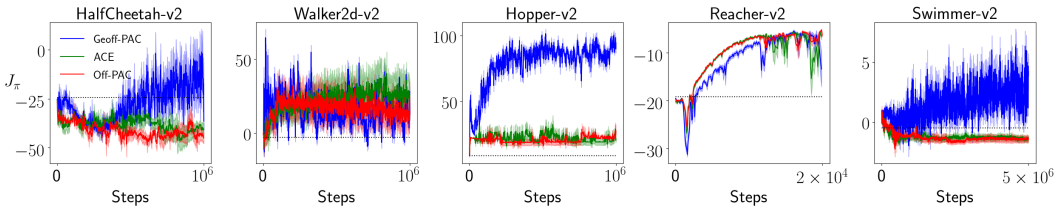


Figure 2: Comparison among Off-PAC, ACE, and Geoff-PAC. Black dash lines are random agents.

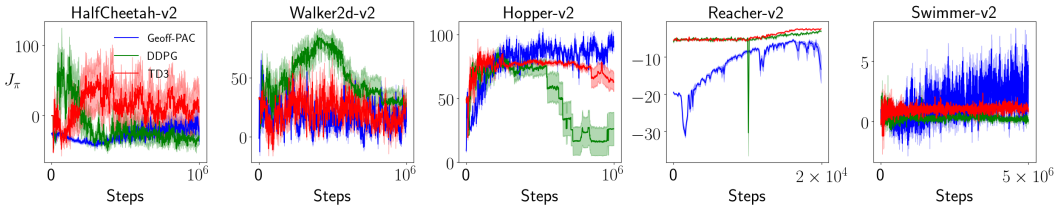


Figure 3: Comparison among DDPG, TD3, and Geoff-PAC

Evaluation: We benchmarked Off-PAC, ACE, DDPG, TD3, and Geoff-PAC on five Mujoco robot simulation tasks from OpenAI gym (Brockman et al., 2016). As all the original tasks are episodic, we adopted similar techniques as White (2017) to compose continuing tasks. We set the discount function γ to 0.99 for all non-termination transitions and to 0 for all termination transitions. The

agent was teleported back to the initial states upon termination. The interest function was always 1. This setting complies with the common training scheme for Mujoco tasks (Lillicrap et al., 2015; Asadi and Williams, 2016). However, we interpret the tasks as continuing tasks. As a consequence, J_π, instead of episodic return, is the proper metric to measure the performance of a policy π. The behavior policy μ is a fixed *uniformly random policy*, same as Gelada and Bellemare (2019). The data generated by μ is significantly different from any meaningful policy in those tasks. Thus, this setting exhibits a high degree of off-policyness. We monitored J_π periodically during training. To evaluate J_π, states were sampled according to π, and v_π was approximated via Monte Carlo return. Evaluation based on the commonly used total undiscounted episodic return criterion is provided in supplementary materials. The relative performance under the two criterion is almost identical.

Implementation: Although emphatic algorithms have enjoyed great theoretical success (Yu, 2015; Hallak et al., 2016; Sutton et al., 2016; Imani et al., 2018), their empirical success is still limited to simple domains (e.g., simple hand-crafted Markov chains, cart-pole balancing) with linear function approximation. To our best knowledge, this is the first time that emphatic algorithms are evaluated in challenging robot simulation tasks with neural network function approximators. To stabilize training, we adopted the A2C (Clemente et al., 2017) paradigm with multiple workers and utilized a target network (Mnih et al., 2015) and a replay buffer (Lin, 1992). All three algorithms share the same architecture and the same parameterization. We first tuned hyperparameters for Off-PAC. ACE and Geoff-PAC inherited common hyperparameters from Off-PAC. For DDPG and TD3, we used the same architecture and hyperparameters as Lillicrap et al. (2015) and Fujimoto et al. (2018) respectively. More details are provided in supplementary materials and all the implementations are publicly available [5].

Results: We first studied the influence of λ_1 on ACE and the influence of $\lambda_1, \lambda_2, \hat{\gamma}$ on Geoff-PAC in HalfCheetah. The results are reported in supplementary materials. We found ACE was not sensitive to λ_1 and set $\lambda_1 = 0$ for all experiments. For Geoff-PAC, we found $\lambda_1 = 0.7, \lambda_2 = 0.6, \hat{\gamma} = 0.2$ produced good empirical results and used this combination for all remaining tasks. All curves are averaged over 10 independent runs and shaded regions indicate standard errors. Figure 2 compares Geoff-PAC, ACE, and Off-PAC. Geoff-PAC significantly outperforms ACE and Off-PAC in 3 out of 5 tasks. The performance on Walker and Reacher is similar. This performance improvement supports our claim that optimizing $J_{\hat{\gamma}}$ can better approximate J_π than optimizing J_μ. We also report the performance of a random agent for reference. Moreover, this is the first time that ACE is evaluated on such challenging domains instead of simple Markov chains. Figure 3 compares Geoff-PAC, DDPG, and TD3. Geoff-PAC outperforms DDPG in Hopper and Swimmer. DDPG with a uniformly random policy exhibits high instability in HalfCheetah, Walker, and Hopper. This is expected because DDPG ignores the discrepancy between $\mathbf{d}_\mu$ and $\mathbf{d}_\pi$. As training progresses, this discrepancy gets larger and finally yields a performance drop. TD3 uses several techniques to stabilize DDPG, which translate into the performance and stability improvement over DDPG in Figure 3. However, Geoff-PAC still outperforms TD3 in Hopper and Swimmer. This is not a fair comparison in that many design choices for DDPG, TD3 and Geoff-PAC are different (e.g., one worker vs. multiple workers, deterministic vs. stochastic policy, network architectures), and we do not expect Geoff-PAC to outperform all applications of OPPG. However, this comparison does suggest GOPPG sheds light on how to improve applications of OPPG.

6 Related Work

The density ratio $\mathbf{c}$ is a key component in Geoff-PAC, which is proposed by Gelada and Bellemare (2019). However, how we use this density ratio is different. Q-Learning (Watkins and Dayan, 1992; Mnih et al., 2015) is a semi-gradient method. Gelada and Bellemare (2019) use the density ratio to reweigh the Q-Learning semi-gradient update directly. The resulting algorithm still belongs to semi-gradient methods. If we would use the density ratio to reweigh the Off-PAC update (7) directly, it would just be an actor-critic analogue of the Q-Learning approach in Gelada and Bellemare (2019). This reweighed Off-PAC, however, will no longer follow the policy gradient of the objective J_μ, yielding instead "policy semi-gradient". In this paper, we use the density ratio to define a new objective, the counterfactual objective, and compute the policy gradient of this new objective directly (Theorem 1). The resulting algorithm, Geoff-PAC, still belongs to policy gradient methods (in the

[5] `https://github.com/ShangtongZhang/DeepRL`

limiting sense). Computing the policy gradient of the counterfactual objective requires computing the policy gradient of the density ratio, which has not been explored in Gelada and Bellemare (2019).

There have been many applications of OPPG, e.g., DPG (Silver et al., 2014), DDPG, ACER (Wang et al., 2016), EPG (Ciosek and Whiteson, 2017), and IMPALA (Espeholt et al., 2018). Particularly, Gu et al. (2017) propose IPG to unify on- and off-policy policy gradients. IPG is a mix of *gradients* (i.e., a mix of ∇J_μ and ∇J_π). To compute ∇J_π, IPG does need on-policy samples. In this paper, the counterfactual objective is a mix of *objectives*, and we do not need on-policy samples to compute the policy gradient of the counterfactual objective. Mixing $\nabla J_{\hat{\gamma}}$ and ∇J_π directly in IPG-style is a possibility for future work.

There have been other policy-based off-policy algorithms. Maei (2018) provide an unbiased estimator (in the limiting sense) for ∇J_μ, assuming the value function is linear. Theoretical results are provided without empirical study. Imani et al. (2018) eliminate this linear assumption and provide a thorough empirical study. We, therefore, conduct our comparison with Imani et al. (2018) instead of Maei (2018). In another line of work, the policy entropy is used for reward shaping. The target policy can then be derived from the value function directly (O'Donoghue et al., 2016; Nachum et al., 2017a; Schulman et al., 2017a). This line of work includes the deep energy-based RL (Haarnoja et al., 2017, 2018), where a value function is learned off-policy and the policy is derived from the value function directly, and path consistency learning (Nachum et al., 2017a,b), where gradients are computed to satisfy certain path consistencies. This line of work is orthogonal to this paper, where we compute the policy gradients of the counterfactual objective directly in an off-policy manner and do not involve reward shaping.

Liu et al. (2018) prove that $\bar{c}$ is the unique solution for a minimax problem, which involves maximization over a function set $\mathcal{F}$. They show that theoretically $\mathcal{F}$ should be sufficiently rich (e.g., neural networks). To make it tractable, they restrict $\mathcal{F}$ to a ball of a reproducing kernel Hilbert space, yielding a closed form solution for the maximization step. SGD is then used to learn an estimate for $\bar{c}$ in the minimization step, which is then used for policy evaluation. In a concurrent work (Liu et al., 2019), this approximate for $\bar{c}$ is used in off-policy policy gradient for J_π, and empirical success is observed in simple domains. By contrast, our $J_{\hat{\gamma}}$ unifies J_π and J_μ, where $\hat{\gamma}$ naturally allows bias-variance trade-off, yielding an empirical success in challenging robot simulation tasks.

7 Conclusions

In this paper, we introduced the counterfactual objective unifying the excursion objective and the alternative life objective in the continuing RL setting. We further provided the Generalized Off-Policy Policy Gradient Theorem and corresponding Geoff-PAC algorithm. GOPPG is the first example that a non-trivial interest function is used, and Geoff-PAC is the first empirical success of emphatic algorithms in prevailing deep RL benchmarks. There have been numerous applications of OPPG including DDPG, ACER, IPG, EPG and IMPALA. We expect GOPPG to shed light on improving those applications. Theoretically, a convergent analysis of Geoff-PAC involving compatible function assumption (Sutton et al., 2000) or multi-timescale stochastic approximation (Borkar, 2009) is also worth further investigation.

Acknowledgments

SZ is generously funded by the Engineering and Physical Sciences Research Council (EPSRC). This project has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation programme (grant agreement number 637713). The experiments were made possible by a generous equipment grant from NVIDIA. The authors thank Richard S. Sutton, Matthew Fellows, Huizhen Yu for the valuable discussion.

References

Asadi, K. and Williams, J. D. (2016). Sample-efficient deep reinforcement learning for dialog control. *arXiv preprint arXiv:1612.06000*.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*.

Borkar, V. S. (2009). *Stochastic approximation: a dynamical systems viewpoint*. Springer.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym. *arXiv preprint arXiv:1606.01540*.

Ciosek, K. and Whiteson, S. (2017). Expected policy gradients. *arXiv preprint arXiv:1706.05374*.

Clemente, A. V., Castejón, H. N., and Chandra, A. (2017). Efficient parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1705.04862*.

Degris, T., White, M., and Sutton, R. S. (2012). Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*.

Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., and Zhokhov, P. (2017). Openai baselines. `https://github.com/openai/baselines`.

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

Fujimoto, S., van Hoof, H., and Meger, D. (2018). Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*.

Gelada, C. and Bellemare, M. G. (2019). Off-policy deep reinforcement learning by bootstrapping the covariate shift. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*.

Ghiassian, S., Patterson, A., White, M., Sutton, R. S., and White, A. (2018). Online off-policy prediction.

Gu, S. S., Lillicrap, T., Turner, R. E., Ghahramani, Z., Schölkopf, B., and Levine, S. (2017). Interpolated policy gradient: Merging on-policy and off-policy gradient estimation for deep reinforcement learning. In *Advances in Neural Information Processing Systems*.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. (2017). Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1352–1361. JMLR. org.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*.

Hallak, A. and Mannor, S. (2017). Consistent on-line off-policy evaluation. In *Proceedings of the 34th International Conference on Machine Learning*.

Hallak, A., Tamar, A., Munos, R., and Mannor, S. (2016). Generalized emphatic temporal difference learning: Bias-variance analysis. In *Proceedins of 30th AAAI Conference on Artificial Intelligence*.

Imani, E., Graves, E., and White, M. (2018). An off-policy policy gradient theorem using emphatic weightings. In *Advances in Neural Information Processing Systems*.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lin, L.-J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*.

Liu, Q., Li, L., Tang, Z., and Zhou, D. (2018). Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *Advances in Neural Information Processing Systems*.

Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. (2019). Off-policy policy gradient with state distribution correction. *arXiv preprint arXiv:1904.08473*.

Maei, H. R. (2018). Convergent actor-critic algorithms under off-policy training and function approximation. *arXiv preprint arXiv:1802.07842*.

Marbach, P. and Tsitsiklis, J. N. (2001). Simulation-based optimization of markov reward processes. *IEEE Transactions on Automatic Control*.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. (2017a). Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*.

Nachum, O., Norouzi, M., Xu, K., and Schuurmans, D. (2017b). Trust-pcl: An off-policy trust region method for continuous control. *arXiv preprint arXiv:1707.01891*.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*.

O'Donoghue, B., Munos, R., Kavukcuoglu, K., and Mnih, V. (2016). Combining policy gradient and q-learning. *arXiv preprint arXiv:1611.01626*.

Osband, I., Aslanides, J., and Cassirer, A. (2018). Randomized prior functions for deep reinforcement learning. In *Advances in Neural Information Processing Systems*.

Precup, D., Sutton, R. S., and Dasgupta, S. (2001). Off-policy temporal-difference learning with function approximation. In *Proceedings of the 18th International Conference on Machine Learning*.

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

Schulman, J., Chen, X., and Abbeel, P. (2017a). Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017b). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Seneta, E. (2006). *Non-negative matrices and Markov chains*. Springer Science & Business Media.

Silver, D. (2015). Policy gradient methods. URL: `http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/pg.pdf`.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on Machine Learning*.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*.

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

Sutton, R. S., Maei, H. R., and Szepesvári, C. (2009). A convergent $o(n)$ temporal-difference algorithm for off-policy learning with linear function approximation. In *Advances in Neural Information Processing Systems*.

Sutton, R. S., Mahmood, A. R., and White, M. (2016). An emphatic approach to the problem of off-policy temporal-difference learning. *The Journal of Machine Learning Research*.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*.

Tsitsiklis, J. N. and Van Roy, B. (1997). Analysis of temporal-diffference learning with function approximation. In *Advances in neural information processing systems*.

Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., and de Freitas, N. (2016). Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*.

Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine Learning*.

White, M. (2017). Unifying task specification in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*.

Yu, H. (2005). A function approximation approach to estimation of policy gradient for pomdp with structured policies. *The 21st Conference on Uncertainty in Artificial Intelligence*.

Yu, H. (2015). On convergence of emphatic temporal-difference learning. In *Conference on Learning Theory*.

A Assumptions and Proofs

A.1 Assumptions

We use the same standard assumptions as Yu (2015) and Imani et al. (2018).

A.2 Proof of the Existence of $\nabla \mathbf{d}_{\hat{\gamma}}$

Proof. (From Yu (2005)) The stationary distribution $\mathbf{d}_{\hat{\gamma}}$ is an eigenvector of $\mathbf{P}_{\hat{\gamma}}$ associated with the eigenvalue 1, which is the largest eigenvalue. For any primitive matrix $\mathbf{X}$, Seneta (2006) states in the proof of his Theorem 1.1 (f) that each row of $Adj(\mathbf{I}-\mathbf{X})$ is an eigenvector associated with the largest eigenvalue of the matrix $\mathbf{X}$, where $Adj(\mathbf{X})$ is the adjugate matrix of $\mathbf{X}$. As $\mathbf{P}_{\hat{\gamma}}$ is ergodic, it is primitive. Consequently, the rows of $Adj(\mathbf{I}-\mathbf{P}_{\hat{\gamma}})$ are the eigenvectors associated with the eigenvalue 1, which is the stationary distribution $\mathbf{d}_{\hat{\gamma}}$. According to fact that each element of $Adj(\mathbf{I}-\mathbf{P}_{\hat{\gamma}})$ is a polynomial of elements in $\mathbf{P}_{\hat{\gamma}}$, $\nabla \mathbf{d}_{\hat{\gamma}}$ exists whenever $\nabla \pi$ exists. Particularly, $\mathbf{d}_{\hat{\gamma}}$ is polynomials of $\hat{\gamma}$, it follows easily that $\lim_{\hat{\gamma}\to 1} \mathbf{d}_{\hat{\gamma}} = \mathbf{d}_{\pi}$. □

A.3 Proof of Lemma 1

Proof. To get $F_t^{(2)}$, we start from $F_{-1}^{(2)}$ and follow μ for t steps. The expectation is taken w.r.t. to this process of following μ for t steps. We use $p(s|\bar{s},k)$ to denote the probability of transitioning to a state s from a state $\bar{s}$ in k steps under the target policy π. We define shorthands:

$$p_t(\bar{s},\bar{a},s) \doteq \Pr_{\mu}(S_{t-1}=\bar{s}, A_{t-1}=\bar{a}|S_t=s), \tag{10}$$

$$f_t(s) \doteq \mathbb{E}_{\mu}[F_t^{(2)}|S_t=s], \tag{11}$$

$$i_t(s) \doteq \mathbb{E}_{\mu}[I_t|S_t=s]. \tag{12}$$

When $t\to\infty$, the agent goes to the stationary distribution d_μ, so

$$\lim_{t\to\infty} p_t(\bar{s},\bar{a},s) \tag{13}$$

$$= \lim_{t\to\infty} \frac{\Pr_{\mu}(S_{t-1}=\bar{s}, A_{t-1}=\bar{a}, S_t=s)}{d_\mu(s)} \quad \text{(Bayes' rule)} \tag{14}$$

$$= d_\mu(s)^{-1} d_\mu(\bar{s})\mu(\bar{a}|\bar{s})p(s|\bar{s},\bar{a}). \tag{15}$$

Consequently,

$$d_\mu(s)\rho(\bar{s},\bar{a}) \lim_{t\to\infty} p_t(\bar{s},\bar{a},s) = d_\mu(\bar{s})\pi(\bar{a}|\bar{s})p(s|\bar{s},\bar{a}). \tag{16}$$

We first compute the limit of the intrinsic interest I_t:

$$\lim_{t\to\infty} i_t(s) \tag{17}$$

$$= \lim_{t\to\infty} \sum_{\bar{s},\bar{a}} \Pr_{\mu}(S_{t-1}=\bar{s}, A_{t-1}=\bar{a}|S_t=s)\mathbb{E}_{\mu}[I_t|S_{t-1}=\bar{s}, A_{t-1}=\bar{a}] \tag{18}$$

(Law of total expectation and Markov property)

$$= d_\mu(s)^{-1} \sum_{\bar{s},\bar{a}} d_\mu(\bar{s})\mu(\bar{a}|\bar{s})p(s|\bar{s},\bar{a})c(\bar{s})\rho(\bar{s},\bar{a})\nabla \log \pi(\bar{a}|\bar{s}) \quad \text{(Eq. 13 and definition of } I_t) \tag{19}$$

$$= d_\mu(s)^{-1} \sum_{\bar{s},\bar{a}} d_\mu(\bar{s})\pi(\bar{a}|\bar{s})p(s|\bar{s},\bar{a})c(\bar{s})\nabla \log \pi(\bar{a}|\bar{s}) \tag{20}$$

$$= d_\mu(s)^{-1} \sum_{\bar{s}} d_\mu(\bar{s})c(\bar{s}) \sum_{\bar{a}} \nabla \pi(\bar{a}|\bar{s})p(s|\bar{s},\bar{a}) \tag{21}$$

$$= d_\mu(s)^{-1} b(s) \quad \text{(Definition of } \mathbf{b}) \tag{22}$$

We then expend f_t recursively:

$$
\begin{aligned}
& f_t(s_k) \\
=& \mathbb{E}_\mu[I_t + \hat{\gamma}\rho_{t-1}F_{t-1}^{(2)}|S_t = s_k] \\
=& i_t(s_k) + \hat{\gamma}\mathbb{E}_\mu[\rho_{t-1}F_{t-1}^{(2)}|S_t = s_k] \\
=& i_t(s_k) + \hat{\gamma} \sum_{s_{k-1},a_{k-1}} p_t(s_{k-1}, a_{k-1}, s_k)\mathbb{E}_\mu[\rho_{t-1}F_{t-1}^{(2)}|S_{t-1} = s_{k-1}, A_{t-1} = a_{k-1}] \\
=& i_t(s_k) + \hat{\gamma} \sum_{s_{k-1},a_{k-1}} p_t(s_{k-1}, a_{k-1}, s_k)\rho(s_{k-1}, a_{k-1})f_{t-1}(s_{k-1})] \\
=& i_t(s_k) + \hat{\gamma} \sum_{s_{k-1},a_{k-1}} p_t(s_{k-1}, a_{k-1}, s_k)\rho(s_{k-1}, a_{k-1})i_{t-1}(s_{k-1})] \\
& + \hat{\gamma}^2 \sum_{s_{k-1},a_{k-1}} p_t(s_{k-1}, a_{k-1}, s_k)\rho(s_{k-1}, a_{k-1}) \sum_{s_{k-2},a_{k-2}} p_{t-1}(s_{k-2}, a_{k-2}, s_{k-1})\rho(s_{k-2}, a_{k-2})f_{t-2}(s_{k-2}) \\
=& i_t(s_k) + \hat{\gamma} \sum_{s_{k-1},a_{k-1}} p_t(s_{k-1}, a_{k-1}, s_k)\rho(s_{k-1}, a_{k-1})i_{t-1}(s_{k-1}) \\
& + \hat{\gamma}^2 \sum_{s_{k-1},a_{k-1}} p_t(s_{k-1}, a_{k-1}, s_k)\rho(s_{k-1}, a_{k-1}) \sum_{s_{k-2},a_{k-2}} p_{t-1}(s_{k-2}, a_{k-2}, s_{k-1})\rho(s_{k-2}, a_{k-2})i_{t-2}(s_{k-2})] \\
& + \dots \\
& + \hat{\gamma}^t \sum_{s_{k-1},a_{k-1}} p_t(s_{k-1}, a_{k-1}, s_k)\rho(s_{k-1}, a_{k-1}) \cdots \sum_{s_{k-t},a_{k-t}} p_1(s_{k-t}, a_{k-t}, s_{k-t+1})\rho(s_{k-t}, a_{k-t})f_0(s_{k-t})
\end{aligned}
$$

This means we can expand $f_t(s_k)$ into $t+1$ terms. For each term, we multiply it by $d_\mu(s_k)$ and compute the limit of the product as:

$$
\begin{aligned}
& \lim_{t\to\infty} d_\mu(s_k) \sum_{s_{k-1},a_{k-1}} p_t(s_{k-1}, a_{k-1}, s_k)\rho(s_{k-1}, a_{k-1})i_{t-1}(s_{k-1}) && (23) \\
=& \sum_{s_{k-1},a_{k-1}} d_\mu(s_k)\rho(s_{k-1}, a_{k-1}) \lim_{t\to\infty} p_t(s_{k-1}, a_{k-1}, s_k) \lim_{t\to\infty} i_{t-1}(s_{k-1}) && (24) \\
=& \sum_{s_{k-1},a_{k-1}} d_\mu(s_{k-1})\pi(a_{k-1}|s_{k-1})p(s_k|s_{k-1}, a_{k-1})d_\mu(s_{k-1})^{-1}b(s_{k-1}) \quad \text{(Eq. 16 and Eq. 17)} && (25) \\
=& \sum_{s_{k-1}} b(s_{k-1})p(s_k|s_{k-1}, 1) && (26)
\end{aligned}
$$

$$
\begin{aligned}
& \lim_{t\to\infty} d_\mu(s_k) \sum_{s_{k-1},a_{k-1}} p_t(s_{k-1}, a_{k-1}, s_k)\rho(s_{k-1}, a_{k-1}) \sum_{s_{k-2},a_{k-2}} p_{t-1}(s_{k-2}, a_{k-2}, s_{k-1})\rho(s_{k-2}, a_{k-2})i_{t-2}(s_{k-2}) \\
=& \sum_{s_{k-1},a_{k-1}} d_\mu(s_k)\rho(s_{k-1}, a_{k-1}) \lim_{t\to\infty} p_t(s_{k-1}, a_{k-1}, s_k) \lim_{t\to\infty} \sum_{s_{k-2},a_{k-2}} p_{t-1}(s_{k-2}, a_{k-2}, s_{k-1})\rho(s_{k-2}, a_{k-2})i_{t-2}(s_{k-2}) \\
=& \sum_{s_{k-1},a_{k-1}} d_\mu(s_{k-1})\pi(a_{k-1}|s_{k-1})p(s_k|s_{k-1}, a_{k-1}) \lim_{t\to\infty} \sum_{s_{k-2},a_{k-2}} p_{t-1}(s_{k-2}, a_{k-2}, s_{k-1})\rho(s_{k-2}, a_{k-2})i_{t-2}(s_{k-2}) \\
=& \sum_{s_{k-1},a_{k-1}} \pi(a_{k-1}|s_{k-1})p(s_k|s_{k-1}, a_{k-1}) \sum_{s_{k-2}} b(s_{k-2})p(s_{k-1}|s_{k-2}, 1) \quad \text{(Eq. 23)} \\
=& \sum_{s_{k-2}} b(s_{k-2})p(s_k|s_{k-2}, 2)
\end{aligned}
$$

Putting all the limits together, we have

$$\begin{aligned}
& f(s_k) \\
=& d_\mu(s_k) \lim_{t\to\infty} f_t(s_k) \\
=& b(s_k) \\
& + \hat{\gamma} \sum_{s_{k-1}} b(s_{k-1}) p(s_k|s_{k-1}, 1) \\
& + \hat{\gamma}^2 \sum_{s_{k-2}} b(s_{k-2}) p(s_k|s_{k-2}, 2) \\
& + \dots
\end{aligned}$$

In a matrix form, we have

$$\mathbf{f} = \mathbf{b} + \hat{\gamma}\mathbf{P}_\pi^\mathrm{T}\mathbf{b} + (\hat{\gamma}\mathbf{P}_\pi^\mathrm{T})^2\mathbf{b} + \dots$$

It follows easily that $\mathbf{f} = (\mathbf{I} - \hat{\gamma}\mathbf{P}_\pi^\mathrm{T})^{-1}\mathbf{b}$ □

A.4 Proof of Proposition 1

Proof. From Proposition 1 in Imani et al. (2018) [6], we have

$$\lim_{t\to\infty} \mathbb{E}_\mu\Big[\rho_t M_t^{(1)} q_\pi(S_t, A_t) \nabla \log \pi(A_t|S_t)\Big] = \textcircled{1}.$$

With $\mathbf{D}_{\hat{i}} \doteq diag(\hat{\mathbf{i}})$, we have,

$$\begin{aligned}
\lim_{t\to\infty} \mathbb{E}_\mu[\hat{\gamma} v_\pi(S_t)\hat{i}(S_t) M_t^{(2)}] &= \hat{\gamma} \lim_{t\to\infty} \mathbb{E}_\mu\Big[\mathbb{E}_\mu[v_\pi(S_t)\hat{i}(S_t) F_t^{(2)}|S_t = s]\Big] \text{ (law of total expectation)} \\
&= \hat{\gamma} \lim_{t\to\infty} \sum_s d_\mu(s) \mathbb{E}_\mu[v_\pi(S_t)\hat{i}(S_t) F_t^{(2)}|S_t = s] \\
&= \hat{\gamma} \sum_s d_\mu(s)\hat{i}(s) v_\pi(s) \lim_{t\to\infty} \mathbb{E}_\mu[F_t^{(2)}|S_t = s] \text{ (conditional independence)} \\
&= \hat{\gamma} \sum_s v_\pi(s)\hat{i}(s) f(s) \\
&= \hat{\gamma} \mathbf{v}_\pi^\mathrm{T} \mathbf{D}_{\hat{i}} \mathbf{f} \\
&= \hat{\gamma} \mathbf{v}_\pi^\mathrm{T} \mathbf{D}_{\hat{i}} \mathbf{D}_\mu \mathbf{D}_\mu^{-1} (\mathbf{I} - \hat{\gamma}\mathbf{P}_\pi^\mathrm{T})^{-1} \mathbf{b} \text{ (Lemma 1)} \\
&= \mathbf{v}_\pi^\mathrm{T} \mathbf{D}_{\hat{i}} \mathbf{D}_\mu \mathbf{g} \\
&= \sum_s d_\mu(s)\hat{i}(s) v_\pi(s) g(s) = \textcircled{2}.
\end{aligned}$$

As $\nabla J_{\hat{\gamma}} = \textcircled{1} + \textcircled{2}$, we have proved $\lim_{t\to\infty} \mathbb{E}_\mu[Z_t] = \nabla J_{\hat{\gamma}}$. □

B Details of Experiments

B.1 Pseudocode of Geoff-PAC

Algorithm 1 provides the pseudocode of Geoff-PAC. `SNLoss` refers to the soft normalization loss for θ_c in Gelada and Bellemare (2019). β is the weight for the `SNLoss`.

B.2 Implementation Details

Task Selection: We use 5 Mujoco tasks from Open AI gym [7](Brockman et al., 2016). Those 5 tasks are of a medium difficulty level. The easy tasks (e.g., the pendulum tasks) and the hard tasks (e.g., the ant task and the humanoid tasks) are excluded.

[6] It can be easily verified that the dependence of i on π does not influence this proposition.

[7] `https://gym.openai.com/`

Algorithm 1: Geoff-PAC with function approximation

Input:
V: value function parameterized by θ_v
C: density ratio estimation parameterized by θ_c
π: policy function parameterized by θ
$\hat{i}$: an interest function

Initialize target networks $\theta_v^- \leftarrow \theta_v, \theta_c^- \leftarrow \theta_c$
Initialize $F^{(1)} \leftarrow 0, F^{(2)} \leftarrow 0, t \leftarrow 0$
while *True* **do**
  Sample a transition $S_t, A_t, R_{t+1}, S_{t+1}$ according to behavior policy μ
  if $t = 0$ **then**
    $t \leftarrow t + 1$
    continue
  end
  $\delta_t = R_{t+1} + \gamma_t V(S_{t+1}; \theta_v^-) - V(S_t; \theta_v)$
  Update θ_v to minimize $\rho_t \delta_t^2$
  Update θ_c to minimize $\left(\left(\hat{\gamma} \rho_t C(S_t; \theta_c^-) + (1 - \hat{\gamma}) - C(S_{t+1}; \theta_c) \right)^2 + \beta \texttt{SNLoss}(\theta_c) \right)$
  $F^{(1)} \leftarrow \gamma \rho_{t-1} F^{(1)} + \hat{i}(S_t) C(S_t; \theta_c)$
  $M^{(1)} \leftarrow (1 - \lambda_1) \hat{i}(S_t) C(S_t; \theta_c) + \lambda_1 F^{(1)}$
  $I \leftarrow C(S_{t-1}; \theta_c) \rho_{t-1} \nabla \log \pi(A_{t-1} | S_{t-1}; \theta)$
  $F^{(2)} \leftarrow \hat{\gamma} \rho_{t-1} F^{(2)} + I$
  $M^{(2)} \leftarrow (1 - \lambda_2) I + \lambda_2 F^{(2)}$
  Update θ in the direction of $\hat{\gamma} \hat{i}(S_t) V(S_t; \theta_v) M^{(2)} + \rho_t M^{(1)} \delta_t \nabla \log \pi(A_t | S_t; \theta)$
  Synchronize θ_v^-, θ_c^- with θ_v, θ_c periodically
  $t \leftarrow t + 1$
end

Function Parameterization: For Off-PAC, ACE, and Geoff-PAC, we use separate two-hidden-layer networks to parameterize C, V and π. Each hidden layer has 64 hidden units and a ReLU (Nair and Hinton, 2010) activation function. Particularly, we parameterized π as a diagonal Gaussian distribution with the mean being the output of the network. The standard derivation is a global state-independent variable. This is a common policy parameterization for continuous-action problems (Schulman et al., 2015, 2017b). For DDPG and TD3, we use the same parameterization as Lillicrap et al. (2015) and Fujimoto et al. (2018) respectively.

Hyperparameter Tuning: Our implementation is based on the A2C (Clemente et al., 2017) architecture. We first tune hyperparameters for Off-PAC based on the A2C implementation from Dhariwal et al. (2017) and previous experiences. Our ACE and Geoff-PAC implementations inherited common hyperparameters from the Off-PAC implementation without further fine-tuning. Previously, Off-PAC and ACE were evaluated on only simple domains with linear function approximation. To our best knowledge, we are the first to demonstrate an empirical success for them in challenging robot simulation tasks.

Hyperparameters of Off-PAC:
Number of workers: 10
Optimizer: RMSProp with an initial learning rate 10^{-3}
Gradient clip by norm: 0.5
Replay buffer size: 10^6
Batch size of the replay buffer: 10
Warm-up steps before learning: 100 environment steps
Target network update frequency: 200 optimization steps
Importance sampling ratio clip: [0, 2]

Additional Hyperparameters of ACE:
λ_1 : 0, tuned over $\{0, 0.1, 0.2, \ldots, 0.9, 1\}$ on HalfCheetah with a grid search

Additional Hyperparameters of Geoff-PAC:
Density ratio (C) clip: [0, 2]
`SNLoss` weight (β): 10^{-3}, suggested by Gelada and Bellemare (2019)
λ_1 : 0.7, tuned over $\{0, 0.1, 0.2, \ldots, 0.9, 1\}$ on HalfCheetah
λ_2 : 0.6, tuned over $\{0, 0.1, 0.2, \ldots, 0.9, 1\}$ on HalfCheetah
$\hat{\gamma}$: 0.2, tuned over $\{0, 0.1, 0.2, \ldots, 0.9\}$ on HalfCheetah
We first select a reasonable $\hat{\gamma}$ based on some preliminary experiments. Then λ_1 and λ_2 are tuned with a grid search.

Hyperparameters of DDPG:
We implemented DDPG ourself. With the normal behavior policy, our implementation matched the reported performance in the literature, e.g., in Fujimoto et al. (2018). We use the same hyperparameters as Lillicrap et al. (2015). We do not use batch normalization.

Hyperparameters of TD3:
We implemented TD3 ourself. With the normal behavior policy, our implementation matched the reported performance in Fujimoto et al. (2018). With a uniformly random behavior policy, our implementation outperformed the implementation from Fujimoto et al. (2018) by a large margin, we therefore use our implementation for comparison. We use the same hyperparameters as Fujimoto et al. (2018).

Computing Infrastructure:
We conducted our experiments on an Nvidia DGX-1 with PyTorch.

B.3 Other Experimental Results

We include a comparison under the total undiscounted episodic return criterion for reference. The results are reported in Figure 4 and Figure 5. All curves are averaged over 10 independent runs, and standard errors are reported as the shadow.

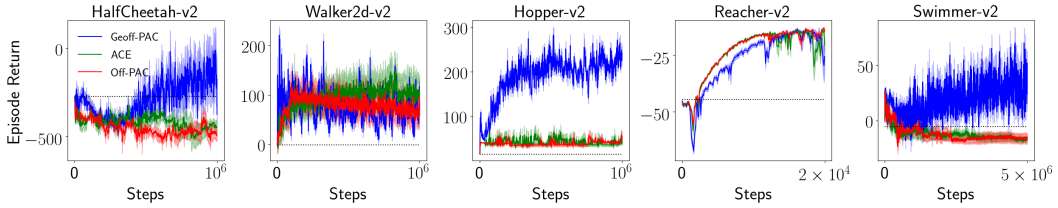


Figure 4: Comparison among Off-PAC, ACE, and Geoff-PAC. Black dash line is a random agent.

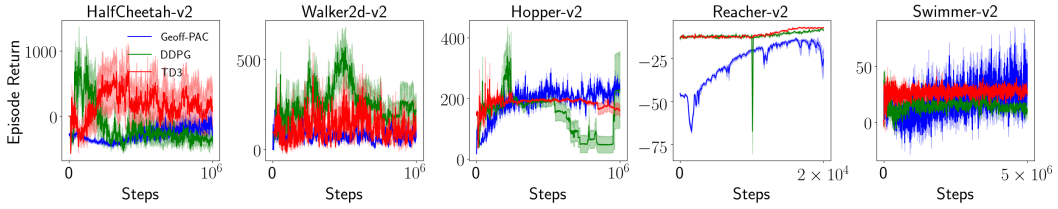
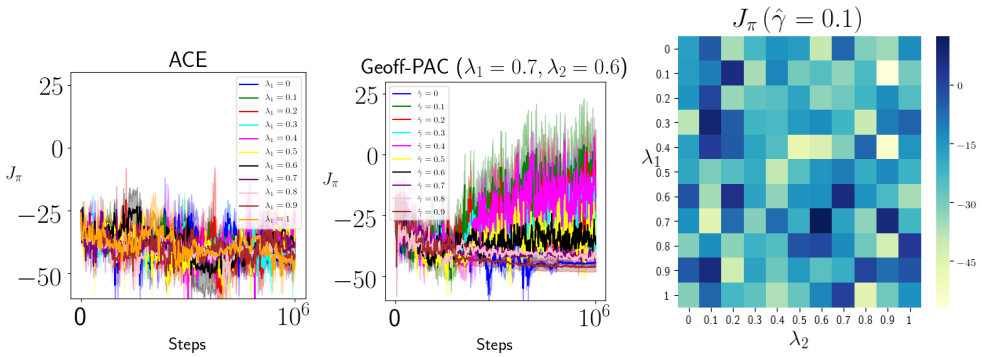


Figure 5: Comparison among DDPG, TD3, and Geoff-PAC

We studied the influence of $(\lambda_1, \lambda_2, \hat{\gamma})$ on ACE and Geoff-PAC in HalfCheetah. Results are reported in Figure 6. Five random seeds were used.



(a) (b) (c)

Figure 6: Hyper-parameter study on HalfCheetah (a) The influence of λ_1 on ACE (b) The influence of $\hat{\gamma}$ on Geoff-PAC (c) The influence of λ_1, λ_2 on Geoff-PAC