Unbiased Policy Gradient with Random Horizon

Rui Yuan
Stellantis, France

Andrii Tretynko
Vehicle Software Development, Ukraine

Simone Rossi
Stellantis, France

Thomas Hannagan
Stellantis, France

Abstract

Policy gradient (PG) methods are widely used in reinforcement learning. However, for infinite-horizon discounted reward settings, practical implementations of PG usually must rely on biased gradient estimators, due to the truncated finite-horizon sampling, which limits actual performance and hinders theoretical analysis. In this work, we introduce a new family of algorithms, *unbiased policy gradient* (UPG), that enables unbiased gradient estimators by considering finite-horizon undiscounted rewards, where the horizon is randomly sampled from a geometric distribution $\mathrm{Geom}(1-\gamma)$ associated to the discount factor γ. Thanks to the absence of bias, UPG achieves the $\mathcal{O}(\epsilon^{-4})$ sample complexity to a stationary point, which is improved by $\mathcal{O}(\log \epsilon^{-1})$, compared to the one of the vanilla PG, and is met with fewer assumptions. Our work also provides a new angle on well-known algorithms such as Q-PGT and RPG. We recover the unbiased Q-PGT algorithm as a special case of UPG, allowing for its first sample complexity analysis. We further show that UPG can be extended to α-UPG, a more generic class of PG algorithms which performs unbiased gradient estimators and notably admits RPG as a special case. The general sample complexity analysis of α-UPG that we present enables to recover the convergence rates of RPG, also with tighter bounds. Finally, we propose and evaluate two new algorithms within the UPG family: unbiased GPOMDP (UGPOMDP) and α-UGPOMDP. We show theoretically and empirically on four different environments that both UGPOMDP and α-UGPOMDP outperform its known vanilla PG counterpart, GPOMDP.

1 Introduction

Policy gradient (PG) methods are popular in reinforcement learning (RL) for computing policies that maximize long-term rewards [54, 49, 5]. The success of PG methods can be attributed to their simplicity and versatility. Indeed, PG methods can be readily implemented to solve a variety of problems, ranging from trajectory planning in non-Markovian and partially-observable environments like autonomous driving [7, 22], to more recent problems arising from the human alignment of Large Language Models [2]. Moreover, PG methods can be effectively combined with other techniques to create more sophisticated algorithms such as natural PG [19], policy mirror descent [50, 53, 16, 3], trust-region based variants [41, 43, 44], and variance-reduced methods [32, 56, 17, 18].

However, a salient issue with PG algorithm is the so-called "horizon discrepancy" – i.e., the difference between the infinite horizon assumed theoretically, and the truncated finite horizon that RL practitioners must resort to in practice when implementing PG. Due to this truncation of the horizon in experimental work, most of the aforementioned algorithms suffer from biased gradient estimators (See Appendix B for the review). This is problematic not only for the performance of the implementation, but also for the formal understanding of PG methods. For instance, Mu and Klabjan [29] recently developed a new second-order stationary point convergence analysis for biased PG: the authors report on analytical difficulties that arise from the bias inherent to horizon truncation, given that the previous analysis involving probabilistic bounds via concentration inequalities relies heavily on the absence of bias of

17th European Workshop on Reinforcement Learning (EWRL 2024).

the gradient estimator. We may ask, then, whether it is possible to obtain an unbiased PG estimator that would remove the horizon discrepancy, without sacrificing any of the other desirable properties of PG.

1.1 Outline and Contributions

In § 2 we review the fundamentals of Markov decision processes (MDPs), and describe the vanilla PG method. Our main contributions start from § 3. First, we introduce a new family of algorithms that are unbiased gradient estimators, referred to as unbiased PG (UPG) in § 3.1, by considering finite-horizon undiscounted MDP with random horizon H sampled from a geometric distribution $\mathrm{Geom}(1-\gamma)$ associated to the discount factor γ. As a special case of UPG, we propose a new algorithm unbiased GPOMDP (UGPOMDP) in (11) and Algorithm 2. The well-known unbiased Q-PGT in (9) and Algorithm 4 belongs to the UPG family as well. Then, we extend UPG to α-UPG in § 3.2 and Algorithm 5, which is also unbiased by design, and we develop two new algorithms – α-UGPOMDP in (12) and Algorithm 6 and α-QPGT in (13) and Algorithm 7, as special cases of α-UPG. When $\alpha = 0$, α-UPG recovers UPG; and when $\alpha = \frac{1}{2}$, α-QPGT recovers RPG [61] as a special case.

In § 4, we present the first-order stationary point (FOSP) convergence results of α-UPG. By leveraging the modern proof techniques of SGD with general expected smoothness Assumption 4 in optimization [20], in § 4.1 we derive a unified $\mathcal{O}(\epsilon^{-4})$ sample complexity of α-UPG, which includes the one of UPG, i.e., α-UPG with $\alpha = 0$ as a special case. Furthermore, we consider the commonly used expected Lipschitz and smooth (E-LS) policies (Assumption 5) in § 4.2 and verify in Theorem 6 that, for each of the instantiations considered in this work, such as UGPOMDP, Q-PGT, α-UGPOMDP, and α-QPGT, E-LS satisfies the expected smoothness assumption. This is the key technical contribution of our work. In particular, these four algorithms all improve the one of GPOMDP by a factor of $\mathcal{O}(\log \epsilon^{-1})$, and UGPOMDP achieves the best sample complexity $\mathcal{O}\big(\frac{1}{(1-\gamma)^6\epsilon^4}\big)$ among the others when γ is close to 1. As a by-product of our approach, we derive the first sample complexity analysis of the unbiased Q-PGT and recover the one of RPG with tighter bounds and a wider range of parameter choices.

Lastly, in § 5 we empirically compare the performance of our new algorithms UGPOMDP and α-UGPOMDP with $\alpha = 0.5$ against both the unbiased methods (Q-PGT and RPG) and the biased method (GPOMDP). Our results show that UGPOMDP and 0.5-UGPOMDP consistently outperform these methods in all four different Gym environments [51], which is consistent with our theoretical findings and supports the benefits of unbiasedness in the algorithm.

Furthermore, the theoretical assumptions we make throughout this paper are standard in the PG literature. In fact, for our main result on sample complexity, we use the weakest assumptions in the literature and match the best known results. Therefore, the limitations of our work are the same as in the PG literature in general, and mainly relate to the fact that our policy is implemented as a non-linear neural network, which does not satisfy the theoretical assumptions. We refer to Appendix A for more details.

2 Preliminaries

2.1 Markov decision process (MDP)

We consider an MDP given by $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma, \rho\}$, where $\mathcal{S}$ is a state space; $\mathcal{A}$ is an action space; $\mathcal{P}$ is a Markovian transition model, where $\mathcal{P}(s' \mid s, a)$ is the transition density from state s to s' under action a; r is the reward function, where $r(s,a) \in [-r_{\max}, r_{\max}]$ is the bounded reward for state-action pair (s,a); $\gamma \in [0,1)$ is the discount factor; and ρ is the initial state distribution. The agent's behavior is modeled as a policy $\pi \in \Delta(\mathcal{A})^{\mathcal{S}}$, where $\pi(\cdot \mid s)$ is the density distribution over action space $\mathcal{A}$ in state $s \in \mathcal{S}$.

We consider the infinite-horizon discounted setting. Let $p(\tau \mid \pi)$ be the probability density of a single trajectory $\tau = (s_0, a_0, r_0, s_1, \cdots)$ with $r_t = r(s_t, a_t)$ being sampled from π. By the Markov property of the MDP, we have $p(\tau \mid \pi) = \rho(s_0) \prod_{t=0}^{\infty} \pi(a_t \mid s_t) \mathcal{P}(s_{t+1} \mid s_t, a_t)$. With a slight abuse of notation, let $r(\tau) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$ be the total discounted reward accumulated along trajectory τ.

In the infinite-horizon discounted setting, the value function of π with an initial state s is defined as

$$V^{\pi}(s) \overset{\text{def}}{=} \mathbb{E}_{\substack{a_t \sim \pi(\cdot|s_t) \\ s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)}} \Big[\sum\nolimits_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s \Big] = \mathbb{E}_{\tau \sim p(\cdot|\pi, s_0 = s)} \Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \Big]. \quad (1)$$

Given an initial state distribution $\rho \in \Delta(\mathcal{S})$, the goal of the agent is to find a policy π that maximizes the expected value function $J(\pi) \stackrel{\text{def}}{=} \mathbb{E}_{s\sim\rho}[V^\pi(s)]$. As with the value function, for each pair $(s,a) \in \mathcal{S}\times\mathcal{A}$, the state-action value function, or *Q-function*, associated with a policy π is defined as $Q^\pi(s,a) \stackrel{\text{def}}{=} \mathbb{E}_\tau\Big[\sum_{t=0}^\infty \gamma^t r(s_t,a_t) \mid s_0 = s, a_0 = a\Big]$ and we have $V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[Q^\pi(s,a)]$.

Given a starting state distribution $\rho \in \Delta(\mathcal{S})$, we define the *state visitation distribution* $d_\rho^\pi \in \Delta(\mathcal{S})$, induced by a policy π, as $d_\rho^\pi(s) \stackrel{\text{def}}{=} (1-\gamma)\,\mathbb{E}_\tau\big[\sum_{t=0}^\infty \gamma^t \Pr^\pi(s_t = s \mid s_0)\big]$, where $\Pr^\pi(s_t = s \mid s_0)$ is the probability that the t-th state is equal to s by following the trajectory generated by π and the transition model $\mathcal{P}$ starting from s_0. Intuitively, the state visitation distribution measures the probability of being at state s across the entire trajectory.

2.2 Policy gradient

We introduce a set of parametrized policies $\{\pi^\theta : \theta \in \Theta\}$, with the assumption that π^θ is differentiable with respect to θ. To simplify notations, we use the shorthand V^θ for V^{π^θ} and similarly Q^θ for Q^{π^θ}, A^θ for A^{π^θ}, d_ρ^θ for $d_\rho^{\pi^\theta}$, $p(\tau \mid \theta)$ for $p(\tau \mid \pi^\theta)$, and $J(\theta)$ for $J(\pi^\theta)$. The policy gradient (PG) methods use gradient ascent in the parametrized space of θ to find the policy that maximizes the expected value function $J(\theta)$. That is, the policy with the *optimal parameters* $\theta^\star \in \arg\max_{\theta\in\Theta} J(\theta)$ would give the *optimal expected value function* $J^\star \stackrel{\text{def}}{=} J(\theta^\star)$. In general, $J(\theta)$ is a non-convex function with respect to θ [see, e.g., 1].

The gradient $G(\theta) \stackrel{\text{def}}{=} \nabla_\theta J(\theta)$ of the expected value function has the following structure

$$G(\theta) = \mathbb{E}_{\tau\sim p(\cdot|\theta)}\Big[\sum_{t=0}^\infty \gamma^t r(s_t,a_t) \sum_{t'=0}^\infty \nabla_\theta \log \pi^\theta(a_{t'} \mid s_{t'})\Big] \tag{2}$$

$$= \mathbb{E}_\tau\Big[\sum_{t=0}^\infty \gamma^t r(s_t,a_t)\Big(\sum_{t'=0}^t \nabla_\theta \log \pi^\theta(a_{t'} \mid s_{t'})\Big)\Big] \tag{3}$$

$$= \mathbb{E}_\tau\Big[\sum_{t=0}^\infty \Big(\sum_{t'=t}^\infty \gamma^{t'} r(s_{t'},a_{t'})\Big) \nabla_\theta \log \pi^\theta(a_t \mid s_t)\Big] \tag{4}$$

$$= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^\theta,\, a\sim\pi^\theta(\cdot|s)}\left[Q^\theta(s,a)\,\nabla_\theta \log \pi^\theta(a \mid s)\right]. \tag{5}$$

The derivations of (2)-(5) are provided in Appendix C (Lemma 4). In the rest of the paper, we omit the θ in ∇_θ for simplicity and we note $\widehat{G}(\theta)$ as an arbitrary empirical gradient estimator of $G(\theta)$.

In practice, we cannot compute the full gradient, since computing the above expectations requires averaging over all possible trajectories $\tau \sim p(\cdot \mid \theta)$. We resort to an empirical estimate of the gradient by sampling m truncated trajectories $\tau_i = \big(s_0^i, a_0^i, r_0^i, s_1^i, \cdots, s_{H-1}^i, a_{H-1}^i, r_{H-1}^i\big)$ with $r_t^i = r(s_t^i, a_t^i)$ obtained by executing π^θ for a given fixed horizon $H \in \mathbb{N}$. The resulting gradient estimator of (2) is

$$\widehat{G}^{\text{REINFORCE}}(\theta) \;=\; \frac{1}{m}\sum_{i=1}^m \sum_{t=0}^{H-1} \gamma^t r(s_t^i,a_t^i) \cdot \sum_{t'=0}^{H-1} \nabla \log \pi^\theta(a_{t'}^i \mid s_{t'}^i). \tag{6}$$

The estimator (6) is known as REINFORCE [54].

The REINFORCE estimator (6) can be simplified by leveraging the fact that future actions do not depend on past rewards. Consequently, half of the terms in (2) are removed, and this leads to the alternative formulations (3) and (4) of the full gradient. In particular, (3) leads to the following estimate of the gradient known as GPOMDP [5]

$$\widehat{G}^{\text{GPOMDP}}(\theta) \stackrel{\text{def}}{=} \frac{1}{m}\sum_{i=1}^m \sum_{t=0}^{H-1} \gamma^t r(s_t^i,a_t^i)\Big(\sum_{t'=0}^t \nabla \log \pi^\theta(a_{t'}^i \mid s_{t'}^i)\Big). \tag{7}$$

Compared to (6), (7) reduces the variance of the policy gradient estimate.

Alternatively, from (4), one can suggest the gradient estimator as

$$\widehat{G}^{\text{PGT}}(\theta) \stackrel{\text{def}}{=} \frac{1}{m}\sum_{i=1}^m \sum_{t=0}^{H-1} \Big(\sum_{t'=t}^{H-1} \gamma^{t'} r(s_{t'}^i,a_{t'}^i)\Big) \nabla \log \pi^\theta(a_t^i \mid s_t^i), \tag{8}$$

known as the policy gradient theorem (PGT) [49]. It has been shown that PGT (8) is equivalent to GPOMDP (7) [34]. Due to their equivalence, we refer to them interchangeably.

Notice that PGT (8) has also action value expressions thanks to (5) , which leads to gradient estimator as

$$\widehat{G}^{\mathrm{Q-PGT}}(\theta) \overset{\text{def}}{=} \frac{1}{m(1-\gamma)} \sum_{i=1}^{m} \widehat{Q}^{\theta}(s^i, a^i) \nabla \log \pi^{\theta}(a^i \mid s^i), \tag{9}$$

where $s^i \sim d_{\rho}^{\theta}$, $a^i \sim \pi^{\theta}(\cdot \mid s^i)$, and $\widehat{Q}^{\theta}(s^i, a^i)$ is an unbiased estimate of $Q^{\theta}(s^i, a^i)$, which can be obtained through roll-outs with random horizon, provided e.g., by Agarwal et al. [1, Algorithm 1] and Yuan et al. [59, Algorithm 3]. For the completeness, we provide Algorithm 4 to sample s^i, a^i and the unbiased estimate $\widehat{Q}^{\theta}(s^i, a^i)$, and to compute $\widehat{G}^{\mathrm{Q-PGT}}(\theta)$ in Appendix D.1.

To simplify notations, we use shorthand $\widehat{G}^{(k)}$ for $\widehat{G}(\theta^{(k)})$ and similarly $G^{(k)}$ for $G(\theta^{(k)})$, and $J^{(k)}$ for $J(\theta^{(k)})$. Equipped with gradient estimators $\widehat{G}(\theta)$ among (6)-(9), at the k-th iteration, policy gradient updates the policy parameters with the stepsize $\eta > 0$ as follows

$$\theta^{(k+1)} = \theta^{(k)} + \eta \widehat{G}^{(k)}. \tag{10}$$

We refer to REINFORCE (6) and GPOMDP (7) as *vanilla policy gradient* [58].

3 Unbiased Policy Gradient Estimators without Truncation

3.1 Unbiased policy gradient – Unbiased GPOMDP and Q-PGT

Notice that REINFORCE (6) and GPOMDP (7) are biased gradient estimator of $J(\theta)$, due to the truncation. Inspired by Zhang et al. [61, Algorithms 3], we propose a general unbiased policy gradient (UPG) algorithm, as shown in Algorithm 1.

Algorithm 1: UPG: Unbiased Policy Gradient

Input: Initial state distribution ρ, policy π^{θ}, discount factor $\gamma \in [0, 1)$

1 Initialize $s_0 \sim \rho$ and $a_0 \sim \pi^{\theta}(\cdot \mid s_0)$, the horizon $H - 1 \sim \mathrm{Geom}(1-\gamma)$

2 **for** $t = 0$ ***to*** $H - 1$ **do**

3 Store the vector $\nabla \log \pi^{\theta}(a_t \mid s_t)$ and the scalar $r(s_t, a_t)$

4 Sample $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$ and $a_{t+1} \sim \pi^{\theta}(\cdot \mid s_{t+1})$

5 Build the undiscounted gradient estimator $\widehat{G}^{\mathrm{UPG}}(\theta)$ from the stored $\nabla \log \pi^{\theta}(a_t \mid s_t)$, $r(s_t, a_t)$

Output: $\widehat{G}^{\mathrm{UPG}}(\theta)$

That is, we consider a finite-horizon undiscounted RL problem where the horizon is random, and introduce the discount factor γ as part of the parameters for the sampling procedure to obtain UPG without truncation. First, we determine the length of the horizon H sampled from a geometric distribution $\mathrm{Geom}(1-\gamma)$ associated to the discount factor γ, which corresponds to Line 1 of Algorithm 1. We have $\Pr(H - 1 = k) = (1-\gamma)\gamma^k$ for $k \in \{0, 1, 2, \ldots\}$. Second, we sample the vectors $\nabla \log \pi^{\theta}(a_t \mid s_t)$ and the scalars $r(s_t, a_t)$ inside the horizon $H - 1$. Lastly, we construct an undiscounted gradient estimator $\widehat{G}^{\mathrm{UPG}}(\theta)$ from the stored $\nabla \log \pi^{\theta}(a_t \mid s_t)$ and $r(s_t, a_t)$, as described in Line 5. There exits different ways to do so as we present next.

As a special case of UPG, based on GPOMDP (7), we consider the following unbiased GPOMDP (UGPOMDP) gradient estimator without the discount factor γ

$$\widehat{G}^{\mathrm{UGPOMDP}}(\theta) \overset{\text{def}}{=} \sum_{t=0}^{H-1} r(s_t, a_t) \left(\sum_{t'=0}^{t} \nabla \log \pi^{\theta}(a_{t'} \mid s_{t'}) \right). \tag{11}$$

We verify in the following lemma that, $\widehat{G}^{\mathrm{UGPOMDP}}(\theta)$ in (11) is unbiased for any π^{θ}, and the expected length of the trajectory is $\frac{1}{1-\gamma}$. Its proof is provided in Appendix E.1.

Lemma 1. *Consider* $\widehat{G}^{\mathrm{UGPOMDP}}(\theta)$ (11). *We have* $\mathbb{E}[H] = \frac{1}{1-\gamma}$ *and* $\mathbb{E}\big[\widehat{G}^{\mathrm{UGPOMDP}}(\theta)\big] = G(\theta)$.

An intuition behind how Lemma 1 works is that, a discounted MDP problem can be viewed as a non-discounted problem with an independent geometric time-horizon. When taking the expectation for the latter, it becomes analytically equivalent to the discounted one in (1). This viewpoint of the

MDP is well-known (e.g., see the example at https://www.tau.ac.il/~mansour/rl-course/scribe4/node17.html). Here, we extend this viewpoint to different UPG algorithms.

From the best of our knowledge, the PG algorithm (10) with UGPOMDP gradient estimator (11) is new. It is unbiased without truncation thanks to the geometric distribution $\mathrm{Geom}(1-\gamma)$ for the horizon sampling, which implicitly injects the discount factor inside the gradient estimator, even though there is no discount factor appearing in (11). See Algorithm 2 an implementation of $\widehat{G}^{\mathrm{UGPOMDP}}(\theta)$ (11) in Appendix D.1.

Notice that the known algorithm Q-PGT in (9) belongs to UPG, as it uses $\mathrm{Geom}(1-\gamma)$ sampling for the horizon and constructs an unbiased estimator of the Q-function with finite-horizon undiscounted rewards [1, Algorithm 1](see Algorithm 4 in Appendix D.1 as well).

Remark 1. The setting of UPG (e.g., $\widehat{G}^{\mathrm{UGPOMDP}}(\theta)$ in (11) and Q-PGT in (9)) is fundamentally different to the one of vanilla PG (e.g., $\widehat{G}^{\mathrm{GPOMDP}}(\theta)$ in (7)). The vanilla PG considers a fixed horizon H, including the infinite horizon $H=\infty$. In practice, H is of order $\mathcal{O}(\frac{1}{1-\gamma})$, which is referred to as *effective horizon* [32]. In contrast, the UPG considers randomized horizons $H-1\sim\mathrm{Geom}(1-\gamma)$, sampled i.i.d. from the batch, which can be arbitrary large and are not fixed for each single trajectory in the batch. Second, the vanilla PG constructs the gradient estimator with discounted rewards, while UPG uses the undiscounted rewards to build the gradient estimator. Consequently, the vanilla PG (e.g., $\widehat{G}^{\mathrm{REINFORCE}}(\theta)$ and $\widehat{G}^{\mathrm{GPOMDP}}(\theta)$) uses a biased gradient estimator due to the truncation, while the UPG (e.g., $\widehat{G}^{\mathrm{Q-PGT}}(\theta)$ and $\widehat{G}^{\mathrm{UGPOMDP}}(\theta)$) uses the unbiased one.

It is worth mentioning that RPG [61, Algorithm 3] shares lots of similarity with Q-PGT (9). Both RPG and Q-PGT have unbiased gradient estimators with random horizon, and both use the action value expression of PGT (5) to construct the gradient estimators. However, RPG is not recovered by Q-PGT/UPG. Compared to Q-PGT, RPG considers the discounted rewards instead of the undiscounted rewards, and uses $\mathrm{Geom}(1-\sqrt{\gamma})$ instead of $\mathrm{Geom}(1-\gamma)$ to sample the horizon, in which case, the stochastic gradient update can be guaranteed to be bounded, while the stochastic gradient estimator Q-PGT in (9) is unbounded. The convergence analysis of RPG in Zhang et al. [61] relies on the boundedness of the stochastic gradient update. Later in § 4.2, we provide the first convergence analysis of the unbiased Q-PGT in (9), even though the stochastic gradient update is unbounded.

Remark 2. There are others ways to construct $\widehat{G}^{\mathrm{UPG}}(\theta)$ for UPG in Line 5 of Algorithm 1. For instance, UPG can replace the unbiased Q-function estimation $\widehat{Q}^{\,\theta}(s,a)$ in (9) by either the difference between $\widehat{Q}^{\,\theta}(s,a)$ and an unbiased estimate of the value function $\widehat{V}^{\,\theta}(s)$ [1, Algorithm 3], or the temporal difference error [48], which involves the unbiased estimate of the value function $\widehat{V}^{\,\theta}(s)$. Similarly, UPG can also replace the unbiased Q-function estimation $\widehat{Q}^{\,\theta}(s,a)$ in (9) by the generalized advantage estimation [42]. We leave the investigation of these alternatives of UPG for future work.

3.2 α-Unbiased policy gradient

UPG in Algorithm 1 does not recover RPG [61, Algorithm 3], as UPG uses undiscounted rewards and RPG uses discounted rewards. However, for RPG, the rewards are $\sqrt{\gamma}$-discounted instead of γ-discounted, in contrast to the vanilla PG. Inspired by RPG, in this section we extend UPG to a more general α-unbiased policy gradient (α-UPG) algorithm. See also Algorithm 5 in Appendix D.2.

That is, given $\alpha\in[0,1)$, we consider finite random horizons H with $H-1\sim\mathrm{Geom}(1-\gamma^{1-\alpha})$, and we construct a discounted gradient estimator $\widehat{G}^{\alpha-\mathrm{UPG}}(\theta)$ using the discounted rewards $\gamma^{\alpha t}r(s_t,a_t)$ for different time step t, where the discount factor is $\gamma^\alpha>\gamma$. Notice that α-UPG recovers UPG when $\alpha=0$. In this case, the reward is undiscounted as the discount factor $\gamma^{\alpha t}=1$ for all t.

Based on UGPOMDP (11) and Q-PGT (9) from UPG, we propose two novel unbiased gradient estimators α-UGPOMDP and α-QPGT as special cases of α-UPG. First, we sample $H-1\sim\mathrm{Geom}(1-\gamma^{1-\alpha})$; and we sample $\nabla\log\pi^\theta(a_t\mid s_t)$ and $r(s_t,a_t)$ inside the horizon $H-1$. Then, we construct the unbiased gradient estimators from the sampled $\nabla\log\pi^\theta(a_t\mid s_t)$ and $r(s_t,a_t)$ as follows,

$$\widehat{G}^{\alpha-\mathrm{UGPOMDP}}(\theta)\overset{\text{def}}{=}\sum_{t=0}^{H-1}\gamma^{\alpha t}r(s_t,a_t)\left(\sum_{t'=0}^{t}\nabla\log\pi^\theta(a_{t'}\mid s_{t'})\right),\tag{12}$$

$$\widehat{G}^{\alpha-\mathrm{QPGT}}(\theta)\overset{\text{def}}{=}\frac{1}{1-\gamma}\left(\sum_{t=0}^{H-1}\gamma^{\alpha t}r(s_t,a_t)\right)\nabla\log\pi^\theta(a_0\mid s_0),\tag{13}$$

where $s_0 \sim d_\rho^\theta$ and $a_0 \sim \pi^\theta(\cdot \mid s_0)$ in (13).

It is straightforward to obtain that, when $\alpha = 0$, α-UGPOMDP (12) recovers UGPOMDP (11) and α-QPGT (13) recovers Q-PGT (9), the undiscounted cases. In particular, α-QPGT (13) recovers RPG with $\alpha = 1/2$. That is, the discount factor $\sqrt{\gamma}$ is considered for the MDP and the random horizon is sampled from $\mathrm{Geom}(1-\sqrt{\gamma})$, as described for RPG previously. See Algorithms 6 and 7 in Appendix D.2 for the implementations of α-UGPOMDP (12) and α-QPGT (13), respectively.

Like Lemma 1, we verify that, $\widehat{G}^{\alpha-\mathrm{UGPOMDP}}(\theta)$ in (12) and $\widehat{G}^{\alpha-\mathrm{QPGT}}(\theta)$ in (13) are unbiased for any π^θ, and the expected length of the trajectory is $\frac{1}{1-\gamma^{1-\alpha}}$, with its proof provided in Appendix E.2:

Lemma 2. *Consider the gradient estimators $\widehat{G}^{\alpha-\mathrm{UGPOMDP}}(\theta)$ in (12) and $\widehat{G}^{\alpha-\mathrm{QPGT}}$ in (13). It follows that $\mathbb{E}[H] = \frac{1}{1-\gamma^{1-\alpha}}$ and $\mathbb{E}\big[\widehat{G}^{\alpha-\mathrm{UGPOMDP}}(\theta)\big] = \mathbb{E}\big[\widehat{G}^{\alpha-\mathrm{QPGT}}(\theta)\big] = G(\theta)$.*

It turns that Lemma 2 implies Lemma 1 with $\alpha = 0$. When α increases from 0 to 1, $\mathbb{E}[H]$ increases from $\frac{1}{1-\gamma}$ to ∞. That is, UPG has the shortest expected horizon, α-UPG gets longer horizon if α increases, and if $\alpha \to 1$, α-UPG will recover the vanilla PG for infinite-horizon γ-discounted rewards.

4 Sample Complexity Analysis

4.1 General sample complexity analysis of α-UPG

In this section, we provide a general sample complexity analysis of PG in (10) with the unbiased gradient estimators α-UPG presented in § 3.2, which includes the one of UPG, i.e., α-UPG with $\alpha = 0$ as a special case. For the forthcoming analysis, the expected return $J(\cdot)$ is assumed to be smooth.
Assumption 3 (Smoothness). There exists $L > 0$ such that, for all $\theta \in \Theta$, we have $\left\|\nabla^2 J(\theta)\right\| \leq L$.
Our analysis builds on an expected smoothness assumption introduced by Khaled and Richtárik [20].
Assumption 4 (Expected smoothness, Assumption 2 in Khaled and Richtárik [20]). There exists $A, B, C \geq 0$ such that for all $\theta \in \mathbb{R}^d$, the policy gradient estimator satisfies

$$\mathbb{E}\big[\|\widehat{G}(\theta)\|^2\big] \leq 2A(J^\star - J(\theta)) + B\|G(\theta)\|^2 + C. \tag{14}$$

This assumption bounds the second moment of the empirical gradient $\widehat{G}(\theta)$ in terms of the suboptimality gap $J^\star - J(\theta)$, the expected gradient $G(\theta)$ and a constant C. It serves as the most general assumption to characterize this quantity, as it recovers a number of popular and more restrictive assumptions commonly used in non-convex optimization: the bounded variance of the stochastic gradient [13], the convex expected smoothness [15, 14], the gradient confusion assumption [39] and other assumptions [40, 6, 52, 23], just to name a few. A more detailed discussion of the assumption for non-convex optimization convergence theory can be found in Khaled and Richtárik [20, Theorem 1].

Notably, Yuan et al. [58] was the first to adapt this assumption to derive the convergence analysis of vanilla PG. The only adaptation made in (14) is that, instead of using the expected unbiased gradient $G(\theta)$ in the second term of (14), Yuan et al. [58] uses the expected gradient but truncated, e.g., $\mathbb{E}\big[\widehat{G}^{\mathrm{REINFORCE}}(\theta)\big]$, which is biased due to the truncation in (6), and consequently develops an analysis from [20, Theorem 2] with a few extra steps. In comparison, we can directly apply (14) and the modern convergence analysis of the unbiased SGD [20, Theorem 2] from optimization into RL to derive the iteration and sample complexity analysis of α-UPG, respectively, as α-UPG is unbiased.

Proposition 1 (Iteration complexity)**.** *Suppose that Assumptions 3 and 4 hold for α-UPG in § 3.2 and in Algorithm 5. Consider the iterates $\theta^{(k)}$ of the PG (10), using the unbiased gradient estimators α-UPG, with constant stepsize $\eta \in \big(0, \frac{2}{LB}\big)$ where $B = 0$ means that $\eta \in (0, \infty)$. It follows that*

$$\min_{0 \leq k \leq K-1} \mathbb{E}\left[\left\|G^{(k)}\right\|^2\right] \leq \frac{2(J^\star - J^{(0)})(1 + L\eta^2 A)^K}{\eta K(2 - LB\eta)} + \frac{LC\eta}{2 - LB\eta}. \tag{15}$$

In particular if $A = 0$, we have

$$\mathbb{E}\left[\|G(\theta_U)\|^2\right] \leq \frac{2(J^\star - J^{(0)})}{\eta K(2 - LB\eta)} + \frac{LC\eta}{2 - LB\eta}, \tag{16}$$

where θ_U is uniformly sampled from the iterates $\{\theta^{(0)}, \cdots, \theta^{(K-1)}\}$.

Corollary 1 (Sample complexity, Corollary 1 in Khaled and Richtárik [20])**.** *Consider the setting of Proposition 1. Given $\epsilon > 0$, let $\eta = \min\left\{\frac{1}{\sqrt{LAK}}, \frac{1}{LB}, \frac{\epsilon}{2LC}\right\}$. If the number of iterations K satisfies*

$$K \geq \frac{12L(J^\star - J^{(0)})}{\epsilon^2} \max\left\{B, \frac{12A(J^\star - J^{(0)})}{\epsilon^2}, \frac{2C}{\epsilon^2}\right\}, \tag{17}$$

then $\min_{0\leq k\leq K-1} \mathbb{E}\left[\left\|G^{(k)}\right\|^2\right] = \mathcal{O}(\epsilon^2)$.

Proposition 1 provides a general characterization of the FOSP convergence of α-UPG as a function of all the constants involved in the assumptions on the problem and the gradient estimator. Refer to Appendix B for a discussion comparing the technical aspects of this result to [20]. The proof is provided in Appendix F.1 for the completeness and Corollary 1 can be derived by [20, Corollary 1].

Thus for an FOSP convergence of α-UPG, from (16) we recover the $1/\sqrt{K}$ convergence rate of vanilla PG for RL problems in the literature [30, 61, 27, 1, 55, 58], which is also the same well-known rate of SGD in non-convex optimization [45, 13]. Furthermore, by choosing a batch size $m = \mathcal{O}(1)$, the expected horizon $\mathbb{E}[H] = \frac{1}{1-\gamma^{1-\alpha}}$ from Lemma 2, and $K = \mathcal{O}(\epsilon^{-4})$ from (17), we recover the $Km\mathbb{E}[H] = \mathcal{O}(\epsilon^{-4})$ sample complexity of RPG [61], which is known as optimal for SGD without extra assumptions on second-order smoothness or disruptive stochastic gradient noise [9, 20].

Compared to [58, Theorem 3.4], our analysis does not require the introduction of an additional assumption [58, Assumption 3.2], which assumes that the difference between the expected unbiased gradient and the expected truncated gradient should be proportional to γ^H, with H being the fixed truncated horizon for vanilla PG. As a result, our analysis not only has tighter bounds, as we avoid the additional error term $\mathcal{O}(\gamma^H)$ present in [58], but also simplifies the process, as we can directly apply the results of [20] without needing the extra analysis steps in Yuan et al. [58]. This is mainly because of the biasedness from truncation. Regarding sample complexity, our tighter iteration complexity bounds lead to improvements in the vanilla PG sample complexity by a factor of $\mathcal{O}(\log \epsilon^{-1})$. This factor originates from the additional error term $\mathcal{O}(\gamma^H) = \mathcal{O}(\epsilon) \Rightarrow H = \mathcal{O}(\log \epsilon^{-1})$ in [58].

Assumption 4 along with Proposition 1 may appear obscure at a first sight, it is indeed a clever way to unify many of the current policy settings used in the RL literature [30, 61, 1, 55], e.g., the softmax policies with or without regularizations [1] and the expected Lipschitz and smooth policies presented in the next section. An extensive study of the assumption used in RL can be found in [58, Section 4].

4.2 Sample complexity analysis of the expected Lipschitz and smooth policy

We consider the commonly used **expected Lipschitz and smooth policy** (E-LS)[1].

Assumption 5 (E-LS policies, Definition 1 in Papini et al. [33]). There exists constants $G, F > 0$ such that for every state $s \in \mathcal{S}$, the expected gradient and Hessian of $\log \pi^\theta(\cdot \mid s)$ satisfy

$$\mathbb{E}_{a\sim\pi^\theta(\cdot|s)}\left[\left\|\nabla \log \pi^\theta(a \mid s)\right\|^2\right] \leq G^2, \quad \text{and} \quad \mathbb{E}_{a\sim\pi^\theta(\cdot|s)}\left[\left\|\nabla^2 \log \pi^\theta(a \mid s)\right\|\right] \leq F. \tag{18}$$

This assumption (or its stronger version without the expectation [32]) is widely adopted in the analysis of PG (see for e.g., [46, 56, 17, 60, 58, 28]). It is satisfied for many classes of policies, e.g., the softmax, Gaussian and Cauchy policies. We refer to Fatkhullin et al. [11, Appendix B] for more details.

Under this assumption, Assumption 3 holds as shown by Yuan et al. [58] in the following lemma.

Lemma 3 (Smoothness, Lemma 4.4 in [58])**.** *Under Assumption 5, $J(\theta)$ is L-smooth, namely $\left\|\nabla^2 J(\theta)\right\| \leq L$ for all θ with $L = \frac{r_{\max}}{(1-\gamma)^2}\left(G^2 + F\right)$.*

In the following theorem, we show that α-UPG with the E-LS policy implies Assumption 4.

Theorem 6. *Consider α-UPG among (11), (9), (12), (13) and RPG [61] with Assumption 5. We have*

$$\mathbb{E}\left[\|\widehat{G}(\theta)\|^2\right] \leq (1 - 1/m)\,\|G(\theta)\|^2 + \nu/m, \tag{19}$$

where m is the batch size, and ν is the upper bound of $\mathbb{E}\left[\|\widehat{G}(\theta)\|^2\right]$ for one single trajectory with $\nu = \frac{3G^2 r_{\max}^2}{(1-\gamma)^3}$ for UGPOMDP (11); $\nu = \frac{2G^2 r_{\max}^2}{(1-\gamma)^4}$ for Q-PGT (9); or $\nu = \frac{2G^2 r_{\max}^2}{(1-\gamma)^3(1-\gamma\sqrt{\gamma})}$ for RPG [61]. As for general α-UGPOMDP (12) and α-QPGT (13), their values of ν are given in (29).

[1] While Papini et al. [33] refers to this assumption as *smoothing policy*, Yuan et al. [58] referred to as the expected Lipschitz and smooth policy.

Notice that the upper bound ν is crucial for the FOSP sample complexity analysis [61, 55, 58], it quantifies the variance of the gradient estimator for one single trajectory. Indeed, from (19), we have

$$\mathbb{V}\mathrm{ar}\left[\|\widehat{G}(\theta)\|^2\right] = \mathbb{E}\left[\|\widehat{G}(\theta)\|^2\right] - \|G(\theta)\|^2 \leq (\nu - \|G(\theta)\|^2)/m \leq \nu/m \qquad (= \nu, \text{ if } m = 1).$$

In previous PG analysis, one of the main challenges is to bound ν. Under Assumption 5, Yuan et al. [58] establishes the best known bound of $\nu = \mathcal{O}(\frac{G^2 r_{\max}^2}{(1-\gamma)^3})$ for GPOMDP (7), while [46, 35] achieve a worse bound $\mathcal{O}(\frac{G^2 r_{\max}^2}{(1-\gamma)^4})$ with more restrictive assumptions (Assumption 5 without expectation). Therefore, our result for UGPOMDP matches the best known bound of ν, even though UGPOMDP uses undiscounted rewards, resulting in an unbounded stochastic gradient estimator.

To the best of our knowledge, $\nu = \frac{2G^2 r_{\max}^2}{(1-\gamma)^4}$ for Q-PGT is novel. Previous results either consider truncated Q-PGT or unbiased gradient estimator RPG [61, 55]. Like UGPOMDP, Q-PGT uses undiscounted rewards, so its stochastic gradient estimator is unbounded, which makes the analysis challenging. Our analysis follows the idea of [59, Corollary 1] by showing that $\mathbb{E}[\widehat{Q}^{\theta}(s,a)^2]$ is bounded, even though $\widehat{Q}^{\theta}(s,a)$ is unbounded. However, the ν for Q-PGT is $\frac{1}{1-\gamma}$ bigger than the one for UGPOMDP.

When analyzing α-QPGT, as a by-product, we obtain ν for RPG, as RPG is in fact α-QPGT with $\alpha = \frac{1}{2}$. Our ν is three times tighter than the one of [61, Theorem 3.4] when γ is close to 1, and is obtained with weaker assumption (Assumption 5). See Remark 9 in Appendix G.1 for more details.

As for general α-UGPOMDP and α-QPGT, ν depends on α as shown in (29). There is no explicit best α to minimize ν and it also depends on γ. Alternatively, we provide in Fig. 1 α-UGPOMDP's heap map of the theoretical value of ν. In this heap map, the x-axis is α between 0 and 1 and the y-axis is γ between 0.9 and 1, and the variance of α-UGPOMDP ν is computed by $f_1(\gamma^\alpha)$ with f_1 defined in (30). Here we consider $G = r_{\max} = 1$ for simplification as the term $G^2 r_{\max}^2$ is common for all the unbiased gradient estimators in (29).

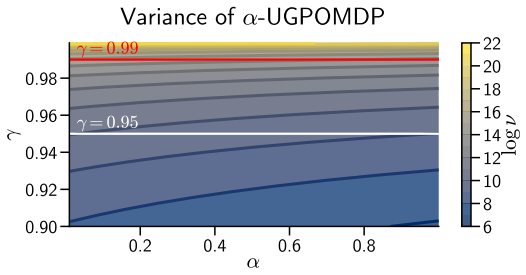

Figure 1: **Variance of α-UGPOMDP.**

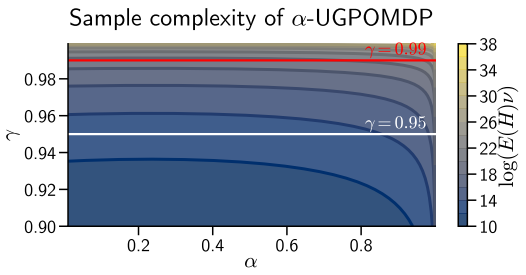

Figure 2: **Sample complexity of α-UGPOMDP.**

From Fig. 1, when γ is fixed, we observe that if α increases, the variance of α-UGPOMDP decreases. The minimal variance achieves when $\alpha \to 1$ and the maximum achieves when $\alpha = 0$. This makes sense. In fact, when α increases, the discount factor γ^α decreases, so the norm of the gradient becomes smaller, which implies smaller variance. However, when $\gamma \to 1$, the contour line becomes horizontal. In this case, changing α will not have impact for the variance of α-UGPOMDP. This is why when $\gamma \to 1$, the variance of UGPOMDP (i.e., α-UGPOMDP with $\alpha = 0$) matches the one of GPOMDP which can be seen as α-UGPOMDP with $\alpha = 1$. Similarly, see Fig. 6 α-QPGT's heap map of ν w.r.t. α and γ in Remark 10 in Appendix G.3.

Now we can establish the sample complexity of α-UPG for the expected Lipschitz and smooth policy assumptions as a corollary of Proposition 1, Lemma 3 and Theorem 6.

Corollary 2. *Suppose that Assumption 5 is satisfied. The α-UPG – UGPOMDP (11), Q-PGT (9), α-UGPOMDP (12) and α-QPGT (13) gradient estimators applied in PG (10) with a batch sampling of size m and constant stepsize $\eta \in (0, 2/(L(1-1/m)))$, satisfy*

$$\mathbb{E}\left[\|\nabla J(\theta_U)\|^2\right] \leq \frac{2(J^\star - J^{(0)})}{\eta K\left(2 - L\eta\left(1 - \frac{1}{m}\right)\right)} + \frac{L\nu\eta}{m\left(2 - L\eta\left(1 - \frac{1}{m}\right)\right)}, \tag{20}$$

where ν and L are provided in Lemma 3 and Theorem 6 respectively.

Similar to Corollary 1, by applying Corollary 2, we obtain sample complexity results for α-UPG.

Corollary 3. *Consider the setting of Corollary 2. For a given $\epsilon > 0$, by choosing the batch size m such that $1 \leq m \leq \frac{2\nu}{\epsilon^2}$, the stepsize $\eta = \frac{\epsilon^2 m}{2L\nu}$, the number of iterations K such that $Km \geq 8L\nu(J^\star - J^{(0)})/\epsilon^4$, then $\mathbb{E}[\|\nabla J(\theta_U)\|^2] = \mathcal{O}(\epsilon^2)$ with the total expected sample complexity $Km\mathbb{E}[H] \geq \mathcal{O}\big(\frac{L\nu\mathbb{E}[H]}{\epsilon^4}\big)$, which is $\mathcal{O}\big(\frac{1}{(1-\gamma)^6\epsilon^4}\big)$ for UGPOMDP (11), $\mathcal{O}\big(\frac{1}{(1-\gamma)^7\epsilon^4}\big)$ for Q-PGT (9), and $\mathcal{O}\big(\frac{1}{(1-\gamma)^5(1-\sqrt{\gamma})(1-\gamma\sqrt{\gamma})\epsilon^4}\big)$ for RPG [61].*

Thus, all UGPOMDP, Q-PGT and RPG improve the sample complexity of GPOMDP [58, Corollary 4.7] by $\mathcal{O}(\log \epsilon^{-1})$, as α-UPG already achieves $\mathcal{O}(\log \epsilon^{-1})$ better sample complexity than vanilla PG (Corollary 1). When γ is close to 1, UGPOMDP achieves the best sample complexity.

Notably, we have developed the sample complexity for Q-PGT for the first time, which is smaller than that of RPG, even though Q-PGT has a larger variance ν than RPG. This is because the expected horizon $\mathbb{E}[H]$ for Q-PGT is $\frac{1}{1-\gamma}$ (Lemma 1) and is shorter than the one for RPG which is $\frac{1}{1-\sqrt{\gamma}}$ (Lemma 2 with $\alpha = \frac{1}{2}$). See in Remark 10 their sample complexity comparison for more details. As a result, this suggests that looking for the lowest variance ν from α does not necessarily lead to the best sample complexity of α-UPG. Instead, optimizing the term $\nu\mathbb{E}[H]$ with α yields the best sample complexity. Thus, we provide α-UGPOMDP's heap map of the sample complexity w.r.t. α and γ in Fig. 2, which is computed by $\nu\mathbb{E}[H]$ with $\mathbb{E}[H] = \frac{1}{1-\gamma^{1-\alpha}}$ given in Lemma 2 and $G = r_{\max} = 1$ as the term $\frac{LG^2 r_{\max}^2}{\epsilon^4}$ is shared in the sample complexity for all the unbiased gradient estimators. Similarly, see Fig. 7 α-QPGT's sample complexity heap map in Remark 10 in Appendix G.3.

From Fig. 2, we observe that, to achieve the lowest sample complexity, the optimal α is between 0 and 1 and it varies w.r.t. γ. The $\alpha = 0.5$ seems to be a good choice to keep the sample complexity low for different γ. The worst sample complexity is achieved when $\alpha \to 1$ as $\mathbb{E}[H] \to \infty$, which is the case of GPOMDP. This theoretically suggests that one should use α-UGPOMDP instead of GPOMDP. Similar to Fig. 2, when $\gamma \to 1$, we observe the same horizontal contour line.

Finally, compared to [61], our RPG sample complexity result is better in terms of the order of $\frac{1}{1-\gamma}$ thanks to the tighter bound of ν . Additionally, we have a range of parameters choices for the batch size and the stepsize η, while [61] do not.

5 Experiments

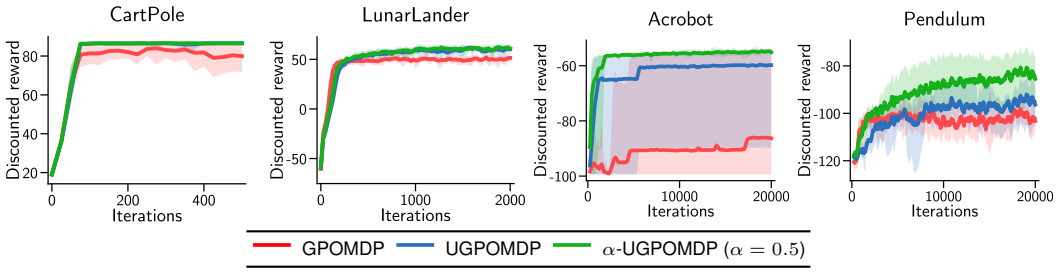

Figure 3: **Comparison between biased and unbiased policy gradient methods.** We compare the evolution of discounted rewards in GPOMDP, UGPOMDP and α-UGPOMDP ($\alpha = 0.5$) on four standard Gym environments.

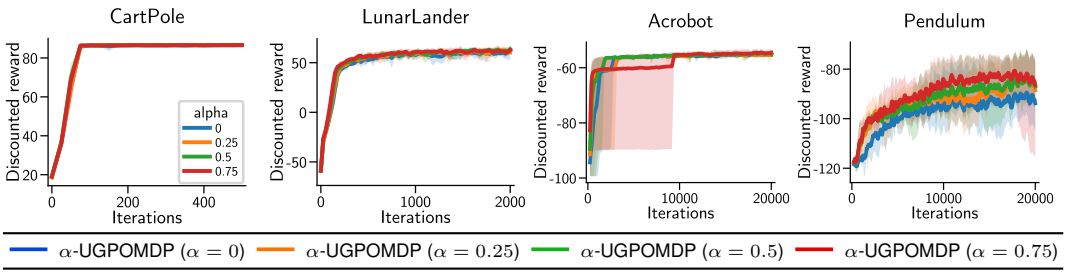

Figure 4: **Comparison among α-UGPOMDP with $\alpha = 0, 0.25, 0.5, 0.75$.**

We provide an empirical evaluation of (α-)UGPOMDP against biased (GPOMDP) and unbiased methods (Q-PGT and RPG) to validate our theoretical findings. Specifically, we test four different Gymnasium environments: Cart Pole, Lunar Lander, Acrobot, and Pendulum [51]. Each algorithm is evaluated with the discounted reward and its averaged performance over 10 runs is shown with 95% confidence interval. The policy is parameterized by a 2-layer MLP with 64 hidden units and Tanh activation function. We use softmax policies for finite action space and Gaussian policies for continuous action space. See Appendix H for the experimental details, additional experiments and plots.

Fig. 3 shows the comparison among the biased GPOMDP, the unbiased UGPOMDP and α-UGPOMDP with $\alpha = 0.5$. In all four environments, both UGPOMDP and 0.5-UGPOMDP outperform GPOMDP, demonstrating the effectiveness of the unbiased gradient methods with random horizon. Notably, 0.5-UGPOMDP outperforms UGPOMDP on Acrobot and Pendulum, and remains competitive on CartPole and LunarLander, which validates the sample complexity analysis in § 4.2 and Fig. 2. As expected from the variance analysis in § 4.2 and Fig. 1, (0.5-)UGPOMDP does not exhibit higher variance than GPOMDP with $\gamma = 0.99$ used in the experiments. This serves as strong empirical evidence that the unbiased methods are superior to biased ones and should be preferred in practice.

We then empirically tested the performance of α-UGPOMDP with $\alpha = 0, 0.25, 0.5, 0.75$ in Fig. 4. For CartPole and LunarLander, they are all competitive; while 0.75-UGPOMDP behaves the worst on Acrobot, the result is the opposite on Pendulum. For Acrobot, we use $\gamma = 0.99$. From the sample complexity analysis in Fig. 2, it is coherent that $\alpha = 0, 0.25$, or 0.5 leads to better sample complexity than $\alpha = 0.75$ with $\gamma = 0.99$ (red line in Fig. 2). For Pendulum, we use $\gamma = 0.95$ as shown in Appendix H.1. From Fig. 2, it also makes sense that $\alpha = 0.75$ leads to better sample complexity than $\alpha = 0, 0.25$, or 0.5 with $\gamma = 0.95$ (white line in Fig. 2). This suggests that our theoretical results for the variance and the sample complexity analysis well support our empirical results.

Additionally, we compare the performance of the unbiased gradient methods against each other. In Fig. 5, we compare UGPOMDP with RPG and Q-PGT. In both CartPole and Pendulum environments, UGPOMDP outperforms both RPG and Q-PGT[2].

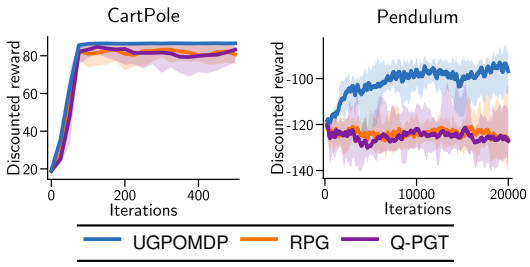

Figure 5: **Comparison between unbiased gradient methods: UGPOMDP, RPG and Q-PGT.**

6 Conclusions

We have introduced α-UPG, a new family of PG algorithms which achieves unbiased gradient estimators by addressing the horizon discrepancy in standard PG, and for which we have obtained strong convergence guarantees. We have exhibited several algorithms from the α-UPG family in order to demonstrate its flexibility, and tested two specialized variants, UGPOMDP and α-UGPOMDP, on several Gym environments with favorable results. The generality of α-UPG has allowed us to propose several new algorithms (e.g., (α-)UGPOMDP and α-QPGT) and also to recover some known ones (e.g., Q-PGT and RPG) along with their consequences, all from the single unified analytic framework of α-UPG. We believe that α-UPG and its convergence analysis will open the way to the design and analysis of a host of new unbiased PG methods, in a way similar to the potential developments mentioned in Remark 2. Further venues of investigation include exploring and testing the performance of different α-UPG, improving our theoretical analysis to the global optimum with faster convergence rates by considering additional assumptions (e.g., the Fisher-non-degenerated policies [11] with the use of the gradient domination property [10]), and building on α-UPG to reach towards more advanced PG algorithms, such as variance reduced PGs [56, 17, 10] and PGs integrated with second-order information [46, 28, 38].

[2] In [61], the authors use a different environment from the one available in Gymnasium and other environment libraries. Specifically, they constrain the action space to $[-20, 20]$, while in Gym the action space is $[-2, 2]$. This makes the problem easier in the setup used in [61] compared to the one used in this work. Hence, the results are not directly comparable.

Bibliography

[1] Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021. (Cited on pages 3, 4, 5, and 7.)

[2] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms, 2024. (Cited on page 1.)

[3] Carlo Alfano, Rui Yuan, and Patrick Rebeschini. A novel framework for policy mirror descent with general parameterization and linear convergence. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. (Cited on pages 1 and 15.)

[4] Anas Barakat, Ilyas Fatkhullin, and Niao He. Reinforcement learning with general utilities: Simpler variance reduction and large state-action space. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 1753–1800. PMLR, 23–29 Jul 2023. (Cited on page 16.)

[5] J. Baxter and P. L. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, Nov 2001. ISSN 1076-9757. doi: 10.1613/jair.806. (Cited on pages 1, 3, and 15.)

[6] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018. ISSN 0036-1445. doi: 10.1137/16M1080173. (Cited on page 6.)

[7] Jianyu Chen, Bodi Yuan, and Masayoshi Tomizuka. Model-free deep reinforcement learning for urban autonomous driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2765–2771, 2019. doi: 10.1109/ITSC.2019.8917306. (Cited on page 1.)

[8] Pierluca D'Oro, Alberto Maria Metelli, Andrea Tirinzoni, Matteo Papini, and Marcello Restelli. Gradient-aware model-based policy search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3801–3808, Apr. 2020. doi: 10.1609/aaai.v34i04.5791. (Cited on page 16.)

[9] Yoel Drori and Ohad Shamir. The complexity of finding stationary points with stochastic gradient descent. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2658–2667. PMLR, 13–18 Jul 2020. (Cited on page 7.)

[10] Ilyas Fatkhullin, Jalal Etesami, Niao He, and Negar Kiyavash. Sharp analysis of stochastic optimization under global Kurdyka-łojasiewicz inequality. In *Advances in Neural Information Processing Systems*, 2022. (Cited on page 10.)

[11] Ilyas Fatkhullin, Anas Barakat, Anastasia Kireeva, and Niao He. Stochastic policy gradient methods: Improved sample complexity for Fisher-non-degenerate policies. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 9827–9869. PMLR, 23–29 Jul 2023. (Cited on pages 7, 10, and 16.)

[12] Jie Feng, Ke Wei, and Jinchi Chen. Global convergence of natural policy gradient with hessian-aided momentum variance reduction, 2024. (Cited on page 16.)

[13] Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368, 2013. ISSN 1052-6234. (Cited on pages 6 and 7.)

[14] Robert M. Gower, Peter Richtárik, and Francis Bach. Stochastic quasi-gradient methods: variance reduction via Jacobian sketching. *Mathematical Programming*, 188(1):135–192, Jul 2021. (Cited on page 6.)

[15] Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. SGD: General analysis and improved rates. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5200–5209. PMLR, 09–15 Jun 2019. (Cited on page 6.)

[16] Jakub Grudzien, Christian A Schroeder De Witt, and Jakob Foerster. Mirror learning: A unifying framework of policy optimisation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 7825–7844. PMLR, 17–23 Jul 2022. (Cited on page 1.)

[17] Feihu Huang, Shangqian Gao, Jian Pei, and Heng Huang. Momentum-based policy gradient methods. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4422–4433. PMLR, 13–18 Jul 2020. (Cited on pages 1, 7, 10, and 16.)

[18] Feihu Huang, Shangqian Gao, and Heng Huang. Bregman gradient policy optimization. In *International Conference on Learning Representations*, 2022. (Cited on pages 1 and 16.)

[19] Sham M Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. (Cited on page 1.)

[20] Ahmed Khaled and Peter Richtárik. Better theory for SGD in the nonconvex world. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. Survey Certification. (Cited on pages 2, 6, 7, 15, 16, 24, and 26.)

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015. (Cited on page 15.)

[22] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, 2022. doi: 10.1109/TITS.2021.3054625. (Cited on page 1.)

[23] Yunwen Lei, Ting Hu, Guiying Li, and Ke Tang. Stochastic gradient descent for nonconvex learning without bounded gradient assumptions. *IEEE Transactions on Neural Networks and Learning Systems*, 31(10):4394–4400, 2020. doi: 10.1109/TNNLS.2019.2952219. (Cited on page 6.)

[24] Yunxiang Li, Rui Yuan, Chen Fan, Mark Schmidt, Samuel Horváth, Robert M. Gower, and Martin Takáč. Enhancing policy gradient with the polyak step-size adaption, 2024. (Cited on page 16.)

[25] Jinsong Liu, Chenghan Xie, Qi Deng, Dongdong Ge, and Yinyu Ye. Stochastic dimension-reduced second-order methods for policy optimization, 2023. (Cited on page 16.)

[26] Rui Liu, Erfaun Noorani, Pratap Tokekar, and John S. Baras. Towards efficient risk-sensitive policy gradient: An iteration complexity analysis, 2024. (Cited on page 16.)

[27] Yanli Liu, Kaiqing Zhang, Tamer Basar, and Wotao Yin. An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods. In *Advances in Neural Information Processing Systems*, volume 33, pages 7624–7636. Curran Associates, Inc., 2020. (Cited on pages 7 and 16.)

[28] Saeed Masiha, Saber Salehkaleybar, Niao He, Negar Kiyavash, and Patrick Thiran. Stochastic second-order methods improve best-known sample complexity of SGD for gradient-dominated functions. In *Advances in Neural Information Processing Systems*, 2022. (Cited on pages 7, 10, and 16.)

[29] Siqiao Mu and Diego Klabjan. On the second-order convergence of biased policy gradient algorithms, 2024. (Cited on pages 1 and 16.)

[30] Matteo Papini. Safe policy optimization. 2020. (Cited on page 7.)

[31] Matteo Papini, Andrea Battistello, Marcello Restelli, Matteo Papini, Andrea Battistello, and Marcello Restelli. Safely exploring policy gradient. In *European Workshop on Reinforcement Learning*, 2018. (Cited on page 16.)

[32] Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirotta, and Marcello Restelli. Stochastic variance-reduced policy gradient. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4026–4035. PMLR, 2018. (Cited on pages 1, 5, 7, and 16.)

[33] Matteo Papini, Matteo Pirotta, and Marcello Restelli. Smoothing policies and safe policy gradients. *Machine Learning*, Oct 2022. ISSN 1573-0565. doi: 10.1007/s10994-022-06232-6. (Cited on pages 7 and 16.)

[34] J. Peters and S. Schaal. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4):682–697, May 2008. (Cited on page 3.)

[35] Nhan Pham, Lam Nguyen, Dzung Phan, Phuong Ha Nguyen, Marten van Dijk, and Quoc Tran-Dinh. A hybrid stochastic policy gradient algorithm for reinforcement learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 374–385. PMLR, 26–28 Aug 2020. (Cited on pages 8 and 16.)

[36] Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Adaptive step-size for policy gradient methods. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. (Cited on page 16.)

[37] Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Policy gradient in lipschitz markov decision processes. *Machine Learning*, 100(2):255–283, Sep 2015. ISSN 1573-0565. doi: 10.1007/s10994-015-5484-1. (Cited on page 16.)

[38] Saber Salehkaleybar, Sadegh Khorasani, Negar Kiyavash, Niao He, and Patrick Thiran. Momentum-based policy gradient with second-order information, 2023. (Cited on pages 10 and 16.)

[39] Karthik Abinav Sankararaman, Soham De, Zheng Xu, W. Ronny Huang, and Tom Goldstein. The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8469–8479. PMLR, 13–18 Jul 2020. (Cited on page 6.)

[40] Mark Schmidt and Nicolas Le Roux. Fast convergence of stochastic gradient descent under a strong growth condition, 2013. (Cited on page 6.)

[41] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1889–1897, Lille, France, 07–09 Jul 2015. PMLR. (Cited on page 1.)

[42] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016. (Cited on page 5.)

[43] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. (Cited on page 1.)

[44] Lior Shani, Yonathan Efroni, and Shie Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 5668–5675, 2020. (Cited on page 1.)

[45] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on Stochastic Programming*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009. doi: 10.1137/1.9780898718751. (Cited on page 7.)

[46] Zebang Shen, Alejandro Ribeiro, Hamed Hassani, Hui Qian, and Chao Mi. Hessian aided policy gradient. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5729–5738. PMLR, 09–15 Jun 2019. (Cited on pages 7, 8, 10, and 16.)

[47] Sebastian U. Stich. Unified optimal analysis of the (stochastic) gradient method, 2019. (Cited on page 25.)

[48] Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, Aug 1988. ISSN 1573-0565. doi: 10.1007/BF00115009. (Cited on page 5.)

[49] Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12, pages 1057–1063. MIT Press, 1999. (Cited on pages 1 and 3.)

[50] Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. In *International Conference on Learning Representations*, 2022. (Cited on page 1.)

[51] Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. Gymnasium, March 2023. URL https://zenodo.org/record/8127025. (Cited on pages 2, 10, and 36.)

[52] Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1195–1204. PMLR, 16–18 Apr 2019. (Cited on page 6.)

[53] Sharan Vaswani, Olivier Bachem, Simone Totaro, Robert Müller, Shivam Garg, Matthieu Geist, Marlos C. Machado, Pablo Samuel Castro, and Nicolas Le Roux. A general class of surrogate functions for stable and efficient reinforcement learning. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 8619–8649. PMLR, 28–30 Mar 2022. (Cited on page 1.)

[54] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. (Cited on pages 1 and 3.)

[55] Huaqing Xiong, Tengyu Xu, Yingbin Liang, and Wei Zhang. Non-asymptotic convergence of adam-type reinforcement learning algorithms under markovian sampling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):10460–10468, May 2021. (Cited on pages 7 and 8.)

[56] Pan Xu, Felicia Gao, and Quanquan Gu. Sample efficient policy gradient methods with recursive variance reduction. In *International Conference on Learning Representations*, 2020. (Cited on pages 1, 7, 10, and 16.)

[57] Huizhuo Yuan, Xiangru Lian, Ji Liu, and Yuren Zhou. Stochastic recursive momentum for policy gradient methods, 2020. (Cited on page 16.)

[58] Rui Yuan, Robert M. Gower, and Alessandro Lazaric. A general sample complexity analysis of vanilla policy gradient. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 3332–3380. PMLR, 28–30 Mar 2022. (Cited on pages 4, 6, 7, 8, 9, 15, 16, 27, and 39.)

[59] Rui Yuan, Simon Shaolei Du, Robert M. Gower, Alessandro Lazaric, and Lin Xiao. Linear convergence of natural policy gradient methods with log-linear policies. In *The Eleventh International Conference on Learning Representations*, 2023. (Cited on pages 4, 8, and 20.)

[60] Junyu Zhang, Chengzhuo Ni, Zheng Yu, Csaba Szepesvari, and Mengdi Wang. On the convergence and sample efficiency of variance-reduced policy gradient method. In *Advances in Neural Information Processing Systems*, 2021. (Cited on page 7.)

[61] Kaiqing Zhang, Alec Koppel, Hao Zhu, and Tamer Başar. Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization*, 58 (6):3586–3612, 2020. doi: 10.1137/19M1288012. (Cited on pages 2, 4, 5, 7, 8, 9, 10, 16, 20, 26, and 33.)

A Limitations

The theoretical assumptions we make throughout this paper are standard in the PG literature. In fact, for our main result on sample complexity, we use the weakest assumptions in the literature and match the best known results. The main point of our paper is precisely to address an assumption that is violated in practice (the "horizon discrepancy"). However, several standard assumptions remain that are violated in our experiments:

- Our theoretical results are for PG (10), while in our experiments we use Adam [21] which empirically allows better performance for all the experiments.
- Our assumptions (Assumption 3, Assumption 4 and Assumption 5 are not satisfied by the neural network parameterized policies that were used in the experiments. More specifically, the assumptions hold for softmax and Gaussian policies. In our experiments, while we also use softmax policies for discrete action space and Gaussian policies for continuous action space, we also add two hidden layers in the policy networks, which breaks the assumption. We use a very small step-size to mitigate the potential issue of the violation of the assumption. To make the theoretical analysis also available with the neural network parameterizations, one needs to consider additional assumptions like approximation error assumption [3, Assumption A1].

Our claims also have a limited scope: the theoretical results obtained are only for first-order stationary point (FOSP) convergence. This is different from second-order stationary point (SOSP) convergence or global optimal convergence. Our FOSP analysis is limited in the sense that it only guarantees that the gradient will converge to zero. For instance, it does not guarantee that the algorithm will find the optimal solution. It may converge to a local optimum, or a saddle point which is even worse. SOSP convergence will guarantee the avoidance of saddle points and global optimal convergence analysis will guarantee to find the optimum which will avoid local optima.

Our approach was tested on 3 unbiased algorithms, UGPOMDP, Q-PGT and RPG, with only the former being novel. We test on 5 environments (4 presented in the main text, 1 in the appendix), with 10 runs for each pair of algorithm/environment tested. In general, we expect our unbiased algorithms to enable learning on the same environments as their biased counterparts. This includes discrete as well as continuous environments, as presented in the main text. The classic environments we use are all fully observable, however, the (biased) GPOMDP algorithm has already been successfully applied to partially observed environments [5], hence we expect our unbiased version to be successful there as well. The environments all have deterministic state transitions, except for Acrobot whose noise applied to actions entails a stochastic transition to next state. Thus, our algorithms can operate with stochastic environments. The algorithms all assume deterministic reward functions, which is the case for all environments tested, but is a limitation for real world applications.

In terms of computational efficiency, our unbiased algorithms do not incur more computations than their biased counterparts. We expect the compute required by our algorithms to scale with the state dimension of the environment and its expected horizon in the same way as their biased counterparts.

B Related Work

Technical contribution and novelty compared to Khaled and Richtárik [20]. Our technical contribution and novelty compared to Khaled and Richtárik [20] can be summarized as follows:

- First, from an algorithmic point of view, we integrate the geometric distribution sampling for the horizon, which is unique to PG methods and has not been considered in Khaled and Richtárik [20].
- Second, compared explicitly to Khaled and Richtárik [20, Theorem 2], our bounds in (15) and (16) share the same rates but different constants due to the choice of the stepsize η. Indeed, their condition was $\eta \in (0, \frac{1}{LB})$, while ours is twice larger in terms of the possible range without loosing the tightness of the bounds, followed by the stepsize choices in [58].
- Furthermore, when considering the results we derived in specific cases in § 4.2, the difference between our work and Khaled and Richtárik [20] is significant. All cases studied in Khaled and Richtárik [20] (e.g., finite-sum structure) are not applicable for PG methods and we

had to derive specific analysis for our specialized settings (expected Lipschitz and smooth policies, such as Gaussian and softmax policies).

- Lastly, our focus is on deriving explicit sample complexity, whereas the results in Khaled and Richtárik [20] are concerned with convergence rates in terms of number of iterations. These dimensions are where most of the technical work was done.

Technical contribution and novelty compared to Yuan et al. [58]. Our technical contribution and novelty compared to Yuan et al. [58] can be summarized as follows:

- First, as mentioned in Remark 1, from an algorithmic point of view, the setting of α-UPG is fundamentally different to the one of vanilla PG, i.e., REINFORCE (6) and GPOMDP (7) studied in [58]. α-UPG considers finite random horizon γ^α-discounted rewards while vanilla PG in [58] considers infinite-horizon γ-discounted rewards with a fixed truncated horizon H.
- From the theoretical point of view,
  - We use a weaker assumption, we find no need to introduce any additional assumption due to the bias from the truncation. As a result, our analysis is easier and we avoid an additional error term $\mathcal{O}(\gamma^H)$.
  - Consequently, we improve on the sample complexity by a factor of $\mathcal{O}(\log \epsilon^{-1})$.
  - The variance of UGPOMDP is of the same order as for GPOMDP in Yuan et al. [58]. This also explains why we can achieve better convergence results, as our methods are unbiased.
  - When considering the results we derived in specific cases in § 4.2, the difference between our work and Yuan et al. [58] is significant. All cases studied in Yuan et al. [58] (e.g., truncated gradient estimators) are not applicable for α-UPG and we had to derive specific results for our specialized settings (such as α-UGPOMDP and α-QPGT).
- Empirically, UPG outperforms the vanilla PG presented in § 5, which validates our theoretical findings. Notice that there is no experiment in Yuan et al. [58].

Technical contribution and novelty compared to Zhang et al. [61]. . Our technical contribution and novelty compared to Zhang et al. [61] can be summarized as follows:

- First, the idea of our work is inspired from RPG [61]. We generalize RPG by introducing α-UPG which is a much more general algorithm and recovers RPG as a special case.
- From an algorithmic point of view, we provide two general new algorithms – α-UGPOMDP and α-QPGT as special cases of α-UPG, which demonstate the great flexibility of α-UPG.
- From the theoretical point of view,
  - We use the most advanced SGD proof techniques, which does not require the boundedness of the stochastic gradient estimators and allows range of parameters choice to achieve the same best performance, while Zhang et al. [61] relies on the boundedness of the stochastic gradient update.
  - We show that first sample complexity analysis of unbiased Q-PGT, even though its gradient estimator is unbounded.

Vanilla policy gradient. The vanilla PG is widely applied in different special RL settings [31, 33, 26] or that investigates the stepsize of the algorithm [36, 37, 8, 24]. However, they are all truncated. Our unbiased approach can be integrated naturally without modifying the rest of the steps, which will improve their results both in theory and in practice.

Variants of policy gradient and beyond. There are many advanced PG methods developed from REINFORCE (6) and GPOMDP (7), such as the variance reduced PGs [32, 56, 57, 17, 35, 27, 18, 4], natural PG variants [27, 12], actor-critic variants and others [8]. As a result, they are all biased due to the truncation. We believe that these variants of PGs can improve their results by simple using α-UPG instead of vanilla PG to construct the desired terms.

Furthermore, not only the gradient but also the Hessian is truncated for PGs with additional second-order information [46, 28, 11, 38, 25, 29]. By using our approach, that is, random horizon H with

$H \sim \mathrm{Geom}(1-\gamma^{1-\alpha})$ and γ^α-discounted reward, the estimates of the Hessian can be improved to be unbiased, which will also improve their results both in theory and in practice.

C Gradient Derivation

Lemma 4. *The full gradient $G(\theta)$ of the expected value function $J(\theta)$ can be written as (2), (3), (4) or (5). That is,*

$$\begin{aligned} G(\theta) &= \mathbb{E}_\tau \left[\sum_{k=0}^{\infty} \gamma^k r(s_k, a_k) \sum_{t=0}^{\infty} \nabla \log \pi^\theta(a_t \mid s_t) \right] \\ &= \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \left(\sum_{k=0}^{t} \nabla \log \pi^\theta(a_k \mid s_k) \right) \gamma^t r(s_t, a_t) \right] \\ &= \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \nabla \log \pi^\theta(a_t \mid s_t) \sum_{t'=t}^{\infty} \gamma^{t'} r(s_{t'}, a_{t'}) \right] \\ &= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\rho^\theta,\, a \sim \pi^\theta(\cdot \mid s)} \left[Q^\theta(s,a) \nabla \log \pi^\theta(a \mid s) \right]. \end{aligned}$$

Proof. First, the gradient $G(\theta)$ of the expected return has the following structure

$$\begin{aligned} G(\theta) &= \int r(\tau) \nabla p(\tau \mid \theta) d\tau \\ &= \int r(\tau) \left(\nabla p(\tau \mid \theta) / p(\tau \mid \theta) \right) p(\tau \mid \theta) d\tau \\ &= \mathbb{E}_{\tau \sim p(\cdot \mid \theta)} \left[r(\tau) \nabla \log p(\tau \mid \theta) \right] \\ &= \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \sum_{t'=0}^{\infty} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}) \right], \end{aligned} \tag{21}$$

which proves (2). To simplify (21), we notice that future actions do not depend on past rewards. That is, for $0 \le k < l$ among terms of the two sums in equation (21), we have

$$\begin{aligned} & \mathbb{E}_\tau \left[\nabla \log \pi^\theta(a_l \mid s_l) \gamma^k r(s_k, a_k) \right] \\ &= \mathbb{E}_{s_{0:l}, a_{0:l}} \left[\nabla \log \pi^\theta(a_l \mid s_l) \gamma^k r(s_k, a_k) \right] \\ &= \mathbb{E}_{s_{0:l}, a_{0:(l-1)}} \left[\gamma^k r(s_k, a_k) \mathbb{E}_{a_l} \left[\nabla \log \pi^\theta(a_l \mid s_l) \,\middle|\, s_{0:l}, a_{0:(l-1)} \right] \right] \\ &= \mathbb{E}_{s_{0:l}, a_{0:(l-1)}} \left[\gamma^k r(s_k, a_k) \int \pi^\theta(a_l \mid s_l) \nabla \log \pi^\theta(a_l \mid s_l)\, da_l \right] \\ &= \mathbb{E}_{s_{0:l}, a_{0:(l-1)}} \left[\gamma^k r(s_k, a_k) \int \nabla \pi^\theta(a_l \mid s_l)\, da_l \right] \\ &= \mathbb{E}_{s_{0:l}, a_{0:(l-1)}} \left[\gamma^k r(s_k, a_k) \nabla \underbrace{\int \pi^\theta(a_l \mid s_l)\, da_l}_{=1} \right] = 0. \end{aligned}$$

Plugging the above property into (21) yields (3) and (4) of the lemma, as half of the terms in (21) can be removed. The equations (3) and (4) are equivalent by changing the order of summation.

To prove (5), we can start from (4). That is,

$$
\begin{aligned}
G(\theta) &= \mathbb{E}_\tau \left[\sum_{t=0}^{\infty} \nabla \log \pi^\theta(a_t \mid s_t) \sum_{t'=t}^{\infty} \gamma^{t'} r(s_{t'}, a_{t'}) \right] \\
&= \mathbb{E}_\tau \left[\nabla \log \pi^\theta(a_0 \mid s_0) \sum_{t'=0}^{\infty} \gamma^{t'} r(s_{t'}, a_{t'}) \right] + \mathbb{E}_\tau \left[\sum_{t=1}^{\infty} \nabla \log \pi^\theta(a_t \mid s_t) \sum_{t'=t}^{\infty} \gamma^{t'} r(s_{t'}, a_{t'}) \right] \\
&= \mathbb{E}_{s_0, a_0} \left[\nabla \log \pi^\theta(a_0 \mid s_0) \mathbb{E}_\tau \left[\sum_{t'=0}^{\infty} \gamma^{t'} r(s_{t'}, a_{t'}) \,\middle|\, s_0, a_0 \right] \right] \\
&\quad + \mathbb{E}_{s_{1:\infty}, a_{1:\infty}} \left[\sum_{t=1}^{\infty} \nabla \log \pi^\theta(a_t \mid s_t) \sum_{t'=t}^{\infty} \gamma^{t'} r(s_{t'}, a_{t'}) \right] \\
&= \mathbb{E}_{s_0, a_0} \left[Q^\theta(s_0, a_0) \nabla \log \pi^\theta(a_0 \mid s_0) \right] \\
&\quad + \gamma \mathbb{E}_{s_{1:\infty}, a_{1:\infty}} \left[\sum_{t=1}^{\infty} \nabla \log \pi^\theta(a_t \mid s_t) \sum_{t'=t}^{\infty} \gamma^{t'-1} r(s_{t'}, a_{t'}) \right] \\
&= \sum_{t=0}^{\infty} \gamma^t \mathbb{E} \left[Q^\theta(s_t, a_t) \nabla \log \pi^\theta(a_t \mid s_t) \right],
\end{aligned}
$$

where the forth equality is obtained through the definition of Q-function, and the last step follows from recursion. The above expectation is computed over the trajectories $\{(s_t, a_t)\}_{t \geq 0}$. Notice that we can also rewrite the expectation over the state and action space $\mathcal{S} \times \mathcal{A}$. That is,

$$
\begin{aligned}
G(\theta) &= \sum_{t=0}^{\infty} \gamma^t \mathbb{E} \left[Q^\theta(s_t, a_t) \nabla \log \pi^\theta(a_t \mid s_t) \right] \\
&= \sum_{t=0}^{\infty} \gamma^t \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \Pr(s_t = s, a_t = a) Q^\theta(s, a) \nabla \log \pi^\theta(a \mid s) \\
&= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s, a_t = a) Q^\theta(s, a) \nabla \log \pi^\theta(a \mid s) \\
&= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \frac{1}{1-\gamma} d_\rho^\theta(s) \pi^\theta(a \mid s) Q^\theta(s, a) \nabla \log \pi^\theta(a \mid s),
\end{aligned}
$$

where the last line is obtained by the definition of the state visitation distribution $d_\rho^\theta(s)$. This completes the proof of the claim. □

D Algorithm Implementations

D.1 UGPOMDP and Q-PGT implementations

We first provide two equivalent Algorithms 2 and 3 to obtain the gradient estimators $\widehat{G}^{\mathrm{UGPOMDP}}(\theta)$ in (11). Recall $\widehat{G}^{\mathrm{UGPOMDP}}(\theta)$ in (11)

$$
\begin{aligned}
\widehat{G}^{\mathrm{UGPOMDP}}(\theta) &= \sum_{t=0}^{H-1} r(s_t, a_t) \left(\sum\nolimits_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}) \right) && (22) \\
&= \sum_{t=0}^{H-1} \left(\sum_{t'=t}^{H-1} r(s_{t'}, a_{t'}) \right) \nabla \log \pi^\theta(a_t \mid s_t), && (23)
\end{aligned}
$$

where the second line (23) is obtained by the change in order of summation, and $H - 1$ is sampled from $\mathrm{Geom}(1 - \gamma)$.

Based on two different but equivalent expressions of $\widehat{G}^{\mathrm{UGPOMDP}}(\theta)$, we propose Algorithms 2 and 3 to implement (22) and (23), respectively.

Algorithm 2: UGPOMDP: UGPOMDP gradient estimator in (11)

Input: Initial state distribution ρ, policy π^θ, discount factor $\gamma \in [0,1)$

Initialize $s_0 \sim \rho$ and $a_0 \sim \pi^\theta(\cdot \mid s_0)$, the vector $v_0 = \nabla \log \pi^\theta(a_0 \mid s_0)$, the horizon $H-1 \sim \text{Geom}(1-\gamma)$

Set the estimate $\widehat{G}^{\text{UGPOMDP}}(\theta) = r(s_0, a_0) v_0$ \\ *Start to estimate* $\widehat{G}^{\text{UGPOMDP}}(\theta)$

for $t = 0$ ***to*** $H-2$ **do**

  Sample $s_{t+1} \sim P(\cdot \mid s_t, a_t)$

  Sample $a_{t+1} \sim \pi^\theta(\cdot \mid s_{t+1})$ and obtain $r(s_{t+1}, a_{t+1})$

  $v_{t+1} \leftarrow v_t + \nabla \log \pi^\theta(a_{t+1} \mid s_{t+1})$

  $\widehat{G}^{\text{UGPOMDP}}(\theta) \leftarrow \widehat{G}^{\text{UGPOMDP}}(\theta) + r(s_{t+1}, a_{t+1}) v_{t+1}$

Output: $\widehat{G}^{\text{UGPOMDP}}(\theta)$

Algorithm 3: UGPOMDP: Equivalent UGPOMDP implementation in (23)

Input: Initial state distribution ρ, policy π^θ, discount factor $\gamma \in [0,1)$

Initialize $s_0 \sim \rho$ and $a_0 \sim \pi^\theta(\cdot \mid s_0)$, the horizon $H-1 \sim \text{Geom}(1-\gamma)$

for $t = 0$ ***to*** $H-1$ **do**

  Store the vector $\nabla \log \pi^\theta(a_t \mid s_a)$ and the scalar $r(s_t, a_t)$

  Sample $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$

  Sample $a_{t+1} \sim \pi^\theta(\cdot \mid s_{t+1})$

Set $R_{H-1} = r(s_{H-1}, a_{H-1})$

Set the estimate $\widehat{G}^{\text{UGPOMDP}}(\theta) = R_{H-1} \nabla \log \pi^\theta(a_{H-1} \mid s_{H-1})$ \\ *Start to estimate* $\widehat{G}^{\text{UGPOMDP}}(\theta)$

for $t = H-2$ ***to*** 0 **do**

  $R_t \leftarrow R_{t+1} + r(s_t, a_t)$

  $\widehat{G}^{\text{UGPOMDP}}(\theta) \leftarrow \widehat{G}^{\text{UGPOMDP}}(\theta) + R_t \nabla \log \pi^\theta(a_t \mid s_t)$

Output: $\widehat{G}^{\text{UGPOMDP}}(\theta)$

Then, we provide Algorithm 4 to implement the unbiased gradient estimator $\widehat{G}^{\text{Q-PGT}}(\theta)$ in (9).

Algorithm 4: Q-PGT: Q-PGT gradient estimator in (9)

Input: Initial state distribution ρ, policy π^θ, discount factor $\gamma \in [0,1)$

Initialize $s_0 \sim \rho$ and $a_0 \sim \pi^\theta(\cdot \mid s_0)$, the horizons $H_1 - 1, H_2 - 1 \sim \text{Geom}(1-\gamma)$

for $t = 0$ ***to*** $H_1 - 2$ **do**

  Sample $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$

  Sample $a_{t+1} \sim \pi^\theta(\cdot \mid s_{t+1})$ \\ *Accept* (s_{H_1-1}, a_{H_1-1})

Set the estimate $\widehat{Q}^{\,\theta}(s_{H_1-1}, a_{H_1-1}) = r(s_{H_1-1}, a_{H_1-1})$ \\ *Start to estimate* $\widehat{Q}^{\,\theta}(s_{H_1-1}, a_{H_1-1})$

for $t = H_1 - 1$ ***to*** $H_1 + H_2 - 3$ **do**

  Sample $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$

  Sample $a_{t+1} \sim \pi^\theta(\cdot \mid s_{t+1})$

  $\widehat{Q}^{\,\theta}(s_{H_1-1}, a_{H_1-1}) \leftarrow \widehat{Q}^{\,\theta}(s_{H_1-1}, a_{H_1-1}) + r(s_{t+1}, a_{t+1})$

Compute Q-PGT in (9): $\widehat{G}^{\text{Q-PGT}}(\theta) = \frac{1}{1-\gamma} \widehat{Q}^{\,\theta}(s_{H_1-1}, a_{H_1-1}) \nabla \log \pi^\theta(a_{H_1-1} \mid s_{H_1-1})$

Output: $\widehat{G}^{\text{Q-PGT}}(\theta)$

The $\widehat{G}^{\mathrm{Q-PGT}}(\theta)$ is an unbiased gradient estimator of $J(\theta)$. Indeed, we have

$$
\begin{aligned}
& \mathbb{E}\left[\widehat{G}^{\mathrm{Q-PGT}}(\theta)\right] \\
\overset{(9)}{=} & \mathbb{E}\left[\frac{1}{1-\gamma}\widehat{Q}^{\theta}(s_{H_1-1}, a_{H_1-1})\nabla \log \pi^{\theta}(a_{H_1-1} \mid s_{H_1-1})\right] \\
= & \mathbb{E}_{s_{H_1-1}, a_{H_1-1}}\left[\mathbb{E}\left[\frac{1}{1-\gamma}\widehat{Q}^{\theta}(s_{H_1-1}, a_{H_1-1})\nabla \log \pi^{\theta}(a_{H_1-1} \mid s_{H_1-1}) \mid s_{H_1-1}, a_{H_1-1}\right]\right] \\
= & \mathbb{E}_{s_{H_1-1}, a_{H_1-1}}\left[\frac{1}{1-\gamma}\mathbb{E}\left[\widehat{Q}^{\theta}(s_{H_1-1}, a_{H_1-1}) \mid s_{H_1-1}, a_{H_1-1}\right]\nabla \log \pi^{\theta}(a_{H_1-1} \mid s_{H_1-1})\right] \\
= & \mathbb{E}_{s_{H_1-1}, a_{H_1-1}}\left[\frac{1}{1-\gamma}Q^{\theta}(s_{H_1-1}, a_{H_1-1})\nabla \log \pi^{\theta}(a_{H_1-1} \mid s_{H_1-1})\right] \\
\overset{(5)}{=} & G(\theta),
\end{aligned}
$$

where the fourth equality uses $\mathbb{E}\left[\widehat{Q}^{\theta}(s_{H_1-1}, a_{H_1-1}) \mid s_{H_1-1}, a_{H_1-1}\right] = Q^{\theta}(s_{H_1-1}, a_{H_1-1})$, and the last line uses (5) with $s_{H_1-1} \sim d_{\rho}^{\theta}$, $a_{H_1-1} \sim \pi^{\theta}(\cdot \mid s_{H_1-1})$. Here, both the unbiased sampling of (s_{H_1-1}, a_{H_1-1}) and the unbiased estimate $\widehat{Q}^{\theta}(s_{H_1-1}, a_{H_1-1})$ of $Q^{\theta}(s_{H_1-1}, a_{H_1-1})$ are shown by Yuan et al. [59, Algorithm 3 and Lemma 4].

D.2 α-UPG implementations

Algorithm 5 provides the general architecture of the implementation of α-UPG.

Algorithm 5: α-UPG: α-Unbiased Policy Gradient

Input: Initial state distribution ρ, policy π^{θ}, discount factor $\gamma \in [0, 1)$, hyperparameter $\alpha \in [0, 1)$

Initialize $s_0 \sim \rho$ and $a_0 \sim \pi^{\theta}(\cdot \mid s_0)$, the horizon $H - 1 \sim \mathrm{Geom}(1 - \gamma^{1-\alpha})$
for $t = 0$ ***to*** $H - 1$ **do**
  Store the vector $\nabla \log \pi^{\theta}(a_t \mid s_t)$ and the scalar $\gamma^{\alpha t} r(s_t, a_t)$
  Sample $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$ and $a_{t+1} \sim \pi^{\theta}(\cdot \mid s_{t+1})$
Build the undiscounted gradient estimator $\widehat{G}^{\alpha-\mathrm{UPG}}(\theta)$ from the stored $\nabla \log \pi^{\theta}(a_t \mid s_t)$, $\gamma^{\alpha t} r(s_t, a_t)$

Output: $\widehat{G}^{\alpha-\mathrm{UPG}}(\theta)$

Then Algorithm 6 implements α-UGPOMDP gradient estimator $\widehat{G}^{\alpha-\mathrm{UGPOMDP}}(\theta)$ in (12), and Algorithm 7 implements the unbiased gradient estimator α-QPGT $\widehat{G}^{\alpha-\mathrm{QPGT}}(\theta)$ in (13), respectively.

Algorithm 6: α-UGPOMDP: α-UGPOMDP gradient estimator in (12)

Input: Initial state distribution ρ, policy π^{θ}, discount factor $\gamma \in [0, 1)$, hyperparameter $\alpha \in [0, 1)$

Initialize $s_0 \sim \rho$ and $a_0 \sim \pi^{\theta}(\cdot \mid s_0)$, the horizon $H - 1 \sim \mathrm{Geom}(1 - \gamma^{1-\alpha})$
Set the estimate $\widehat{G}^{\alpha-\mathrm{UGPOMDP}}(\theta) = r(s_0, a_0) v_0$ \\ *Start to estimate* $\widehat{G}^{\alpha-\mathrm{UGPOMDP}}(\theta)$
for $t = 0$ ***to*** $H - 2$ **do**
  Sample $s_{t+1} \sim P(\cdot \mid s_t, a_t)$
  Sample $a_{t+1} \sim \pi^{\theta}(\cdot \mid s_{t+1})$ and obtain $r(s_{t+1}, a_{t+1})$
  $v_{t+1} \leftarrow v_t + \nabla \log \pi^{\theta}(a_{t+1} \mid s_{t+1})$
  $\widehat{G}^{\alpha-\mathrm{UGPOMDP}}(\theta) \leftarrow \widehat{G}^{\alpha-\mathrm{UGPOMDP}}(\theta) + \gamma^{\alpha(t+1)} r(s_{t+1}, a_{t+1}) v_{t+1}$

Output: $\widehat{G}^{\alpha-\mathrm{UGPOMDP}}(\theta)$

Remark 7. Notice that in Algorithm 7, H_1 and H_2 are sampled from different geometric distributions, which is the same case as in RPG [61]. Indeed, $H_1 - 1 \sim \mathrm{Geom}(1 - \gamma)$, which is the same as in

Algorithm 7: α**-QPGT:** α-QPGT gradient estimator in (13)

Input: Initial state distribution ρ, policy π^θ, discount factor $\gamma \in [0,1)$, hyperparameter $\alpha \in [0,1)$

1 Initialize $s_0 \sim \rho$ and $a_0 \sim \pi^\theta(\cdot \mid s_0)$, the horizons $H_1 - 1 \sim \text{Geom}(1-\gamma)$ and $H_2 - 1 \sim \text{Geom}(1-\gamma^{1-\alpha})$

2 **for** $t = 0$ ***to*** $H_1 - 2$ **do**

3 Sample $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$

4 Sample $a_{t+1} \sim \pi^\theta(\cdot \mid s_{t+1})$ \\ *Accept* (s_{H_1-1}, a_{H_1-1})

5 Set the estimate $\widehat{Q}^\theta(s_{H_1-1}, a_{H_1-1}) = r(s_{H_1-1}, a_{H_1-1})$ \\ *Start to estimate* $\widehat{Q}^\theta(s_{H_1-1}, a_{H_1-1})$

6 **for** $t = H_1 - 1$ ***to*** $H_1 + H_2 - 3$ **do**

7 Sample $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$

8 Sample $a_{t+1} \sim \pi^\theta(\cdot \mid s_{t+1})$

9 $\widehat{Q}^\theta(s_{H_1-1}, a_{H_1-1}) \leftarrow \widehat{Q}^\theta(s_{H_1-1}, a_{H_1-1}) + \gamma^{\alpha((t+1)-(H_1-1))} r(s_{t+1}, a_{t+1})$

10 Compute Q-PGT in (9): $\widehat{G}^{\text{Q-PGT}}(\theta) = \frac{1}{1-\gamma} \widehat{Q}^\theta(s_{H_1-1}, a_{H_1-1}) \nabla \log \pi^\theta(a_{H_1-1} \mid s_{H_1-1})$

Output: $\widehat{G}^{\text{Q-PGT}}(\theta)$

Algorithm 4, because Algorithm 7 first simulates the state visitation distribution d_ρ^θ of the original discounted MDP with the discount factor γ. The $H_2 - 1 \sim \text{Geom}(1-\gamma^{1-\alpha})$, which is different to the one in Algorithm 4 with $\text{Geom}(1-\gamma)$, because Algorithm 7 will construct the discounted gradient estimator for α-UPG starting from Line 5, where the discount factor is specifically equal to γ^α.

E Proofs of Section 3

E.1 Proof of Lemma 1

Proof. The expected length H of sampling the trajectory $\tau = \{s_0, a_0, s_1, a_1, \cdots, s_{H-1}, a_{H-1}\}$ is

$$\mathbb{E}[H] = \sum_{k=0}^{\infty} \Pr(H = k+1)(k+1) = (1-\gamma) \sum_{k=0}^{\infty} \gamma^k (k+1) = \frac{1}{1-\gamma}.$$

Now we verify that $\widehat{G}^{\text{UGPOMDP}}(\theta)$ in (11) is an unbiased estimate of (3). Indeed, from (11) (or from Algorithm 2),

$$\widehat{G}^{\text{UGPOMDP}}(\theta) = \sum_{t=0}^{H-1} r(s_t, a_t) v_t,$$

where H is the length of the trajectory τ, sampled from $\text{Geom}(1-\gamma)$, and

$$v_t = \sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}).$$

Taking expectation with respect to the randomness of sampling τ from (11), we have

$$\begin{aligned}
\mathbb{E}\left[\widehat{G}^{\text{UGPOMDP}}(\theta)\right] &= \mathbb{E}\left[\sum_{t=0}^{H-1} r(s_t, a_t) v_t\right] \\
&= \sum_{k=0}^{\infty} \Pr(H-1=k)\mathbb{E}\left[\sum_{t=0}^{H-1} r(s_t, a_t) v_t \mid H-1=k\right] \\
&= \sum_{k=0}^{\infty} (1-\gamma)\gamma^k \mathbb{E}\left[\sum_{t=0}^{k} r(s_t, a_t) v_t\right] \\
&= (1-\gamma)\mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \sum_{t=0}^{k} r(s_t, a_t) v_t\right] \\
&= (1-\gamma)\mathbb{E}\left[\sum_{t=0}^{\infty} r(s_t, a_t) v_t \sum_{k=t}^{\infty} \gamma^k\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) v_t\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \left(\sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'})\right)\right],
\end{aligned}$$

where the fifth line is obtained by the change in order of summation, and the last line is obtained by the definition of v_t, and it recovers $G(\theta)$ in (3).

□

E.2 Proof of Lemma 2

Proof. By $H - 1 \sim \text{Geom}(1 - \gamma^{1-\alpha})$, we have that

$$\mathbb{E}[H] = \sum_{k=0}^{\infty} \Pr(H = k+1)(k+1) = (1-\gamma^{1-\alpha}) \sum_{k=0}^{\infty} \gamma^{k(1-\alpha)}(k+1) = \frac{1}{1-\gamma^{1-\alpha}},$$

where the last equality is obtained by applying (58) in Lemma 7 with $\gamma^{1-\alpha}$, where $\gamma^{1-\alpha}$ is between 0 and 1 as $\alpha \in [0, 1)$.

Now we verify that $\widehat{G}^{\alpha-\text{UGPOMDP}}(\theta)$ in (12) is an unbiased estimate of (3). Indeed, from (12) (or from Algorithm 6),

$$\widehat{G}^{\alpha-\text{UGPOMDP}}(\theta) = \sum_{t=0}^{H-1} \gamma^{\alpha t} r(s_t, a_t) v_t,$$

where H is the length of the trajectory τ, sampled from $\text{Geom}(1 - \gamma^{(1-\alpha)})$, and

$$v_t = \sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}).$$

Taking expectation with respect to the randomness of sampling τ from (12), we have

$$\begin{aligned}
\mathbb{E}\left[\widehat{G}^{\alpha-\mathrm{UGPOMDP}}(\theta)\right] &= \mathbb{E}\left[\sum_{t=0}^{H-1}\gamma^{\alpha t} r(s_t,a_t) v_t\right] \\
&= \sum_{k=0}^{\infty}\Pr(H-1=k)\mathbb{E}\left[\sum_{t=0}^{H-1}\gamma^{\alpha t} r(s_t,a_t) v_t \mid H-1=k\right] \\
&= \sum_{k=0}^{\infty}(1-\gamma^{(1-\alpha)})\gamma^{k(1-\alpha)}\mathbb{E}\left[\sum_{t=0}^{k}\gamma^{\alpha t} r(s_t,a_t) v_t\right] \\
&= (1-\gamma^{(1-\alpha)})\mathbb{E}\left[\sum_{k=0}^{\infty}\gamma^{k(1-\alpha)}\sum_{t=0}^{k}\gamma^{\alpha t} r(s_t,a_t) v_t\right] \\
&= (1-\gamma^{(1-\alpha)})\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^{\alpha t} r(s_t,a_t) v_t \sum_{k=t}^{\infty}\gamma^{k(1-\alpha)}\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^{\alpha t}\gamma^{t(1-\alpha)} r(s_t,a_t) v_t\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^{t} r(s_t,a_t)\left(\sum_{t'=0}^{t}\nabla\log\pi^{\theta}(a_{t'}\mid s_{t'})\right)\right],
\end{aligned}$$

where the fifth line is obtained by the change in order of summation, and the last line is obtained by the definition of v_t, and it recovers $G(\theta)$ in (3).

Lastly, we verify that $\widehat{G}^{\alpha-\mathrm{QPGT}}(\theta)$ in (13) is an unbiased estimate of (4).

Let

$$\widehat{Q}^{\,\theta}(s_{H_1-1},a_{H_1-1}) \overset{\text{def}}{=} \sum_{t=H_1-1}^{H_1+H_2-2}\gamma^{\alpha(t-(H_1-1))} r(s_t,a_t).$$

Similar to the derivation of the unbiased gradient estimator $\widehat{G}^{\mathrm{Q-PGT}}(\theta)$ in (9) for Q-PGT right after Algorithm 4, from (13) and Algorithm 7, we have

$$\begin{aligned}
&\mathbb{E}\left[\widehat{G}^{\alpha-\mathrm{QPGT}}(\theta)\right] \\
\overset{(13)}{=}\ &\mathbb{E}\left[\Big(\frac{1}{1-\gamma}\sum_{t=H_1-1}^{H_1+H_2-2}\gamma^{\alpha(t-(H_1-1))} r(s_t,a_t)\Big)\nabla\log\pi^{\theta}(a_{H_1-1}\mid s_{H_1-1})\right] \\
=\ &\mathbb{E}\left[\frac{1}{1-\gamma}\widehat{Q}^{\,\theta}(s_{H_1-1},a_{H_1-1})\nabla\log\pi^{\theta}(a_{H_1-1}\mid s_{H_1-1})\right] \\
=\ &\mathbb{E}_{s_{H_1-1},a_{H_1-1}}\left[\mathbb{E}\left[\frac{1}{1-\gamma}\widehat{Q}^{\,\theta}(s_{H_1-1},a_{H_1-1})\nabla\log\pi^{\theta}(a_{H_1-1}\mid s_{H_1-1})\mid s_{H_1-1},a_{H_1-1}\right]\right], \\
=\ &\mathbb{E}_{s_{H_1-1},a_{H_1-1}}\left[\frac{1}{1-\gamma}\mathbb{E}\left[\widehat{Q}^{\,\theta}(s_{H_1-1},a_{H_1-1})\mid s_{H_1-1},a_{H_1-1}\right]\nabla\log\pi^{\theta}(a_{H_1-1}\mid s_{H_1-1})\right].
\end{aligned} \tag{24}$$

Now we are going to show that $\widehat{Q}^{\,\theta}(s_{H_1-1},a_{H_1-1})$ is an unbiased estimate of the Q-function $Q^{\,\theta}(s_{H_1-1},a_{H_1-1})$, knowing s_{H_1-1} and a_{H_1-1}.

Using the fact that $H_2 - 1 \sim \text{Geom}(1-\gamma^{1-\alpha})$, we have

$$
\begin{aligned}
& \mathbb{E}\left[\widehat{Q}^{\theta}(s_{H_1-1}, a_{H_1-1}) \mid s_{H_1-1}, a_{H_1-1}\right] \\
= \; & \mathbb{E}\left[\sum_{t=0}^{H_2-1} \gamma^{\alpha t} r(s_t, a_t) \mid s_0 = s_{H_1-1}, a_0 = a_{H_1-1}\right] \\
= \; & \sum_{k=0}^{\infty} \Pr(H_2 - 1 = k) \mathbb{E}\left[\sum_{t=0}^{H_2-1} \gamma^{\alpha t} r(s_t, a_t) \mid H_2 - 1 = k, s_0 = s_{H_1-1}, a_0 = a_{H_1-1}\right] \\
= \; & (1-\gamma^{1-\alpha}) \sum_{k=0}^{\infty} \gamma^{k(1-\alpha)} \mathbb{E}\left[\sum_{t=0}^{k} \gamma^{\alpha t} r(s_t, a_t) \mid s_0 = s_{H_1-1}, a_0 = a_{H_1-1}\right] \\
= \; & (1-\gamma^{1-\alpha}) \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{\alpha t} r(s_t, a_t) \sum_{k=t}^{\infty} \gamma^{k(1-\alpha)} \mid s_0 = s_{H_1-1}, a_0 = a_{H_1-1}\right] \\
= \; & \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{\alpha t} \gamma^{t(1-\alpha)} r(s_t, a_t) \mid s_0 = s_{H_1-1}, a_0 = a_{H_1-1}\right] \\
= \; & \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{t} r(s_t, a_t) \mid s_0 = s_{H_1-1}, a_0 = a_{H_1-1}\right] \\
= \; & Q^{\theta}(s_{H_1-1}, a_{H_1-1}),
\end{aligned}
$$

where the first equality is obtained by the Markov property of the MDP, and the fourth equality is obtained by changing the order of the summation.

So, we can rewrite (24) thanks to the result above that

$$
\begin{aligned}
\mathbb{E}\left[\widehat{G}^{\alpha-\text{QPGT}}(\theta)\right] &= \mathbb{E}_{s_{H_1-1}, a_{H_1-1}}\left[\frac{1}{1-\gamma} Q^{\theta}(s_{H_1-1}, a_{H_1-1}) \nabla \log \pi^{\theta}(a_{H_1-1} \mid s_{H_1-1})\right] \\
&\overset{(5)}{=} G(\theta),
\end{aligned}
$$

as we have $(s_{H_1-1}, a_{H_1-1}) \sim d_{\rho}^{\theta} \circ \pi^{\theta}(\cdot \mid s_{H_1-1})$ as in Algorithm 4, since H_1 follows the same geometric distribution in both Algorithms 4 and 7, which is mentioned in Remark 7. □

F Proofs of Section 4.1

F.1 Proof of Proposition 1

Here we apply the proof techniques from Khaled and Richtárik [20, Theorem 2] to prove Proposition 1.

Proof. We start with L-smoothness of J from Assumption 3, which implies

$$
\begin{aligned}
J^{(k+1)} &\geq J^{(k)} + \left\langle G^{(k)}, \theta^{(k+1)} - \theta^{(k)} \right\rangle - \frac{L}{2}\left\|\theta^{(k+1)} - \theta^{(k)}\right\|^2 \\
&= J^{(k)} + \eta \left\langle G^{(k)}, \widehat{G}^{(k)} \right\rangle - \frac{L\eta^2}{2}\left\|\widehat{G}^{(k)}\right\|^2.
\end{aligned}
$$

Taking expectations conditioned on $\theta^{(k)}$ with the shorthand $\mathbb{E}_k[\cdot]$ for $\mathbb{E}\left[\cdot \mid \theta^{(k)}\right]$, and noticing that $\mathbb{E}_k\left[\widehat{G}^{(k)}\right] = G^{(k)}$, as α-UPG is unbiased gradient estimator of $J(\cdot)$, we get

$$
\begin{aligned}
\mathbb{E}_k\left[J^{(k+1)}\right] &\geq J^{(k)} + \eta\left\|G^{(k)}\right\|^2 - \frac{L\eta^2}{2}\mathbb{E}_k\left[\left\|\widehat{G}^{(k)}\right\|^2\right] \\
&\overset{(14)}{\geq} J^{(k)} + \eta\left\|G^{(k)}\right\|^2 - \frac{L\eta^2}{2}\left(2A(J^{\star} - J^{(k)}) + B\left\|G^{(k)}\right\|^2 + C\right) \\
&= J^{(k)} + \eta\left(1 - \frac{LB\eta}{2}\right)\left\|G^{(k)}\right\|^2 - L\eta^2 A(J^{\star} - J^{(k)}) - \frac{LC\eta^2}{2}.
\end{aligned}
$$

Subtracting $J^\star$ from both sides gives

$$-\left(J^\star - \mathbb{E}_k\left[J^{(k+1)}\right]\right) \;\geq\; -(1+L\eta^2 A)(J^\star - J^{(k)}) + \eta\left(1-\frac{LB\eta}{2}\right)\left\|G^{(k)}\right\|^2 - \frac{LC\eta^2}{2}.$$

Taking the total expectation and rearranging, we get

$$\mathbb{E}\left[J^\star - J^{(k+1)}\right] + \eta\left(1-\frac{LB\eta}{2}\right)\mathbb{E}\left[\left\|G^{(k)}\right\|^2\right] \;\leq\; (1+L\eta^2 A)\mathbb{E}\left[J^\star - J^{(k)}\right] + \frac{LC\eta^2}{2}.$$

Letting $\delta^{(k)} \overset{\text{def}}{=} \mathbb{E}\left[J^\star - J^{(k)}\right]$ and $g^{(k)} \overset{\text{def}}{=} \mathbb{E}\left[\left\|G^{(k)}\right\|^2\right]$, we can rewrite the last inequality as

$$\eta\left(1-\frac{LB\eta}{2}\right)g^{(k)} \;\leq\; (1+L\eta^2 A)\delta^{(k)} - \delta^{(k+1)} + \frac{LC\eta^2}{2}. \tag{25}$$

We now introduce a sequence of weights $w_{-1}, w_0, w_1, \cdots, w_{K-1}$ based on a technique developed by Stich [47]. Let $w_{-1} > 0$. Define $w_k \overset{\text{def}}{=} \frac{w_{k-1}}{1+L\eta^2 A}$ for all $k \geq 0$. Notice that if $A = 0$, we have $w_k = w_{k-1} = \cdots = w_{-1}$. Multiplying (25) by w_k/η,

$$\begin{aligned}\left(1-\frac{LB\eta}{2}\right)w_k g^{(k)} &\leq \frac{w_k(1+L\eta^2 A)}{\eta}\delta^{(k)} - \frac{w_k}{\eta}\delta^{(k+1)} + \frac{LC\eta}{2}w_k \\ &= \frac{w_{t-1}}{\eta}\delta^{(k)} - \frac{w_k}{\eta}\delta^{(k+1)} + \frac{LC\eta}{2}w_k.\end{aligned}$$

Summing up both sides as $k = 0, 1, \cdots, K-1$ and using telescopic sum, we have,

$$\begin{aligned}\left(1-\frac{LB\eta}{2}\right)\sum_{k=0}^{K-1} w_k g^{(k)} &\leq \frac{w_{-1}}{\eta}\delta^{(0)} - \frac{w_{K-1}}{\eta}\delta^{(K)} + \frac{LC\eta}{2}\sum_{k=0}^{K-1} w_k \\ &\leq \frac{w_{-1}}{\eta}\delta^{(0)} + \frac{LC\eta}{2}\sum_{k=0}^{K-1} w_k.\end{aligned} \tag{26}$$

Let $W_K \overset{\text{def}}{=} \sum_{k=0}^{K-1} w_k$. Dividing both sides by W_K, we have,

$$\left(1-\frac{LB\eta}{2}\right)\min_{0\leq k\leq K-1} g^{(k)} \;\leq\; \frac{1}{W_k}\cdot\left(1-\frac{LB\eta}{2}\right)\sum_{k=0}^{K-1} w_k g^{(k)} \;\leq\; \frac{w_{-1}}{W_k}\frac{\delta^{(0)}}{\eta} + \frac{LC\eta}{2}. \tag{27}$$

Note that,

$$W_K \;=\; \sum_{k=0}^{K-1} w_k \geq \sum_{k=0}^{K-1}\min_{0\leq i\leq K-1} w_i \;=\; K w_{K-1} \;=\; \frac{K w_{-1}}{(1+L\eta^2 A)^K}.$$

Using this in (27),

$$\left(1-\frac{LB\eta}{2}\right)\min_{0\leq k\leq K-1} g^{(k)} \;\leq\; \frac{\delta^{(0)}(1+L\eta^2 A)^K}{\eta K} + \frac{LC\eta}{2}. \tag{28}$$

Our choice of stepsize guarantees that no matter $B > 0$ or $B = 0$, we have $1 - \frac{LB\eta}{2} > 0$. Dividing both sides by $1 - \frac{LB\eta}{2}$ and rearranging yields the proposition's claim in the case when $A > 0$.

If $A = 0$, we know that $\{w_k\}_{k\geq -1}$ is a constant sequence. In this case, $W_K = K w_{-1}$. Dividing both sides of (26) by W_K, we have,

$$\left(1-\frac{LB\eta}{2}\right)\frac{1}{K}\sum_{k=0}^{K-1} g^{(k)} \;\leq\; \frac{\delta^{(0)}}{\eta K} + \frac{LC\eta}{2}.$$

Dividing both sides by $1 - \frac{LB\eta}{2}$ and rearranging yields the proposition's claim in the case when $A = 0$. □

F.2 Proof of Corollary 1

Proof. Given $\epsilon > 0$, from Corollary 1 in Khaled and Richtárik [20], we know that if $\eta = \min\left\{\frac{1}{\sqrt{LAK}}, \frac{1}{LB}, \frac{\epsilon}{2LC}\right\}$ and the number of iterations K satisfies

$$K \geq \frac{12L(J^\star - J^{(0)})}{\epsilon^2} \max\left\{B, \frac{12A(J^\star - J^{(0)})}{\epsilon^2}, \frac{2C}{\epsilon^2}\right\},$$

we have

$$\min_{0\leq k\leq K-1} \mathbb{E}\left[\left\|G^{(k)}\right\|^2\right] \leq \frac{2(J^\star - J^{(0)})(1+L\eta^2 A)^K}{\eta K(2-LB\eta)} + \frac{LC\eta}{2-LB\eta} \leq \epsilon^2.$$

□

G Proofs of Section 4.2

G.1 Proof of Theorem 6

First, we provide the complete version of Theorem 6 as follows.

Theorem 8. *Under Assumption 5, consider an α-UPG among (11), (9), (12) and (13). We have*

$$\mathbb{E}\left[\|\widehat{G}(\theta)\|^2\right] \leq (1-1/m)\,\|G(\theta)\|^2 + \nu/m,$$

where m is the batch size, and

$$\nu = \begin{cases} \frac{3G^2 r_{\max}^2}{(1-\gamma)^3} & \text{for UGPOMDP (11)} \\ \frac{2G^2 r_{\max}^2}{(1-\gamma)^4} & \text{for Q-PGT (9)} \\ G^2 r_{\max}^2 f_1(\gamma^\alpha) & \text{for } \alpha\text{-UGPOMDP (12)} \\ \frac{G^2 r_{\max}^2 f_2(\gamma^\alpha)}{(1-\gamma)^2} & \text{for } \alpha\text{-QPGT (13)} \\ \frac{2G^2 r_{\max}^2}{(1-\gamma)^3(1-\gamma\sqrt{\gamma})} & \text{for } \alpha\text{-QPGT (i.e., RPG [61]) in (13) with } \alpha = \frac{1}{2} \end{cases}, \tag{29}$$

where $f_1 : (\gamma, 1) \to \mathbb{R}$ and $f_2 : (\gamma, 1) \to \mathbb{R}$ are scalar functions defined as follows,

$$f_1(x) \overset{def}{=} \frac{(1-x)^2 - (1-x)(x-\gamma) - (x-\gamma)^2}{(1-\gamma)^2(1-x)^3} + \frac{x(x-\gamma)\left((1-x)+(1-\gamma x)\right)}{(1-x)^3(1-\gamma x)^2}, \tag{30}$$

$$f_2(x) \overset{def}{=} \frac{(1-\gamma) - 2(x-\gamma)}{(1-\gamma)(1-x)^2} + \frac{x(x-\gamma)}{(1-x)^2(1-\gamma x)}. \tag{31}$$

Proof. Let $g(\tau \mid \theta)$ be a stochastic gradient estimator of one single sampled trajectory τ. Thus $\widehat{G}(\theta) = \frac{1}{m}\sum_{i=1}^m g(\tau_i \mid \theta)$. By Lemmas 1 and 2, both $\widehat{G}(\theta)$ and $g(\tau \mid \theta)$ are unbiased gradient estimators of $J(\theta)$ for all (11), (9), (12) and (13), which is $G(\theta)$. By following the derivation of

Equation (68) in Yuan et al. [58], we have

$$\begin{aligned}
\mathbb{E}\left[\left\|\widehat{G}(\theta)\right\|^2\right] &= \mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=0}^{m-1} g(\tau_i \mid \theta)\right\|^2\right] \\
&= \mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=0}^{m-1} g(\tau_i \mid \theta) - G(\theta) + G(\theta)\right\|^2\right] \\
&= \|G(\theta)\|^2 + \mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=0}^{m-1} (g(\tau_i \mid \theta) - G(\theta))\right\|^2\right] \\
&= \|G(\theta)\|^2 + \frac{1}{m^2}\sum_{i=0}^{m-1} \mathbb{E}\left[\|g(\tau_i \mid \theta) - G(\theta)\|^2\right] \\
&= \|G(\theta)\|^2 + \frac{1}{m}\mathbb{E}\left[\|g(\tau_1 \mid \theta) - G(\theta)\|^2\right] \\
&= \left(1 - \frac{1}{m}\right)\|G(\theta)\|^2 + \frac{\mathbb{E}\left[\|g(\tau_1 \mid \theta)\|^2\right]}{m},
\end{aligned} \tag{32}$$

where the third, the fourth and the fifth lines are all obtained by using $G(\theta) = \mathbb{E}\left[g(\tau_i \mid \theta)\right]$. It remains to show $\mathbb{E}_\tau\left[\|g(\tau \mid \theta)\|^2\right]$ is bounded under Assumption 5 for UGPOMDP in (11), Q-PGT in (9) with Algorithm 4, α-UGPOMDP in (12) when $\alpha \neq 0$, and α-QPGT in (13) with Algorithm 7 when $\alpha \neq 0$, respectively.

Part I: UGPOMDP in (11).

First, consider $\widehat{G}^{\mathrm{UGPOMDP}}(\theta)$ from (11) and with random horizon $H - 1 \sim \mathrm{Geom}(1-\gamma)$:

$$g(\tau \mid \theta) = \widehat{G}^{\mathrm{UGPOMDP}}(\theta) \overset{(11)}{=} \sum_{t=0}^{H-1} r(s_t, a_t)\left(\sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'})\right).$$

Taking expectation with respect to the randomness of sampling τ from (11), we have

$$
\begin{aligned}
& \mathbb{E}_\tau \left[\|g(\tau \mid \theta)\|^2 \right] \\
\overset{(11)}{=} \; & \mathbb{E}_\tau \left[\left\| \sum_{t=0}^{H-1} r(s_t, a_t) \left(\sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}) \right) \right\|^2 \right] \\
= \; & \sum_{k=0}^{\infty} \Pr(H-1=k) \mathbb{E} \left[\left\| \sum_{t=0}^{H-1} r(s_t, a_t) \left(\sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}) \right) \right\|^2 \mid H-1=k \right] \\
= \; & \sum_{k=0}^{\infty} (1-\gamma)\gamma^k \mathbb{E} \left[\left\| \sum_{t=0}^{k} r(s_t, a_t) \left(\sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}) \right) \right\|^2 \right] \\
\leq \; & (1-\gamma) \sum_{k=0}^{\infty} \gamma^k \mathbb{E} \left[\left(\sum_{t=0}^{k} r(s_t, a_t)^2 \right) \left(\sum_{t=0}^{k} \left\| \sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}) \right\|^2 \right) \right] \\
\leq \; & r_{\max}^2 (1-\gamma) \sum_{k=0}^{\infty} \gamma^k (k+1) \sum_{t=0}^{k} \mathbb{E} \left[\left\| \sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}) \right\|^2 \right] \\
\overset{(56)}{=} \; & r_{\max}^2 (1-\gamma) \sum_{k=0}^{\infty} \gamma^k (k+1) \sum_{t=0}^{k} \sum_{t'=0}^{t} \mathbb{E} \left[\|\nabla \log \pi^\theta(a_{t'} \mid s_{t'})\|^2 \right] \\
\overset{(54)}{\leq} \; & G^2 r_{\max}^2 (1-\gamma) \sum_{k=0}^{\infty} \gamma^k (k+1) \sum_{t=0}^{k} (t+1) \\
= \; & \frac{1}{2} G^2 r_{\max}^2 (1-\gamma) \sum_{k=0}^{\infty} \gamma^k (k+1)^2 (k+2) \\
\overset{(60)}{\leq} \; & \frac{3G^2 r_{\max}^2}{(1-\gamma)^3},
\end{aligned}
\tag{33}
$$

where the fourth line (the first inequality) is obtained from the Cauchy-Schwarz inequality, the fifth line (the second inequality) is obtained by using $|r(s_t, a_t)| \leq r_{\max}$ and the last line is obtained by (60) in Lemma 7.

The above together with (32) imply that the expected smoothness assumption holds for the batch version of UGPOMDP with

$$
\mathbb{E} \left[\left\| \widehat{G}^{\mathrm{UGPOMDP}}(\theta) \right\|^2 \right] \overset{(32),\,(33)}{\leq} \left(1 - \frac{1}{m} \right) \|G(\theta)\|^2 + \frac{3G^2 r_{\max}^2}{m(1-\gamma)^3}.
$$

Part II: Q-PGT in (9) with Algorithm 4.

Now, consider $\widehat{G}^{\mathrm{Q-PGT}}(\theta)$ from (9) sampled from Algorithm 4. With random horizon $H_1 - 1 \sim \mathrm{Geom}(1-\gamma)$, we have

$$
g(\tau \mid \theta) = \widehat{G}^{\mathrm{Q-PGT}}(\theta) \overset{(9)}{=} \frac{1}{1-\gamma} \widehat{Q}^{\,\theta}(s_{H_1-1}, a_{H_1-1}) \nabla \log \pi^\theta(a_{H_1-1} \mid s_{H_1-1}),
$$

where, from Algorithm 4, $\widehat{Q}^{\,\theta}(s_{H_1-1}, a_{H_1-1})$ is computed as

$$
\widehat{Q}^{\,\theta}(s_{H_1-1}, a_{H_1-1}) = \sum_{t=H_1-1}^{H_1+H_2-2} r(s_t, a_t). \tag{34}
$$

Taking expectation with respect to the randomness of sampling τ from (9) in Algorithm 4, we have

$$\begin{aligned}
&\mathbb{E}\left[\|g(\tau \mid \theta)\|^2\right] \\
&= \mathbb{E}\left[\left\|\frac{1}{1-\gamma}\widehat{Q}^{\theta}(s_{H_1-1}, a_{H_1-1})\nabla \log \pi^{\theta}(a_{H_1-1} \mid s_{H_1-1})\right\|^2\right] \\
&= \mathbb{E}\left[\frac{1}{(1-\gamma)^2}\widehat{Q}^{\theta}(s_{H_1-1}, a_{H_1-1})^2\left\|\nabla \log \pi^{\theta}(a_{H_1-1} \mid s_{H_1-1})\right\|^2\right] \\
&= \mathbb{E}_{s_{H_1-1}, a_{H_1-1}}\left[\mathbb{E}\left[\frac{\widehat{Q}^{\theta}(s_{H_1-1}, a_{H_1-1})^2}{(1-\gamma)^2}\left\|\nabla \log \pi^{\theta}(a_{H_1-1} \mid s_{H_1-1})\right\|^2 \mid s_{H_1-1}, a_{H_1-1}\right]\right] \\
&= \mathbb{E}_{s_{H_1-1}, a_{H_1-1}}\left[\frac{\left\|\nabla \log \pi^{\theta}(a_{H_1-1} \mid s_{H_1-1})\right\|^2}{(1-\gamma)^2}\mathbb{E}\left[\widehat{Q}^{\theta}(s_{H_1-1}, a_{H_1-1})^2 \mid s_{H_1-1}, a_{H_1-1}\right]\right].
\end{aligned} \tag{35}$$

If $\mathbb{E}\left[\widehat{Q}^{\theta}(s_{H_1-1}, a_{H_1-1})^2 \mid s_{H_1-1}, a_{H_1-1}\right]$ inside (35) is bounded, by using $\mathbb{E}\left[\left\|\nabla \log \pi^{\theta}(a_{H_1-1} \mid s_{H_1-1})\right\|^2\right] \leq G^2$ in (18), we obtain (35) bounded.

From (34) and by using the Markov property of the MDP, we rewrite $\widehat{Q}^{\theta}(s_{H_1-1}, a_{H_1-1})$ as

$$\widehat{Q}^{\theta}(s_0, a_0) = \sum_{t=0}^{H_2-1} r(s_t, a_t), \tag{36}$$

with $(s_0, a_0) = (s_{H_1-1}, a_{H_1-1})$ and H_2 is the length of the trajectory for estimating $Q^{\theta}(s_{H_1-1}, a_{H_1-1})$.

Thus, we have

$$\begin{aligned}
&\mathbb{E}\left[\widehat{Q}^{\theta}(s_0, a_0)^2 \mid s_0 = s_{H_1-1}, a_0 = a_{H_1-1}\right] \\
\overset{(36)}{=}\ &\mathbb{E}\left[\left(\sum_{t=0}^{H_2-1} r(s_t, a_t)\right)^2 \mid s_0 = s_{H_1-1}, a_0 = a_{H_1-1}\right] \\
=\ &\sum_{k=0}^{\infty} \Pr(H_2 - 1 = k)\mathbb{E}\left[\left(\sum_{t=0}^{H_2-1} r(s_t, a_t)\right)^2 \mid H_2 - 1 = k, s_0 = s_{H_1-1}, a_0 = a_{H_1-1}\right] \\
=\ &(1-\gamma)\sum_{k=0}^{\infty}\gamma^k \mathbb{E}\left[\left(\sum_{t=0}^{k} r(s_t, a_t)\right)^2 \mid s_0 = s_{H_1-1}, a_0 = a_{H_1-1}\right] \\
\leq\ &r_{\max}^2(1-\gamma)\sum_{k=0}^{\infty}\gamma^k(k+1)^2 \\
\overset{(59)}{\leq}\ &\frac{2r_{\max}^2}{(1-\gamma)^2},
\end{aligned} \tag{37}$$

where the first inequality in the second last line is obtained as $|r(s_t, a_t)| \in r_{\max}$ for all $(s_t, a_t) \in \mathcal{S} \times \mathcal{A}$.

Combining (35) and (37) and using (18), we have

$$\mathbb{E}\left[\|g(\tau \mid \theta)\|^2\right] \overset{(35),\,(37),\,(18)}{\leq} \frac{2G^2 r_{\max}^2}{(1-\gamma)^4}. \tag{38}$$

The above together with (32) imply that the expected smoothness assumption holds for the batch version of Q-PGT with

$$\mathbb{E}\left[\left\|\widehat{G}^{\mathrm{Q-PGT}}(\theta)\right\|^2\right] \overset{(32),\,(38)}{\leq} \left(1 - \frac{1}{m}\right)\|G(\theta)\|^2 + \frac{2G^2 r_{\max}^2}{m(1-\gamma)^4}.$$

Part III: α-UGPOMDP in (12) when $\alpha \neq 0$.

Now, consider $\widehat{G}^{\alpha-\text{UGPOMDP}}(\theta)$ in (12) when $\alpha \neq 0$, and with random horizon $H-1 \sim \text{Geom}(1-\gamma^{1-\alpha})$, we have

$$g(\tau \mid \theta) = \widehat{G}^{\alpha-\text{UGPOMDP}}(\theta) \overset{(12)}{=} \sum_{t=0}^{H-1} \gamma^{\alpha t} r(s_t, a_t) \left(\sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}) \right).$$

Taking expectation with respect to the randomness of sampling τ from (12), we have

$$\begin{aligned}
& \mathbb{E}_\tau \left[\|g(\tau \mid \theta)\|^2 \right] \\
\overset{(12)}{=} \; & \mathbb{E}_\tau \left[\left\| \sum_{t=0}^{H-1} \gamma^{\alpha t} r(s_t, a_t) \left(\sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}) \right) \right\|^2 \right] \\
= \; & \sum_{k=0}^{\infty} \Pr(H-1=k) \mathbb{E} \left[\left\| \sum_{t=0}^{H-1} \gamma^{\alpha t} r(s_t, a_t) \left(\sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}) \right) \right\|^2 \mid H-1=k \right] \\
= \; & \sum_{k=0}^{\infty} (1-\gamma^{1-\alpha}) \gamma^{(1-\alpha)k} \mathbb{E} \left[\left\| \sum_{t=0}^{k} \gamma^{\alpha t} r(s_t, a_t) \left(\sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}) \right) \right\|^2 \right] \\
= \; & \sum_{k=0}^{\infty} (1-\gamma^{1-\alpha}) \gamma^{(1-\alpha)k} \mathbb{E} \left[\left\| \sum_{t=0}^{k} \gamma^{\alpha t/2} r(s_t, a_t) \gamma^{\alpha t/2} \left(\sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}) \right) \right\|^2 \right] \\
\leq \; & (1-\gamma^{1-\alpha}) \sum_{k=0}^{\infty} \gamma^{(1-\alpha)k} \mathbb{E} \left[\left(\sum_{t=0}^{k} \gamma^{\alpha t} r(s_t, a_t)^2 \right) \left(\sum_{t=0}^{k} \gamma^{\alpha t} \left\| \sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}) \right\|^2 \right) \right] \\
\leq \; & r_{\max}^2 (1-\gamma^{1-\alpha}) \sum_{k=0}^{\infty} \gamma^{(1-\alpha)k} \left(\sum_{t=0}^{k} \gamma^{\alpha t} \right) \sum_{t=0}^{k} \gamma^{\alpha t} \mathbb{E} \left[\left\| \sum_{t'=0}^{t} \nabla \log \pi^\theta(a_{t'} \mid s_{t'}) \right\|^2 \right] \\
\overset{(56)}{=} \; & r_{\max}^2 (1-\gamma^{1-\alpha}) \sum_{k=0}^{\infty} \gamma^{(1-\alpha)k} \cdot \frac{1-\gamma^{\alpha(k+1)}}{1-\gamma^\alpha} \sum_{t=0}^{k} \gamma^{\alpha t} \sum_{t'=0}^{t} \mathbb{E} \left[\|\nabla \log \pi^\theta(a_{t'} \mid s_{t'})\|^2 \right] \\
\overset{(54)}{\leq} \; & G^2 r_{\max}^2 (1-\gamma^{1-\alpha}) \sum_{k=0}^{\infty} \gamma^{(1-\alpha)k} \cdot \frac{1-\gamma^{\alpha(k+1)}}{1-\gamma^\alpha} \sum_{t=0}^{k} \gamma^{\alpha t} (t+1) \\
\overset{(57)}{=} \; & \frac{G^2 r_{\max}^2 (1-\gamma^{1-\alpha})}{(1-\gamma^\alpha)^2} \underbrace{\sum_{k=0}^{\infty} \gamma^{(1-\alpha)k} (1-\gamma^{\alpha(k+1)}) \left(\frac{1-\gamma^{\alpha(k+1)}}{1-\gamma^\alpha} - (k+1)\gamma^{\alpha(k+1)} \right)}_{(*)}, \qquad (39)
\end{aligned}$$

where the first inequality is obtained from the Cauchy-Schwarz inequality, the second inequality is obtained by using $|r(s_t, a_t)| \leq r_{\max}$, and the last line is obtained by applying (57) with γ^α in Lemma 7.

From (39), we have

$$\begin{aligned}
(*) &= \sum_{k=0}^{\infty} \left(\gamma^{(1-\alpha)k} - \gamma^{k+\alpha} \right) \left(\frac{1-\gamma^{\alpha(k+1)}}{1-\gamma^\alpha} - (k+1)\gamma^{\alpha(k+1)} \right) \\
&= \sum_{k=0}^{\infty} \frac{\gamma^{(1-\alpha)k}}{1-\gamma^\alpha} - \frac{\gamma^{k+\alpha}}{1-\gamma^\alpha} - \frac{\gamma^{k+\alpha}}{1-\gamma^\alpha} + \frac{\gamma^{k+\alpha(k+2)}}{1-\gamma^\alpha} - (k+1)\gamma^{k+\alpha} + (k+1)\gamma^{k+\alpha(k+2)} \\
&= \frac{1}{(1-\gamma^\alpha)(1-\gamma^{1-\alpha})} - \frac{2\gamma^\alpha}{(1-\gamma^\alpha)(1-\gamma)} + \frac{\gamma^{2\alpha}}{(1-\gamma^\alpha)(1-\gamma^{1+\alpha})} - \frac{\gamma^\alpha}{(1-\gamma)^2} \\
&\quad + \frac{\gamma^{2\alpha}}{(1-\gamma^{1+\alpha})^2},
\end{aligned}$$

where in the last equality, we apply (58) in Lemma 7 twice with γ and $\gamma^{1+\alpha}$ to obtain the last two terms, respectively.

So, (39) is upper bounded by

$$\begin{aligned}
&\mathbb{E}_\tau\left[\|g(\tau\mid\theta)\|^2\right]\\
&\le G^2r_{\max}^2\left(\frac{1}{(1-\gamma^\alpha)^3}-\frac{2(\gamma^\alpha-\gamma)}{(1-\gamma^\alpha)^3(1-\gamma)}+\frac{\gamma^{2\alpha}-\gamma^{1+\alpha}}{(1-\gamma^\alpha)^3(1-\gamma^{1+\alpha})}-\frac{\gamma^\alpha-\gamma}{(1-\gamma^\alpha)^2(1-\gamma)^2}\right.\\
&\quad\left.+\frac{\gamma^{2\alpha}-\gamma^{1+\alpha}}{(1-\gamma^\alpha)^2(1-\gamma^{1+\alpha})^2}\right)\\
&= G^2r_{\max}^2\Bigg(\underbrace{\frac{(1-\gamma^\alpha)^2-(1-\gamma^\alpha)(\gamma^\alpha-\gamma)-(\gamma^\alpha-\gamma)^2}{(1-\gamma)^2(1-\gamma^\alpha)^3}+\frac{\gamma^\alpha(\gamma^\alpha-\gamma)\left((1-\gamma^\alpha)+(1-\gamma^{1+\alpha})\right)}{(1-\gamma^\alpha)^3(1-\gamma^{1+\alpha})^2}}_{=f_1(\gamma^\alpha)}\Bigg),
\end{aligned}$$

where the last line is obtained from (30) with $x=\gamma^\alpha$, that is, we have, for all $x\in(\gamma,1)$,

$$f_1(x) \;=\; \frac{(1-x)^2-(1-x)(x-\gamma)-(x-\gamma)^2}{(1-\gamma)^2(1-x)^3}+\frac{x(x-\gamma)\left((1-x)+(1-\gamma x)\right)}{(1-x)^3(1-\gamma x)^2}.$$

So we have

$$\mathbb{E}_\tau\left[\|g(\tau\mid\theta)\|^2\right]\le G^2r_{\max}^2f_1(\gamma^\alpha).$$

The above together with (32) imply that the expected smoothness assumption holds for the batch version of α-UGPOMDP with

$$\mathbb{E}\left[\left\|\widehat{G}^{\alpha-\mathrm{UGPOMDP}}(\theta)\right\|^2\right]\overset{(32)}{\le}\left(1-\frac{1}{m}\right)\|G(\theta)\|^2+\frac{G^2r_{\max}^2f_1(\gamma^\alpha)}{m}.$$

Part IV: α-QPGT in (13) with Algorithm 7 when $\alpha\neq 0$.

Lastly, consider $\widehat{G}^{\alpha-\mathrm{QPGT}}(\theta)$ from (13) sampled from Algorithm 7. With random horizon $H_1-1\sim\mathrm{Geom}(1-\gamma)$, we have

$$g(\tau\mid\theta)=\widehat{G}^{\alpha-\mathrm{QPGT}}(\theta)\overset{(13)}{=}\frac{1}{1-\gamma}\widehat{Q}^{\,\theta}(s_{H_1-1},a_{H_1-1})\nabla\log\pi^\theta(a_{H_1-1}\mid s_{H_1-1}),$$

where, from (13) and Algorithm 7, $\widehat{Q}^{\,\theta}(s_{H_1-1},a_{H_1-1})$ is computed as

$$\widehat{Q}^{\,\theta}(s_{H_1-1},a_{H_1-1})=\sum_{t=H_1-1}^{H_1+H_2-2}\gamma^{\alpha(t-(H_1-1))}r(s_t,a_t), \tag{40}$$

with $H_2-1\sim\mathrm{Geom}(1-\gamma^{1-\alpha})$.

Following the same derivation for (35), when taking expectation with respect to the randomness of sampling τ from (13) in Algorithm 7, we have

$$\begin{aligned}
&\mathbb{E}\left[\|g(\tau\mid\theta)\|^2\right]\\
&=\mathbb{E}_{s_{H_1-1},a_{H_1-1}}\left[\frac{\left\|\nabla\log\pi^\theta(a_{H_1-1}\mid s_{H_1-1})\right\|^2}{(1-\gamma)^2}\mathbb{E}\left[\widehat{Q}^{\,\theta}(s_{H_1-1},a_{H_1-1})^2\mid s_{H_1-1},a_{H_1-1}\right]\right].
\end{aligned}\tag{41}$$

Again, following the same derivation for (36), we rewrite $\widehat{Q}^{\,\theta}(s_{H_1-1},a_{H_1-1})$ as

$$\widehat{Q}^{\,\theta}(s_0,a_0)=\sum_{t=0}^{H_2-1}\gamma^{\alpha t}r(s_t,a_t), \tag{42}$$

with $(s_0, a_0) = (s_{H_1-1}, a_{H_1-1})$ and H_2 is the length of the trajectory for estimating $Q^\theta(s_{H_1-1}, a_{H_1-1})$.

From (41) and (42), we have

$$
\begin{aligned}
& \mathbb{E}\left[\widehat{Q}^\theta(s_0,a_0)^2 \mid s_0 = s_{H_1-1}, a_0 = a_{H_1-1}\right] \\
\overset{(42)}{=} & \mathbb{E}\left[\left(\sum_{t=0}^{H_2-1}\gamma^{\alpha t} r(s_t,a_t)\right)^2 \mid s_0 = s_{H_1-1}, a_0 = a_{H_1-1}\right] \\
= & \sum_{k=0}^{\infty}\Pr(H_2-1=k)\mathbb{E}\left[\left(\sum_{t=0}^{H_2-1}\gamma^{\alpha t} r(s_t,a_t)\right)^2 \mid H_2-1=k, s_0 = s_{H_1-1}, a_0 = a_{H_1-1}\right] \\
= & (1-\gamma^{1-\alpha})\sum_{k=0}^{\infty}\gamma^{(1-\alpha)k}\mathbb{E}\left[\left(\sum_{t=0}^{k}\gamma^{\alpha t} r(s_t,a_t)\right)^2 \mid s_0 = s_{H_1-1}, a_0 = a_{H_1-1}\right] \\
\leq & r_{\max}^2(1-\gamma^{1-\alpha})\sum_{k=0}^{\infty}\gamma^{(1-\alpha)k}\left(\sum_{t=0}^{k}\gamma^{\alpha t}\right)^2 \\
= & \frac{r_{\max}^2(1-\gamma^{1-\alpha})}{(1-\gamma^\alpha)^2}\sum_{k=0}^{\infty}\gamma^{(1-\alpha)k}(1-\gamma^{\alpha(k+1)})^2 \\
= & \frac{r_{\max}^2(1-\gamma^{1-\alpha})}{(1-\gamma^\alpha)^2}\sum_{k=0}^{\infty}\gamma^{(1-\alpha)k}(1-2\gamma^{\alpha(k+1)}+\gamma^{2\alpha(k+1)}) \\
= & \frac{r_{\max}^2(1-\gamma^{1-\alpha})}{(1-\gamma^\alpha)^2}\sum_{k=0}^{\infty}\gamma^{(1-\alpha)k}-2\gamma^{k+\alpha}+\gamma^{(1+\alpha)k+2\alpha} \\
= & \frac{r_{\max}^2(1-\gamma^{1-\alpha})}{(1-\gamma^\alpha)^2}\left(\frac{1}{1-\gamma^{1-\alpha}}-\frac{2\gamma^\alpha}{1-\gamma}+\frac{\gamma^{2\alpha}}{1-\gamma^{1+\alpha}}\right) \\
\overset{(31)}{=} & r_{\max}^2\underbrace{\left(\frac{(1-\gamma)-2(\gamma^\alpha-\gamma)}{(1-\gamma)(1-\gamma^\alpha)^2}+\frac{\gamma^\alpha(\gamma^\alpha-\gamma)}{(1-\gamma^\alpha)^2(1-\gamma^{1+\alpha})}\right)}_{=f_2(\gamma^\alpha)} = r_{\max}^2 f_2(\gamma^\alpha),
\end{aligned}
\tag{43}
$$

where the inequality is obtained as $|r(s_t,a_t)| \in r_{\max}$ for all $(s_t,a_t) \in \mathcal{S}\times\mathcal{A}$, and for the last line, from (31) we have, for $x \in (\gamma, 1)$,

$$
f_2(x) = \frac{(1-\gamma)-2(x-\gamma)}{(1-\gamma)(1-x)^2}+\frac{x(x-\gamma)}{(1-x)^2(1-\gamma x)}.
$$

Combining (41) and (43) and using (18), we have

$$
\mathbb{E}\left[\|g(\tau \mid \theta)\|^2\right] \overset{(41),\,(43),\,(18)}{\leq} \frac{G^2 r_{\max}^2 f_2(\gamma^\alpha)}{(1-\gamma)^2}. \tag{44}
$$

The above together with (32) imply that the expected smoothness assumption holds for the batch version of Q-PGT with

$$
\mathbb{E}\left[\left\|\widehat{G}^{\alpha-\mathrm{QPGT}}(\theta)\right\|^2\right] \overset{(32),\,(44)}{\leq} \left(1-\frac{1}{m}\right)\|G(\theta)\|^2+\frac{G^2 r_{\max}^2 f_2(\gamma^\alpha)}{m(1-\gamma)^2}.
$$

Part V: RPG, which is α-QPGT in (13) with Algorithm 7 and $\alpha = \frac{1}{2}$.

In particular, when $\alpha = \frac{1}{2}$, α-QPGT recovers RPG. To compute $f_2(\sqrt{\gamma})$ in (43), we have

$$\begin{aligned}
f_2(\sqrt{\gamma}) &= \frac{(1-\gamma) - 2(\sqrt{\gamma}-\gamma)}{(1-\gamma)(1-\sqrt{\gamma})^2} + \frac{\sqrt{\gamma}(\sqrt{\gamma}-\gamma)}{(1-\sqrt{\gamma})^2(1-\gamma\sqrt{\gamma})} \\
&= \frac{(1-\gamma) - (\sqrt{\gamma}-\gamma)}{(1-\gamma)(1-\sqrt{\gamma})^2} - \left(\frac{(\sqrt{\gamma}-\gamma)}{(1-\gamma)(1-\sqrt{\gamma})^2} - \frac{\sqrt{\gamma}(\sqrt{\gamma}-\gamma)}{(1-\sqrt{\gamma})^2(1-\gamma\sqrt{\gamma})}\right) \\
&= \frac{1}{(1-\gamma)(1-\sqrt{\gamma})} - \left(\frac{\sqrt{\gamma}}{(1-\gamma)(1-\sqrt{\gamma})} - \frac{\gamma}{(1-\sqrt{\gamma})(1-\gamma\sqrt{\gamma})}\right) \\
&= \frac{1}{(1-\gamma)(1-\sqrt{\gamma})} - \frac{\sqrt{\gamma}-\gamma}{(1-\gamma)(1-\sqrt{\gamma})(1-\gamma\sqrt{\gamma})} \\
&= \frac{1}{(1-\gamma)}\left(\frac{1}{1-\sqrt{\gamma}} - \frac{\sqrt{\gamma}}{1-\gamma\sqrt{\gamma}}\right) \\
&= \frac{1+\gamma}{(1-\gamma)(1-\gamma\sqrt{\gamma})}.
\end{aligned}$$

So, in this case we have

$$\nu = \frac{2G^2 r_{\max}^2}{(1-\gamma)^3(1-\gamma\sqrt{\gamma})},$$

and

$$\mathbb{E}\left[\left\|\widehat{G}^{\frac{1}{2}-\mathrm{QPGT}}(\theta)\right\|^2\right] \quad \leq \quad \left(1-\frac{1}{m}\right)\|G(\theta)\|^2 + \frac{2G^2 r_{\max}^2}{m(1-\gamma)^3(1-\gamma\sqrt{\gamma})}.$$

□

Remark 9. Notice that for α-QPGT with $\alpha = \frac{1}{2}$ (i.e., RPG [61]), from (29), our $\nu = \frac{2G^2 r_{\max}^2}{(1-\gamma)^3(1-\gamma\sqrt{\gamma})}$ improves the result ν of [61, Theorem 3.4], which is $\frac{G^2 r_{\max}^2}{(1-\gamma)^2(1-\sqrt{\gamma})^2}$, and is obtained with more restrictive assumptions (Assumption 5 without expectation). Thus, the improvement is by a factor of

$$\begin{aligned}
\frac{G^2 r_{\max}^2}{(1-\gamma)^2(1-\sqrt{\gamma})^2} \quad / \quad \frac{2G^2 r_{\max}^2}{(1-\gamma)^3(1-\gamma\sqrt{\gamma})} &= \frac{(1-\gamma)(1-\gamma\sqrt{\gamma})}{2(1-\sqrt{\gamma})^2} \\
&= \frac{(1-\sqrt{\gamma})(1+\sqrt{\gamma})(1-\sqrt{\gamma}+\sqrt{\gamma}-\gamma\sqrt{\gamma})}{2(1-\sqrt{\gamma})^2} \\
&= \frac{(1+\sqrt{\gamma})((1-\sqrt{\gamma})+\sqrt{\gamma}(1-\gamma))}{2(1-\sqrt{\gamma})} \\
&= \frac{(1+\sqrt{\gamma})((1-\sqrt{\gamma})+\sqrt{\gamma}(1-\sqrt{\gamma})(1+\sqrt{\gamma}))}{2(1-\sqrt{\gamma})} \\
&= \frac{(1+\sqrt{\gamma})(1-\sqrt{\gamma})(1+\sqrt{\gamma}(1+\sqrt{\gamma}))}{2(1-\sqrt{\gamma})} \\
&= \frac{(1+\sqrt{\gamma})(1+\gamma+\sqrt{\gamma})}{2} \\
&= \frac{1+2\gamma+2\sqrt{\gamma}+\gamma\sqrt{\gamma}}{2} \approx 3, \quad \text{when } \gamma \longrightarrow 1.
\end{aligned}$$

See also Fig. 6 in Remark 10 for the variance analysis of α-QPGT to have its interpretation.

G.2 Proof of Corollary 2

Proof. From Lemma 3, we know that J is L-smooth. Consider policy gradient with a batch sampling of size m. From Theorem 6, we have Assumption 4 holds with $A = 0$, $B = 1 - \frac{1}{m}$ and $C = \nu/m$. By Proposition 1, plugging $A = 0$, $B = 1 - \frac{1}{m}$ and $C = \nu/m$ in (16) yields the corollary's claim with stepsize $\eta \in \left(0, \frac{2}{L\left(1-\frac{1}{m}\right)}\right)$. □

G.3 Proof of Corollary 3

Proof. Consider α-UPG with stepsize $\eta \in \left(0, \frac{1}{L\left(1-\frac{1}{m}\right)}\right)$ and a batch sampling of size m. We have

$$\begin{aligned}\mathbb{E}\left[\left\|\nabla J(\theta_U)\right\|^2\right] &\overset{(20)}{\leq} \frac{2(J^\star - J^{(0)})}{\eta K\left(2 - L\eta\left(1-\frac{1}{m}\right)\right)} + \frac{L\nu\eta}{m\left(2 - L\eta\left(1-\frac{1}{m}\right)\right)} \\ &\leq \frac{2(J^\star - J^{(0)})}{\eta K} + \frac{L\nu\eta}{m},\end{aligned}$$

where the second inequality is obtained by $\frac{1}{2-L\eta\left(1-\frac{1}{m}\right)} \leq 1$ with $\eta \in \left(0, \frac{1}{L\left(1-\frac{1}{m}\right)}\right)$.

To get $\mathbb{E}\left[\left\|\nabla J(\theta_U)\right\|^2\right] \leq \epsilon^2$, it suffices to have

$$\epsilon^2 \geq \frac{2(J^\star - J^{(0)})}{\eta K} + \frac{L\nu\eta}{m}. \tag{45}$$

To make the right hand side of (45) smaller than ϵ^2, we require

$$\frac{L\nu\eta}{m} \leq \frac{\epsilon^2}{2} \iff \eta \leq \frac{\epsilon^2 m}{2L\nu}. \tag{46}$$

Similarly, for the first term of the right hand side of (45), we require

$$\frac{2(J^\star - J^{(0)})}{\eta K} \leq \frac{\epsilon^2}{2} \iff \frac{4(J^\star - J^{(0)})}{\epsilon^2 K} \leq \eta. \tag{47}$$

Combining the above two inequalities gives

$$\frac{4(J^\star - J^{(0)})}{\epsilon^2 K} \leq \eta \leq \frac{\epsilon^2 m}{2L\nu}. \tag{48}$$

This implies

$$Km \geq \frac{8L\nu(J^\star - J^{(0)})}{\epsilon^4}. \tag{49}$$

The condition on the stepsize $\eta \in \left(0, \frac{1}{L\left(1-\frac{1}{m}\right)}\right)$ requires that the batch size satisfies

$$\frac{\epsilon^2 m}{2L\nu} \leq \frac{1}{L\left(1-\frac{1}{m}\right)} \implies m \leq \frac{2\nu}{\epsilon^2}.$$

To conclude, it suffices to choose the stepsize $\eta = \frac{4(J^\star - J^{(0)})}{\epsilon^2 K} = \frac{\epsilon^2 m}{2L\nu}$, a batch size m between 1 and $\frac{2\nu}{\epsilon^2}$, and the number of iterations $K = \frac{8(J^\star - J^{(0)})L\nu}{m\epsilon^4}$, so that the inequalities (46), (47), (48) and (49) hold, which guarantee $\mathbb{E}\left[\left\|\nabla J(\theta_U)\right\|^2\right] \leq \epsilon^2$.

Thus, the total expected sample complexity is

$$Km \times \mathbb{E}[H] = \frac{8L\nu(J^\star - J^{(0)})\mathbb{E}[H]}{\epsilon^4} = \mathcal{O}(\epsilon^{-4}), \tag{50}$$

where $\mathbb{E}[H] = \mathcal{O}(1/(1-\gamma))$ is obtained from Lemma 1 for UGPOMDP and Q-PGT, and $\mathbb{E}[H] = \mathcal{O}(1/(1-\sqrt{\gamma}))$ is obtained from Lemma 2 for α-QPGT with $\alpha = \frac{1}{2}$. Indeed, for UGPOMDP, $\mathbb{E}[H] = 1/(1-\gamma)$ is directly obtained from Lemma 1; for Q-PGT in Algorithm 4, $\mathbb{E}[H] = 2/(1-\gamma)$, as $H = H_1 + H_2$ with $\mathbb{E}[H_1] = \mathbb{E}[H_2] = 1/(1-\gamma)$, applied Lemma 1 twice; similarly, for α-QPGT with $\alpha = \frac{1}{2}$ in Algorithm 7, $\mathbb{E}[H] = 2/(1-\sqrt{\gamma})$, as $H = H_1 + H_2$ with $\mathbb{E}[H_1] = \mathbb{E}[H_2] = 1/(1-\sqrt{\gamma})$, applied Lemma 2 twice.

More precisely, from Lemma 3, $L = \frac{r_{\max}}{(1-\gamma)^2}(G^2+F)$. When using UGPOMDP gradient estimator (11), from Theorem 6, $\nu = \frac{3G^2 r_{\max}^2}{(1-\gamma)^3}$. Thus, when γ is close to 1, the sample complexity of UGPOMDP is

$$\frac{24(J^\star - J^{(0)})G^2 r_{\max}^3 (G^2+F)}{(1-\gamma)^6 \epsilon^4} = \mathcal{O}\left((1-\gamma)^{-6}\epsilon^{-4}\right). \tag{51}$$

In this case, we can choose the batch size $m \in \left[1; \frac{2\nu}{\epsilon^2}\right]$, i.e., from 1 to $\mathcal{O}\left((1-\gamma)^{-3}\epsilon^{-2}\right)$ and the constant stepsize $\eta = \frac{\epsilon^2 m}{2L\nu}$ varies from $\mathcal{O}\left((1-\gamma)^5\epsilon^2\right)$ to $\mathcal{O}\left((1-\gamma)^2\right)$ accordingly.

When using Q-PGT gradient estimator (9), from Theorem 6, $\nu = \frac{2G^2 r_{\max}^2}{(1-\gamma)^4}$. Thus, when γ is close to 1, the sample complexity is

$$\frac{16(J^\star - J^{(0)})G^2 r_{\max}^3 (G^2+F)}{(1-\gamma)^7 \epsilon^4} = \mathcal{O}\left((1-\gamma)^{-7}\epsilon^{-4}\right). \tag{52}$$

In this case, we can choose the batch size $m \in \left[1; \frac{2\nu}{\epsilon^2}\right]$, i.e., from 1 to $\mathcal{O}\left((1-\gamma)^{-4}\epsilon^{-2}\right)$ and the constant stepsize $\eta = \frac{\epsilon^2 m}{2L\nu}$ is proportional to the batch size m from $\mathcal{O}\left((1-\gamma)^6\epsilon^2\right)$ to $\mathcal{O}\left((1-\gamma)^2\right)$ accordingly.

Lastly, when using α-QPGT with $\alpha = \frac{1}{2}$ gradient estimator (13), from Theorem 6, $\nu = \frac{2G^2 r_{\max}^2}{(1-\gamma)^3(1-\gamma\sqrt{\gamma})}$. Thus, when γ is close to 1, the sample complexity is

$$\frac{16(J^\star - J^{(0)})G^2 r_{\max}^3 (G^2+F)}{(1-\gamma)^5(1-\sqrt{\gamma})(1-\gamma\sqrt{\gamma})\epsilon^4} = \mathcal{O}\left((1-\gamma)^{-5}(1-\sqrt{\gamma})^{-1}(1-\gamma\sqrt{\gamma})^{-1}\epsilon^{-4}\right). \tag{53}$$

□

Remark 10. Similar to Figs. 1 and 2, we evaluate the same heap maps in Figs. 6 and 7 for α-QPGT's variance and sample complexity. It is coherent that Q-PGT (i.e., α-QPGT with $\alpha = 0$) has bigger variance than RPG (i.e., α-QPGT with $\alpha = 0.5$), as from Fig. 6 we observe that increasing α induces decreasing variance for ν. However, Q-PGT has lower sample complexity than RPG, independent to the choice of γ, as from Fig. 7 we observe that increasing α will also increase the sample complexity.

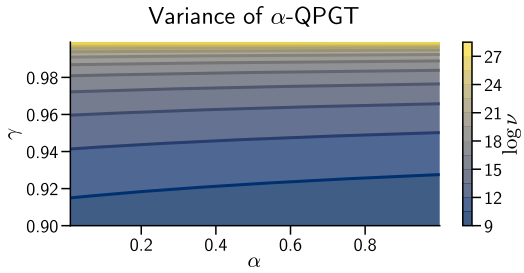

Figure 6: **Variance of α-QPGT.**

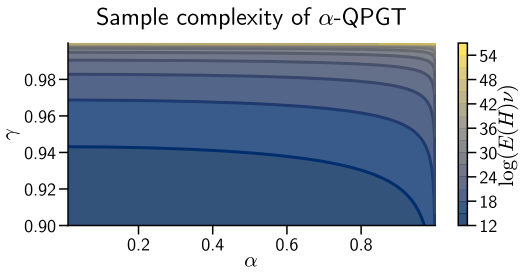

Figure 7: **Sample complexity of α-QPGT.**

Like in Remark 9, from (52) and (53) in Corollary 3, the improvement of the sample complexity is by a factor of

$$\begin{aligned}
&\frac{16(J^\star - J^{(0)})G^2 r_{\max}^3 (G^2+F)}{(1-\gamma)^5(1-\sqrt{\gamma})(1-\gamma\sqrt{\gamma})\epsilon^4} \;/\; \frac{16(J^\star - J^{(0)})G^2 r_{\max}^3 (G^2+F)}{(1-\gamma)^7\epsilon^4} \\
&= \frac{(1-\gamma)^2}{(1-\sqrt{\gamma})(1-\gamma\sqrt{\gamma})} = \frac{((1-\sqrt{\gamma})(1+\sqrt{\gamma}))^2}{(1-\sqrt{\gamma})(1-\sqrt{\gamma})(1+\sqrt{\gamma}+\gamma)} = \frac{(1+\sqrt{\gamma})^2}{1+\sqrt{\gamma}+\gamma} \\
&= \frac{1+2\sqrt{\gamma}+\gamma}{1+\sqrt{\gamma}+\gamma} = 1 + \frac{\sqrt{\gamma}}{1+\sqrt{\gamma}+\gamma} \approx \frac{4}{3}, \quad \text{when } \gamma \longrightarrow 1.
\end{aligned}$$

H Experimental Details of Section 5 and Additional Experiments

H.1 Experimental details

Table 1 provides the details of the hyperparameters choices of the experiments for each environment presented in § 5. In particular, each method has the same train batch size of 64 and all the algorithms are evaluated every 25 episodes with a test batch size of 128. The discount factor γ is set to 0.99 for Cart Pole, Lunar Lander, and Acrobot, and 0.95 for Pendulum, and the size of the horizon for the truncation is $H = (1-\gamma)^{-1}$.

Furthermore, we share some other experimental details in the following.

Gymnasium Vector. To ensure a robust evaluation of PG and UPG algorithms, it is essential to conduct hundreds of iterative training experiments. To expedite this process, we opted to utilize Gymnasium Vector capabilities [51]. According to the documentation, "Vector environments can offer a linear speed-up in the steps taken per second by sampling multiple sub-environments simultaneously. To prevent terminated environments from waiting until all sub-environments have terminated or been truncated, the vector environments automatically reset sub-environments after termination or truncation."

While the simultaneous simulation of several environments is a clear advantage, the auto-reset of sub-environments after termination can complicate the derivation of gradient estimates. Essentially, our objective is to capture only the interaction sequence, depicting a single trajectory from the initial state to terminal/truncation, to calculate estimates based on equations such as equation (15). Therefore, collecting multiple trajectories (even the last one, which may be truncated) in a single sequence would not be a suitable option. Another crucial aspect is to maintain the correct batch size across iterations. Hence, we devised a specific procedure for simulating and parsing trajectories:

- Set the number of sub-environments equal to the batch size.
- Simulate the vector environment until each sub-environment terminates once.
- For each sub-environment, monitor action probabilities (and gradients) and rewards only until the first termination.

This approach ensures the proper implementation of batch optimization.

Infinite horizon setting adaptation. When implementing alpha-UGPOMDP or alpha-QPGT, it is required to sample $H_1 - 1$ and $H_2 - 1$ from a geometric distribution. These values determine the trajectory slices used to calculate gradient estimation. It is important to note that actual environments do not perfectly match the infinite horizon trajectory setup. The difference is that infinite horizon trajectories do not have terminal states; therefore, rewards are defined for every possible state, action pair, and timestep. In contrast, Gymnasium environments have termination conditions for every environment, so we need to adapt our algorithms to this limitation. Since it is not possible to sample actions after the terminal state, we decided to stop the estimation procedure once termination is reached.

Let's call the trajectory length obtained during the simulation as H_s. In practice, there could be several scenarios:

1. $H_s < H - 2$, relevant to alpha-UGPOMDP, especially, when trajectories tend to be shorter. For this case, we stop the estimation process once reaching the terminal state, equivalently, $H - 2 = min(H_s, H - 2)$.
2. $H_s <= H_1 - 2$, relevant for alpha-QPGT. In this case the estimation process stops once reaching the terminal state and $\widehat{Q}^{\theta}(s_{H_s}, a_{H_s}) = r(s_{H_s}, a_{H_s})$.
3. $H_1 - 2 < H_S <= H_1 + H_2 - 3$, relevant for alpha-QPGT. For this case the estimation process stops once reaching the terminal state and the estimate of $\widehat{Q}^{\theta}$ is calculated based on steps $H_1 - 2, ..., H_S$.

Those adaptations are not ideal and can lead to gradient estimate corruption due to slice truncation. Depending on the properties of the environment, there might be better ways to adapt the algorithms. This issue requires additional research.

Table 1: Hyperparameters for each environment

	Acrobot-v1	CartPole-v1	LunarLander-v2	MountainCarContinuous-v0	Pendulum-v1
eval_batch_size	128	128	128	128	128
gamma	0.99	0.99	0.99	0.99	0.95
horizon	100	100	100	100	20
lr	0.001	0.001	0.001	0.001	0.001
max_eval_steps	500	200	1000	999	200
n_iterations	10001	501	2001	10001	10001
train_batch_size	64	64	64	64	64

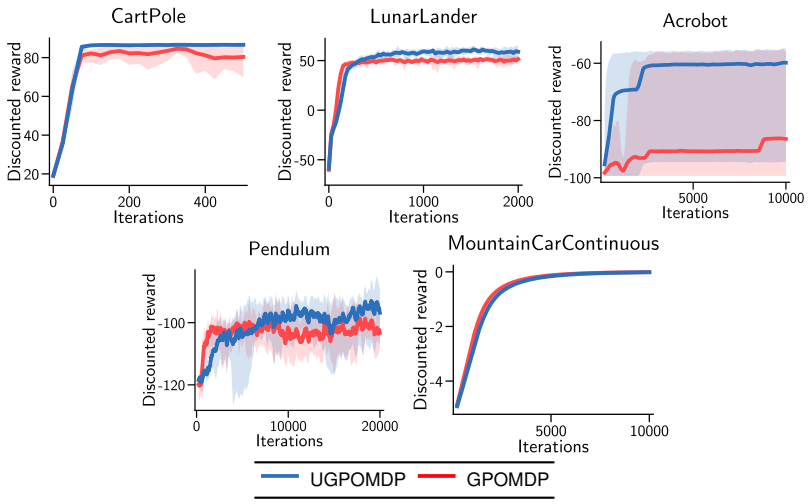

Figure 8: **Comparison between biased and unbiased policy gradient methods**. We compare the evolution of discounted rewards in GPOMDP (biased) and UGPOMDP (unbiased) on five standard Gym environments.

Evaluation. Each method uses a training batch size of 64, and all algorithms are evaluated every 25 episodes with a test batch size of 128. The discount factor gamma is set to 0.99 for CartPole, Lunar Lander and Acrobot. For Pendulum, it is set to 0.95.

The truncation horizon, used for biased algorithms during training, is $H = \frac{1}{1-\gamma}$. Each algorithm is evaluated using the same 10 randomly sampled seeds, and we report the discounted episodic reward. Evaluation trajectories are truncated to ensure the same maximum length for each algorithm, and therefore, the same objective function (discounted episodic reward). The horizons used for evaluation are as follows:

- CartPole-v1: 200
- LunarLander-v2: 1000
- Acrobot-v1: 500
- Pendulum-v1: 200
- MountainCarContinuous-v0: 999

H.2 Results compared to biased PG with double-horizons

Previously, we compared UGPOMDP and GPOMDP using the concept of the "effective horizon". The "effective horizon" means that for GPOMDP, training trajectories were truncated at $H = \frac{1}{1-\gamma}$, which is the expected trajectory length of UGPOMDP. The idea is to use the same amount of information on average for both algorithms to ensure a fair comparison. The similar amount of information consumed for training is also confirmed by Fig. 9, which shows performance in terms of environment interactions. It is important to note that UGPOMDP can still simulate trajectories longer than $\frac{1}{1-\gamma}$, thus benefiting from deeper environment exploration.

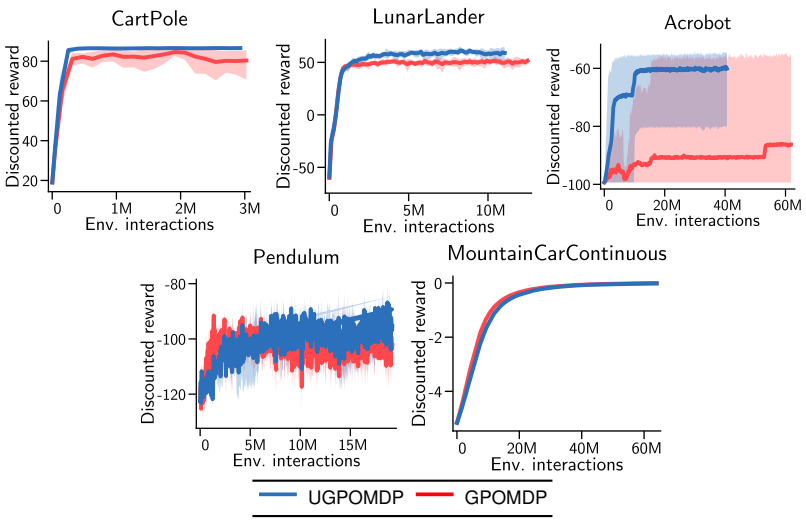

Figure 9: **Comparison between biased and unbiased policy gradient methods as a function of number of environment interactions**. We compare the evolution of discounted rewards in GPOMDP (biased) and UGPOMDP (unbiased) on five standard Gym environments.

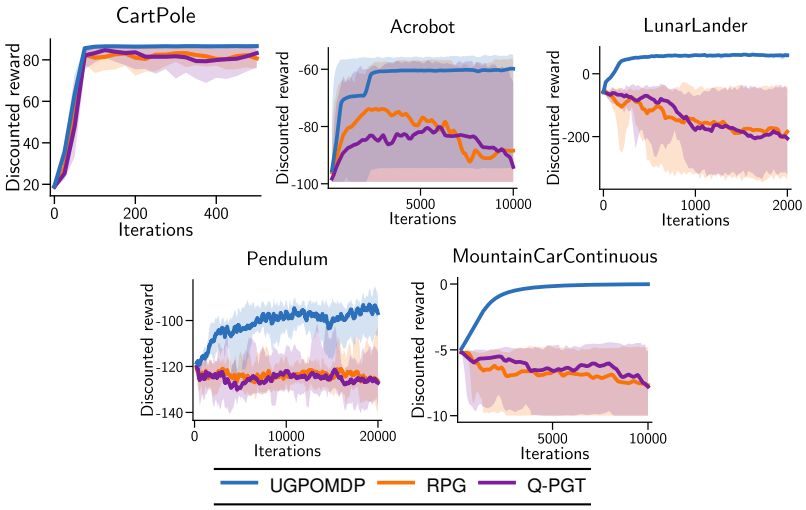

Figure 10: **Comparison between unbiased gradient methods: UGPOMDP, RPG and Q-PGT.** We compare the evolution of discounted rewards in UGPOMDP, RPG and Q-PGT on five standard Gym environments.

This is why it is interesting to compare UGPOMDP performance with GPOMDP trained with a larger horizon. Such a comparison is depicted in Fig. 11. With a horizon twice as large, GPOMDP shows a similar convergence speed in terms of the number of iterations and a similar discounted episodic reward but achieves a considerably better discounted episodic reward for Acrobot, and a slightly better discounted episodic reward for CartPole and LunarLander. The Pendulum environment cannot be strictly compared between figures due to different gamma values, but in Fig. 8, GPOMDP is generally worse than UGPOMDP, whereas in Fig. 11, it is competitive or even better. In the Acrobot environment, UGPOMDP still wins by a significant margin, but the maximum reward for GPOMDP is considerably higher compared to the "effective horizon" maximum reward.

Examining Figs. 8 to 12, it is clear that UGPOMDP shows better sample efficiency compared to GPOMDP for Acrobot, Pendulum, and MountainCarContinuous, and performs practically the same for CartPole and Lunar Lander (even when using a twice larger horizon on average). These experimental results align well with the theoretical expectations.

Table 2: Hyperparameters for each environment (double horizon).

env_name	Acrobot-v1	CartPole-v1	LunarLander-v2	MountainCarContinuous-v0	Pendulum-v1
eval_batch_size	128	128	128	128	128
gamma	0.99	0.99	0.99	0.99	0.99
horizon	200	200	200	200	200
lr	0.001	0.001	0.001	0.001	0.001
max_eval_steps	500	200	1000	999	200
n_iterations	10001	501	2001	10001	10001
train_batch_size	64	64	64	64	64

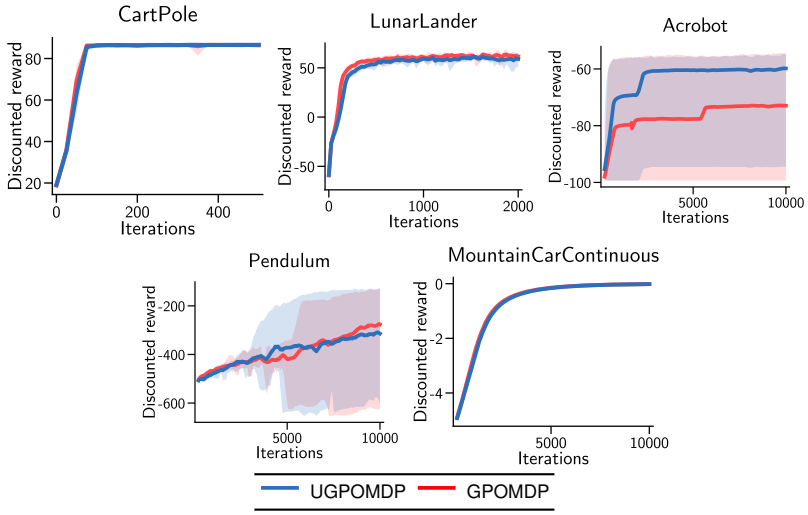

Figure 11: **Comparison between biased and unbiased policy gradient methods using double size of horizon for the biased one.** We compare the evolution of discounted rewards in GPOMDP (biased) and UGPOMDP (unbiased) on five standard Gym environments.

H.3 Additional RPG / Q-PGT evaluation

We evaluated RPG and Q-PGT using 3 new environments - LunarLander, Acrobot, MountainCarContinious, results described in Fig. 10. For all tested environments adaptaion described in H.1.2 was applied. Addressing Fig. 10 we can conclude that with applied adaption only CartPole suits theoretical setting well. Another environments does not work well, so experiments with other parameters, like gamma and thus sampled trajectories lengths should be continued.

I Auxiliary Lemmas

Lemma 5 (Lemma B.4 in Yuan et al. [58]). *Under Assumption 5, for all non negative integer t and any state-action pair $(s_t, a_t) \in \mathcal{S} \times \mathcal{A}$ at time t of a trajectory $\tau \sim p(\cdot \mid \theta)$ sampled under the parametrized policy π^θ, we have that*

$$\mathbb{E}_{\tau\sim p(\cdot|\theta)}\left[\left\|\nabla \log \pi^\theta(a_t \mid s_t)\right\|^2\right] \leq G^2, \tag{54}$$

$$\mathbb{E}_{\tau\sim p(\cdot|\theta)}\left[\left\|\nabla^2 \log \pi^\theta(a_t \mid s_t)\right\|\right] \leq F. \tag{55}$$

Lemma 6 (Lemma B.6 in Yuan et al. [58]). *For all non negative integers $0 \leq t$, and any state-action pairs $(s_h, a_h) \in \mathcal{S} \times \mathcal{A}$ at time $0 \leq h \leq t$ of the same trajectory $\tau \sim p(\cdot \mid \theta)$ sampled under the parametrized policy π^θ, we have*

$$\mathbb{E}_\tau\left[\left\|\sum_{h=0}^{t} \nabla \log \pi^\theta(a_h \mid s_h)\right\|^2\right] = \sum_{h=0}^{t} \mathbb{E}_\tau\left[\left\|\log \pi^\theta(a_h \mid s_h)\right\|^2\right]. \tag{56}$$

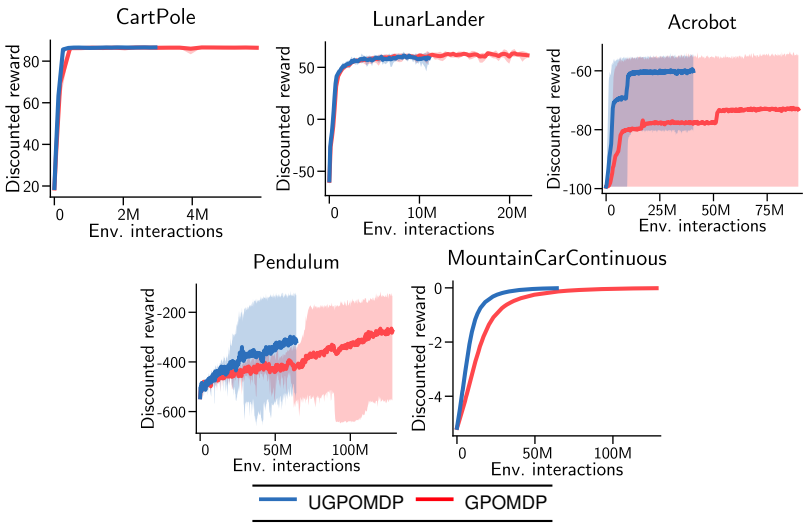

Figure 12: **Comparison between biased and unbiased policy gradient methods using double size of horizon for the biased one as a function of number of environment interactions.** We compare the evolution of discounted rewards in GPOMDP (biased) and UGPOMDP (unbiased) on five standard Gym environments.

Lemma 7. *For all $\gamma \in [0,1)$, we have that*

$$\sum_{k=0}^{K}(k+1)\gamma^k = \frac{1-\gamma^{K+1}}{(1-\gamma)^2} - \frac{(K+1)\gamma^{K+1}}{1-\gamma}, \tag{57}$$

$$\sum_{k=0}^{\infty}(k+1)\gamma^k = \frac{1}{(1-\gamma)^2}, \tag{58}$$

$$\sum_{k=0}^{\infty}(k+1)^2\gamma^k = \frac{2}{(1-\gamma)^3} - \frac{1}{(1-\gamma)^2} \leq \frac{2}{(1-\gamma)^3}, \tag{59}$$

$$\sum_{k=0}^{\infty}(k+1)^2(k+2)\gamma^k = \frac{6}{(1-\gamma)^4} - \frac{4}{(1-\gamma)^3} \leq \frac{6}{(1-\gamma)^4}. \tag{60}$$

Proof. Let

$$S_1 \overset{\text{def}}{=} \sum_{k=0}^{K}(k+1)\gamma^k.$$

We have

$$\gamma S_1 = \sum_{k=0}^{K}(k+1)\gamma^{k+1} = \sum_{k=1}^{K+1} k\gamma^k.$$

Subtracting of the above two equations gives

$$\begin{aligned}(1-\gamma)S_1 &= \sum_{k=0}^{K}(k+1)\gamma^k - \sum_{k=1}^{K+1} k\gamma^k = 1 + \sum_{k=1}^{K}\gamma^k - (K+1)\gamma^{K+1} \\ &= \sum_{k=0}^{K}\gamma^k - (K+1)\gamma^{K+1} = \frac{1-\gamma^{K+1}}{1-\gamma} - (K+1)\gamma^{K+1}.\end{aligned}$$

Finally, we obtain S_1 for (57) by dividing $1-\gamma$ on both hand side.

Then, from (57), letting $K \to \infty$ yields (58).

Now let

$$S_2 \overset{\text{def}}{=} \sum_{t=0}^{\infty}(t+1)^2\gamma^t.$$

We have

$$\gamma S_2 = \sum_{t=0}^{\infty}(t+1)^2\gamma^{t+1} = \sum_{t=1}^{\infty}t^2\gamma^t.$$

Thus, the subtraction of the above two equations gives

$$\begin{aligned}(1-\gamma)S_2 &= \sum_{t=0}^{\infty}(t+1)^2\gamma^t - \sum_{t=1}^{\infty}t^2\gamma^t = 1 + \sum_{t=1}^{\infty}((t+1)^2 - t^2)\gamma^t = \sum_{t=0}^{\infty}(2t+1)\gamma^t \\ &= 2\sum_{t=0}^{\infty}(t+1)\gamma^t - \sum_{t=0}^{\infty}\gamma^t \overset{(58)}{=} \frac{2}{(1-\gamma)^2} - \frac{1}{1-\gamma} \leq \frac{2}{(1-\gamma)^2}.\end{aligned}$$

Finally, we obtain S_2 for (59) by dividing $1-\gamma$ on both hand side.

Lastly, let

$$S_3 \overset{\text{def}}{=} \sum_{k=0}^{\infty}\gamma^k(k+1)^2(k+2).$$

We have

$$\gamma S_3 = \sum_{k=0}^{\infty}\gamma^{k+1}(k+1)^2(k+2) = \sum_{k=1}^{\infty}\gamma^k k^2(k+1).$$

Thus, the subtraction of the above two equations gives

$$\begin{aligned}(1-\gamma)S_3 &= \sum_{k=0}^{\infty}\gamma^k(k+1)^2(k+2) - \sum_{k=1}^{\infty}\gamma^k k^2(k+1) \\ &= 2 + \sum_{k=1}^{\infty}\left((k+1)^2(k+2) - k^2(k+1)\right)\gamma^k \\ &= 2 + \sum_{k=1}^{\infty}(k+1)(3k+2)\gamma^k \\ &= \sum_{k=0}^{\infty}(k+1)\left((3k+3)-1\right)\gamma^k \\ &= 3\sum_{k=0}^{\infty}(k+1)^2\gamma^k - \sum_{k=0}^{\infty}(k+1)\gamma^k \\ &\overset{(59),\,(58)}{=} \frac{6}{(1-\gamma)^3} - \frac{4}{(1-\gamma)^2}\end{aligned}$$

Finally, we obtain S_3 for (60) by dividing $1-\gamma$ on both hand side. □