A Temporal-Difference Approach to Policy Gradient Estimation

Samuele Tosatto[1] **Andrew Patterson**[1] **Martha White**[1 2] **A. Rupam Mahmood**[1 2]

Abstract

The policy gradient theorem (Sutton et al., 2000) prescribes the usage of a cumulative discounted state distribution under the target policy to approximate the gradient. Most algorithms based on this theorem, in practice, break this assumption, introducing a distribution shift that can cause the convergence to poor solutions. In this paper, we propose a new approach of reconstructing the policy gradient from the start state without requiring a particular sampling strategy. The policy gradient calculation in this form can be simplified in terms of a *gradient critic*, which can be recursively estimated due to a new Bellman equation of gradients. By using temporal-difference updates of the gradient critic from an off-policy data stream, we develop the first estimator that side-steps the distribution shift issue in a model-free way. We prove that, under certain realizability conditions, our estimator is unbiased regardless of the sampling strategy. We empirically show that our technique achieves a superior bias-variance trade-off and performance in presence of off-policy samples. The implementation of the experiments can be found at https://github.com/SamuelePolimi/temporal-difference-gradient.

1. Introduction

Policy gradient methods provide an elegant approach to learn a parameterized policy in reinforcement learning (Deisenroth et al., 2013). The policy gradient theorem (Sutton et al., 2000) provides a form for the gradient of the policy objective that can be sampled in a model-free way. This early work laid the foundation for practical methods, but as yet there is much more work to be done to provide effective approximations for the gradient. This is specially important in the off-policy setting, where we need the gradient under the current policy (the target policy) but the agent's experience is generated under a different policy (the behavior policy). Addressing this gap is critical for building sample-efficient methods that permit re-use of the agent's experience either from past policies as replay (Mnih et al., 2015; Lillicrap et al., 2016), offline datasets (Levine et al., 2020), or human demonstrations. The difficulty of constructing policy gradient approximators arises from the need to sample states from the discounted state-distribution that is induced by the target policy. Although such sampling can be achieved in an on-policy manner, this approach is rarely used in practice, as it requires samples to be used only once, causing high variance and sample inefficiency. Instead, most methods reuse data, introducing some bias but compensating it with better sample efficiency.

An alternative approach consists in correcting the state distribution. The most straightforward choice is to use importance sampling to reweight states, as if they had been sampled proportionally to the target policy (Shelton, 2001; Peshkin & Shelton, 2002). These methods are usually unbiased but affected by prohibitively large variance (Owen, 2013). Many recent papers aim to lower the variance of pure importance sampling correction. Liu et al. (2018) and Liu et al. (2019) introduce the concept of state-wise importance sampling. Imani et al. (2018) proposes to combine semi-gradient with an emphatic weighting. Notably, they come across the gradient Bellman equation in their derivation. AlgaeDICE (Nachum et al., 2019) incorporates a correction of the off-policy distribution by relying on the dual problem of a modified objective that incorporates an f-divergence regularization. But as yet more work is needed to make state-reweighting a practical choice.

The most common choice has been to simply omit any correction to the state distribution such as is done in OffPAC (Degris et al., 2012), DDPG (Lillicrap et al., 2016), A3C (Mnih et al., 2016), TD3 (Fujimoto et al., 2018) and SAC (Haarnoja et al., 2018). These methods are referred to as *semi-gradient* methods since the shift in the distribution can be seen as a result of the omission of a term in the gradient computation (Imani et al., 2018). Though most state-of-the-art policy gradient algorithms use this semi-gradient approach, there are well known counterexamples

[*] Equal contribution [1] Department of Computer Science, University of Alberta, Edmonton, Canada [2] CIFAR AI Chair, Alberta Machine Intelligence Institute (Amii). Correspondence to: Samuele Tosatto <tosatto@ualberta.ca>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

showing that the bias can result in poor solutions, in both on-policy (Nota & Thomas, 2020; Thomas, 2014) and off-policy settings (Imani et al., 2018; Liu et al., 2019). These counterexamples are not pathological and indicate issues that can arise under reasonable state aliasing. (Fujimoto et al., 2019) suggests that the effectiveness of aforementioned approaches can be hindered when the distribution shift is more pronounced.

In this work, we side-step the issue of the state weighting by pursuing an alternative form for the policy gradient. We propose learning a parametric representation of the cumulative discounted sequence of gradients generated from the target policy, which we call *the gradient critic*. The gradient critic can be learned from off-policy data using classic temporal-difference (TD) approaches. We will see that the gradient critic satisfies a Bellman equation (which we call *gradient Bellman equation*), allowing us to leverage the rich body of literature of value function estimators, including the ones for off-policy setting. The gradient critic can be queried on the starting states, allowing us to predict the policy gradient without constraining ourselves to the need of a particular state distribution (i.e., the target-policy state-distribution). We show that the gradient estimate is unbiased when the gradient function is realizable. To our knowledge, our method is the first to allow unbiased and model-free estimation of the policy gradient without using a state distribution reweighting[1].

2. The Issue of State-Reweighting

We consider a Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, p, \gamma, \mu_0)$, where $\mathcal{S}$ represents a finite set of states, $\mathcal{A}$ a finite set of actions[2], $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ a reward signal, $p(s'|s, a)$ a probability density of transitioning to state s' after the application of action a in state s, γ is the discount factor, and μ_0 is the distribution of the starting state. The parameterized policy π_θ, with parameters $\theta \in \mathbb{R}^{n_p}$, is a stochastic mapping, with density $\pi_\theta(a|s)$ over actions. We assume π_θ to be differentiable w.r.t. θ. We denote with S_t, A_t, R_t the random variables representing the state, action, and reward at time t. We denote a sequence of state, action, and reward with τ_π, when they are on-policy and with τ_β otherwise.

The objective in the episodic setting is to maximize the expected discounted return from the start states

$$J(\theta) = (1-\gamma)\mathop{\mathbb{E}}_{\tau_\pi}\left[\sum_{t=0}^{\infty}\gamma^t R_t\right] = (1-\gamma)\mathop{\mathbb{E}}_{\substack{S_0\sim\mu_0\\ A_0\sim\pi_\theta}}\left[Q^\pi(S_0, A_0)\right],$$

where the action-values $Q^\pi(S, A)$ are the expected return under the policy from a given state and action, defined recursively as for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$

$$Q^\pi(s, a) = r(s, a) + \gamma \sum_{s', a'} Q^\pi(s', a') p(s'|s, a)\pi_\theta(a'|s'). \quad (1)$$

Policy Gradient Theorem. One of the most important results in reinforcement learning is the policy gradient theorem (PGT), which allows us to estimate the policy gradient via samples:

$$\nabla_\theta J(\theta) = \mathop{\mathbb{E}}_{S\sim\mu_\gamma^\pi, A\sim\pi_\theta}\left[Q^{\pi_\theta}(S, A)\nabla_\theta \log \pi_\theta(A|S)\right]. \quad (2)$$

Note that this gradient has states sampled from the discounted state districution, i.e., $S \sim \mu_\gamma^\pi$, which is defined as follows. The state-distribution $\mu_t^\pi(s) = p_\pi(S_t = s)$ indicates the density or the probability of the state s being observed at time t when following π. The discounted state distribution is $\mu_\gamma^\pi(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mu_t^\pi(s)$.

Semi-Gradient. Most algorithms do not sample from the discounted state distribution μ_γ^π, as they do not perform proper discounting (Nota & Thomas, 2019) and they reuse past experience collected in the replay buffer. The semi-gradient estimate can be seen as

$$\nabla_\theta^{SG} J(\theta) = \mathop{\mathbb{E}}_{\substack{S\sim\mu^\beta\\ A\sim\beta}}\left[\frac{\pi_\theta(A|S)}{\beta(A|S)} Q^{\pi_\theta}(S, A)\nabla_\theta \log \pi_\theta(A|S)\right], \quad (3)$$

where β is a behavior policy, and μ_β its induced state-distribution. Notice that the importance sampling in Equation (3) only corrects the mistach in the action distribution but not the off-policy distribution μ_β. Examples of semi-gradient approaches are OffPAC (Degris et al., 2012), DDPG (Silver et al., 2014) and SAC (Haarnoja et al., 2018; Heess et al., 2015).

To avoid the semi-gradient problem, most approaches propose to perform a state-reweighting. The simplest version, proposed by (Shelton, 2001) and (Peshkin & Shelton, 2002), consists of multiplying all the importance sampling corrections along the trajectory, causing high variance. Recent work (Imani et al., 2018; Liu et al., 2018; 2019) aims to lower the variance, but still relies on forms of importance sampling corrections. A proper and practical way of state-reweighting remains to be one of the critical issues for effective policy gradient estimation.

3. Policy Gradient Using a Gradient Critic

In this section, we pursue another path to estimating the policy gradient, by introducing the notion of a *gradient critic*. This gradient critic is the discounted accumulation of gradients, and as we show later, can be estimated using standard temporal-difference methods. This approach avoids the need to reweighting state distribution or incorporate high-variance importance sampling ratios, without incurring the high bias of semi-gradient approaches.

[1] At the time of our submission, Ni et al. (2022) indipendently released on Arxiv a similar idea corroborating our findings.

[2] We provide a continuous state-action formulation in appendix.

To obtain our alternative gradient estimator, we use a different formulation of the policy gradient theorem

$$\nabla_\theta J(\theta) \propto \mathbb{E}_{\tau_\pi}\Big[\sum_{t=0}^{\infty}\gamma^t \mathbf{g}_t\Big], \tag{4}$$

where $\mathbf{g}(S, A) = Q^\pi(S, A)\nabla_\theta \log \pi_\theta(A|S)$, $\mathbf{g}_t = \mathbf{g}(S_t, A_t)$[3]. Equation (4) is equivalent to Equation (2), with a constant factor $1-\gamma$ omitted. This form, however, highlights that the policy gradient can be seen as the discounted cumulation of gradients induced by *on-policy trajectories*.

We derive this form here, and connect it to what we call the gradient critic $\mathbf{\Gamma}^\pi(s, a) \doteq \nabla_\theta Q^\pi(s, a)$. Let us return to the definition of the objective J and attempt to naively compute the gradient, using the chain rule,

$$\nabla_\theta J(\theta) \propto \sum_{s,a}\mu_0(s)\big(\pi_\theta(a|s)\nabla_\theta Q^{\pi_\theta}(s,a) + Q^{\pi_\theta}(s,a)\nabla_\theta \pi_\theta(a|s)\big)$$
$$= \sum_{s,a}\mu_0(s)\pi_\theta(a|s)[\mathbf{g}(s, a) + \mathbf{\Gamma}^\pi(s, a)] \tag{5}$$

We can derive a formula for $\nabla_\theta Q^\pi(s, a)$ by taking the derivative of the Bellman equation in Equation (1)

$$\nabla_\theta Q^\pi(s, a) = \gamma\sum_{s',a'}\Big(Q^\pi(s', a')\nabla_\theta \log \pi_\theta(a'|s') + \nabla_\theta Q^\pi(s', a')\Big)\pi_\theta(a'|s')p(s'|s, a).$$

By substituting $\nabla_\theta Q^\pi$ with $\mathbf{\Gamma}^\pi$ and the integral with an expectation, we obtain the following recursive form

$$\mathbf{\Gamma}^\pi(s,a) = \gamma\mathbb{E}[\mathbf{g}_{t+1} + \mathbf{\Gamma}^\pi(S_{t+1}, A_{t+1})|S_t = s, A_t = a]. \tag{6}$$

We can unroll this recursion, expanding $\mathbf{\Gamma}^\pi(S_{t+1}, A_{t+1})$,

$$\begin{aligned}\mathbf{\Gamma}^\pi(s,a) &= \gamma\mathbb{E}[\mathbf{g}_{t+1} + \mathbf{\Gamma}^\pi(S_{t+1}, A_{t+1})|S_t = s, A_t = a]\\ &= \gamma\mathbb{E}[\mathbf{g}_{t+1} + \gamma\mathbf{g}_{t+2} + \gamma\mathbf{\Gamma}^\pi(S_{t+2}, A_{t+2})|S_t = s, A_t = a]\\ &= \mathbb{E}\Big[\sum_{t=1}^{\infty}\gamma^t\mathbf{g}_t|S_t = s, A_t = a\Big].\end{aligned} \tag{7}$$

From Equations (5) and (7) we can verify that

$$\nabla_\theta J(\theta) \propto \mathop{\mathbb{E}}_{\substack{S\sim\mu_0\\ A\sim\pi_\theta}}\big[\mathbf{g}(S, A) + \mathbf{\Gamma}^\pi(S, A)\big]. \tag{8}$$

Note that the gradient critic can recover the policy gradient just by computing an expectation over the starting-state distribution. In other words, given an estimated gradient critic $\hat{\mathbf{\Gamma}}^\pi$, an estimated value critic $\hat{Q}^\pi$, and a start state s_0, the policy can be updated by using $a_0^\pi \sim \pi_\theta(\cdot|s_0)$ and

$$\theta \leftarrow \theta + \eta[\hat{\mathbf{g}}(s_0, a_0^\pi) + \hat{\mathbf{\Gamma}}^\pi(s_0, a_0^\pi)], \tag{9}$$

where $\hat{\mathbf{g}}(s_0, a_0^\pi) = \hat{Q}^\pi(s_0, a_0^\pi)\nabla_\theta \log \pi_\theta(a_0^\pi|s_0)$. This policy gradient estimator is naturally model-free and off-policy, does not require state distribution reweighting, and has less variance than the classic policy gradient, as it involves overall less stochasticity.

There is, of course, a big caveat: we require an estimate of this gradient critic. Poor estimates may introduce significant bias, overriding the benefits of this variance reduction. Even worse, we compound two approximations: an approximate value critic $\hat{Q}^\pi$ and gradient critic $\hat{\mathbf{\Gamma}}^\pi$. Remarkably, we find that we can actually obtain an unbiased gradient estimate, under linear function approximation with realizability, using a (batch) TD approach for learning both the value critic and gradient critic. We prove this later, in Theorem 2, after introducing the gradient critic estimation approaches. This theorem is particularly surprising because semi-gradient approaches remain biased, even with the knowledge of a perfect critic (Imani et al., 2018; Liu et al., 2019).

Aside the independent work of Ni et al. (2022), to the best of our knowledge, this is the first unbiased policy gradient approach estimator, with function approximation, that does not rely on state distribution reweighting. Notably, Tosatto et al. (2020; 2021) derived a similar approach to Equation (9) based on nonparametric statistics. Their method, however, do not scale with samples and requires infinitesimal bandwidth of the kernels to ensure unbiasedness.

In practice, of course, we may not have realizability and we need to understand when this approach will succeed and when it will fail. In the remainder of this paper, we investigate the properties of this approach, particularly focusing on different estimation approaches for the gradient critic and assessing those approaches empirically.

4. Estimating the Gradient Critic

In this section, we discuss the gradient Bellman equation and how we can use it to estimate the gradient critic $\mathbf{\Gamma}^\pi$. Notice that $\mathbf{\Gamma}^\pi(s, a) = \nabla_\theta Q^\pi(s, a)$ represents the differentiation of the state-action value with respect to the policy's parameters, is different from reparameterization gradient, and cannot, in general, be found in closed form or via automatic differentiation. Instead, we leverage the gradient Bellman equation and explain how to use the a TD algorithm to estimate this gradient critic. The basic idea is that the gradient estimator can be used with Equation (9) to update the policy gradient. Later in Section 6, we outline an n-step estimator that is robust to the bias of the gradient critic.

4.1. The Gradient Bellman Equation

The *gradient Bellman equation* was already shown in Equation (6), though we had not yet given it a name. The equation is $\mathbf{\Gamma}^\pi(s, a) = \gamma\mathbb{E}[\mathbf{g}_{t+1} + \mathbf{\Gamma}^\pi(S_{t+1}, A_{t+1})|S_t = s, A_t = a]$.

[3] As discussed in Appendix B, the likelihood-ratio gradient (LR) in $\mathbf{g}$ can, in principle, be replaced with reparametrization gradient (RP) (similar to SAC (Haarnoja et al., 2018)), compositions of LR and RP (Lan et al., 2022), or others (Carvalho et al., 2021).

For the i-th element g_i of $\mathbf{g}$, we have that

$$\Gamma_i^\pi(s,a) = \sum_{s',a'} (g_i(s',a') + \gamma\Gamma_i^\pi(s',a')\pi(a'|s'))p(s'|s,a),$$

which is a Bellman equation for a scalar $\Gamma_i^\pi(s,a)$. Therefore, Equation (6) is a Bellman equation for the vector $\mathbf{\Gamma}^\pi(s,a)$, composed of this set of independent Bellman equations. That is why we call (6) the *gradient Bellman equation*.

Bellman equations are well studied, giving us broad literature about approximation techniques and theoretical results. For example, $\mathbf{\Gamma}^\pi(s,a)$ can be estimated using bootstrapping approaches, like temporal-difference learning. One key subtlety is that the term $g_i(s,a)$ involves Q^π, which also needs to be estimated. Fortunately, we already estimate this term, the standard value critic, in actor-critic methods.

4.2. An Online Estimator using TDRC

The full algorithm involves 1) estimating a value critic, 2) using the value critic to estimate the gradient critic, and 3) using both the value and the gradient critics to estimate the policy gradient update. In this section, we explain an algorithm based on TD with regularized correction (Ghiassian et al., 2020), and detailed in Algorithm 1.

We can use TD to estimate both the standard action-value critic, as well as the gradient critic. We now have two temporal-difference errors: δ_t for the value critic, and the vector $\boldsymbol{\delta}_t^g$ for the gradient critic, which is the size of the number of policy parameters. Let $\boldsymbol{\omega}_t$ be the parameters for the value critic $\hat{Q}_t$ and $\mathbf{G}_t$ the parameters for the gradient critic $\hat{\mathbf{\Gamma}}$. The updates for the value critic, with step-size α_t, are the standard TD updates:

$$\begin{aligned}\delta_t &= R_{t+1} + \gamma\hat{Q}_t(S_{t+1},A_{t+1}) - \hat{Q}_t(S_t,A_t),\\ \boldsymbol{\omega}_{t+1} &= \boldsymbol{\omega}_t + \alpha_t\delta_t\nabla_{\boldsymbol{\omega}_t}\hat{Q}_t(S_t,A_t). \end{aligned} \tag{10}$$

Similarly, we can get vector-valued TD updates for the gradient critic as follows:

$$\begin{aligned}\mathbf{g}_t &= \hat{Q}_t(S_t,A_t^\pi)\nabla_\theta \log\pi_\theta(A_t^\pi|S_t),\\ \boldsymbol{\delta}_t^g &= \mathbf{g}_{t+1} + \gamma\hat{\mathbf{\Gamma}}(S_{t+1},A_{t+1}) - \hat{\mathbf{\Gamma}}(S_t,A_t),\\ \mathbf{G}_{t+1} &= \mathbf{G}_t + \alpha_t\boldsymbol{\delta}_t^g\nabla_{\mathbf{G}_t}\hat{\mathbf{\Gamma}}(S_t,A_t). \end{aligned} \tag{11}$$

Simple TD techniques (Sutton, 1988) are sample efficient but are not guaranteed to converge with off-policy data (Baird, 1995). Gradient TD methods (Sutton et al., 2008) and TD with gradient corrections (TDC) (Sutton et al., 2009) are guaranteed to converge under general conditions, however are often less sample efficient than TD (Ghiassian et al., 2020). A recent approach called TDRC (temporal-difference with regularized correction, Ghiassan et al., (2020)) proposes to mix regular TD with TDC, allowing convergence with off-policy samples without losing sample efficiency.

Once the gradient critic has been estimated, it can be used to update the policy parameters as in Equation (9). Algorithm 1 details a pseudocode of TDRC with policy improvement.

5. Unbiased Estimation Under Realizability

In this section we analyze the properties of the gradient critic, obtained with TD under linear function approximation. In particular, we show that the gradient critic given by the TD fixed-point solution in the realizable setting—the case where the features are sufficient to represent the value critic—gives an unbiased estimate of the policy gradient.

The TD fixed-point solution of the projected Bellman equation induced by (6) is as follows:

$$\begin{aligned}&\hat{\mathbf{\Gamma}}_{TDQ}^\pi(s,a) = \boldsymbol{\phi}^\intercal(s,a)\mathbf{G}_{TDQ} \text{ with } \mathbf{G}_{TDQ} = \mathbf{A}_\pi^{-1}\mathbf{B}_Q,\\ &\mathbf{A}_\pi = \mathbb{E}_\zeta\left[\boldsymbol{\phi}(S,A)\left(\boldsymbol{\phi}^\intercal(S,A) - \gamma\boldsymbol{\phi}^\intercal(S',A')\right)\right],\\ &\mathbf{B}_Q = \gamma\mathbb{E}_\zeta\left[\boldsymbol{\phi}(S,A)Q^\pi(S',A')\nabla_\theta\log\pi_\theta(A'|S')\right]. \end{aligned} \tag{12}$$

We first consider approximation error of the gradient critic assuming access to the true value critic Q^π.

Lemma 1 (Gradient Critic with Perfect Value Critic)**.** *Let us consider finite state and action sets. For an irreducible Markov chain induced jointly by the transition function p and the policy π_θ having steady distribution μ. Let ζ be a process where $S \sim \mu_\beta$, $A \sim \beta(\cdot|S)$, $S' \sim p(S'|S,A)$ and $A' \sim \pi_\theta(\cdot|S')$. If $p(S,A) = \mu_\beta(S)\beta(A|S)$ satisfies the inequality introduced by (Kolter, 2011), then*

$$\begin{aligned}&\|\hat{\mathbf{\Gamma}}_{TDQ}^\pi(s,a) - \nabla_\theta Q^\pi(s,a)\|_\zeta \leq\\ &\frac{1+\kappa\gamma}{1-\gamma}\min_{\mathbf{G}}\|\boldsymbol{\phi}^\intercal(s,a)\mathbf{G} - \nabla_\theta Q^\pi(s,a)\|_\zeta, \end{aligned} \tag{13}$$

with $\kappa = \max_{s,a} h(s,a)/\min_{s,a} h(s,a)$ where $h(s,a) = \sqrt{\mu(s)\pi_\theta(a|s)}/\sqrt{\mu_\beta(s)\beta(a|s)}$.

Next we consider the more realistic setting where we estimate the value critic. Again, because we use TD methods, we will use the TD-fixed point solution $\hat{Q}_{TD}^\pi(s,a) = \boldsymbol{\varphi}^\intercal(s,a)\boldsymbol{\omega}_{TD}$. Namely, we have $\boldsymbol{\omega}_{TD} = \mathbf{C}_\pi^{-1}\mathbf{b}$ with

$$\begin{aligned}&\mathbf{C}_\pi = \mathbb{E}_\zeta\left[\boldsymbol{\varphi}(S,A)\left(\boldsymbol{\varphi}^\intercal(S,A) - \gamma\boldsymbol{\varphi}^\intercal(S',A')\right)\right],\\ &\mathbf{b} = \mathbb{E}_\zeta[\boldsymbol{\varphi}(S,A)r(S,A)], \text{ and}\\ &\hat{\mathbf{\Gamma}}_{TD}^\pi(s,a) = \boldsymbol{\phi}^\intercal(s,a)\mathbf{G}_{TD}, \text{ with } \mathbf{G}_{TD} = \mathbf{A}_\pi^{-1}\mathbf{B}, \end{aligned} \tag{14}$$

where $\mathbf{B} = \gamma\mathbb{E}_\zeta[\phi(S,A)\hat{Q}_{TD}^\pi(S',A')\nabla_\theta\log\pi_\theta(A'|S')]$ is different from $\mathbf{B}_Q$ in (12) and $\mathbf{A}_\pi$ is as before.

The approximation error of the gradient critic when using the TD fixed-point solution for approximating both the critic can be bounded.

Theorem 1 (Error Analysis)**.** *Consider the assumption in Lemma 1 and Proposition 1 (Appendix A.2). The TD fixed-*

point solution $\hat{\mathbf{\Gamma}}^{\pi}_{TD}$ of the gradient function defined in Equation 14 satisfies

$$\|\hat{\mathbf{\Gamma}}^{\pi}_{TD}(s,a) - \nabla_\theta Q^\pi(s,a)\|_\zeta \leq$$
$$\frac{1+\gamma\kappa}{1-\gamma}\min_{\mathbf{G}}\|\boldsymbol{\phi}^\intercal(s,a)\mathbf{G} - \nabla_\theta Q^\pi(s,a)\|_\zeta +$$
$$\gamma n_p b\kappa\frac{(1+\gamma\kappa)^2}{(1-\gamma)^2}\min_{\boldsymbol{\omega}}\|\boldsymbol{\varphi}^\intercal(s,a)\boldsymbol{\omega} - Q^\pi(s,a)\|_\zeta,$$

with $b = |\max_{a,s,i}\partial/\partial\theta_i \log\pi(a|s)|$, n_p the number of the policy's parameters and κ as in Lemma 1.

The proof of Theorem 1 relies on the results in (Kolter, 2011) and on Lemma 2 (Appendix A.3).

Remark: Theorem 1 shows that the approximation error of the TD fixed-point solution of the gradient critics is bounded by the projection error of both critics: if the feature spaces of both the value and the gradient critic are good enough, both the projection errors goes to zero, ensuring an unbiased gradient estimate.

5.1. Shared Features

The gradient critic is inherently more complex than the value critic, since it predicts a high-dimensional quantity. It seems resonable that the feature space should also be larger (w.r.t. the value critic's one) to compensate this complexity. Surprisingly, in this linear setting, when the feature space of the value critic allows an unbiased value estimate, then it can be also reused by the gradient critic to obtain an unbiased gradient estimate.

Consider sharing the features between value and gradient critic, i.e., $\boldsymbol{\phi} = \boldsymbol{\varphi}$. Notice that, in this case, $\hat{Q}^\pi_{TD}(s,a) = \boldsymbol{\phi}^\intercal(s,a)\boldsymbol{\omega}_{TD}$, and $\boldsymbol{\omega}_{TD} = \mathbf{A}_\pi^{-1}\mathbf{b}$ where $\mathbf{b} = \mathbb{E}_\zeta[\boldsymbol{\phi}(S,A)r(S,A)]$.

In this case, it is possible to show that the TD fixed-point solution of the gradient critic is the gradient of TD fixed-point solution of the value critic.

Lemma 2. *When $\boldsymbol{\phi} = \boldsymbol{\varphi}$, the gradient approximation $\hat{\mathbf{\Gamma}}^\pi_{TD}(s,a)$ and the TD fixed-point critic $\hat{Q}^\pi_{TD}(s,a)$ satisfies*

$$\hat{\mathbf{\Gamma}}^\pi_{TD}(s,a) = \nabla_\theta\hat{Q}^\pi_{TD}(s,a) \qquad \forall s\in\mathcal{S},\forall a\in\mathcal{A}.$$

Proof. Consider $\nabla_\theta\hat{Q}^\pi_{TD}(s,a) = \boldsymbol{\phi}^\intercal(s,a)\nabla_\theta\boldsymbol{\omega}_{TD}$ and

$$\begin{aligned}\nabla_\theta\boldsymbol{\omega}_{TD} &= -\mathbf{A}_\pi^{-1}\left(\nabla_\theta\mathbf{A}_\pi\right)\mathbf{A}_\pi^{-1}\mathbf{b}\\ &= -\mathbf{A}_\pi^{-1}\left(\nabla_\theta\mathbf{A}_\pi\right)\boldsymbol{\omega}_{TD}\\ &= \gamma\mathbf{A}_\pi^{-1}\mathbb{E}_\zeta\left[\boldsymbol{\phi}(S,A)\boldsymbol{\phi}^\intercal(S',A')\boldsymbol{\omega}_{TD}\nabla_\theta\log\pi_\theta(A'|S')\right]\\ &= \gamma\mathbf{A}_\pi^{-1}\mathbb{E}_\zeta\left[\boldsymbol{\phi}(S,A)\hat{Q}^\pi_{TD}(S',A')\nabla_\theta\log\pi_\theta(A'|S')\right].\\ &= \mathbf{A}_\pi^{-1}\mathbf{B} = \mathbf{G}_{TD},\end{aligned}$$

implying $\nabla_\theta\hat{Q}^\pi_{TD}(s,a) = \boldsymbol{\phi}^\intercal(s,a)\mathbf{G}_{TD} = \hat{\mathbf{\Gamma}}^\pi_{TD}(s,a)$. □

This identity shows that the gradient predicted by the gradient critic is *consistent* with the gradient of the value critic. Usually, policy gradient algorithms do not guarantee this consistency. In fact, after the policy update, the policy might improve, but this improvement might be not representable by the critic, causing instability. This issue is well known in value iteration, called *delusional bias* (Lu et al., 2018). When features are shared, the converged gradient critic predicts the gradient of *the approximated value critic* w.r.t. the policy parameters, guaranteeing its improvement.

The benefit of sharing features is emphasized in Theorem 2, where the realizability of the value critic implies the realizability of the gradient critic.

Theorem 2 (Perfect Features)**.** *Let $\Phi \equiv \{\phi(s,a)|\forall s\in\mathcal{S}\wedge a\in\mathcal{A}\}$ and $\Phi' \equiv \{\phi(s,a) - \gamma\sum_{s',a'}\phi(s',a')\pi_\theta(a'|s)p(s'|s,a)|\forall s\in\mathcal{S}\wedge a\in\mathcal{A}\}$ be n_f-dimensional vector spaces (they both admit at least one basis of dimension d_f). Let μ_β be such that $\mathbf{A}_\pi$ is invertible. If we assume that for any policy parameter θ exists a parameter $\boldsymbol{\omega}_\pi$ such that*

$$\boldsymbol{\phi}^\intercal(s,a)\boldsymbol{\omega}_\pi = Q^\pi(s,a) \qquad \forall s\in\mathcal{S}\wedge a\in\mathcal{A},$$

then $$\hat{\mathbf{\Gamma}}^\pi_{TD}(s,a) = \nabla_\theta Q^\pi(s,a) \qquad \forall s\in\mathcal{S}\wedge a\in\mathcal{A}.$$

The proof can be found in Appendix A.4. This theorem further empasizes that the gradient critic can be unbiased. In particular, even though the gradient critic predicts a higher-dimensional vector compared to the value critic, it can still achieve a good approximation with the features used by the classic value critic.

6. Controlling the Bias and Variance

The proposed estimator fully relies on the gradient critic. We can instead reduce this reliance, by incorporating sampled gradient components and then bootstrapping.

It is important to notice that Equations 4 and 8 represent two extremes: the first is a full Monte-Carlo rollout, while the second uses full bootstrapping. By applying recursively the definition of the gradient critic, as we have done in the derivation of Equation 7, we can rewrite the policy gradient as a n-step estimator:

$$\nabla_\theta J(\theta) \propto \mathop{\mathbb{E}}_{\tau_\pi}\left[\sum_{t=0}^{n}\gamma^t\mathbf{g}_t + \gamma^n\mathbf{\Gamma}^\pi(S_n,A_n^\pi)\right].$$

The advantage of this approach is that we can either immediately bootstrap off of our estimate of the gradient ($n=0$), or we can wait one step to bootstrap, or we can wait n steps. This perspective highlights even more the role of $\mathbf{\Gamma}^\pi(s,a)$ as a critic function.

We can also express the n-step estimator under off-policy sampling. We can do so with the standard strategy of pathwise importance sampling corrections (Shelton, 2001). Under behavior policy β, it yields the following

$$\nabla_\theta J(\theta) \propto \mathbb{E}_{\tau_\beta}\left[\sum_{t=0}^{n}\gamma^t\rho_t\mathbf{g}_t + \gamma^n\rho_n\boldsymbol{\Gamma}^\pi(S_n, A_n^\pi)\right],$$

where $\rho_0 = 1$, $\rho_t = \prod_{i=0}^{t-1}\pi(A_i|S_i)/\beta(A_i|S_i)$, τ_β are off-policy trajectories, $\mathbf{g}_t = \mathbf{g}(S_t, A_t^\pi)$ and $A_n^\pi \sim \pi_\theta(\cdot|S_n)$ are on-policy actions. These actions are sampled on-policy after n steps and used to reduce the gradient's variance.

The utility of the n-step form for the PG is twofold: (a) it allows us to trade off bias and variance in our PG estimator, and (b) it allows us to mitigate the role of the state reweighting and the associated variance issues. Using $n = 1$ means that we rely heavily on our gradient critic, which might be biased. However, we avoid the variance of sequences of sampled $\mathbf{g}_t$, which we have for larger n. This effect is pronounced in the off-policy setting, where for $n > 1$, we correct the whole trajectory distribution. As n gets larger, we approach the classic PG estimator, with state reweighting given by the products of importance sampling ratios and γ. Therefore, the n-step estimator allows us to reduce the variance due both to sampled $\mathbf{g}_t$ and state reweighting. In the extreme, at $n = 0$, we do not need to use any reweighting, because $\boldsymbol{\Gamma}^\pi(s, a)$ allows us to query the gradient from any state and action.

Once we have this n-step estimator, it is straightforward to extend it to eligibility traces (Appendix A.1),

$$\nabla_\theta J(\theta) \propto \mathbb{E}_{\tau_\beta}\left[\sum_{t=0}^{\infty}\lambda^t\gamma^t\rho_t\big(\mathbf{g}_t + (1-\lambda)\boldsymbol{\Gamma}^\pi(S_t, A_t^\pi)\big)\right],$$

with trace parameter $\lambda \in [0, 1]$. With $\lambda = 0$, we obtain the $n = 0$ estimator, where we immediately bootstrap off of $\boldsymbol{\Gamma}^\pi$. As $\lambda \to 1$, we recover the classic PGT. This trace gradient is actually an exponential average, with weighting λ, of all n-step estimators, and so provides a smoother trade-off between bias and variance.

Finally, we can further reduce variance, at the cost of bias, by considering the PG without any state reweighting. Namely, we can instead blend between the semi-gradient and our approach, rather than the corrected gradient and our approach,

$$\nabla_\theta J(\theta) \approx \mathbb{E}_{\tau_\beta}\left[\sum_{t=0}^{\infty}\lambda^t\gamma^t\big(\mathbf{g}_t + (1-\lambda)\boldsymbol{\Gamma}^\pi(S_t, A_t^\pi)\big)\right]. \quad (15)$$

As $\lambda \to 1$, we recover the semi-gradient, because we are effectively sampling $S \sim \mu_\gamma^\beta$. The bias-variance trade-off in this estimator is more subtle. For larger λ, we are more robust to bias in the gradient critic, but also suffer more from bias due to the omission of the importance sampling ratios. Therefore, when the gradient critic is quite accurate, a lower λ might result in less bias. We showed in Theorem 2 that in some cases, the gradient critic can be unbiased, even when estimated under off-policy samples. This result highlights that this generalized estimator can provide improvements on the classic gradient estimation, allowing us to avoid reweighting and potentially reducing the bias significantly. Algorithm 1 depicts a policy improvement scheme unifying the gradient critic estimate presented in Section 4.2 with this extension to eligibility traces.

Algorithm 1 TDRC$\boldsymbol{\Gamma}$

```
Input: Set of features $\phi$, policy $\pi_\theta$, learning rates $\alpha_t$
  and $\eta$, TDRC regularization factor $\beta$, eligibility trace $\lambda$,
  initial parameters $\boldsymbol{\omega}_0$ and $\mathbf{G}_0$
$\nu_0 = 1$, $s_0 \sim \mu_0$
for $t = 0$ to $T - 1$: do
    Apply $a_t \sim \beta(\cdot|s_t)$ on the environment
    Observe state $s_t$ and reward $\mathbf{r}_{t+1}$
    Draw actions $a_t^\pi \sim \pi_\theta(\cdot|s_t)$, $a_{t+1}^\pi \sim \pi_\theta(\cdot|s_{t+1})$
    $\hat{Q}_t = \boldsymbol{\phi}^\intercal(s_t, a_t^\pi)\boldsymbol{\omega}_t$, $\hat{\boldsymbol{\Gamma}}_t = \boldsymbol{\phi}^\intercal(s_t, a_t^\pi)\mathbf{G}_t$
    $\theta \leftarrow \theta + \eta\nu_t\left(Q_t\nabla_\theta\log\pi_\theta(a_t^\pi|s_t) + (1-\lambda)\hat{\boldsymbol{\Gamma}}_t\right)$
    Compute $\boldsymbol{\omega}_{t+1}$, and $\mathbf{G}_{t+1}$ using TDRC (see Appendix C.2)
    if $s_t'$ is a terminal state: then
        $\nu_{t+1} = 1$, $s_{t+1} \sim \mu_0$
    else
        $\nu_{t+1} = \lambda\gamma\nu_t$, $s_{t+1} = s_t$
    end if
end for
```

7. Extension to Deep Reinforcement Learning

The primary goal of this paper is to introduce the theoretical foundations of gradient critic algorithms. To this end, we focused on linear function approximation; however, the concepts presented in can be extended to function approximation with deep neural networks. The primary challenge is that the gradient critic estimates a vector of size d—the number of parameters in the neural network—resulting in a very large output. To overcome this issue, we propose that the gradient critic can learn only a subset of the gradient, while still achieving a favorable bias-variance trade-off.

In fact, past literature—in addition to our own experiments—suggests that some gradients are more susceptible than others to distribution shift. Imani et al. (2018) show that the distribution shift becomes detrimental when united with state aliasing. When analyzing the estimation bias of the semi-gradient approach on their toy MDP (Figure 2a), we find that the gradient update was biased mainly for the parameters responsible for selecting the action corresponding to the aliased states (Section 1c, d, details in Appendix C.4). In MDPs, the state is usually fully informativebut as information flows from the bottom to the top layers of the neural

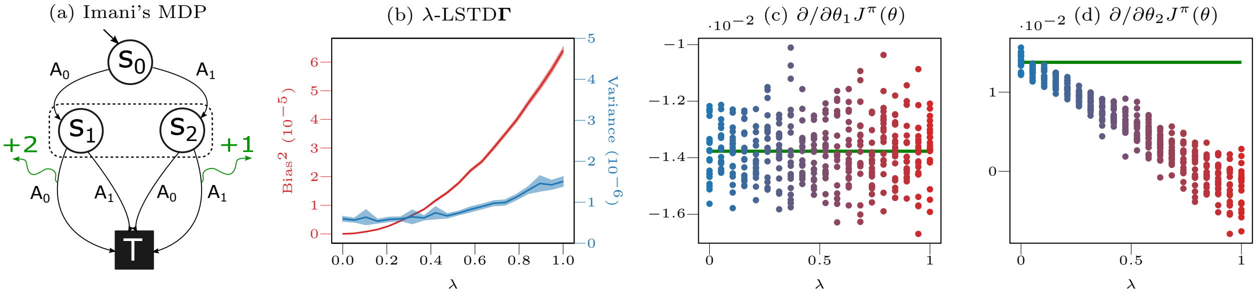

Figure 1. (a) Imani's MDP (Imani et al., 2018). (b) Bias and variance of gradient evaluation with LSTD**Γ** in Imani's MDP. Lower λ achieves lower bias and variance, showing that the gradient critic helps delivering a better estimate. (c, d) the scatter plots show single estimates of LSTD**Γ** and green lines the ground truth. While low λ helps the estimate of $\partial/\partial\theta_2$ it does not improve $\partial/\partial\theta_1$, suggesting that the gradient critic used on a convenient subset of parameters could still be beneficial.

network, the learned features may introduce state aliasing in higher-level of abstraction. We argue that learning the gradient of the last layer of the actor network will potentially trade off the complexity of learning of a high-dimensional gradient, with the benefit introduced by our approach.

8. Empirical Analysis

We want to show that 1) the semi-gradient is generally biased, whereas the gradient critic is unbiased provided realizability, 2) this unbiasedness helps the convergence to better solutions, and 3) even when applied to a subset of the actor parameters, the gradient critic helps to attain higher performance. We test four different algorithms: a classic semi-gradient algorithm, OffPAC (Degris et al., 2012), an actor critic algorithm with full importance sampling correction ACE(1) (Graves et al., 2021), a simple and new policy gradient scheme called LSTD**Γ** that uses the least-squares temporal-difference solution for the gradient critic computed from offline data using Equations 12, and TDRC**Γ** as described in Algorithm 1.

Imani's MDP (Figure 1a) is designed to show the fallacy of semi-gradient methods under off-policy distribution. In their work, Imani et al. assumed a perfect critic but aliased states for the actor. In agreement with their setup, we use a behavior policy that samples with probability 0.25 action A_0 and 0.75 action A_1. The critic's features have sufficient information for all state-action pairs $\phi(s,a) = \texttt{one-hot-encode}(s,a)$. The optimization policy is initialized with probabilities 0.9 and 0.1 for actions A_0 and A_1 respectively.[4]

Randomly Generated MDPs. The MDP mentioned above is designed appositely to show the flaws of semi-gradient algorithms, and it assumes fully informative critic features. We want to test the gradient function in a more generic setting. To this end, we randomly generate 2500 MDPs with 30 states and 2 actions. We use this task to study the effect of the application of the gradient critic restricted to a subset of the parameters.

8.1. Analysis on Imani's MDP

The goal of this set of experiments is to analyze the effect of the gradient critic estimator on Imani's MDP. In particular we analyze the estimation bias, and variance and performance of our method for different value of λ.

Bias-Variance Analysis. We generate datasets of 500 samples beforehand, using the behavioral policy. The target policy's parameters are initialized to match the condition described earlier. We estimate the gradient using LSTD**Γ** with $\lambda \in \{0, 0.05, 0.1, \dots, 1\}$. For each value of λ, we compute 1000 estimates of the bias and the variance accompanied with confidence intervals at 95%. Figure 1b shows that both the bias and the variance of the estimator increase as λ increases. This means that the gradient critic, which is most used at $\lambda \to 0$, helps in delivering high quality estimation of the gradient. Notably, the bias of semi-gradient affects only a subset of gradient vector (Figure 1c and d), suggesting that the gradient critic could be tailored to learn only a subset of the gradient (more details in Appendix C.4).

Performance of LSTDΓ. The previous analysis supports the unbiasedness discussed in (Section 5). However, this does not automatically imply an increase in performance. To provide an analysis on the performance, we generated datasets of 500 samples, and we trained the policy for 1000 steps using the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 0.01. We repeat the process 20 times for each value of λ. Figure 2a depicts the final performance of the algorithm for the different values of λ. Observe that high values of λ, like 1 or 0.9, lead to a poor solution, while lower values of λ reach high performance. This enhancement suggests that the contribution of the gradient critic is beneficial. Interestingly, even a weak mixing of the gradient critic helps the performance dramatically.

[4] Differently from their setup, we measure the performance by using the return in Section 2, instead of their proposed off-policy objective. Furthermore, our returns are discounted by $1-\gamma$.

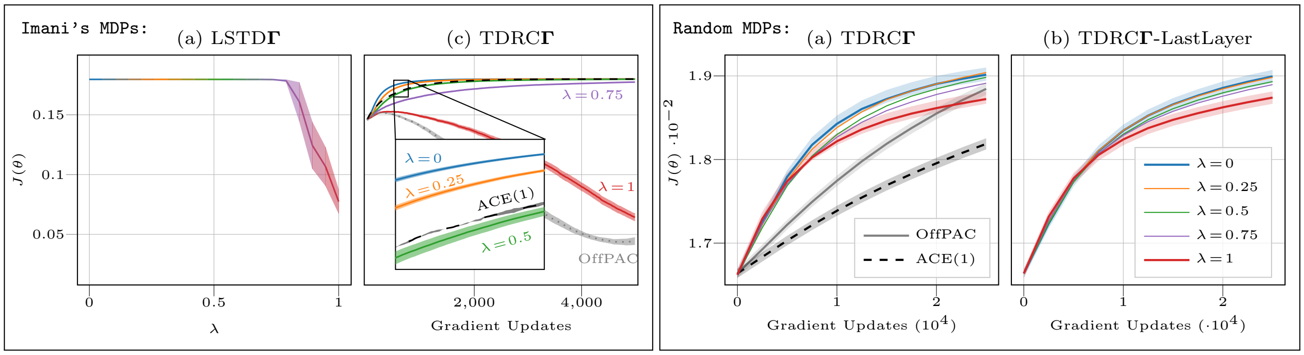

Figure 2. `Imani's MPD`: (a) final performance of LSTDΓ and (b) learning curve of TDRCΓ. `Random MPDs`: (c) learning curve of classic TDRCΓ compared with OffPAC and ACE(1) (d) learning curves of TDRCΓ with the gradient critic applied only to the last layer of the actor. We notice that lower values of λ improves the performance in both the tasks. Using the critic only on the last layer does not degrade the performance sensibly. Shaded areas show the standard error.

Performance of TDRCΓ. The online estimator of the gradient function TDRCΓ requires a separate validation as it is more subject to noise in the data. We used a similar settings as for LSTDΓ, except that samples are streamed, Adam's learning rate is set to 0.001, and the optimization takes 5000 steps. We used $\beta = 1$ as regularization factor and constant learning rate for both critic and gradient critic $\alpha = 0.1$. Figure 2b shows that the algorithm behaves similarly to LSTDΓ, enforcing the idea that the gradient critic also helps when its approximation is more pronouced. The effect of the delayed gradient estimation does not impact negatively the performance (more details in Appendix C.6).

8.2. Analysis on Randomly Generated MDPs

Previous analyses show us a clear use-case where the gradient critic helps to solve the issue of semi-gradient approaches. Despite the convergence issue discussed by Imani et al. (2018) and Fujimoto et al. (2019), semi-gradients are widely used since they perform reasonably well when the distribution shift is not too marked. Hence, we test TDRCΓ on 2500 randomly generated MDPs. We try to replicate realistic conditions and show that our method works well across different models. Our MDPs have 30 states and 2 actions. The structure of each MDP is generated randomly using a low-entropy distribution that ensures the sparsity of both mean reward and transitions. This sparsity ensures a diversification in the different MDPs, sometimes creating cycles and absorbing states. We also add a Gaussian noise to the reward to make the setting more challenging. The discount factor is 0.95, while the episode length is 50 steps.

In this set of experiments, we do not provide a direct source of state aliasing. Instead, we codify each state with its numerical, 1-dimensional value. The actor, a neural network with one hidden layer of 5 neurons, receives complete information about the state. However, its under-parametrization (26 parameters in an MDP of 30 states and 2 actions per state) can cause a similar and more realistic aliasing effect (details in Appendix C.7).

We test OffPAC, ACE(1), TDRCΓ. In addition, we include TDRCΓ-LastLayer, which uses the gradient critic only to update the last layer of the actor, while the remaining weights are updated with TDRCΓ with λ=1. Figure 2c, shows that TDRCΓ outperforms both OffPAC and ACE(1) in this setting. Furthermore, lower values of λ still obtain higher returns, showing that the gradient critic effectively improves the performance also in this scenario. It is interesting to notice that there is no substantial performance degradation between TDRCΓ and TDRCΓ-LastLayer (Figure 2d), corroborating the intuition that applying the gradient critic only to the last layer of the actor is still beneficial.

9. Conclusion and Future Work

Most policy gradient algorithms use off-policy samples without correcting the state distribution, causing a biased gradient estimate. Such bias deteriorates the algorithm's performance. Instead of resorting to importance sampling, we proposed to learn the policy gradient using a *gradient critic*. Like the classic value critic, our gradient critic is expressible with a Bellman equation, hence learnable via temporal-difference under off-policy distribution. The ability of the *gradient critic* to predict the gradient cumulation overcomes the need for sample reweighting. The gradient critic can provide an unbiased policy gradient estimator using arbitrary experience without resorting to importance sampling. Further, we introduced an approach based on eligibility traces that smoothly combines it with classic semi-gradient estimation. We showed empirically that our approach mitigates the high bias of semi-gradients, boosting its performance. Future work will focus on the extension of the gradient critic to deep reinforcement learning, using our technique to predict a subset of the policy gradient.

Acknowledgment

This research is financially supported by the Alberta Machine Intelligence Institute (Amii), the Canadian Institute for Advanced Research (CIFAR), the Reinforcement Learning and Artificial Laboratory (RLAI) in the university of Alberta, Canada. We would like to thank Compute Canada for providing the computational resources needed.

References

Baird, L. Residual Algorithms: Reinforcement Learning with Function Approximation. *Machine Learning Proceedings*, pp. 30–37, 1995.

Carvalho, J., Tateo, D., Muratore, F., and Peters, J. An Empirical Analysis of Measure-Valued Derivatives for Policy Gradients. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10. IEEE, 2021.

Degris, T., White, M., and Sutton, R. S. Off-Policy Actor-Critic. In *Proceedings of the 29th International Coference on Machine Learning*, pp. 179–186. Omnipress, 2012.

Deisenroth, M. P., Neumann, G., and Peters, J. *A Survey on Policy Search for Robotics*. Now Publishers, 2013.

Fujimoto, S., van Hoof, H., and Meger, D. Addressing Function Approximation Error in Actor-Critic Methods. *Journal of Machine Learning Research*, 80, 2018.

Fujimoto, S., Meger, D., and Precup, D. Off-Policy Deep Reinforcement Learning without Exploration. In *Proceeding of the 36th International Conference on Machine Learning*, pp. 2052–2062, 2019.

Ghiassian, S., Patterson, A., Garg, S., Gupta, D., White, A., and White, M. Gradient Temporal-Difference Learning with Regularized Corrections. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 3524–3534. PMLR, 2020.

Graves, E., Imani, E., Kumaraswamy, R., and White, M. Off-Policy Actor-Critic with Emphatic Weightings. *arXiv preprint arXiv:2111.08172*, 2021.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceeding of the 35th International Conference on Machine Learning*, pp. 1856–1865, 2018.

Heess, N., Wayne, G., Silver, D., Lillicrap, T., Erez, T., and Tassa, Y. Learning Continuous Control Policies by Stochastic Value Gradients. *Advances in Neural Information Processing Systems*, 28:2944–2952, 2015.

Imani, E., Graves, E., and White, M. An Off-Policy Policy Gradient Theorem Using Emphatic Weightings. In *Advances in Neural Information Processing Systems*, pp. 96–106, 2018.

Kingma, D. P. and Ba, J. ADAM: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kolter, J. The Fixed Points of Off-Policy TD. In *Advances in Neural Information Processing Systems*, volume 24, pp. 2169–2177, 2011.

Lagoudakis, M. G. and Parr, R. Least-Squares Policy Iteration. *Journal of Machine Learning Research*, 4:1107–1149, 2003. Publisher: JMLR. org.

Lan, Q., Tosatto, S., Farrahi, H., and Mahmood, A. R. Model-Free Policy Learning with Reward Gradients. In *Proceeding of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Virtual, 2022.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *arXiv preprint arXiv:2005.01643*, 2020.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous Control with Deep Reinforcement Learning. In *International Conference on Learning Representations*, 2016. URL http://arxiv.org/abs/1509.02971. arXiv: 1509.02971.

Liu, Q., Li, L., Tang, Z., and Zhou, D. Breaking the Curse of Horizon: Infinite-Horizon Off-Policy Estimation. In *Advances in Neural Information Processing Systems*, pp. 5356–5366, 2018.

Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. Off-Policy Policy Gradient with State Distribution Correction. *arXiv:1904.08473*, 2019. URL http://arxiv.org/abs/1904.08473. arXiv: 1904.08473.

Lu, T., Schuurmans, D., and Boutilier, C. Non-Delusional Q-learning and Value-Iteration. In *Advances in Neural Information Processing Systems*, pp. 9949–9959. Curran Associates, Inc., 2018.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-Level Control Through Deep Reinforcement Learning. *Nature*, 518(7540):529–533, 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature14236. URL http://www.nature.com/articles/nature14236.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1928–1937, 2016.

Nachum, O., Dai, B., Kostrikov, I., Chow, Y., Li, L., and Schuurmans, D. AlgaeDICE: Policy Gradient from Arbitrary Experience. *arXiv:1912.02074v1*, 2019.

Ni, C., Zhang, R., Ji, X., Zhang, X., and Wang, M. Optimal Estimation of Off-Policy Policy Gradient via Double Fitted Iteration. *arXiv preprint arXiv:2202.00076*, 2022.

Nota, C. and Thomas, P. S. Is the Policy Gradient a Gradient? *arXiv preprint arXiv:1906.07073*, 2019.

Nota, C. and Thomas, P. S. Is the Policy Gradient a Gradient? In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, 2020.

Owen, A. B. *Monte Carlo Theory, Methods and Examples*. 2013.

Peshkin, L. and Shelton, C. R. Learning from Scarce Experience. In *Proceedings of the Nineteenth International Conference on Machine Learning*, 2002. URL http://arxiv.org/abs/cs/0204043. arXiv: cs/0204043.

Shelton, C. R. Policy Improvement for POMDPs Using Normalized Importance Sampling. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, pp. 496–503. Morgan Kaufmann Publishers Inc., 2001. ISBN 978-1-55860-800-9. event-place: Seattle, Washington.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic Policy Gradient Algorithms. In *Proceedings of the 31 st International Conference on Machine Learning*, 2014.

Sutton, R. S. Learning to Predict by the Methods of Temporal Differences. *Journal of Machine Learning Research*, 3(1):9–44, 1988. Publisher: Springer.

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

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Advances in Neural Information Processing Systems*, pp. 1057–1063, 2000.

Sutton, R. S., Szepesvári, C., and Maei, H. R. A Convergent O(n) Algorithm for Off-Policy temporal-Difference Learning with Linear Function Approximation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 21, pp. 1609–1616. MIT Press, 2008.

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C., and Wiewiora, E. Fast Gradient-Descent Methods for Temporal-Difference Learning with Linear Function Approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 993–1000, 2009.

Thomas, P. Bias in Natural Actor-Critic Algorithms. In *Proceeding of the 31st International Conference on Machine Learning*, pp. 441–448, 2014.

Tosatto, S., Carvalho, J., Abdulsamad, H., and Peters, J. A Nonparametric Off-Policy Policy Gradient. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, Palermo, Italy, 2020.

Tosatto, S., Carvalho, J., and Peters, J. Batch Reinforcement Learning with a Nonparametric Off-Policy Policy Gradient. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. doi: 10.1109/TPAMI.2021.3088063.

Tsitsiklis, J. N. and Van Roy, B. An Analysis of Temporal-Difference Learning with Function Approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997. Publisher: IEEE.

Algorithm 2 Policy Gradient with LSTD$\boldsymbol{\Gamma}$

```
**Input:** Policy $\pi_\theta$, set of features $\phi$, learning rate $\eta$, and dataset $D$ of off-policy transitions $(s_i, a_i, r_i, s'_i)$.
$\hat{\mathbf{b}} = 1/N \sum_i \phi(s_i, a_i) r_i$
**while** not converged **do**
    For each $s'_i$ sample $a'_i \sim \pi_\theta(\cdot|s'_i)$
    $\hat{\mathbf{A}} = 1/N \sum_i \phi(s_i, a_i) \left(\phi(s_i, a_i) - \gamma \phi(s'_i, a'_i)\right)^\intercal$
    $\hat{Q}(s, a) = \boldsymbol{\phi}^\intercal(s, a) \hat{\boldsymbol{\omega}}_{TD}$; $\hat{\boldsymbol{\omega}}_{TD} = \hat{\mathbf{A}}^{-1} \hat{\mathbf{b}}$;
    $\hat{\mathbf{B}} = 1/N \sum_i \phi(s_i, a_i) \hat{Q}(s'_i, a'_i) \nabla_\theta \log \pi_\theta(a'_i|s'_i)$
    $\hat{\boldsymbol{\Gamma}}(s, a) = \boldsymbol{\phi}^\intercal(s, a) \hat{\mathbf{G}}_{TD}$; $\hat{\mathbf{G}}_{TD} = \hat{\mathbf{A}}^{-1} \hat{\mathbf{B}}$
    Sample $s_0 \sim \mu_0$ (from dataset), $a_0^\pi \sim \pi_\theta(\cdot|s_0)$
    $\mathbf{g} = \hat{Q}(s_0, a_0^\pi) \nabla_\theta \log \pi_\theta(a_0^\pi|s_0)$
    $\theta \leftarrow \theta + \eta(\mathbf{g} + \hat{\boldsymbol{\Gamma}}(s_0, a_0^\pi))$
**end while**
```

A. Supplement to the Theoretical Analysis

This appendix is structured as follows: we introduce a *generalized* temporal-difference in Appendix A.2, which will be useful to precisely determine Bellman equations and their least-squares solution. We prove Lemma 1 and Theorem 1 in Appendix A.3. We prove Theorem 2 in Appendix A.4.

A.1. Gradient Function and Eligibility Traces

In this section, we detail all the passages to show the n-step view of the $\boldsymbol{\Gamma}$-function, and the eligibility-trace view. It is interesting to see the parallel between the n-step view and eligibility traces in critic estimation (Sutton & Barto, 2018).

A.1.1. n-Step View of the Policy Gradient and Gradient Function

To start, let us remind that $\boldsymbol{\Gamma}^\pi(s, a) = \nabla_\theta Q^\pi(s, a)$ by definition. We report here the *gradient* Bellman equation introduced in (6), and that can be seen as the application on both sides of (1),

$$\boldsymbol{\Gamma}^\pi(s, a) = \gamma \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}} \left(Q^\pi(s', a') \nabla_\theta \log \pi_\theta(a'|s') + \boldsymbol{\Gamma}^\pi(s', a')\right) \pi(a'|s') p(s'|s, a).$$

The LHS of the gradient Bellman equation can be expanded, by using the recursive definition of $\boldsymbol{\Gamma}(s, a)$,

$$\begin{aligned}\boldsymbol{\Gamma}^\pi(s, a) = & \gamma \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}} \Big(Q^\pi(s', a') \nabla_\theta \log \pi_\theta(a'|s') \\ & + \gamma \sum_{s'' \in \mathcal{S}} \sum_{a'' \in \mathcal{A}} \Big(Q^\pi(s'', a'') \nabla_\theta \log \pi_\theta(a''|s'') + \gamma \boldsymbol{\Gamma}^\pi(s'', a'')\Big) \pi(a''|s'') p(s''|s', a')\Big) \pi(a'|s') p(s'|s, a).\end{aligned}$$

Taking in consideration that $\sum_{s'' \in \mathcal{S}} \sum_{a'' \in \mathcal{A}} \pi(a''|s'') p(s''|s', a') = 1$, we can reformulate the gradient Bellman equation as

$$\begin{aligned}\boldsymbol{\Gamma}^\pi(s, a) = & \sum_{s', a', s'', a''} \Big(\gamma Q^\pi(s', a') \nabla_\theta \log \pi_\theta(a'|s') + \gamma^2 Q^\pi(s'', a'') \nabla_\theta \log \pi_\theta(a''|s'') \\ & + \gamma^2 \boldsymbol{\Gamma}^\pi(s'', a'')\Big) \pi(a''|s'') p(s''|s', a') \pi(a'|s') p(s'|s, a),\end{aligned}$$

which is equivalent to

$$\boldsymbol{\Gamma}^\pi(s, a) = \mathbb{E}_{\pi_\theta}\left[\gamma Q^\pi(S_1, A_1) \nabla_\theta \log \pi_\theta(A_1|S_1) + \gamma^2 Q^\pi(S_2, A_2) \nabla_\theta \log \pi_\theta(A_2|S_2) + \gamma^2 \boldsymbol{\Gamma}^\pi(S_2, A_2) \Big| S_0 = s, A_0 = a\right],$$

This process can be repeated a finite number of time n, to find out that

$$\boldsymbol{\Gamma}^\pi(s, a) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=1}^{n} \gamma^t Q^\pi(S_t, A_t) \nabla_\theta \log \pi_\theta(A_t|S_t) + \gamma^n \boldsymbol{\Gamma}^\pi(S_n, A_n) \Big| S_0 = s, A_0 = a\right]. \tag{16}$$

Equation 5 united with Equation 16 yields

$$\nabla_\theta J(\theta) = (1-\gamma)\mathop{\mathbb{E}}_{\tau_\pi}\left[\sum_{t=0}^{n-1} \gamma^t Q^\pi(S_t, A_t)\nabla_\theta \log \pi_\theta(A_t|S_t) + \gamma^{n-1}\mathbf{\Gamma}^\pi(S_{n-1}, A_{n-1})\right], \tag{17}$$

which is equivalent to the last passage in the derivation 8.

A.1.2. Eligibility-Trace View of Policy Gradient

Consider $0 \leq \lambda < 1$. We know that $\sum_n^\infty \lambda^n = 1/(1-\gamma)$, hence $(1-\lambda)\sum_{n=0}^\infty \lambda^n = 1$. Consider not an enumerable set of expressions $\{x_n\}_i^\infty$ which are all mathematically equivalent to a value x, i.e., $x_0 = x_1 = x_2 = \dots = x$. We can say that $(1-\gamma)\sum_{n=0}^\infty \lambda^n x_n = x$. Let

$$y_t := \mathbb{E}_{\tau_\pi}\left[Q^\pi(S_t, A_t)\nabla_\theta \log \pi_\theta(A_t|S_t)\right] \text{ and } z_t := \mathbb{E}_{\tau_\pi}[\mathbf{\Gamma}(S_t, A_t)].$$

In consideration of Equation 17, we can say that

$$\nabla_\theta J(\theta) = (1-\gamma)(1-\lambda)\sum_{n=0}^{\infty}\lambda^n \sum_{t=0}^{n}\gamma^t y_t + \gamma^n z_t.$$

The equation above can be rewritten by “unrolling” the innermost summation

$$\begin{aligned}\nabla_\theta J(\theta) = (1-\gamma)(1-\lambda)\Big(&y_0 + z_0\\ &+\lambda y_0 + \lambda\gamma y_1 + \lambda\gamma z_1\\ &+\lambda^2 y_0 + \lambda^2\gamma y_1 + \lambda^2\gamma^2 y_2 + \lambda^2\gamma^2 z_2\\ &+\lambda^3 y_0 + \lambda^3\gamma y_1 + \lambda^3\gamma^2 y_2 + \lambda^3\gamma^3 y_3 + \lambda^3\gamma^3 z_3\\ &+\lambda^4 y_0 + \lambda^4\gamma y_1 + \lambda^4\gamma^2 y_2 + \lambda^4\gamma^3 y_3 + \lambda^4\gamma^4 y_4 + \lambda^4\gamma^4 z_4 + \dots\Big)\end{aligned}$$

The equation has be graphically arranged to highlight its structure. In particular, we can see the right hand side as a summation of y_n terms that can be collected together column-wise, plus a summation of z_n,

$$\begin{aligned}\nabla_\theta J(\theta) =&(1-\gamma)\left((1-\lambda)\sum_{n=0}^{\infty}\lambda^n y_0 + (1-\lambda)\sum_{n=1}^{\infty}\lambda^n\gamma y_1 + (1-\lambda)\sum_{n=2}^{\infty}\lambda^n\gamma^2 y_2 + \dots + (1-\lambda)\sum_{n=0}^{\infty}\lambda^n\gamma^n z_n\right)\\ =&(1-\gamma)\left((1-\lambda)\sum_{n=0}^{\infty}\lambda^n y_0 + (1-\lambda)\lambda\sum_{n=0}^{\infty}\lambda^n\gamma y_1 + (1-\lambda)\lambda^2\sum_{n=0}^{\infty}\lambda^n\gamma^2 y_2 + \dots + (1-\lambda)\sum_{n=0}^{\infty}\lambda^n\gamma^n z_n\right)\\ =&(1-\gamma)\left((y_0 + \lambda\gamma y_1 + \lambda^2\gamma^2 y_2 + \dots)(1-\lambda)\sum_{n=0}^{\infty}\lambda^n + (1-\lambda)\sum_{n=0}^{\infty}\lambda^n\gamma^n z_n\right)\\ =&(1-\gamma)\left((y_0 + \lambda\gamma y_1 + \lambda^2\gamma^2 y_2 + \dots) + (1-\lambda)\sum_{n=0}^{\infty}\lambda^n\gamma^n z_n\right)\\ =&(1-\gamma)\sum_{n=0}^{\infty}\lambda^n\gamma^n y_n + (1-\lambda)\lambda^n\gamma^n z_n\\ =&(1-\gamma)\sum_{n=0}^{\infty}\lambda^n\gamma^n\left(y_n + (1-\lambda)z_n\right)\end{aligned}$$

Looking back at the definitions of y_n and z_n, we can state

$$\nabla_\theta J(\theta) =(1-\gamma)\mathop{\mathbb{E}}_{\tau_\pi}\left[\sum_{n=0}^{\infty}\lambda^n\gamma^n\big(Q^\pi(S_n, A_n)\nabla_\theta \log \pi_\theta(A_n|S_n) + (1-\lambda)\mathbf{\Gamma}^\pi(S_n, A_n)\big)\right].$$

A.2. Generalized Least-Squares Temporal-Difference

This section provides a generalization of least-square temporal-difference. We introduce a setting that abstract the concepts of state and action (which will be seen as a conglomerate variable x), and that unifies a finite set of "Bellman" equations that share same dynamics but different "rewards" in a compact vectorial notation. Eventually, we report the error analysis conduced by (Kolter, 2011) using our vectorial notation.

Proposition 1 (Generalized Least Squares)**.** *Let us consider finite set $\{x_1, x_2, \dots, x_n\} \equiv \mathcal{X}$. Let us consider an irreducible Markov chain induced by the transition function $g(x'|x)$ with steady distribution μ. Let us consider K stochastic mappings $c_k : \mathcal{X} \to \Omega(\mathbb{R})$ where $\Omega(\mathbb{R})$ denotes the set of all probability distributions over $\mathbb{R}$. Let us assume that $\overline{c}_k(x) = \mathbb{E}[c(x)]$ exists and it is finite for all $x \in \mathcal{X}$ and $k \in \{1, \dots, K\}$. Consider $\gamma \in [0, 1)$. Consider the Bellman-like equations*

$$f_k(x) = \overline{c}(x) + \gamma \sum_{x' \in \mathcal{X}} f_k(x') g(x'|x),$$

where each $f_k : \mathcal{X} \to \mathbb{R}$ exists and is unique. The equations above can be rewritten as

$$\mathbf{f}(x) = \overline{\mathbf{c}}(x) + \gamma \sum_{x' \in \mathcal{X}} \mathbf{f}(x') g(x'|x), \tag{18}$$

where $\mathbf{f} : \mathcal{X} \to \mathbb{R}^K$ exists and is unique. Consider a function $\hat{\mathbf{f}}_t(x) = \boldsymbol{\phi}^\intercal(x)\mathbf{H}_t$ where $\boldsymbol{\phi} : \mathcal{X} \to \mathbb{R}^{n_f}$ is a feature vector and $\mathbf{H} \in \mathbb{R}^{n_f \times K}$. Furtermore, consider a matrix $\boldsymbol{\Phi}$ where each row i is $\boldsymbol{\phi}^\intercal(x_i)$ and assume that all the columns of $\boldsymbol{\Phi}$ are linearly independend. Consider a process that starts with a desired parameter $\mathbf{H}_0$, and that updates

$$\mathbf{H}_{t+1} = \arg\min_{\mathbf{H}} \|\boldsymbol{\phi}^\intercal(x)\mathbf{H} - \overline{\mathbf{c}}(x) - \gamma \sum_{x' \in \mathcal{X}} \hat{\mathbf{f}}_t(x') g(x'|x)\|_d, \tag{19}$$

where $\|\mathbf{x}\|_d = \mathbb{E}_d[\langle \mathbf{x}, \mathbf{x} \rangle]$. It is possible to verify that the process described in (19) *is equivalent to*

$$\mathbf{h}_{t+1,i} = \arg\min_{\mathbf{h}} \|\boldsymbol{\phi}^\intercal(x)\mathbf{h} - \overline{c}_i(x) - \gamma \sum_{x' \in \mathcal{X}} \hat{f}_{t,i}(x') g(x'|x)\|_d \tag{20}$$

with $\mathbf{H}_{t+1} = [\mathbf{h}_{t+1,1}, \mathbf{h}_{t+1,2}, \dots, \mathbf{h}_{t+1,k}]$. As reported by (Lagoudakis & Parr, 2003), the fixed point of (20) *is*

$$\mathbf{h}_i^* = \mathop{\mathbb{E}}_{\substack{x \sim d, \\ x' \sim g(x)}} \left[\boldsymbol{\phi}(x)(\boldsymbol{\phi}(x) - \gamma\boldsymbol{\phi}(x'))^\intercal\right]^{-1} \mathbb{E}_{x \sim d}\left[\boldsymbol{\phi}(x)\overline{c}_i(x)\right]$$

, which can be compactly rewritten in vectorial notation

$$\mathbf{H}^* = \mathop{\mathbb{E}}_{\substack{x \sim d, \\ x' \sim g(x)}} \left[\boldsymbol{\phi}(x)(\boldsymbol{\phi}(x) - \gamma\boldsymbol{\phi}(x'))^\intercal\right]^{-1} \mathbb{E}_{x \sim d}\left[\boldsymbol{\phi}(x)\mathbf{c}^\intercal(x)\right]. \tag{21}$$

Thanks to the work of (Kolter, 2011), we are able to bound the "scalar" fixed point solution, i.e.,

$$\|\boldsymbol{\phi}^\intercal(x)\mathbf{h}^* - f_i(x)\|_d \leq \frac{1 + \kappa\gamma}{1 - \gamma} \min_{\mathbf{h}} \|\boldsymbol{\phi}^\intercal(x)\mathbf{h} - f_i(x)\|_d.$$

where $\kappa = \max_i \sqrt{d(x_i)/\mu(x_i)} / \min_i \sqrt{d(x_i)/\mu(x_i)}$ and d satisties the inequality in (Kolter, 2011). Knowing that $\|\mathbf{x}\|_d = \mathbb{E}_d[\langle \mathbf{x}, \mathbf{x} \rangle] = \sum_i \mathbb{E}_d[x_i^2]$, we can see that

$$\begin{aligned} &\sum_i \|\boldsymbol{\phi}^\intercal(x)\mathbf{h}_i^* - f_i(x)\|_d \leq \sum_i \frac{1 + \kappa\gamma}{1 - \gamma} \min_{\mathbf{h}} \|\boldsymbol{\phi}^\intercal(x)\mathbf{h} - f_i(x)\|_d \\ \implies &\|\boldsymbol{\phi}^\intercal(x)\mathbf{H}^* - \mathbf{f}(x)\|_d \leq \frac{1 + \kappa\gamma}{1 - \gamma} \min_{\mathbf{H}} \|\boldsymbol{\phi}^\intercal(x)\mathbf{H} - \mathbf{f}(x)\|_d. \end{aligned} \tag{22}$$

A.3. Least Squares Solution for the Gradient Function

Proof of Lemma 1. Let us analize Equation 6 for a single parameter θ_k and for finite state-action space,

$$\Gamma^{\pi}_{k}(s,a) = \gamma \sum_{s'}\sum_{a'} \left(Q^{\pi}(s',a') \frac{\partial}{\partial \theta_k} \log \pi_{\theta}(a'|s') + \Gamma^{\pi}_{k}(s',a') \right) \pi(a'|s') p(s'|s,a) \tag{23}$$

Let us set $\mathbf{x} = (s,a)$ and $g(\mathbf{x}'|\mathbf{x}) = \pi(a'|s')p(s'|s,a)$, $e(\mathbf{x}) = Q(s,a)\partial/\partial\theta_k \pi_\theta(a|s)$, and $\overline{c}(\mathbf{x}) = \sum_i e_i(\mathbf{x}')g(\mathbf{x}'|\mathbf{x})$, and, by posing $f(\mathbf{x}) = \Gamma^{\pi}_{i}(s,a)$, we realize that

$$\begin{aligned} \Gamma^{\pi}_{k}(s,a) &= \gamma \sum_{s'}\sum_{a'} \left(Q^{\pi}(s',a') \frac{\partial}{\partial \theta_k} \log \pi_{\theta}(a'|s') + \Gamma^{\pi}_{k}(s',a') \right) \pi(a'|s') p(s'|s,a) \\ \implies f(\mathbf{x}) &= \overline{c}(\mathbf{x}) + \gamma f(\mathbf{x}_i) p(\mathbf{x}_i|\mathbf{x}). \end{aligned} \tag{24}$$

Notice that in Lemma 1, we consider the stationary distribution μ_π w.r.t. the transition $p(s'|s,a)$. we notice that $\mu_\pi(s)\pi(a|s)$ is the stationary distribution w.r.t. thetransition $g(\mathbf{x}'|\mathbf{x})$. Taking in consideration Equation 21, we can prove that $\hat{\mathbf{\Gamma}}^{\pi}_{TDQ}$ is a fixed point of the approximated gradient Bellman equation. Furthermore, it is possible to prove, thank to (Kolter, 2011), that

$$\left\| \mathbf{\Gamma}^{\pi}_{TDQ,k}(s,a) - \frac{\partial}{\partial \theta_k} Q^{\pi}(s,a) \right\|_{\zeta} \leq \frac{1-\gamma\kappa}{1-\gamma} \min_{\mathbf{g}} \left\| \boldsymbol{\phi}^{\intercal}(s,a)\mathbf{g} - \frac{\partial}{\partial \theta_k} Q^{\pi}(s,a) \right\|_{\zeta},$$

and, therefore,

$$\left\| \mathbf{\Gamma}^{\pi}_{TDQ}(s,a) - \nabla_\theta Q^{\pi}(s,a) \right\|_{\zeta} \leq \frac{1-\gamma\kappa}{1-\gamma} \min_{\mathbf{G}} \left\| \boldsymbol{\phi}^{\intercal}(s,a)\mathbf{G} - \nabla_\theta Q^{\pi}(s,a) \right\|_{\zeta}. \tag{25}$$

where $\kappa = \max_{s,a} h(s,a)/\min_{s,a} h(s,a)$, $h(s,a) = \sqrt{\mu(s)\pi_\theta(a|s)}/\sqrt{\mu_\beta(s)\beta(a|s))}$ and $\mu_\pi(s)\pi_\theta(a|s)$ must comply the matrix inequality defined in (Kolter, 2011). □

Proof of Theorem 1. Let us consider an arbitrary order of state and action pairs, and a feature matrix

$$\mathbf{\Phi} = \begin{bmatrix} \boldsymbol{\phi}^{\intercal}(s_1,a_1) \\ \boldsymbol{\phi}^{\intercal}(s_1,a_2) \\ \vdots \\ \boldsymbol{\phi}^{\intercal}(s_n,a_m) \end{bmatrix}, \boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\varphi}^{\intercal}(s_1,a_1) \\ \boldsymbol{\varphi}^{\intercal}(s_1,a_2) \\ \vdots \\ \boldsymbol{\varphi}^{\intercal}(s_n,a_m) \end{bmatrix}$$

where n is the number of states and m is the number of actions. $\mathbf{\Phi} \in \mathbb{R}^{nm \times n_f}$. Consider an arbitrary parameter vector $\boldsymbol{\omega}$. $\boldsymbol{\Phi}\boldsymbol{\omega}$ returns a vector of values for each state-action pairs. We denote the TD solution of the Q-function with $\hat{\mathbf{q}} = \boldsymbol{\Phi}\boldsymbol{\omega}_{TD}$. Pairwise, we denote a matrix representing the matrix function with $\hat{\boldsymbol{\nu}} = \mathbf{\Phi}\mathbf{G}_{TD}$. The true Q-function and $\mathbf{\Gamma}$-function are, in vector notation,

$$\mathbf{q}^{\pi} = \begin{bmatrix} Q^{\pi}(s_1,a_1) \\ Q^{\pi}(s_1,a_2) \\ \vdots \\ Q^{\pi}(s_n,a_m) \end{bmatrix}, \quad \boldsymbol{\nu}^{\pi} = \begin{bmatrix} \mathbf{\Gamma}^{\pi}(s_1,a_1) \\ \mathbf{\Gamma}^{\pi}(s_1,a_2) \\ \vdots \\ \mathbf{\Gamma}^{\pi}(s_n,a_m) \end{bmatrix}.$$

Similarly to (Lagoudakis & Parr, 2003), we introduce the transition matrix $\mathbf{P}$ and the policy $\mathbf{\Pi}$

$$\mathbf{\Pi} = \mathbf{I}_n \otimes \pi^{\intercal} \quad \text{where} \quad \boldsymbol{\pi} = [\pi_\theta(a_1|s_1), \pi_\theta(a_2|s_1), \dots, \pi_\theta(a_m|s_n)]^{\intercal}$$

and

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_n \end{bmatrix} \quad \text{where} \quad \mathbf{P}_i = \begin{bmatrix} p(s_i|s_1,a_1) & p(s_i|s_2,a_1) & \dots & p(s_i|s_n,a_1) \\ p(s_i|s_1,a_2) & p(s_i|s_2,a_2) & \dots & p(s_i|s_n,a_2) \\ & & \vdots & \\ p(s_i|s_1,a_m) & p(s_i|s_2,a_m) & \dots & p(s_i|s_n,a_m) \end{bmatrix} \tag{26}$$

Let D a diagonal matrix where at each entry we have $\mu_\beta(x_i)\beta(a_j|x_i)$ where the indexes follow the enumeration introduced above. Let us introduce the norm $\|\mathbf{M}\|_D$ of a matrix $\mathbf{M}$,

$$\|\mathbf{M}\|_D = \sqrt{\sum_i D_{i,i}\langle \mathbf{M}_i, \mathbf{M}_i\rangle}$$

The least squares solution of the gradient inder the norm $\|\cdot\|_D$ is the unique solution of

$$\hat{\boldsymbol{\nu}} = \boldsymbol{\Psi}\left(\gamma \boldsymbol{\Pi}\mathbf{P}(\nabla_\theta \log \boldsymbol{\pi}) \odot \hat{\mathbf{q}} + \gamma \boldsymbol{\Pi}\mathbf{P}\hat{\boldsymbol{\nu}}\right) \tag{27}$$

where

$$\nabla_\theta \log \boldsymbol{\pi} = \begin{bmatrix} \nabla_\theta^\intercal \log \pi_\theta(a_1|s_1) \\ \nabla_\theta^\intercal \log \pi_\theta(a_2|s_1) \\ \vdots \\ \nabla_\theta^\intercal \log \pi_\theta(a_m|s_n) \end{bmatrix},$$

$(\mathbf{A}\odot\mathbf{b})_i = \mathbf{A}_i b_i$ is a row-wise product and $\boldsymbol{\Psi}_D = \boldsymbol{\Phi}(\boldsymbol{\Phi}^\intercal D \boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\intercal D$ is a non-expansion under the norm $\|\cdot\|_D$ as shown by (Tsitsiklis & Van Roy, 1997). Remember, that $\boldsymbol{\Psi}$ is a least-square projection under the norm $\|\cdot\|_D$, and, therefore, $\|\boldsymbol{\Psi}\mathbf{M} - \mathbf{M}\|_D = \min_{\mathbf{H}} \|\boldsymbol{\Phi}\mathbf{H} - \mathbf{M}\|$. The true gradient function is the fixed point of the gradient Bellman equation,

$$\boldsymbol{\nu}^\pi = \gamma\boldsymbol{\Pi}\mathbf{P}(\nabla_\theta \log \boldsymbol{\pi}) \odot \mathbf{q}^\pi + \gamma\boldsymbol{\Pi}\mathbf{P}\boldsymbol{\nu}. \tag{28}$$

We want now to bound $\|\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}^\pi\|_D$.

$$\begin{aligned}
\|\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}^\pi\|_D &= \|\hat{\boldsymbol{\nu}} - \boldsymbol{\Psi}\boldsymbol{\nu}^\pi + \boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi\|_D \\
&\le \|\hat{\boldsymbol{\nu}} - \boldsymbol{\Psi}\boldsymbol{\nu}^\pi\|_D + \|\boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi\|_D \\
&= \|\boldsymbol{\Psi}\hat{\boldsymbol{\nu}} - \boldsymbol{\Psi}\boldsymbol{\nu}^\pi\|_D + \|\boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi\|_D \\
&= \|\boldsymbol{\Psi}\left(\gamma\boldsymbol{\Pi}\mathbf{P}(\nabla_\theta \log \boldsymbol{\pi}) \odot \hat{\mathbf{q}} + \gamma\mathbf{P}\Pi\hat{\boldsymbol{\nu}}\right) - \boldsymbol{\Psi}\left(\gamma\boldsymbol{\Pi}\mathbf{P}(\nabla_\theta \log \boldsymbol{\pi}) \odot \hat{\mathbf{q}} + \gamma\mathbf{P}\Pi\boldsymbol{\nu}^\pi\right)\|_D + \|\boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi\|_D \\
&\le \gamma\underbrace{\|\boldsymbol{\Psi}\mathbf{P}\Pi(\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}^\pi)\|_D}_{\text{A}} + \gamma\underbrace{\|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\nabla_\theta \log \boldsymbol{\pi}) \odot (\hat{\mathbf{q}} - \mathbf{q}^\pi)\|_D}_{\text{B}} + \|\boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi\|_D
\end{aligned}$$

Upperbound of term A. Since D satisfies, by assumption, the inequality in (Kolter, 2011), then $\|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}\boldsymbol{\Phi}\boldsymbol{\omega}\|\|_D \le \|\boldsymbol{\phi}\boldsymbol{\omega}\|_D$,

$$\|\boldsymbol{\Psi}\mathbf{P}\boldsymbol{\Pi}(\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}^\pi)\|_D = \|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\boldsymbol{\Phi}\mathbf{G}_{TD} - \boldsymbol{\nu}^\pi)\|_D \le \|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\boldsymbol{\Phi}\mathbf{G}_{TD} - \boldsymbol{\Psi}\boldsymbol{\nu}^\pi)\|_D + \|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi)\|_D \tag{29}$$

knowing that exists some $\overline{\mathbf{G}}$ such that $\boldsymbol{\Psi}\boldsymbol{\nu}^\pi = \boldsymbol{\Phi}\overline{\mathbf{G}}$, we have that

$$\begin{aligned}
&\|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\boldsymbol{\Phi}\mathbf{G}_{TD} - \boldsymbol{\Psi}\boldsymbol{\nu}^\pi)\|_D + \|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi)\|_D \\
=&\|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}\boldsymbol{\Phi}(\mathbf{G}_{TD} - \overline{\mathbf{G}})\|_D + \|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi)\|_D \\
\le&\|\boldsymbol{\Phi}\mathbf{G}_{TD} - \boldsymbol{\Phi}\overline{\mathbf{G}}\|_D + \|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi)\|_D
\end{aligned}$$

furthermore, thanks to the convexity of the spanning set, we have that

$$\|\boldsymbol{\Phi}\mathbf{G}_{TD} - \boldsymbol{\Phi}\overline{\mathbf{G}}\|_D \le \|\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}^\pi\|_D.$$

Furthermore,

$$\|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi)\|_D \le \|\boldsymbol{\Pi}\mathbf{P}\|_D\|\boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi\|_D,$$

which yields

$$\|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}^\pi)\|_D \le \|\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}^\pi\|_D + \|\boldsymbol{\Pi}\mathbf{P}\|_D\|\boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi\|_D \tag{30}$$

Table 1. Description of symbols used in proof of Theorem 1.

Symbol	Dimension	Meaning
n	-	Number of states
m	-	Number of actions
n_p	-	Number of policy parameters
n_p	-	Number of features
γ	-	Discount factor
b	-	$\lvert \max_{a,s,i} \partial/\partial\theta_i \log \pi(a\|s)\rvert$
κ	-	Defined in Lemma 1
D	$nm \times nm$	Diagonal matrix containing off-policy probabilities $\mu_\beta(s_i)\beta(a_j\|s_i)$
q	$nm \times 1$	Vector of Q-values
$\hat{\mathrm{q}}$	$nm \times 1$	TD-solution of q
$\boldsymbol{\Phi}$	$nm \times n_f$	Matrix of features
$\boldsymbol{\Psi}$	$nm \times nm$	Orthogonal projection onto $\|\cdot\|_D$
$\mathbf{P}$	$n \times nm$	Transition matrix
$\boldsymbol{\pi}$	$nm \times 1$	Vector represention of the policy
$\boldsymbol{\Pi}$	$nm \times n$	Matrix representation of the olicy
$\nabla_\theta \log \boldsymbol{\pi}$	$nm \times n_p$	Matrix of gradients of $\log \boldsymbol{\pi}$
$\boldsymbol{\nu}$	$nm \times n_p$	Matrix representing the true $\boldsymbol{\Gamma}$ per state-action pairs
$\hat{\boldsymbol{\nu}}$	$nm \times n_p$	TD-solution of $\boldsymbol{\Gamma}$ per state-action pairs

Upperbound of term B. The upperbound of the term B follows a very similar structure to term A, with the addition that we need to deal with the weighting $\nabla_\theta \log \boldsymbol{\pi}$. Recalling the non-expansion from (Kolter, 2011) and the definition of the row-wise product $\odot$ given earlier,

$$\begin{aligned}\|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\nabla_\theta \log \boldsymbol{\pi}) \odot (\hat{\mathbf{q}} - \mathbf{q}^\pi)\|_D \leq& \|\boldsymbol{\Pi}\mathbf{P}\|_D \|(\nabla_\theta \log \boldsymbol{\pi}) \odot (\hat{\mathbf{q}} - \mathbf{q}^\pi)\|_D \\ \leq& n_p \|\nabla_\theta \log \boldsymbol{\pi}\|_\infty \|\boldsymbol{\Pi}\mathbf{P}\|_D \|(\hat{\mathbf{q}} - \mathbf{q}^\pi)\|_D,\end{aligned}$$

where n_d is the number of parameters of the policy. The quantity $\|\hat{\mathbf{q}} - \mathbf{q}^\pi\|_D$ has been bounded in Lemma 1,

$$\|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\nabla_\theta \log \boldsymbol{\pi}) \odot (\hat{\mathbf{q}} - \mathbf{q}^\pi)\|_D \leq \|\nabla_\theta \log \boldsymbol{\pi}\|_\infty \|\boldsymbol{\Pi}\mathbf{P}\|_D \frac{1+\gamma\kappa}{1-\gamma} \min_{\boldsymbol{\omega}} \|\boldsymbol{\Phi}\boldsymbol{\omega} - \mathbf{q}^\pi\|.$$

we notice that the term b introduced in Theorem 1 is actually $\|\nabla_\theta \log \boldsymbol{\pi}\|_\infty$, hence,

$$\|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\nabla_\theta \log \boldsymbol{\pi}) \odot (\hat{\mathbf{q}} - \mathbf{q}^\pi)\|_D \leq b \|\boldsymbol{\Pi}\mathbf{P}\|_D \frac{1+\gamma\kappa}{1-\gamma} \min_{\boldsymbol{\omega}} \|\boldsymbol{\Phi}\boldsymbol{\omega} - \mathbf{q}^\pi\|.$$

Collecting both upperbound of terms A and B.

$$\|\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}^\pi\|_D \leq \gamma \|\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}^\pi\|_D + \gamma \underbrace{\|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\nabla_\theta \log \boldsymbol{\pi}) \odot (\hat{\mathbf{q}} - \mathbf{q}^\pi)\|_D}_{\text{B}} + (1 + \gamma \|\boldsymbol{\Pi}\mathbf{P}\|_D) \|\boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi\|_D$$

Notice that the term $\|\boldsymbol{\Pi}\mathbf{P}\|_D$ can be bounded by κ, as illustrated in (Kolter, 2011), and therefore

$$\|\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}^\pi\|_D \leq \gamma \|\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}^\pi\|_D + \gamma \underbrace{\|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\nabla_\theta \log \boldsymbol{\pi}) \odot (\hat{\mathbf{q}} - \mathbf{q}^\pi)\|_D}_{\text{B}} + (1 + \gamma\kappa) \|\boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi\|_D.$$

Projection errors like $\|\boldsymbol{\Psi}\mathbf{q}^\pi - \mathbf{q}^\pi\|_D$ and $\|\boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi\|_D$ can be bounded by

$$\|\boldsymbol{\Psi}\mathbf{q}^\pi - \mathbf{q}^\pi\|_D \leq \min_{\boldsymbol{\omega}} \|\boldsymbol{\Phi}\boldsymbol{\omega} - \mathbf{q}^\pi\|_D \quad \text{and} \quad \|\boldsymbol{\Psi}\boldsymbol{\nu}^\pi - \boldsymbol{\nu}^\pi\|_D \leq \min_{\mathbf{G}} \|\boldsymbol{\Phi}\mathbf{G} - \boldsymbol{\nu}^\pi\|_D.$$

Hence,

$$(1-\gamma)\|\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}^\pi\|_D \leq \gamma \underbrace{\|\boldsymbol{\Psi}\boldsymbol{\Pi}\mathbf{P}(\nabla_\theta \log \boldsymbol{\pi}) \odot (\hat{\mathbf{q}} - \mathbf{q}^\pi)\|_D}_{\text{B}} + (1 + \gamma\kappa) \min_{\mathbf{G}} \|\boldsymbol{\Phi}\mathbf{G} - \boldsymbol{\nu}^\pi\|_D.$$

$$\leq \gamma n_p b \kappa \frac{1+\gamma\kappa}{1-\gamma} \min_{\boldsymbol{\omega}} \|\boldsymbol{\Phi}\boldsymbol{\omega} - \mathbf{q}^\pi\| + (1+\gamma\kappa) \min_{\mathbf{G}} \|\mathbf{\Phi}\mathbf{G} - \boldsymbol{\nu}^\pi\|_D.$$

which yields

$$\|\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}^\pi\|_D \leq \gamma n_p b \kappa \frac{1+\gamma\kappa}{(1-\gamma)^2} \min_{\boldsymbol{\omega}} \|\boldsymbol{\Phi}\boldsymbol{\omega} - \mathbf{q}^\pi\| + \frac{(1+\gamma\kappa)}{1-\gamma} \min_{\mathbf{G}} \|\mathbf{\Phi}\mathbf{G} - \boldsymbol{\nu}^\pi\|_D.$$

□

A.4. Unbiased Gradient with Perfect Features

Proof of ***Theorem 2***. We start the proof by showing that

$$\boldsymbol{\phi}^\top(s,a)\omega_{TD} = Q^\pi(s,a)$$

Notice that by assumption, there must be a vector $\boldsymbol{\omega}$ such that

$$\boldsymbol{\xi}^\top(s,a)\boldsymbol{\omega} = r(s,a). \tag{31}$$

where

$$\boldsymbol{\xi}(s,a) = \boldsymbol{\phi}(s,a) - \gamma \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \boldsymbol{\phi}(s',a')\pi_\theta(a'|s')p(s'|s,a)\,\mathrm{d}a'\,\mathrm{d}s' \tag{32}$$

Since Φ' is a n_f-dimensional vector space, there is one and only one ω satisfating the relation above. Given the fact that (31) is a linear equation, we can take a set of linearly independent features $\boldsymbol{\xi}$ to solve it. Since Φ' admits a n_f-dimensional basis, there exist $\{(s_i, s_i)\}_{i=1}^{n_f}$ such that we can construct a set of n_d linearly independent vectors $\mathbf{e}_i = \boldsymbol{\xi}(s_i, a_i)$. Let us construct a basis matrix $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{n_f}]$. The unique solution of (31) is determined by

$$\boldsymbol{\omega}^* = \mathbf{E}^{-\top}\mathbf{r}, \tag{33}$$

where $\mathbf{r} = [r(s_1,a_1), r(s_2,a_2), \dots, r(s_{n_f}, a_{n_f})]^\top$. The TD solution satisfies

$$\begin{aligned} &\mathbb{E}_\zeta\left[\boldsymbol{\phi}(S,A)\left(\boldsymbol{\phi}^\top(S,A) - \gamma\boldsymbol{\phi}^\top(S',A')\right)\right]\boldsymbol{\omega}_{TD} = \mathbb{E}_\zeta\left[\boldsymbol{\phi}(S,A)r(S,A)\right] \\ \implies &\mathbb{E}_\zeta\left[\boldsymbol{\phi}(S,A)\boldsymbol{\xi}^\top(S,A)\right]\omega_{TD} = \mathbb{E}_\zeta\left[\boldsymbol{\phi}(S,A)r(S,A)\right], \end{aligned} \tag{34}$$

where ζ is a process generating $S \sim \mu_\beta, A \sim \beta(\cdot|S), S' \sim p(\cdot|S,A)$ and $A' \sim \pi_\theta(\cdot|S')$. Notice that, thanks to the property of vector spaces, there are two functions $\mathbf{f} : \mathcal{S}\times\mathcal{A} \to \mathbb{R}^{n_f}$ and $\mathbf{h} : \mathcal{S}\times\mathcal{A} \to \mathbb{R}^{n_f}$ such that

$$\boldsymbol{\xi}(s,a) = \mathbf{E}\mathbf{f}(s,a), \boldsymbol{\phi}(s,a) = \mathbf{B}\mathbf{h}(s,a) \qquad \forall s \in \mathcal{S} \wedge a \in \mathcal{A},$$

where $\mathbf{B}$ is a basis function for the set Φ (defined in Theorem 2). We can rewrite (34) as

$$\mathbb{E}_\zeta\left[\mathbf{B}\mathbf{h}(S,A)\mathbf{f}^\top(S,A)\mathbf{E}^\top\right]\omega_{TD} = \mathbb{E}_\zeta\left[\mathbf{B}\mathbf{h}(S,A)r(S,A)\right]$$

looging back to Equation 31, we notice that $r(S,A) = f^\top(S,A)\mathbf{E}^\top\omega = f^\top(S,A)\mathbf{E}^\top\mathbf{E}^{-\top}\mathbf{r}$, and, therefore,

$$\begin{aligned} &\mathbb{E}_\zeta\left[\mathbf{B}\mathbf{h}(S,A)\mathbf{f}^\top(S,A)\mathbf{E}^\top\right]\boldsymbol{\omega}_{TD} = \mathbb{E}_\zeta\left[\mathbf{B}\mathbf{h}(S,A)f^\top(S,A)\mathbf{E}^\top\mathbf{E}^{-1}\mathbf{r}\right] \\ \implies &\boldsymbol{\omega}_{TD} = \mathbf{E}^{-\top}\mathbf{r}. \end{aligned}$$

Therefore, looking back at Equation 33

$$\boldsymbol{\phi}^\top(s,a)\boldsymbol{\omega}_{TD} = \boldsymbol{\phi}^\top(s,a)\boldsymbol{\omega}^* = Q^\pi(s,a) \qquad \forall s \in \mathcal{S} \wedge a \in \mathcal{A}.$$

This result is valid for any policy π, and state-action pairs. This implies that

$$\nabla_\theta\boldsymbol{\phi}^\top(s,a)\boldsymbol{\omega}_{TD} = \nabla_\theta Q^\pi(s,a) \qquad \forall s \in \mathcal{S} \wedge a \in \mathcal{A},$$

which, thanks to Lemma 2, implies that

$$\mathbf{\Gamma}(s,a) = \nabla_\theta Q^\pi(s,a) \qquad \forall s \in \mathcal{S} \wedge a \in \mathcal{A}.$$

□

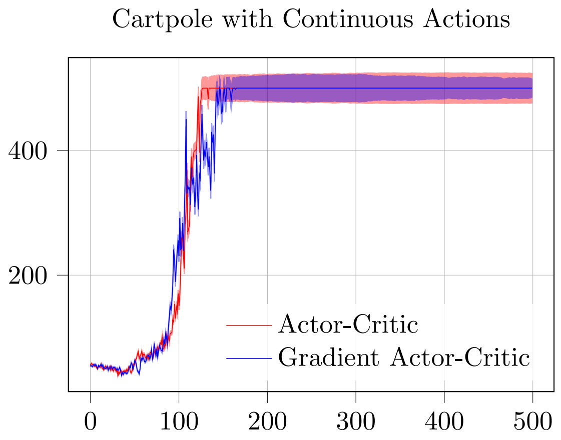

Figure 3. We run two algorithms: a classic actor-critic architecture and our gradient actor-critic architecture. On this task, both the algorithms exhibit similar performace, showing, nevertheless that our gradient actor-critic successfully solves the task.

B. Extension to the Continuous State-Action Space

Consider a Markov decision process formed by the tuple $(\mathcal{S}, \mathcal{A}, r, p, \gamma, \mu_0)$ where $\mathcal{S}$ and $\mathcal{A}$ represent the set of states and actions, $r : \mathcal{S} \times \mathcal{A} \to [-R_{\max}, R_{\max}]$ is a bounded reward function, $p : \mathcal{S} \times \mathcal{A} \to \mathcal{M}(\mathcal{S})$ is a transition probability, $\gamma \in [0, 1)$ the discount factor and $\mu_0 \in \mathcal{M}(\mathcal{S})$ a distribution of starting states. We assume that the policy $\pi_\theta : \mathcal{S} \to \mathcal{M}(\mathcal{A})$ is differentiable w.r.t. its parameters θ. We denoted with $\mathcal{M}(X)$ the set of probability measures over a σ-algebra on a set X.

Informal Extension of the Proofs to Continuous State-Action Space. The proofs in Appendix A.1.1, A.4 remain valid in the continuous case, since they only require substituting summations with integrals. The proofs in Appendix A.2 A.3 can also be arranged in the continuous state-action spaces by rewriting the norm operator $\|\mathrm{B}\|_d = \sqrt{d_i \langle \mathbf{b}_i, \mathbf{b}_i \rangle}$ as $\sqrt{\mathbb{E}_{\mathbf{b} \sim d(\mathbf{b})} [\langle \mathbf{b}, \mathbf{b} \rangle]}$ with $\mathbf{b} \in \mathcal{B}$ where d is a probability measure over a σ-algebra on $\mathcal{B}$.

B.1. Reparametrization Gradient

The gradient Bellman equation can be framed also in terms of reparametrization gradient. Suppose that we have a function $f(s, \epsilon)$ with $\epsilon \sim p$ such that

$$A = f_\theta(s, \epsilon) \overset{d}{=} A \sim \pi_\theta(\cdot|s). \tag{35}$$

We can rewrite the classic Bellman eqution as

$$Q^\pi(s, a) = r(s, a) + \mathbb{E}_{s', \epsilon} \left[Q^\pi(s', f_\theta(s', \epsilon)) \right],$$

and taking the gradients on both the sides yields

$$\begin{aligned} \nabla_\theta Q^\pi(s, a) &= \gamma \mathbb{E}_{s', \epsilon} \left[\nabla_{a'} Q^\pi(s', a') \big|_{a' = f_\theta(s', \epsilon)} \nabla_\theta f_\theta(s', \epsilon) + \nabla_\theta Q^\pi(s', a') \big|_{a' = f_\theta(s', \epsilon)} \right] \\ \implies \mathbf{\Gamma}^\pi(s, a) &= \gamma \mathbb{E}_{s', \epsilon} \left[\mathbf{g}_{REP}(s', a') + \mathbf{\Gamma}(s', a') \right], \end{aligned} \tag{36}$$

where the *immediate reparametrization gradient* is

$$\mathbf{g}_{REP}(s', a') = \nabla_{a'} Q^\pi(s', a') \big|_{a' = f_\theta(s', \epsilon)} \nabla_\theta f_\theta(s', \epsilon). \tag{37}$$

B.2. An Experiment with Continuous Action Space

We demonstrate the applicability of the Gradient Actor-Critic method on a simple control task with continuous state-action spaces. Our goal is two-fold, first to show that a simple heuristic allows for the use of neural network function approximation without an excessive computational cost, and the second to show that the proposed methodology can successfully solve a continuous control task. We perform this demonstration using the continuous action Cartpole environment with a scalar action $u \in [-1, 1]$ which applies a lateral force on the cart for a one second period. We cut off the episode after a maximum of 500 steps, then reinitialize the cart with a random velocity and the pole with a random pole angle and angular velocity.

To estimate the action-value function for the critic, we use a two hidden-layer neural network with tanh activations and 64 units per layer. We feed the observable state and action into the neural network and have two heads attached to the penultimate layer, one head for the standard critic and another for the gradient critic. The policy is likewise parameterized by a two hidden-layer neural network with tanh activations and 64 units per layer. We swept the hyperparameters for both the Gradient Actor-Critic and Actor-Critic baseline, selecting the maximizing hyperparameter setting using 30 random seeds. The swept hyperparameters are reported in the table below. We then reran each algorithm for 100 random seeds for the maximizing hyperparameter setting in order to minimize maximization bias.

In Figure 3 we show that both the Gradient Actor-Critic and Actor-Critic methods are able to successfully learn a near-optimal policy on this task, with the optimal return being 500. Although the Cartpole task is too simplistic to induce differences between these algorithms, it does highlight that the Gradient Actor-Critic method can easily solve a continuous control problem using neural network function approximation. Both methods incurred near-identical computational cost on this problem setting, taking on average three minutes per run using a modern desktop processor.

Hyperparameters for continuous control experiment:

Optimizer	ADAM$(\beta_1 = 0.9, \beta_2 = 0.999)$
Target network moving average	$\{0.99, 0.9\}$
Learning rate for the critic	$\{0.1, 0.01, 0.001, 0.0001\}$
Learning rate for the actor	$\{0.1, 0.01, 0.001, 0.0001\}$
Eligibility Trace	$\{0.9, 0.75, 0.5, 0.1\}$

Algorithm 3 LSTD$\mathbf{\Gamma}$

```
Input: Set of features $\phi$, dataset $D$ of transitions $(s_i, a_i, r_i, s_i', t_i)$ where $r_i$ are the rewards, $s_i'$ the next states and $t_i$ is the time-step, policy $\pi_\theta$, learning rate $\eta$
$\hat{\mathbf{b}} = 1/N \sum_i \phi(s_i, a_i) r_i$
while not converged do
    Fore each $s_i'$ sample $a_i' \sim \pi_\theta(\cdot|s_i')$
    $\hat{\mathbf{A}} = 1/N \sum_i \phi(s_i, a_i)\left((s_i, a_i) - \gamma\phi(s_i', a_i')\right)^\intercal$
    $\hat{Q}(s, a) = \boldsymbol{\phi}^\intercal(s, a)\hat{\boldsymbol{\omega}}_{TD}$; $\hat{\boldsymbol{\omega}}_{TD} = \hat{\mathbf{A}}^{-1}\hat{\mathbf{b}}$;
    $\hat{\mathbf{B}} = 1/N \sum_i \phi(s_i, a_i)\hat{Q}(s_i', a_i')\nabla_\theta \log \pi_\theta(a_i'|s_i')$
    $\hat{\mathbf{\Gamma}}(s, a) = \boldsymbol{\phi}^\intercal(s, a)\hat{\mathbf{G}}_{TD}$; $\hat{\mathbf{G}}_{TD} = \hat{\mathbf{A}}^{-1}\hat{\mathbf{B}}$
    Sample $s_i$ from dataset and $a_i \sim \pi_\theta(\cdot|s_i)$
    $\mathbf{g}_i = \hat{Q}(s_i, a_i)\nabla_\theta \log \pi_\theta(a_i|s_i)$
    $\theta \leftarrow \theta + \eta\lambda^{t_i}\gamma^{t_i}(\mathbf{g}_i + \hat{\mathbf{\Gamma}}(s_i, a_i))$
end while
```

Algorithm 4 LSTD$\mathbf{\Gamma}$ with Automatic Differentiation

```
Input: Set of features $\phi$, dataset $D$ of transitions $(s_i, a_i, r_i, s_i')$ where $r_i$ are the rewards and $s_i'$ the next states, policy $\pi_\theta$, learning rate $\eta$
$\hat{\mathbf{b}} = 1/N \sum_i \phi(s_i, a_i) r_i$
while not converged do
    Fore each $s_i'$ sample $a_i' \sim \pi_\theta(\cdot|s_i')$
    $\hat{\mathbf{A}} = 1/N \sum_i \phi(s_i, a_i)\left((s_i, a_i) - \gamma\phi(s_i', a_i')\right)^\intercal$
    $\hat{Q}(s, a) = \boldsymbol{\phi}^\intercal(s, a)\hat{\boldsymbol{\omega}}_{TD}$; $\hat{\boldsymbol{\omega}}_{TD} = \hat{\mathbf{A}}^{-1}\hat{\mathbf{b}}$;
    Sample $s_t$ from dataset and $a_i \sim \pi_\theta(\cdot|s_i)$
    $\mathbf{g}_i = \hat{Q}(s_i, a_i)\nabla_\theta \log \pi_\theta(a_i|s_i)$
    $\theta \leftarrow \theta + \eta\lambda^{t_i}\gamma^{t_i}\nabla_\theta\hat{Q}(s_i, a_i)$
end while
```

C. Supplement to the Empirical Analysis

This section introduces some notes on the practical implementation of the algorithms, environments, and hyperparameters and settings used in the experiments. We finally present some complementary results to the one presented in the main paper.

C.1. LSTD$\mathbf{\Gamma}$

Algorithm 2 illustrates "pure" LSTD$\mathbf{\Gamma}$ ($\lambda = 0$). This section discusses how to incorporate the eligibility traces in practice and how to write a simple 'pytorch' spinnet to compute the gradient.

Eligibility Trace. To implement eligibility traces, we need a dataset where for each transition $s\ a\ r\ s'$, we have also accompanied with t, a variable indicating the number of steps that occurred since the beginning of the current episode. Hence, we first fit the matrix $\mathbf{A}_\pi$, and we compute the parameter matrix $\mathbf{G}$ and then, we compute the gradient as in Equation 15. A schematic representation of LSTD$\mathbf{\Gamma}$ can be found in Algorithm 3.

Using Automatic Differentiation. One can actually avoid to compute $\mathbf{G}$. When we look bach to Lemma 2, we see that $\mathbf{G}_{TD} = \nabla_\theta \boldsymbol{\omega}_{TD}$. Automatic differentiation via `pytorch` is actually able to derive that step automatically. Therefore, instead of computing $\mathbf{G}$ explicitly in the code, one can simply compute $\boldsymbol{\omega}_{TD}$ and let the automatic differentiation tto find $\nabla_\theta \boldsymbol{\omega}_{TD}$, as in Algorithm 4.

C.2. TDRC$\mathbf{\Gamma}$.

This algorithm, described in Algorithm 1, uses TDRC to estimate both critic and gradient critic. To do so, we simply replace the semi-gradient TD update rule with the following TDRC update

$$\delta_t = R_t + \gamma\hat{Q}_t^\pi(S_{t+1}, A_{t+1}) - \hat{Q}_t^\pi(S_t, A_t)$$

$$\begin{aligned}\boldsymbol{\chi}_{t+1} &= \boldsymbol{\chi}_t + \alpha_t \boldsymbol{\phi}_t \left(\delta_t - \boldsymbol{\phi}_t^\intercal \boldsymbol{\chi}_t\right) - \alpha_t \beta \boldsymbol{\chi}_t \\ \boldsymbol{\omega}_t &= \boldsymbol{\omega}_t + \alpha_t \boldsymbol{\phi} \delta_t - \alpha \gamma \boldsymbol{\phi}_t' \boldsymbol{\phi}_t^\intercal \boldsymbol{\chi}_t,\end{aligned} \tag{38}$$

where χ_t are a secondary set of weights to perform gradient correction and β is the TDRC regularization factor.

Similarly, we can estimate the gradient critic with a vector form of TDRC,

$$\begin{aligned}\boldsymbol{\varepsilon}_t &= \gamma \hat{Q}_t^\pi(S_{t+1}, A_{t+1}) \nabla_\theta \log \pi_\theta(A_{t+1}|S_{t+1}) \\ &\quad + \gamma \hat{\boldsymbol{\Gamma}}_t^\pi(S_{t+1}, A_{t+1}) - \hat{\boldsymbol{\Gamma}}_t^\pi(S_t, A_t) \\ \mathbf{H}_{t+1} &= \mathbf{H}_t + \alpha_t \boldsymbol{\phi} \left(\boldsymbol{\varepsilon}_t^\intercal - \boldsymbol{\phi}_t^\intercal \mathbf{H}_t\right) - \alpha_t \beta \mathbf{H}_t \\ \mathbf{G}_{t+1} &= \mathbf{G}_t + \alpha_t \boldsymbol{\phi} \boldsymbol{\varepsilon}_t^\intercal - \alpha \gamma \boldsymbol{\phi}' \boldsymbol{\phi}^\intercal \mathbf{H}_t,\end{aligned} \tag{39}$$

where $\hat{\boldsymbol{\Gamma}}_t^\pi(S, A) = \boldsymbol{\phi}^\intercal(S, A)\mathbf{G}_t$ and $\hat{Q}$ is an estimate of the critic. $\mathbf{H}_t$ have the same role as $\boldsymbol{\chi}_t$ in (38). The samples $S_t, A_t, S_{t+1}, A_{t+1}$ are sampled i.i.d. according to ζ.

Because the critic and gradient critic estimations have no circular dependencies, we can easily prove convergence of the gradient critic to $\hat{\boldsymbol{\Gamma}}_{TD}^\pi$ by simply allowing TRDC to first converge to $\hat{Q}_{TD}^\pi$ and subsequently iterating (39) using $\hat{Q}_{TD}^\pi$, converging therefore to $\hat{\boldsymbol{\Gamma}}_{TD}^\pi$. However, such an approach is not practical. To obtain faster convergence, we propose to interleave both the updates in (38), (39), and of the target polcy. We call this algorithm TDRC$\boldsymbol{\Gamma}$ (Algorithm 1).

To have as few hyperparameters as possible, we set the same learning rate for both the critic and the gradient critic. Across all the experiments, we use $\beta = 1$. To be precise, in `Imani's MDP`, one could avoid using a full-gradient TD technique (like TDC, GTC, ...) since the critic features are perfect. However, we preferred to maintain consistency between different experiments.

C.3. Imani's MDP.

There are a few choices that can be made to implement this MDP. We opted to implement this MDP as a four-state MDP where the terminal state is absorbing. We did this because our current code computes the policy gradient in closed form without knowing terminal states. This modification is not an issue. Making T an absorbing state changes the discounted stationary distribution, leaving the ratio of visitation between S_0, S_1, and S_2 unchanged, which is, after all, what matters. Furthermore, the gradient on the absorbing state is always 0.

To allow generality, our policy, therefore, accepts the input of 4 different states, and, since the possible actions per state are two, the tabular policy is encoded with 8 parameters. In the presence of state-aliasing, however, when the MDP is in state S_2, state S_1 is fed to the policy instead. For this reason, from the policy perspective, state S_2 is never visited, causing the gradient of the parameters that correspond to state S_2 to be always zero.

These implementation choices do not change the math and the effects of the original MDP of Imani's et al.

The parameters that matter are $\theta_0, \theta_1, \theta_2$ and θ_3, corresponding to state S_0 and S_1 (which is aliased with S_2).

C.4. Bias-Variance Tradeoff in Figure 1b, c, and d

`Imani's MDP` has a closed-form solution of the policy gradient. We use this solution to compute the bias of the estimators. While the experiment's setting has been already described in the paper, here we provide fewer details on how the bias and the variance have been estimated.

We build both variance and bias estimates for each value of λ by sampling 20 instances of the estimators (e.g., running 20 times the algorithm to estimate the gradient). After, we compute the squared bias and the variance per component, i.e.,

$$\hat{\mathbf{b}} = \left(\frac{1}{20}\sum_{i=1}^{20}\left(\hat{\mathbf{g}}_i - \nabla_\theta J(\theta)\right)\right)^2; \quad \hat{\mathbf{v}} = \frac{1}{20}\sum_{i=1}^{20}\left(\hat{\mathbf{g}}_i - \overline{\mathbf{g}}\right)^2,$$

where $\mathbf{g}_i$ are the single estimates of the gradient, $\nabla_\theta J(\theta)$ the true gradient, and $\overline{\mathbf{g}}$ the empirical average of the gradient estimate. The vectors of empirical bias $\mathbf{b}$ and variance $\mathbf{v}$ are then transformed to scalars by taking the mean over the components, i.e.,

$$\hat{b} = \frac{1}{8}\sum_{i}^{8} \hat{\mathbf{g}}_i; \quad \hat{v} = \frac{1}{8}\sum_{i}^{8} \hat{\mathbf{v}}_i.$$

Now, $\hat{b}$ and $\hat{v}$ are also estimates. Therefore, we repeat this process 50 times to compute an empirical average of the estimates and build confidence intervals. Therefore, for each value of λ, we compute 1000 estimates of the gradient. We show the estimate both on a circular plot, which shows the ground truth and the single estimates compactly, and we also report the single estimates of $\{\partial/\partial\theta_i\}_{i=0}^{3}$ (since the remaining partial derivatives are all equal to zero).

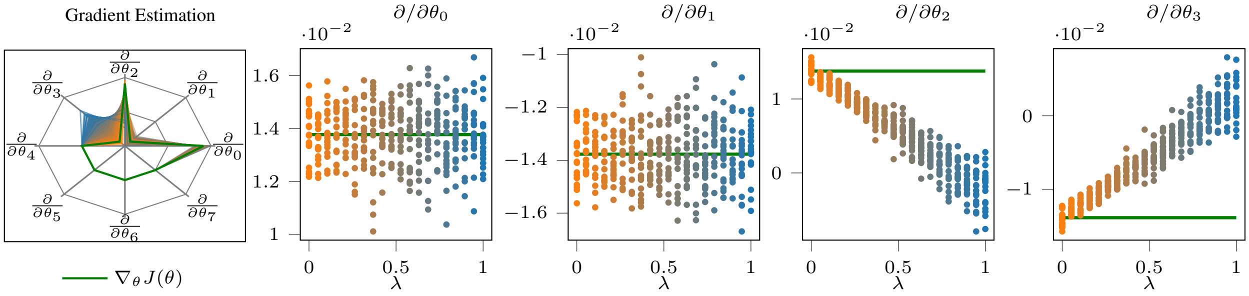

Figure 4. On the left, a plot with all the gradient estimates. In orange color, we have low values of λ (hence, full use of the gradient critic); in blue color, we have the semi-gradient estimator. We denote in green the ground truth. In the plots on the left, we show the actual gradient estimates for the first four parameters. Parameters θ_2 and θ_3 are critical, as they are subject to state aliasing. The gradient critic delivers an unbiased estimate, while semi-gradient exhibits high bias.

C.5. LSTDΓ - Figure 2a

The performance of LSTDΓ on `Imani's MDP` has been shown on Figures 2a. In this experiments, we sampled a dataset of 500 using the behavioral policy, and we applied LSTDΓ for 1000 steps. More in particular, the estimated gradient has been used with Adam (with learning rage 0.01). At each step, the return of the target policy is computed in closed form. We inspect 20 values of λ in the interval $[0, 1]$, performing 10 different indipendent runs of the algorithm to appreciate confidence intervals at 95%. Since most values of λ tend to have similar return, we defided both to show the final performance (at the 1000th iteration), and a few learning curves. We also computed the learning curve of pure semi-gradient and pure LSTD**Λ** using the gradients in closed-form. Figure 5 depicts the learning curve obtained for a fewer values of λ.

C.6. TDRCΓ - Figure 2c

This figure has been produced by running TDRC with parameters $\beta = 1$, $\alpha = 0.1$, and Adam with a learning rate 0.001 for the actor update. Surprisingly, the curve is almost identical to the one obtained in Figure 2b. The confidence intervals have been obtained by running 20 instances for each value of λ. Moreover, in Figure 5, we show the performance at the last iteration step for 20 values of λ in the range $[0, 1]$. Interestingly, λ behaves similarly to LSTD, as in Figure 2a.

OffPAC:

Learning rate for the critic	0.1
Learning rate for the actor	0.001
GDT Regulariztion	0.1
Eligibility Trace	0.1

ACE($\eta = 1$):

Learning rate for the critic	0.1
Learning rate for the actor	0.001
Entropic Regularization	0
GDT Regulariztion	0.1
Eligibility Trace	0.1

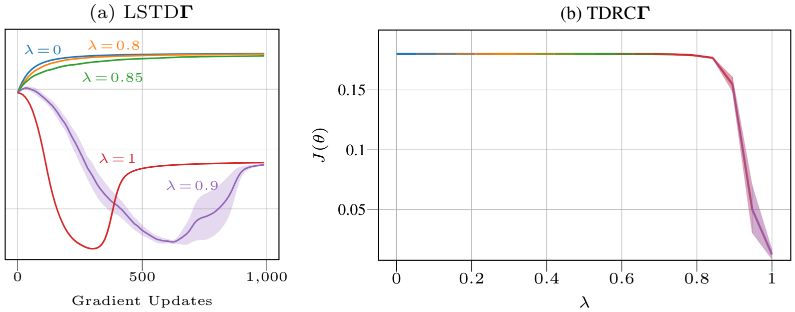

Figure 5. (a) Learningt curves of LSTD$\boldsymbol{\Gamma}$ on Imani's MDP (b) TDRC$\boldsymbol{\Gamma}$ applied on `Imani's MDP` and evaluated at the last training step. The hyper-parameter λ behaves similarly to LSTD$\boldsymbol{\Gamma}$ in Figure 2a .

Note that we estimate the critic using GDT both for OffPAC and ACE.

TDRC$\boldsymbol{\Gamma}$:

Learning rate for the value critic	0.1
Learning rate for the actor critic	0.1
Learning rate for the actor	0.001
TDRC Regulariztion	1.0

C.7. Experiments on Random MDPs- Figure 2c and d

To enerate the transition and the reward model, we first sample a uniform vector, and then we feed it in a soft-max function

$$\mathrm{SoftMax}(\mathbf{x})_i = \frac{\exp T x_i}{\sum_j \exp T x_j}, \tag{40}$$

where the temperature T controls the entropy of the overall distribution. With high T we tend to have sparse reward and deterministic transition, while with low T, uniform transitions and reward model. In our experiments, where we use 30 states and 2 action, a temperature $T = 10$ seems to be a good balance to generate interesting models. As explained in the main paper, when interacting with the MDP, the agent observes the reward with an addition of a Gaussian noise with standard deviation of 0.1.

In the following, we describe the setting used for this experiment.

OffPAC:

Learning rate for the critic	$(\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert)^{-1}$
Learning rate for the actor	$10^{-2}(\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert)^{-1}$
GDT Regulariztion	0.1
Eligibility Trace	0

ACE($\eta = 1$):

Learning rate for the critic	$(\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert)^{-1}$
Learning rate for the actor	$10^{-2}(\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert)^{-1}$
Entropic Regularization	0
GDT Regularization	0.1
Eligibility Trace	0

Note that we estimate the critic using GDT both for OffPAC and ACE.

TDRC$\boldsymbol{\Gamma}$:

Learning rate for the value critic	$(\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert)^{-1}$
Learning rate for the actor critic	$(\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert)^{-1}$
Learning rate for the actor	$10^{-2}(\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert)^{-1}$
TDRC Regulariztion	1.0