Closing the Gap: Achieving Global Convergence (Last Iterate) of Actor-Critic under Markovian Sampling with Neural Network Parametrization

Mudit Gaur [1] **Amrit Singh Bedi** [2] **Di Wang** [3] **Vaneet Aggarwal** [4]

Abstract

The current state-of-the-art theoretical analysis of Actor-Critic (AC) algorithms significantly lags in addressing the practical aspects of AC implementations. This crucial gap needs bridging to bring the analysis in line with practical implementations of AC. To address this, we advocate for considering the MMCLG criteria: **M**ulti-layer neural network parametrization for actor/critic, **M**arkovian sampling, **C**ontinuous state-action spaces, the performance of the **L**ast iterate, and **G**lobal optimality. These aspects are practically significant and have been largely overlooked in existing theoretical analyses of AC algorithms. In this work, we address these gaps by providing the first comprehensive theoretical analysis of AC algorithms that encompasses all five crucial practical aspects (covers MMCLG criteria). We establish global convergence sample complexity bounds of $\tilde{\mathcal{O}}\left(\epsilon^{-3}\right)$. We achieve this result through our novel use of the weak gradient domination property of MDP's and our unique analysis of the error in critic estimation.

1. Introduction

Actor-Critic (AC) algorithms (Konda & Tsitsiklis, 1999) have emerged as a cornerstone in modern reinforcement learning, showcasing remarkable versatility and effectiveness across a diverse range of applications such as games (Zhou et al., 2022), network scheduling (Agarwal et al., 2022), robotics (Chen et al., 2022), autonomous driving (Tang et al., 2022), and video streaming (Zhang et al., 2022). At their core, AC algorithms aim to maximize the expected returns, denoted by $J(\lambda)$, where $\lambda \in \mathbb{R}^d$ is the policy parameter and $J(\lambda)$ is the expected reward for a policy parameterised by λ. The algorithms involves an interplay between the gradient ascent for the actor parameter and the estimation of the action value function or critic. Theoretical analysis of AC algorithms, however, often lags behind their practical implementations. Due to the practical use of multi-layer neural network parameterizations for the actor as well as critic, an important observation in recent research is the widening gap between theoretical models and real-world applicability. True insights are gained when theoretical analysis mirrors practical complexities, even if it means accepting more conservative bounds for the sake of realism. Our focus in this work is to close this gap between theoretical analysis and practical implementations of AC algorithms.

To align closely with practical settings, it is essential that any theoretical analysis of Actor-Critic (AC) algorithms thoroughly considers five crucial aspects. These are (a) **M**ulti-layer neural network parametrization for actor/critic, (b) **M**arkovian sampling, (c) **C**ontinuous state-action spaces, (d) performance of the **L**ast iterate, and (e) **G**lobal optimality. The significance of the MMCLG criteria lies in its alignment with practical implementations: Deep neural networks are commonly used for actor-critic implementations (Lee et al., 2020), data in real-world scenarios is typically sampled in a Markovian fashion (Zhong et al., 2019), and most applications, like robotics, operate in continuous spaces (Dankwa & Zheng, 2019). Furthermore, in practice, the last iterate of the algorithms is used to evaluate performance (Wang & Hu, 2023). Finally for training neural networks, the global convergence truly matters, as local convergence can be misleading in evaluating the effectiveness of the trained network (Swirszcz et al., 2016).

The existing literature (cf. Table 1), while extensive, does not have any works that simultaneously address the above-mentioned five dimensions (MMCLG). There exist works which have achieved local convergence (which upper bound the quantity $\frac{1}{T}\sum_{i=1}^{T}||\nabla J(\lambda_t)||^2$) for finite state spaces for **M**ulti-layer settings with **M**arkovian sampling (Tian et al., 2023)]. Another set of results have

[1]Department of Statistics, Purdue University, West Lafayette, IN, U.S.A [2]Department of Computer Science, University of Central Florida [3]Department of Computer Science, KAUST [4]School of IE and School of ECE, Purdue University, West Lafayette, IN, U.S.A. Correspondence to: Mudit Gaur <mgaur@purdue.edu>, Vaneet Aggarwal <vaneet@purdue.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Table 1. This table summarizes the features of different actor-critic convergence results. Our result is the first to provide last iterate sample complexity results of AC for an MDP setting with multi layer neural network for the actor-critic, continuous state and action space, Markovian sampling and global optimality results on the last iterate performance.

References	Per. of **L**ast Iterate	**G**lobal Optimality	**C**ontinuous State Action Space	**M**ulti Layer NN AC	**M**arkovian Sampling	Sample Complexity
(Xu et al., 2020b)	✗	✓	✓	✗	✓	$\tilde{\mathcal{O}}(\epsilon^{-4})$
(Khodadadian et al., 2021)	✗	✓	✗	✗	✓	$\tilde{\mathcal{O}}(\epsilon^{-3})$
(Xu et al., 2020a)	✗	✓	✓	✗	✓	$\tilde{\mathcal{O}}(\epsilon^{-2})$
(Xu et al., 2021)	✗	✓	✓	✗	✓	$\tilde{\mathcal{O}}(\epsilon^{-4})$
(Wang et al., 2020)	✗	✓	✗	✗	✗	$\tilde{\mathcal{O}}(\epsilon^{-4})$
(Cayci et al., 2022)	✗	✓	✗	✗	✗	$\tilde{\mathcal{O}}(\epsilon^{-4})$
(Fu et al., 2021)	✗	✓	✗	✓	✗	$\tilde{\mathcal{O}}(\epsilon^{-6})$
(Tian et al., 2023)	✗	✗	✗	✓	✓	$\tilde{\mathcal{O}}(\epsilon^{-2})$
This work	✓	✓	✓	✓	✓	$\tilde{\mathcal{O}}(\epsilon^{-3})$

obtained **G**lobal optimality results known as average iterate complexity bounds or regret bounds. These establish an upper bound on the quantity $\frac{1}{T}\sum_{i=1}^{T}(J^* - J(\lambda_t))$ known as the regret. For a linear critic parametrization, such **G**lobal optimality bounds have been established in (Xu et al., 2020a) for **M**arkovian sampling. For a neural network actor and critic parameterization (Cayci et al., 2022), establishes **G**lobal optimality where the actor and critic neural networks have a single hidden layer, while (Fu et al., 2021) does so for a **M**ulti Layer neural network of arbitrary depth. Notably, none of the above works focus on the last iterate convergence aspect, which requires an upper bound on $J^* - J(\lambda_T)$. Also none of the works having a **M**ulti Layer neural network of arbitrary depth work for **C**ontinuous state-action spaces. Hence, in this work, we ask this question

Is it possible to develop a theoretical analysis of actor-critic algorithm that covers all MMCLG criteria in one analysis?

The above question essentially implies that can we obtain **G**lobal sample complexity bounds for the **L**ast iterate convergence of the actor-critic algorithm with a **M**ulti-layer neural network parametrization of the critic, without assuming i.i.d. sampling (under **M**arkovian) for **C**ontinuous spaces? We answer the above question in the affirmative in this work. Our main contributions as listed as follows.

- We establish an upper bound on the performance of the **L**ast Iterate $J^* - J(\lambda_T)$ in terms of the sum of the errors incurred in the estimation of the critic. This is done using our novel analysis combining the smoothness assumption on the policy parametrization and the weak gradient bound condition for MDP. This analysis is different from existing works such as (Fatkhullin et al., 2023), (Masiha et al., 2022), where an upper bound on the last iterate performance is obtained using an unbiased estimator of the policy gradient using sample trajectories. This is not available to us in actor critic where a parametric estimate of the critic is used and we have to account for the error incurred in critic estimation. Our analysis does not rely on the cardinality of the action space, unlike in existing multi layer critic analyses such as (Fu et al., 2021) thereby, our **G**lobal convergence bound works for **C**ontinous state-action spaces.

  In the analysis of critic error estimation, we derived a novel decomposition of error. The error is split into the error incurred due to the limited approximation ability of the class of function representing the critic and the error incurred due to the limited sample size to estimate the critic, as well as the error incurred in solving the critic estimate in a finite number of steps. This decomposition allows us to consider the **M**arkov dependence of samples and a **M**ulti-layer neural network parametrization of the critic, which is the first result that achieves this. This is in contrast to (Cayci et al., 2022; Fu et al., 2021), where i.i.d. sampling was assumed.

- We derive a last iterate convergence sample complexity bound of $\tilde{\mathcal{O}}(\epsilon^{-3})$ for the actor-critic algorithm with neural network parameterizations for the critic and the actor. True to our knowledge, this work is the first to present a last iterate global convergence result for an actor-critic algorithm with neural network critic parametrization. It is also the best sample complexity for global convergence in terms of ϵ for an actor-critic algorithm with a neural network parametrization of the critic.

2. Related Works

Policy Gradient: Policy gradient algorithms, first conceptualized in (Sutton et al., 1999) perform a gradient step on the parameters to obtain an estimate of the optimal policy. The estimate of the action value (or advantage) function

is obtained by following the current estimate of the policy and calculating the action value function from the obtained rewards. In such a case, sample complexity estimates are possible without the need to assume parametric form of the action value function as is done in (Agarwal et al., 2021). It obtained a sample complexity bound of $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^4}\right)$. Further improvements have been obtained in (Liu et al., 2020b; Mondal & Aggarwal, 2024), where the proposed algorithm in (Mondal & Aggarwal, 2024) achieves a sample complexity of $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2}\right)$. Note that these results were all average iterate convergence results. More recently, works such as (Fatkhullin et al., 2023; Masiha et al., 2022) have obtained last iterate convergence of policy gradient algorithms.

Actor Critic: Actor critic methods aim to combine the benefits of the policy gradient methods and Q-learning based methods. Local convergence results results for Actor Critic were obtained in (Castro & Meir, 2010) and (Maei, 2018). More recently average iterate convergence results actor critic using a linear critic have been obtained in (Xu et al., 2020b;a) with the best known sample complexity of $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2}\right)$ in (Xu et al., 2020a). Average iterate convergence for Actor Critic where neural networks are used to represent the actor and critic are obtained in works such as (Gaur et al., 2023b; Wang et al., 2020) which obtain sample complexities of $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^4}\right)$ and $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^6}\right)$ respectively . More recently works such as (Tian et al., 2023) obtain local convergence results with a sample complexity of $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2}\right)$, with a finite state and action space, limiting the practicality of the algorithm.

Last Iterate Convergence: In optimization literature, the strongest convergence bound for an algorithm that can be obtained is known as a last iterate convergence bound. It can be written as

$$f(\lambda^*) - f(\lambda_t) \leq \mathcal{O}(h(t)). \tag{1}$$

Here, f is the objective function of interest to be maximized. λ_t is the parameter obtained at the tth iteration of the algorithm and λ^* is the optimal parameter corresponding to the highest possible value of the objective function and h is some function of the number of iteration and possibly the sample size needed at each iteration. Such results were typically proven for gradient descent-type algorithms for convex objective functions (Boyd & Vandenberghe, 2004; Balkanski & Singer, 2017). In many modern machine learning applications, the objective function of interest is non-convex. This means many convergence results only demonstrate a local convergence, which can be written as $\frac{1}{t}\sum_{i=1}^{t}||\nabla f(\lambda_i)||^2 \leq \mathcal{O}(\tilde{h}(t))$, which only guarantees that the algorithm will converge to a local optimum. Example of such results are in works such as (Li & Orabona, 2019; Chen et al., 2023) and (Tian et al., 2023).

For value iteration based methods such as Q learning and fitted Q iteration, last iterate convergence results with neural network parametrization have been obtained in works such as (Cai et al., 2019) and (Gaur et al., 2023a) respectively.

In policy iteration methods such as gradient methods, with additional 'compatible function approximation' assumptions (Sutton et al., 1999), an upper bound is established on the *regret* defined as

$$\frac{1}{t}\sum_{i=1}^{t}(f(\lambda^*) - f(x_i)) \leq \mathcal{O}(\tilde{h}(t)) \tag{2}$$

Such a result is the only type of global convergence shown so far for actor critic methods.

In order to establish an upper bound on $f(\lambda^*) - f(\lambda_t)$ for non-convex $f(x)$, (Polyak, 1963) established the notion of *weak gradient bound* defined as

$$\mu \cdot ||\nabla f(x)||^{\alpha} \leq f(\lambda^*) - f(\lambda) \tag{3}$$

where μ is a positive constant that depends on the function f and $\alpha \in [1, 2]$. We note that using this condition, last iterate convergence results have been demonstrated for non-convex optimization in works such as (Yue et al., 2023) and (Doan, 2022). For an MDP setup, under standard assumptions, a PL like condition was established in (Ding et al., 2022). This has been used in policy gradient works such as (Masiha et al., 2022; Fatkhullin et al., 2023) to establish last iterate convergence.

I.I.D. vs Markov Sampling in AC: Prior analyses of AC algorithms with neural network actor and critic algorithms such as (Wang et al., 2020; Cayci et al., 2022; Fu et al., 2021) all rely on *local linearization* of the neural networks. These techniques assume that the samples are drawn independently from the stationary distribution of a fixed policy. However, as is shown in (Mnih et al., 2013), Q-learning algorithms with deep neural networks require the use of *target networks* and *experience replay* to converge. (Lillicrap et al., 2015) showed the same is the case for actor critic algorithms particularly for cases with continuous action spaces. Thus, the sampling approach (which is iid) in existing analyses for AC with neural network parametrizations is not applicable in practice, and we need convergence analysis under Markovian sampling to better reflect the practical cases in theory.

3. Problem Formulation

We consider a discounted Markov Decision Process (MDP) given by the tuple $\mathcal{M} := (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is a bounded measurable state space, $\mathcal{A}$ is the set of actions which is also a bounded measurable space. Note that for our setup, both the state and action space can be infinite. $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the probability transition function,

$r : \mathcal{S} \times \mathcal{A} \to ([0, 1])$ is the reward function on the state action space and $0 < \gamma < 1$ is the discount factor. A policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ maps a state to a probability distribution over the action space. The action value function for a given policy π is given by

$$Q^\pi(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)|s_0 = s, a_0 = a\right], \quad (4)$$

where $a_t \sim \pi(\cdot|s_t)$ and $s_{t+1} \sim P(\cdot|s_t, a_t)$ for $t = \{0, \cdots, \infty\}$. For a discounted MDP, we define the optimal action value functions as

$$Q^*(s, a) = \sup_\pi Q^\pi(s, a), \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}. \quad (5)$$

A policy that achieves the optimal action-value functions is known as the *optimal policy* and is denoted as π^*. We define $\rho^\pi_\nu(s)$ as the stationary state distribution induced by the policy π starting at state distribution ν and $\zeta^\pi_\nu(s, a)$ is the corresponding stationary state action distribution defined as $\zeta^\pi_\nu(s, a) = \rho^\pi_\nu(s) \cdot \pi(a|s)$. We can define the state action visitation distribution as $d^\pi_\nu(s, a) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t Pr^\pi(s_t = s, a_t = a|(s_0, a_0) \sim \nu)$, where $Pr^\pi(s_t = s, a_t = a|(s_0, a_0) \sim \nu)$ denotes the probability that the state action pair at time t is (s, a) when following the policy π with starting state action distribution of ν.

We additionally define the Bellman operator for a policy π on a function $Q : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \times \mathcal{A}$ is defined as

$$\begin{aligned}(T^\pi Q)(s, a) &= r(s, a) \\ &\quad + \gamma \int Q(s', \pi(s')) P(ds'|s, a) \quad (6)\end{aligned}$$

Further, operator P^π is defined as

$$P^\pi Q(s, a) = \mathbb{E}[Q(s', a')|s' \sim P(\cdot|s, a), a' \sim \pi(\cdot|s')] \quad (7)$$

This is the one step Markov transition operator for policy π for the Markov chain defined on $\mathcal{S} \times \mathcal{A}$ with the transition dynamics given by $S_{t+1} \sim P(\cdot|S_t, A_t)$ and $A_{t+1} \sim \pi(\cdot|S_{t+1})$. It defines a distribution on the state action space after one transition from the initial state. Similarly, $P^{\pi_t} P^{\pi_{t-1}} \cdots P^{\pi_1}$ is the m-step Markov transition operator following policy π_t at steps $1 \leq t \leq m$.

Neural Network Parametrization. We define a neural network of D layers with m neurons per layer as follows

$$y = \frac{1}{\sqrt{m}} \cdot \sigma(b_D^T x_{D-1}) \quad (8)$$

$$x^h = \frac{1}{\sqrt{m}} \cdot \sigma(W_h^T x_{h-1}), \quad (9)$$

$$h \in \{1, \cdots, D-1\}, \quad (10)$$

where m is the number of neurons in each neural network layer, D represents the number of layers in the neural network, W_h represents the weight matrix for the hth layer, where b_D represents the weight vectors of the final. Also, x_0 is the input to the neural network, σ is an activation function, typically a function such as sigmoid or ReLU may be used here. We denote the neural network representing the critic corresponding to a set of parameters $\theta = \{W_1, \cdots, W_{D-1}, b_D\}$ for a given state action pair as (s, a) as $Q_\theta(s, a)$. For notational convenience we denote the input to this neural networks as $(s, a) \in \mathcal{S} \times \mathcal{A}$. For the actor-network, we use a Gaussian policy given by $\pi_\lambda(s) = \mathcal{N}(\mu_{\lambda_1}(s), \kappa_{\lambda_2}(s))$. Here $\mu_{\lambda_1}(s)$ and $\kappa_{\lambda_2}(s)$ represent the mean and variance of a normal distribution represented by neural networks with the state $s \in \mathcal{S}$ as an input. We denote the set of parameters as $\lambda = \{\lambda_1, \lambda_2\}$. The activation functions for the actor network are required to be smooth. Both these networks has the structure defined in equations (8) to (10). Actor and critic networks sharing the same architecture is implemented in practical Actor Critic implementations like (Liu et al., 2020a).

4. Actor Critic Algorithm

In an actor-critic algorithm (Konda & Tsitsiklis, 1999), the aim is to maximize the expected return given by

$$J(\lambda) = \mathbb{E}_{s \sim \nu, a \sim \pi_\lambda(.|s)} Q^{\pi_\lambda}(s, a) \quad (11)$$

A policy gradient step is performed to update the policy parameters of the actor. For our setup, the policy is parameterized as $\{\pi_\lambda, \lambda \in \Lambda\}$ and $\Lambda \subset \mathbb{R}^d$ where d is a positive integer. We have K total iterations of the Algorithm. At iteration k, the policy parameters are updated using a natural policy gradient step given by

$$\lambda_{k+1} = \lambda_k + \alpha_k \nabla_\lambda J(\lambda_k), \quad (12)$$

From the policy gradient theorem in (Sutton et al., 1999) we have

$$\nabla_\lambda J(\lambda_k) = \mathbb{E}_{s,a}(\nabla_\lambda \log(\pi_{\lambda_k}(a|s)) Q^{\pi_{\lambda_k}}(s, a)), \quad (13)$$

where $(s, a) \sim d^{\pi_{\lambda_k}}_\nu$. This policy update requires us to calculate the Q function for the current estimate of the optimal policy. Thus, we maintain a parameterized estimate of the Q-function, which is updated at each step and is used to approximate $Q^{\pi_{\lambda_k}}$. An estimate of its parameters is obtained by solving an optimization of the form

$$\arg\min_{\theta \in \Theta} \mathbb{E}_{s,a}(Q^{\pi_{\lambda_k}} - Q_\theta)^2, \quad (14)$$

where $(s, a) \sim d^{\pi_{\lambda_k}}_\nu$, Θ is the space of parameters for the critic neural networks, and Q_θ is the neural network corresponding to the parameter θ. This step is known as the critic step.

Algorithm 1 Actor Critic with Neural Parametrization

Input: $\mathcal{S}, \mathcal{A}, \gamma$, Time Horizon $K \in \mathcal{Z}$, sample batch size $n \in \mathcal{Z}$, resample batch size $L \in \mathcal{Z}$, Updates per time step $J \in \mathcal{Z}$,starting state sampling distribution ν, Actor step size α, Critic step size $\beta^{'}$, starting actor parameter λ_1,

1: **for** $k \in \{1, \cdots, K\}$ **do**
2: Sample n tuples $(s, a, r, s^{'})$ by following the policy π^{λ_k} from a starting state distribution ν and store the tuples.
3: **for** $j \in \{1, \cdots, J\}$ **do**
4: Initialize θ_0 using a standard Gaussian for all elements of weight matrices and b_D from Unif$(-1, 1)$ for the bias vector.
5: **for** $i \in \{1, \cdots, L\}$ **do**
6: Sample a tuple $(s_i, a_i, r_i, s_i^{'})$ with equal probability from the stored dataset
7: Sample $a_i^{'}$ using $\pi^{\lambda_k}(.|s_i^{'})$
8: Set $y_i = r_i + \gamma Q_{k,j-1}(s_i^{'}, a_i^{'})$,
9: $\theta_i^{'} = \theta_{i-1} + \beta^{'}(y_i - Q_{\theta_i}(s_i, a_i))\nabla Q_{\theta_i}(s_i, a_i)$
10: $\theta_i = \Gamma_{\theta_0, \frac{1}{(1-\gamma)}}\left(\theta_i^{'}\right)$
11: **end for**
12: $Q_{k,j} = Q_{\theta'}$ where $\theta^{'} = \frac{1}{L}\sum_{i=1}^{L}(\theta_i)$
13: **end for**
14: $d_k = \frac{1}{n}\sum_{i=1}^{n} \nabla \log(\pi(a_i|s_i))Q_{k,J}(s_i, a_i)$
15: Update $\lambda_{k+1} = \lambda_k + \left(\frac{\alpha}{k}\right)\frac{d_k}{||d_k||}$
16: **end for**

Output: $\pi_{\lambda_{K+1}}$

We summarize the Actor-Critic approach in Algorithm 1. It has one outer for loop indexed by the iteration counter k. The first inner for loop indexed by j is the loop where the critic step is performed. At a fixed iteration k of the main for loop and iteration j of the first inner for loop, we solve the following optimization problem

$$\arg\min_{\theta \in \Theta} \mathbb{E}_{s,a}(T^{\pi_{\lambda_k}}Q_{k,j-1}(s, a) - Q_\theta(s, a))^2, \qquad (15)$$

Where $(s, a) \sim d_\nu^{\pi_{\lambda_k}}$. This is the *target network* technique. For the inner loop at iteration j, the target is $T^{\pi_{\lambda_k}}Q_{k,j-1}(s, a)$. The first inner for loop has a nested inner for loop indexed by i where the optimization step for the current target is performed. Note that in line 10 $\Gamma_{\theta_0,(1-\gamma)^{-1}}$ represents the projection operator on the ball of radius $(1-\gamma)^{-1}$ centered on θ_0 in the space Θ. Formally this set is defined as $\Theta^{'} = \{\theta \in \Theta : ||W_h - W_h^0|| \leq (1-\gamma)^{-1}, \forall h \in \{1, \cdots, D-1\}\}$.

Here the tuples $(s_i, a_i, r_i, s_i^{'})$ are sampled randomly from the stored dataset and the optimization is performed using this sampled data. This random sampling on line 6 is the *experience replay* technique. The target network is updated in the second inner for loop indexed by i. The inner loop indexed by j controls how many times the target network is updated.

The algorithm laid out here is a simplified version of the Twin Delayed Deep Deterministic Policy Gradient algorithm (Dankwa & Zheng, 2020).

5. Theoretical Analysis: Global Convergence

5.1. Assumptions

Before stating the main result, we formally describe the required assumptions in this subsection.

Assumption 1. *For any $\lambda, \lambda_1, \lambda_2 \in \Lambda$ and $(s, a) \in (\mathcal{S} \times \mathcal{A})$ we have*

(i) $\|\nabla log(\pi_{\lambda_1})(a|s) - \nabla log(\pi_{\lambda_2})(a|s)\| \leq \beta\|\lambda_1 - \lambda_2\|$,

(ii) $\|\nabla log(\pi_{\lambda_1})(a|s)\| \leq M_g$,

(iii) $\mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_1}}}(\nabla \log \pi_{\lambda_1}(a|s))(\nabla \log \pi_{\lambda_1}(a|s))^T \succcurlyeq \mu_f I_d$

where $\beta, M_g, \mu_f \geq 0$.

Such assumptions have been utilized in prior policy gradient based works such as (Masiha et al., 2022; Fatkhullin et al., 2023) and actor critic using linear critic such as (Xu et al., 2020a). and global convergence results for neural critic parameterization such as (Fu et al., 2021; Wang et al., 2020), which restrict their analysis to energy-based policies for finite action spaces. Works such as (Cayci et al., 2022) only considers soft-max policies which are a simplification of energy based policies. However, as (Fatkhullin et al., 2023) points out, soft-max and energy based policies do not always satisfy (iii), while Gaussian policies are shown to satisfy this assumption for sufficiently deep neural networks.

Assumption 2. *For any $\lambda \in \Lambda$, let π_λ be the corresponding policy, ν be the starting distribution over the state action space, and let $\zeta_\nu^{\pi_\lambda}$ be the corresponding stationary state action distribution. We assume that there exists a positive integer p such that for every positive integer τ and for any set $\cdot \in \mathcal{S} \times \mathcal{A}$ and any $(s) \in \mathcal{S}$*

$$d_{TV}\left(\mathbb{P}((s_\tau, a_\tau) \in \cdot|(s_0) = (s)), \zeta_\nu^{\pi_\lambda}(\cdot)\right) \leq p\rho^\tau,$$

This assumption implies that the Markov chain is geometrically mixing. Such assumption is widely used both in the analysis of stochastic gradient descent literature such as (Doan, 2022; Sun et al., 2018), as well as finite time analysis of RL algorithms such as (Xu et al., 2020a). In (Fu et al., 2021), it is assumed that data can be sampled

from the stationary distribution of a given policy. Instead, in practice we can only sample from a Markov chain which has a stationary distribution as the desired distribution to sample from.

Assumption 3. *For any fixed $\lambda \in \Lambda$ we have*

$$\mathbb{E}\left(A^{\pi_\lambda}(s,a) - (1-\gamma)w^*(\lambda)^\top \nabla \log(\pi_\lambda)(a|s)\right)^2 \leq \epsilon_{bias}$$

Here, the expectation is over $(s,a) \sim d_\nu^{\pi^*}$ where π^* is the optimal policy. We also have $w^*(\lambda) = F(\lambda)^\dagger \nabla J(\lambda)$ where $F(\lambda) = \mathbb{E}_{(s,a)\sim d_\nu^{\pi_\lambda}}(\nabla \log \pi_\lambda(a|s))(\nabla \log \pi_\lambda(a|s))^T$.

(Wang et al., 2020) proves that this error goes to zero if both the actor and critic are represented by overparametrised neural networks. This assumption allows us to establish the weak gradient bound property for our MDP setup. It is used in policy gradient works such as (Yuan et al., 2022; Masiha et al., 2022; Fatkhullin et al., 2023) to establish last iterate convergence.

Assumption 4. *For any fixed $\lambda \in \Lambda$ we have*

$$\min_{\theta_1 \in \Theta'} \mathbb{E}_{s,a\sim\zeta_\nu^{\pi_{\lambda_k}}} \left(Q_{\theta_1}(s,a) - T^{\pi_\lambda} Q_\theta(s,a)\right)^2 \leq \epsilon_{approx}$$

This assumption ensures that a class of neural networks are able to approximate the function obtained by applying the Bellman operator to a neural network of the same class. Similar assumptions are taken in (Fu et al., 2021; Wang et al., 2020). In works such as (Cayci et al., 2022), stronger assumptions are made wherein the function class used for critic parametrization is assumed to be able to approximate any smooth function.

Before we move on to the main result, we want to state the key lemma proved in (Ding et al., 2022), that is used to obtain the last iterate convergence.

Lemma 1. *If Assumptions 1 and 3 hold then for any fixed $\lambda \in \Lambda$ we have*

$$\sqrt{\mu}(J(\lambda^*) - J(\lambda)) \leq \epsilon^{'} + \|\nabla J(\lambda)\|,$$

where $\epsilon^{'} = \frac{\mu_f \sqrt{\epsilon_{bias}}}{M_g(1-\gamma)}$ and $\mu = \frac{\mu_f^2}{2M_g^2}$

5.2. Main Result

Next, we present the main result.

Theorem 1. *Suppose Assumptions 1-4 hold and we have $\alpha = \frac{7}{2\sqrt{\mu}}$ and $\beta^{'} = \frac{1}{\sqrt{L}}$ then from Algorithm 1 we obtain*

$$\begin{aligned} J(\lambda^*) - J(\lambda_t) \leq & \tilde{\mathcal{O}}\left(\frac{1}{t}\right) + \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{n}}\right) \\ & + \tilde{\mathcal{O}}\left(\frac{1}{L^{\frac{1}{4}}}\right) + \tilde{\mathcal{O}}(m^{-\frac{1}{12}} D^{\frac{7}{2}}) \\ & + \tilde{\mathcal{O}}(\gamma^J) + \tilde{\mathcal{O}}(\sqrt{\epsilon_{bias}}) \\ & + \tilde{\mathcal{O}}(\sqrt{\epsilon_{approx}}). \end{aligned}$$

Hence, for $t = \tilde{\mathcal{O}}(\epsilon^{-1})$, $J = \tilde{\mathcal{O}}\left(\log\left(\frac{1}{\epsilon}\right)\right)$, $n = \tilde{\mathcal{O}}\left(\epsilon^{-2}\right)$, $L = \tilde{\mathcal{O}}\left(\epsilon^{-4}\right)$ we have

$$\begin{aligned} J(\lambda^*) - J(\lambda_t) \leq & \epsilon + \tilde{\mathcal{O}}(m^{-\frac{1}{12}} D^{\frac{7}{2}}) \\ & + \tilde{\mathcal{O}}\left(\sqrt{\epsilon_{bias}}\right) + \tilde{\mathcal{O}}\left(\sqrt{\epsilon_{approx}}\right), \end{aligned}$$

which implies a sample complexity of $t \cdot n = \tilde{\mathcal{O}}\left(\epsilon^{-3}\right)$.

The fourth term is the consequence of the finite size of the critic neural network. Such terms are present in other results where multi layer neural network parametrizations are used such as (Fu et al., 2021) and (Tian et al., 2023). We can see that as the width of the neural network tends to infinity these terms go to zero. This is in keeping with Neural Tanget Kernet (NTK) theory (Jacot et al., 2018), which states that in the infinite width limit neural networks converge to linear functions.

6. Proof Sketch of Theorem 1

The proof is split into two stages. In the first stage, we show how to obtain the last iterate performance gap as a function of the errors incurred in estimating the critic at each step. The second part decomposes the critic estimation error into its constituent components, which are bounded separately.

Upper Bounding Last Iterate Performance Gap: Under Assumption 1, (Yuan et al., 2022) proves that the expected return is a smooth function. Thus, we have

$$\begin{aligned} -J(\lambda_{t+1}) \geq & -J(\lambda_t) - \langle \nabla J(\lambda_t), \lambda_{t+1} - \lambda_t \rangle \\ & + L_J ||\lambda_{t+1} - \lambda_t||^2. \end{aligned} \tag{16}$$

Here, L_J is the smoothness parameter of the expected return and λ_t denotes the critic parameters ate iteration t of Algorithm 1 . From this, using Assumption 2, the weak gradient domination property proved in (Ding et al., 2022), we obtain the following

$$\begin{aligned} J^* - J(\lambda_{t+1}) \leq & \left(1 - \frac{\eta_t \sqrt{\mu'}}{3}\right)(J^* - J(\lambda_t)) \\ & + \frac{\alpha}{t} ||\nabla J(\lambda_t) - d_t|| \\ & + L_J ||\lambda_{t+1} - \lambda_t||^2 + \frac{\alpha}{t}\epsilon^{'}. \end{aligned} \tag{17}$$

The key step now is recursively applying this condition and and substitute the result back in Equation (17). We thus obtain the following result.

$$J(\lambda^*) - J(\lambda_t) \leq \frac{\alpha.M_g}{t} \sum_{k=0}^{k=t-2} (\mathbb{E}|Q^{\lambda_k}(s,a) - Q_{k,J}(s,a)|) + \left(\frac{1}{t}\right)(J(\lambda^*) - J(\lambda_2)) + \frac{L_J \alpha^2}{t} + \mathcal{O}(\sqrt{\epsilon_{bias}}). \tag{18}$$

Here, we have used the identity $\nabla J(\lambda_k)(s,a) = \mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}} \nabla \log(\lambda_k(a|s)) Q^{\lambda_k}(s,a)$. The details of this are given in Appendix C.

We thus obtain an upper bound on the last iterate optimality gap in terms of the average of the difference between the true gradient $\nabla J(\lambda_t)$ and our estimate of the true gradient d_t so far. Thus far we obtained an upper bound on the **G**lobal optimality of the **P**erformance of the last iterate. Since we do not have any terms that are functions of the cardinality of the state and action space, the analysis is valid for a **C**ontinous Action Space. Since the only restriction on the policy parametrization is smoothness, this analysis holds for a **M**ulti Layer Neural Network actor parametrization.

Upper Bounding Error in Critic Step:

The critic error at each step is equivalent to solving the following optimization problem

$$\arg\min_{\theta\in\Theta'} \mathbb{E}_{s,a}(Q^{\pi_{\lambda_k}} - Q_\theta)^2, \tag{19}$$

where $(s,a) \sim d_\nu^{\pi_{\lambda_k}}$. We denote as $Q_{k,J}$ as our estimate of $Q^{\pi_{\lambda_k}}$ at k^{th} iteration of Algorithm 1 and iteration j of the first inner for loop. We obtain the following result for the for the action value function Q

$$\mathbb{E}_{s,a}|Q^{\pi_{\lambda_k}} - Q_{k,J}| \leq \sum_{j=0}^{J-2} \gamma^{J-j-1}(P^{\pi_{\lambda_k}})^{J-j-1}\mathbb{E}|\epsilon_{k,j+1}| + \tilde{\mathcal{O}}(\gamma^J), \tag{20}$$

where $\epsilon = T^{\pi_{\lambda_k}} Q_{k,j-1} - Q_{k,j}$ is error incurred at iteration j of the first inner for loop and iteration k of the outer for loop of Algorithm 1. In doing so, we have split the error incurred in estimating the critic at each step into the errors in estimating the target function at each iteration of the inner loop indexed by j. The first term on the right hand side of Equation (20) denotes this error, in works such as (Farahmand et al., 2010) this is known as the algorithmic error. The second term on the right hand side is called as the statistical error, which is the error incurred due to the random nature of the system. Intuitively, the error in estimating the target function depends on how much data is collected at each iteration, how many samples we take in the buffer replay step and how well our neural network function class can approximate $T^{\pi_{\lambda_k}} Q_{k,j-1}$, i.e., the target function. Building upon this intuition, we split $\epsilon_{k,j}$ into four different components as follows.

$$\begin{aligned}\epsilon_{k,j} =& T^{\pi_{\lambda_k}} Q_{k,j-1} - Q_{k,j} \\ =& \underbrace{T^{\pi_{\lambda_k}} Q_{k,j-1} - Q^1_{k,j}}_{\epsilon^1_{k,j}} + \underbrace{Q^1_{k,j} - Q^2_{k,j}}_{\epsilon^2_{k,j}} \\ &+ \underbrace{Q^2_{k,j} - Q^3_{k,j}}_{\epsilon^3_{k,j}} + \underbrace{Q^3_{k,j} - Q_{k,j}}_{\epsilon^4_{k,j}} \\ =& \epsilon^1_{k,j} + \epsilon^2_{k,j} + \epsilon^3_{k,j} + \epsilon^4_{k,j}.\end{aligned} \tag{21}$$

The first two components are dependent on the approximating power of the class of neural networks. The third term is dependent on the number of samples collected from the policy, for which the corresponding Q function is to be measured. This is the term that will account for the **M**arkovian Sampling. The last term is dependent on the number of samples in the buffer replay step and accounts for the **M**ulti Layer Neural Network critic parametrization.

We now define the various Q-functions and then define the corresponding errors. We start by defining the best possible approximation of the function $T^{\pi_{\lambda_k}} Q_{k,j-1}$ possible from the class of neural networks, with respect to the square loss function with the target being $T^{\pi_{\lambda_k}} Q_{k,j-1}$.

Definition 1. *For iteration k of the outer for loop and iteration j of the first inner for loop of Algorithm 1, we define*

$$Q^1_{k,j} = \arg\min_{Q_\theta, \theta\in\Theta'} \mathbb{E}(Q_\theta(s,a) - T^{\pi_{\lambda_k}} Q_{k,j-1}(s,a))^2,$$

where $(s,a) \sim \zeta_\nu^{\pi_{\lambda_k}}$.

Note that we do not have access to the transition probability kernel P, hence we do not know $T^{\pi_{\lambda_k}}$. To alleviate this, we use the observed next state and actions instead. Using this, we define $Q^2_{k,j}$ as,

Definition 2. *For iteration k of the outer for loop and iteration j of the first inner for loop of Algorithm 1, we define*

$$\begin{aligned} Q^2_{k,j} &= \arg\min_{Q_\theta, \theta\in\Theta'} \mathbb{E}(Q_\theta(s,a) \\ &- (r(s,a) + \gamma Q_{k,j-1}(s',a'))^2, \end{aligned}$$

Here the expectation is with respect to $(s,a) \sim \zeta_\nu^{\pi_{\lambda_k}}, s' \sim P(\cdot|s,a)$ and $a' \sim \pi^{\lambda_k}(.|s')$. To obtain $Q^2_{k,j}$, we still need to compute the true expected value. However, we still do not know the transition function P. We thus sample transitions obtained from

following the policy π^{λ_k} and minimize the corresponding empirical loss function as follows. Consider the set of n state-action pairs sampled by starting from a state action distribution ν and following policy π^{λ_k}, using which we define $Q^3_{k,j}$ as,

Definition 3. *For the set of n state action pairs sampled in iteration k of the outer for loop of Algorithm 1 and at iteration j of the first inner for loop we define*

$$Q^3_{k,j} = \arg\min_{Q_\theta, \theta \in \Theta'} \frac{1}{n} \sum_{i=1}^{n} \Big(Q_\theta(s_i, a_i) - \big(r_i + \gamma Q_{k,j-1}(s_i^{'}, a_i^{'})\big)\Big)^2$$

$Q^3_{k,j}$ is the best possible approximation for Q-value function which minimizes the sample average of the square loss functions with the target values as $\big(r_i + \gamma Q_{k,j-1}(s_i^{'}, a_i^{'})\big)$. In other words, this is the optimal solution for fitting the observed data.

We now defined the errors using the Q functions just defined. We start by defining the approximation error which represents the difference between the function $T^{\pi_{\lambda_k}} Q_{k,j-1}$ and its best approximation possible from the class of neural networks used for critic parametrization denoted by $Q^1_{k,j}$.

Definition 4 (Approximation Error)**.** *For a given iteration k of the outer for loop and iteration j of the first inner for loop of Algorithm 1, we define, $\epsilon^1_{k,j} = T^{\pi_{\lambda_k}} Q_{k,j-1} - Q^1_{k,j}$.*

This error is a measure of the approximation power of the class of neural networks we use to represent the critic. We upper bound this error in Lemma 4 in Appendix B.

We also define Estimation Error which denotes the error between the best approximation of $T^{\pi_{\lambda_k}} Q_{k,j-1}$ possible from the class of neural networks denoted by $Q^1_{k,j}$ and the minimizer of the loss function in Definition 2 denoted by $Q^2_{k,j}$.

Definition 5 (Estimation Error)**.** *For a given iteration k of the outer for loop and iteration j of the first inner for loop of Algorithm 1, we define, $\epsilon^2_{k,j} = Q^1_{k,j} - Q^2_{k,j}$.*

We demonstrate that this error is zero in Lemma 5 in Appendix B.

We now define the Sampling error, which denotes the difference between the minimizer of expected loss function in Definition (2) denoted by $Q^2_{k,j}$ and the minimizer of the empirical loss function in Definition (3) denoted by $Q^3_{k,j}$. We can see that intuitively, the more samples we have the closer these two functions will be. We use Rademacher complexity results to upper bound this error. Thus this error is a function of the number of samples of transitions collected. We account for the Markov dependence of the transitions in this error.

Definition 6 (Sampling Error)**.** *For a given iteration k of the outer for loop and iteration j of the first inner for loop of Algorithm 1, we define, $\epsilon^3_{k,j} = Q^3_{k,j} - Q^2_{k,j}$.*

An upper bound on this error is established in Lemma 6 in Appendix B.

Finally, we define optimization error which denotes the difference between the minimizer of the empirical square loss function in Definition (3) denoted by $Q^3_{k,j}$, and our estimate of this minimizer that is obtained from the gradient descent algorithm that is implemented in lines $5 - 11$ of Algorithm (1).

Definition 7 (Optimization Error)**.** *For a given iteration k of the outer for loop and iteration j of the first inner for loop of Algorithm 1, we define, $\epsilon^4_{k,j} = Q^3_{k,j} - Q_{k,j}$.*

The key insight we use to bound this error is the fact that the loss function in Definition (3) can be treated as an expected loss function, with a weight of $\frac{1}{n}$ over all n transition samples. We thus bound this error using tools established in (Fu et al., 2021) in Lemma 7 in Appendix B.

7. Conclusions

In this paper, we study an actor critic algorithm with a neural network used to represent both the the critic and find the sample complexity guarantees for the algorithm. We show that our approach achieves a last iterate convergence sample complexity of $\tilde{\mathcal{O}}(\epsilon^{-3})$. We do so without assuming i.i.d. sampling and without the restriction of the finite action space. To our knowledge this is the first work that achieves such a result.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are no potential societal consequences of our work.

References

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Agarwal, M., Aggarwal, V., and Lan, T. Multi-objective reinforcement learning with non-linear scalarization. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 9–17, 2022.

Balkanski, E. and Singer, Y. The sample complexity of optimizing a convex function. In *Conference on Learning Theory*, pp. 275–301, 2017.

Bertail, P. and Portier, F. Rademacher complexity for markov chains: Applications to kernel smoothing and metropolis–hastings. *Bernoulli*, 25:3912–3938, 11 2019. doi: 10.3150/19-BEJ1115.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, March 2004. ISBN 0521833787.

Cai, Q., Yang, Z., Lee, J. D., and Wang, Z. Neural temporal-difference learning converges to global optima. *Advances in Neural Information Processing Systems*, 32, 2019.

Castro, D. D. and Meir, R. A convergent online single time scale actor critic algorithm. *The Journal of Machine Learning Research*, 11:367–410, 2010.

Cayci, S., He, N., and Srikant, R. Finite-time analysis of entropy-regularized neural natural actor-critic algorithm. *arXiv preprint arXiv:2206.00833*, 2022.

Chen, J., Chen, J., Lan, T., and Aggarwal, V. Scalable multi-agent covering option discovery based on kronecker graphs. *Advances in Neural Information Processing Systems*, 35:30406–30418, 2022.

Chen, X., Karimi, B., Zhao, W., and Li, P. On the convergence of decentralized adaptive gradient methods. In Khan, E. and Gonen, M. (eds.), *Proceedings of The 14th Asian Conference on Machine Learning*, volume 189 of *Proceedings of Machine Learning Research*, pp. 217–232. PMLR, 12–14 Dec 2023.

Dankwa, S. and Zheng, W. Twin-delayed ddpg: A deep reinforcement learning technique to model a continuous movement of an intelligent robot agent. In *Proceedings of the 3rd international conference on vision, image and signal processing*, pp. 1–5, 2019.

Dankwa, S. and Zheng, W. Twin-delayed ddpg: A deep reinforcement learning technique to model a continuous movement of an intelligent robot agent. In *Proceedings of the 3rd International Conference on Vision, Image and Signal Processing*, ICVISP 2019, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450376259. doi: 10.1145/3387168.3387199.

Ding, Y., Zhang, J., and Lavaei, J. On the global optimum convergence of momentum-based policy gradient. In *International Conference on Artificial Intelligence and Statistics*, pp. 1910–1934, 2022.

Doan, T. T. Finite-time analysis of markov gradient descent. *IEEE Transactions on Automatic Control*, 68(4): 2140–2153, 2022.

Farahmand, A.-m., Szepesvári, C., and Munos, R. Error propagation for approximate policy and value iteration. *Advances in Neural Information Processing Systems*, 23, 2010.

Fatkhullin, I., Barakat, A., Kireeva, A., and He, N. Stochastic policy gradient methods: Improved sample complexity for fisher-non-degenerate policies. In *International Conference on Machine Learning*, 2023.

Fu, Z., Yang, Z., and Wang, Z. Single-timescale actor-critic provably finds globally optimal policy. In *International Conference on Learning Representations*, 2021.

Gaur, M., Aggarwal, V., and Agarwal, M. On the global convergence of fitted q-iteration with two-layer neural network parametrization. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11013–11049. PMLR, 23–29 Jul 2023a.

Gaur, M., Bedi, A. S., Wang, D., and Aggarwal, V. On the global convergence of natural actor-critic with two-layer neural network parametrization, 2023b.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Khodadadian, S., Chen, Z., and Maguluri, S. T. Finite-sample analysis of off-policy natural actor-critic algorithm. In *International Conference on Machine Learning*, pp. 5420–5431, 2021.

Konda, V. and Tsitsiklis, J. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

Lee, A. X., Nagabandi, A., Abbeel, P., and Levine, S. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, 33:741–752, 2020.

Li, X. and Orabona, F. On the convergence of stochastic gradient descent with adaptive stepsizes. In *The 22nd international conference on artificial intelligence and statistics*, pp. 983–992, 2019.

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. *arXiv preprint arXiv:1509.02971*, 2015.

Liu, C.-L., Chang, C.-C., and Tseng, C.-J. Actor-critic deep reinforcement learning for solving job shop scheduling problems. *IEEE Access*, 8:71752–71762, 2020a. doi: 10.1109/ACCESS.2020.2987820.

Liu, Y., Zhang, K., Basar, T., and Yin, W. An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods. *Advances in Neural Information Processing Systems*, 33:7624–7636, 2020b.

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

Masiha, S., Salehkaleybar, S., He, N., Kiyavash, N., and Thiran, P. Stochastic second-order methods improve best-known sample complexity of SGD for gradient-dominated functions. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

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

Mondal, W. U. and Aggarwal, V. Improved sample complexity analysis of natural policy gradient algorithm with general parameterization for infinite horizon discounted reward markov decision processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 3097–3105, 2024.

Polyak, B. Gradient methods for the minimisation of functionals. *Ussr Computational Mathematics and Mathematical Physics*, 3:864–878, 1963.

Rudin, W. *Real and complex analysis, 3rd ed.* McGraw-Hill, Inc., USA, 1987. ISBN 0070542341.

Sun, T., Sun, Y., and Yin, W. On markov chain gradient descent. *Advances in neural information processing systems*, 31, 2018.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Swirszcz, G., Czarnecki, W. M., and Pascanu, R. Local minima in training of neural networks. *arXiv preprint arXiv:1611.06310*, 2016.

Tang, X., Huang, B., Liu, T., and Lin, X. Highway decision-making and motion planning for autonomous driving via soft actor-critic. *IEEE Transactions on Vehicular Technology*, 71(5):4706–4717, 2022.

Tian, H., Olshevsky, A., and Paschalidis, I. Convergence of actor-critic with multi-layer neural networks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Wang, D. and Hu, M. Deep deterministic policy gradient with compatible critic network. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):4332–4344, 2023. doi: 10.1109/TNNLS.2021.3117790.

Wang, L., Cai, Q., Yang, Z., and Wang, Z. Neural policy gradient methods: Global optimality and rates of convergence. *International Conference on Learning Representations*, 2020.

Xu, T., Wang, Z., and Liang, Y. Improving sample complexity bounds for (natural) actor-critic algorithms. *Advances in Neural Information Processing Systems*, 33: 4358–4369, 2020a.

Xu, T., Wang, Z., and Liang, Y. Non-asymptotic convergence analysis of two time-scale (natural) actor-critic algorithms. *arXiv preprint arXiv:2005.03557*, 2020b.

Xu, T., Yang, Z., Wang, Z., and Liang, Y. Doubly robust off-policy actor-critic: Convergence and optimality. In *International Conference on Machine Learning*, pp. 11581–11591, 2021.

Yuan, R., Gower, R. M., and Lazaric, A. A general sample complexity analysis of vanilla policy gradient. In *International Conference on Artificial Intelligence and Statistics*, pp. 3332–3380, 2022.

Yue, P., Fang, C., and Lin, Z. On the lower bound of minimizing polyak-Łojasiewicz functions. In Neu, G. and Rosasco, L. (eds.), *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pp. 2948–2968. PMLR, 12–15 Jul 2023.

Zhang, G., Zhang, J., Liu, K., Guo, J., Lee, J. Y., Hu, H., and Aggarwal, V. Duasvs: A mobile data saving strategy in short-form video streaming. *IEEE Transactions on Services Computing*, 16(2):1066–1078, 2022.

Zhong, C., Lu, Z., Gursoy, M. C., and Velipasalar, S. A deep actor-critic reinforcement learning framework for dynamic multichannel access. *IEEE Transactions on Cognitive Communications and Networking*, 5(4):1125–1139, 2019. doi: 10.1109/TCCN.2019.2952909.

Zhou, H., Lan, T., and Aggarwal, V. Pac: Assisted value factorization with counterfactual predictions in multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 35:15757–15769, 2022.

Contents

1 Introduction **1**

2 Related Works **2**

3 Problem Formulation **3**

4 Actor Critic Algorithm **4**

5 Theoretical Analysis: Global Convergence **5**

5.1 Assumptions 5

5.2 Main Result 6

6 Proof Sketch of Theorem 1 **6**

7 Conclusions **8**

A Supplementary Lemmas **13**

B Supporting Lemmas **15**

C Proof of Theorem 1 **15**

D Proof of Supporting Lemmas **21**

D.1 Proof Of Lemma 4 21

D.2 Proof Of Lemma 5 22

D.3 Proof Of Lemma 6 23

D.4 Proof Of Lemma 7 26

Appendix

A. Supplementary Lemmas

Here we provide some definitions and results that will be used to prove the lemmas stated in the paper.

Definition 8. *For a given set $Z \subset \mathbb{R}^n$, we define the Rademacher complexity of the set Z as*

$$Rad(Z) = \mathbb{E}\left(\sup_{z \in Z} \frac{1}{n} \sum_{i=1}^{d} \Omega_i z_i\right) \tag{22}$$

where Ω_i is random variable such that $P(\Omega_i = 1) = \frac{1}{2}$, $P(\Omega_i = -1) = \frac{1}{2}$ and z_i are the co-ordinates of z which is an element of the set Z

Lemma 2. *Consider a set of observed data denoted by $z = \{z_1, z_2, \cdots z_n\} \in Z$, a parameter space Θ, a loss function $\{l : Z \times \Theta \to \mathbb{R}\}$ where $0 \leq l(\theta, z) \leq 1 \; \forall (\theta, z) \in \Theta \times Z$. The empirical risk for a set of observed data as $R(\theta) = \frac{1}{n}\sum_{i=1}^{n} l(\theta, z_i)$ and the population risk as $r(\theta) = \mathbb{E}l(\theta, \tilde{z}_i)$, where $\tilde{z}_i$ sampled from some distribution over Z.*

We define a set of functions denoted by $\mathcal{L}$ as

$$\mathcal{L} = \{z \in Z \to l(\theta, z) \in \mathbb{R} : \theta \in \Theta\} \tag{23}$$

Given $z = \{z_1, z_2, z_3 \cdots, z_n\}$ we further define a set $\mathcal{L} \circ z$ as

$$\mathcal{L} \circ z \; = \{(l(\theta, z_1), l(\theta, z_2), \cdots, l(\theta, z_n)) \in \mathbb{R}^n : \theta \in \Theta\} \tag{24}$$

Then, we have

$$\mathbb{E} \sup_{\theta \in \Theta} |\{r(\theta) - R(\theta)\}| \leq 2\mathbb{E}\left(Rad(\mathcal{L} \circ z)\right) \tag{25}$$

If the data is of the form $z_i = (x_i, y_i), x \in X, y \in Y$ and the loss function is of the form $l(a_\theta(x), y)$, is L lipschitz and $a_\theta : \Theta \times X \to \mathbb{R}$, then we have

$$\mathbb{E} \sup_{\theta \in \Theta} |\{r(\theta) - R(\theta)\}| \leq 2L\mathbb{E}\left(Rad(\mathcal{A} \circ \{x_1, x_2, x_3, \cdots, x_n\})\right) \tag{26}$$

where

$$\mathcal{A} \circ \{x_1, x_2, \cdots, x_n\} \; = \{(a(\theta, x_1), a(\theta, x_2), \cdots, a(\theta, x_n)) \in \mathbb{R}^n : \theta \in \Theta\} \tag{27}$$

The detailed proof of the above statement is given in (Rebeschini, 2022)[1]. The upper bound for $\mathbb{E}\sup_{\theta \in \Theta}(\{r(\theta) - R(\theta)\})$ is proved in the aformentioned reference. However, without loss of generality the same proof holds for the upper bound for $\mathbb{E}\sup_{\theta \in \Theta}(\{R(\theta) - r(\theta)\})$. Hence the upper bound for $\mathbb{E}\sup_{\theta \in \Theta}|\{r(\theta) - R(\theta)\}|$ can be established.

Lemma 3. *Consider three random random variable $x \in \mathcal{X}$ and $y, y^{'} \in \mathcal{Y}$. Let $\mathbb{E}_{x,y}, \mathbb{E}_x$ and $\mathbb{E}_{y|x}$, $\mathbb{E}_{y^{'}|x}$ denote the expectation with respect to the joint distribution of (x, y), the marginal distribution of x, the conditional distribution of y given x and the conditional distribution of $y^{'}$ given x respectively . Let $f_\theta(x)$ denote a bounded measurable function of x parameterised by some parameter θ and $g(x, y)$ be bounded measurable function of both x and y.*

Then we have

[1] Algorithmic Foundations of Learning [Lecture Notes].https://www.stats.ox.ac.uk/∼rebeschi/teaching/AFoL/22/

$$\arg\min_{f_\theta} \mathbb{E}_{x,y}\left(f_\theta(x) - g(x,y)\right)^2 = \arg\min_{f_\theta}\left(\mathbb{E}_x\left(f_\theta(x) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)^2\right) \tag{28}$$

Proof. Denote the left hand side of Equation (28) as $\mathbb{X}_\theta$, then add and subtract $\mathbb{E}_{y|x}(g(x,y^{'}))$ to it to get

$$\begin{aligned}
\mathbb{X}_\theta =& \arg\min_{f_\theta}\left(\mathbb{E}_{x,y}\left(f_\theta(x) - \mathbb{E}_{y'|x}(g(x,y^{'})) + \mathbb{E}_{y'|x}(g(x,y^{'})) - g(x,y)\right)^2\right) && (29)\\
=& \arg\min_{f_\theta}\left(\mathbb{E}_{x,y}\left(f_\theta(x) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)^2 + \mathbb{E}_{x,y}\left(g(x,y) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)^2\right.\\
&\left. - 2\mathbb{E}_{x,y}\left(f_\theta(x) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)\left(g(x,y) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)\right). && (30)
\end{aligned}$$

Consider the third term on the right hand side of Equation (30)

$$\begin{aligned}
&2\mathbb{E}_{x,y}\left(f_\theta(x) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)\left(g(x,y) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)\\
&=2\mathbb{E}_x\mathbb{E}_{y|x}\left(f_\theta(x) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)\left(g(x,y) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right) && (31)\\
&=2\mathbb{E}_x\left(f_\theta(x) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)\mathbb{E}_{y|x}\left(g(x,y) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right) && (32)\\
&=2\mathbb{E}_x\left(f_\theta(x) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)\left(\mathbb{E}_{y|x}(g(x,y)) - \mathbb{E}_{y|x}\left(\mathbb{E}_{y'|x}(g(x,y^{'}))\right)\right) && (33)\\
&=2\mathbb{E}_x\left(f_\theta(x) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)\left(\mathbb{E}_{y|x}(g(x,y)) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right) && (34)\\
&=0 && (35)
\end{aligned}$$

Equation (31) is obtained by writing $\mathbb{E}_{x,y} = \mathbb{E}_x\mathbb{E}_{y|x}$ from the law of total expectation. Equation (32) is obtained from (31) as the term $f_\theta(x) - \mathbb{E}_{y'|x}(g(x,y^{'}))$ is not a function of y. Equation (33) is obtained from (32) as $\mathbb{E}_{y|x}\left(\mathbb{E}_{y'|x}(g(x,y^{'}))\right) = \mathbb{E}_{y'|x}(g(x,y^{'}))$ because $\mathbb{E}_{y'|x}(g(x,y^{'}))$ is not a function of y hence is constant with respect to the expectation operator $\mathbb{E}_{y|x}$.

Thus plugging in value of $2\mathbb{E}_{x,y}\left(f_\theta(x) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)\left(g(x,y) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)$ in Equation (30) we get

$$\begin{aligned}
\arg\min_{f_\theta} \mathbb{E}_{x,y}\left(f_\theta(x) - g(x,y)\right)^2 =& \arg\min_{f_\theta}(\mathbb{E}_{x,y}\left(f_\theta(x) - \mathbb{E}_{x,y'}(g(x,y^{'}))\right)^2\\
&+ \mathbb{E}_{x,y}\left(g(x,y) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)^2). && (36)
\end{aligned}$$

Note that the second term on the right hand side of Equation (36) des not depend on $f_\theta(x)$ therefore we can write Equation (36) as

$$\arg\min_{f_\theta} \mathbb{E}_{x,y}\left(f_\theta(x) - g(x,y)\right)^2 = \arg\min_{f_\theta}\left(\mathbb{E}_{x,y}\left(f_\theta(x) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)^2\right) \tag{37}$$

Since the right hand side of Equation (37) is not a function of y we can replace $\mathbb{E}_{x,y}$ with $\mathbb{E}_x$ to get

$$\arg\min_{f_\theta} \mathbb{E}_{x,y}\left(f_\theta(x) - g(x,y)\right)^2 = \arg\min_{f_\theta}\left(\mathbb{E}_x\left(f_\theta(x) - \mathbb{E}_{y'|x}(g(x,y^{'}))\right)^2\right) \tag{38}$$

□

B. Supporting Lemmas

We will now state the key lemmas that will be used for finding the sample complexity of the proposed algorithm.

Lemma 4. *For a given iteration k of the outer for loop and iteration j of the first inner for loop of Algorithm 1, the approximation error denoted by $\epsilon^1_{k,j}$ in Definition 4, we have*

$$\mathbb{E}\left(|\epsilon^1_{k,j}|\right) \leq \sqrt{\epsilon_{approx}}, \tag{39}$$

Where the expectation is with respect to and $(s,a) \sim \zeta_\nu^{\pi_{\lambda_k}}(s,a)$

Proof Sketch: We use Assumption 4 and the definition of the variance of a random variable to obtain the required result. The detailed proof is given in Appendix D.1.

Lemma 5. *For a given iteration k of the outer for loop and iteration j of the first inner for loop of Algorithm 1, $Q^1_{k,j} = Q^2_{k,j}$, or equivalently $\epsilon^2_{k,j} = 0$*

Proof Sketch: We use Lemma 3 in Appendix A and use the definitions of $Q^1_{k,j}$ and $Q^2_{k,j}$ to prove this result. The detailed proof is given in Appendix D.2.

Lemma 6. *For a given iteration k of the outer for loop and iteration j of the first inner for loop of Algorithm 1, if the number of state action pairs sampled are denoted by n, then the error $\epsilon^3_{k,j}$ defined in Definition 6 is upper bounded as*

$$\mathbb{E}\left(|\epsilon^3_{k,j}|\right) \leq \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{n}}\right), \tag{40}$$

Where the expectation is with respect to and $(s,a) \sim \zeta_\nu^{\pi_{\lambda_k}}(s,a)$

Proof Sketch: First we note that For a given iteration k of Algorithm 1 and iteration j of the first for loop of Algorithm 1, $\mathbb{E}(R_{X_{k,j},Q_{k,j-1}}(\theta)) = L_{Q_{j,k-1}}(\theta)$ where $R_{X_{k,j},Q_{j,k-1}}(\theta)$ and $L_{Q_{j,k-1}}(\theta)$ are defined in Appendix D.3. We use this to get a probabilistic bound on the expected value of $|(Q^2_{j,k}) - (Q^3_{j,k})|$ using Rademacher complexity theory when the samples are drawn from a Markov chain satisfying Asssumption 2. The detailed proof is given in Appendix D.3.

Lemma 7. *For a given iteration k of the outer for loop and iteration j of the first inner for loop of Algorithm 1, let the number of gradient descent steps be denoted by L, and the gradient descent step size $\beta^{'}$ satisfy*

$$\beta^{'} = \frac{1}{\sqrt{L}}, \tag{41}$$

Then with probability at least $1 - \Omega\left(exp - \left(m^{\frac{2}{3}}D\right)\right)$ the error ϵ_{k_4} defined in Definition 7 is upper bounded as

$$\mathbb{E}(|\epsilon^4_{k,j}|) \leq \mathcal{O}\left(L^{-\frac{1}{4}}\right) + \mathcal{O}\left(m^{-\frac{1}{12}}D^{\frac{7}{2}}\right), \tag{42}$$

Where the expectation is with respect to $(s,a) \sim \zeta_\nu^{\pi_{\lambda_k}}(s,a)$.

Proof Sketch: They key insight we use here is that the loss function in Definition 3 can be considered as an expectation over the state action transitions samples at iteration k with an equal probability of selecting each transitions. This is then combined with results from (Fu et al., 2021) to obtain the desired result.

C. Proof of Theorem 1

Proof. From the smoothness property of the expected return we have

$$\begin{aligned} -J(\lambda_{t+1}) &\leq -J(\lambda_t) - \langle \nabla J(\lambda_t), \lambda_{t+1} - \lambda_t \rangle + L_J||\lambda_{t+1} - \lambda_t||^2 && (43) \\ &\leq -J(\lambda_t) - \alpha_t \frac{\langle \nabla J(\lambda_t), d_t \rangle}{||d_t||} + L_J||\lambda_{t+1} - \lambda_t||^2 && (44) \end{aligned}$$

Here α_t is the actor step size at iteration t of Algorithm 1. Now define the term $e_t = d_t - \nabla J(\lambda_t)$.

Consider two cases, first if $||e_t|| \leq \frac{1}{2}||\nabla J(\lambda_t)||$, then we have

$$
\begin{aligned}
-\frac{\langle \nabla J(\lambda_t), d_t \rangle}{||d_t||} &= \frac{-||\nabla J(\lambda_t)||^2 - \langle \nabla J(\lambda_t), e_t \rangle}{||d_t||} && (45)\\
&\leq \frac{-||\nabla J(\lambda_t)||^2 + ||\nabla J(\lambda_t)||\cdot||e_t||}{||d_t||} && (46)\\
&\leq \frac{-||\nabla J(\lambda_t)||^2 + ||\nabla J(\lambda_t)||\cdot||e_t||}{||d_t||} && (47)\\
&\leq \frac{-||\nabla J(\lambda_t)||^2 + \frac{1}{2}||\nabla J(\lambda_t)||^2}{||d_t||} && (48)\\
&\leq -\frac{||\nabla J(\lambda_t)||^2}{2(||e_t|| + ||\nabla J(\lambda_t)||)} && (49)\\
&\leq -\frac{1}{3}||\nabla J(\lambda_t)|| && (50)
\end{aligned}
$$

If $||e_t|| \geq \frac{1}{2}||\nabla J(\lambda_t)||$, then we have

$$
\begin{aligned}
\frac{\langle \nabla J(\lambda_t), d_t \rangle}{||d_t||} &\leq ||\nabla J(\lambda_t)|| && (51)\\
&= -\frac{1}{3}||\nabla J(\lambda_t)|| + \frac{4}{3}||\nabla J(\lambda_t)|| && (52)\\
&\leq -\frac{1}{3}||\nabla J(\lambda_t)|| + \frac{8}{3}||e_t|| && (53)
\end{aligned}
$$

This inequality has been established in (Fatkhullin et al., 2023). Now using Equation (53) in Equation (44) we get

$$
-J(\lambda_{t+1}) \leq -J(\lambda_t) - \frac{\alpha_t}{3}||\nabla J(\lambda_t)|| + \frac{8\alpha_t}{3}||e_t|| + L_J||\lambda_{t+1} - \lambda_t||^2 \tag{54}
$$

Now from Lemma 4 we have

$$
\begin{aligned}
-J(\lambda_{t+1}) &\leq -J(\lambda_t) - \frac{\alpha_t\sqrt{\mu'}}{3}(J^* - J(\lambda_t)) + \frac{8\alpha_t}{3}||\nabla J(\lambda_t) - d_t||\\
&+ L_J||\lambda_{t+1} - \lambda_t||^2 + \frac{\alpha_t}{3}\epsilon' && (55)\\
J^* - J(\lambda_{t+1}) &\leq J^* - J(\lambda_t) - \frac{\alpha_t\sqrt{\mu'}}{3}(J^* - J(\lambda_t)) + \frac{8\alpha_t}{3}||\nabla J(\lambda_t) - d_t||\\
&+ L_J||\lambda_{t+1} - \lambda_t||^2 + \frac{\alpha_t}{3}\epsilon' && (56)\\
\delta_{\lambda_{t+1}} &\leq \left(1 - \frac{\alpha_t\sqrt{\mu'}}{3}\right)\delta_{\lambda_t} + \frac{8\alpha_t}{3}||\nabla J(\lambda_t) - d_t||\\
&+ L_J||\lambda_{t+1} - \lambda_t||^2 + \frac{\alpha_t}{3}\epsilon' && (57)\\
&\leq \left(1 - \frac{\alpha_t\sqrt{\mu'}}{3}\right)\delta_{\lambda_t} + \frac{8\alpha_t}{3}||\nabla J(\lambda_t) - d_t||\\
&+ L_J{\alpha_t}^2 + \frac{\alpha_t}{3}\epsilon' && (58)
\end{aligned}
$$

Where $\delta_t = J^* - J(\lambda_t)$Now, in equation (58), if we plug in the value of δ_{λ_t} by evaluating equation (58) for $t-1$, we get the following

$$\begin{aligned}
\delta_{\lambda_{t+1}} &\leq \left(1 - \frac{\alpha_t \cdot \sqrt{\mu'}}{3}\right)\left(1 - \frac{\alpha_{t-1} \cdot \sqrt{\mu'}}{3}\right)\delta_{\lambda_{t-1}} \\
&+ \left(1 - \frac{\alpha_t \sqrt{\mu'}}{3}\right)\alpha_{t-1}(||\nabla J(\lambda_{t-1}) - d_{t-1}|| + \epsilon') + \alpha_t(||\nabla J(\lambda_t) - d_t|| + \epsilon') \\
&+ \left(1 - \frac{\alpha_t \sqrt{\mu'}}{3}\right) L_J {\alpha_{t-1}}^2 + L_J {\alpha_t}^2 \qquad (59)
\end{aligned}$$

We got rid of the factor of $\frac{8}{3}$ and $\frac{1}{3}$ on $||\nabla J(\lambda_t) - d_t||$ and ϵ' respectively as it will be absorbed in the $\tilde{\mathcal{O}}$ notation. If we repeat the substitution starting from t and going back till $t = 2$ we get

$$\begin{aligned}
\delta_{\lambda_t} &\leq \underbrace{\Pi_{k=2}^{k=t}\left(1 - \frac{\alpha_k \sqrt{\mu'}}{3}\right)\delta_{\lambda_2}}_{A} \\
&+ \underbrace{\sum_{k=0}^{k=t-2}\left(\Pi_{i=0}^{k-1}\left(1 - \frac{\alpha_{(t-i)}\sqrt{\mu'}}{3}\right)\right)^{\mathbb{1}(k\geq 1)} \alpha_{t-k}(||\nabla J(\lambda_{t-k}) - d_{t-k}|| + \epsilon')}_{B} \\
&+ \underbrace{L_J \sum_{k=0}^{k=t-2}\left(\Pi_{i=0}^{i=k-1}\left(1 - \frac{\alpha_{(t-i)}\sqrt{\mu'}}{3}\right)\right)^{\mathbb{1}(k\geq 1)} (\alpha_{t-k})^2}_{C} \qquad (60)
\end{aligned}$$

Now let us consider the term A is equation (60), if $\alpha_k = \frac{\alpha}{k}$ where $\alpha = \frac{7}{2\sqrt{\mu'}}$, then we have

$$\begin{aligned}
1 - \frac{\alpha_k \sqrt{\mu'}}{3} &= 1 - \frac{7}{6k} \qquad (61) \\
&\leq 1 - \frac{1}{k} \qquad (62) \\
&\leq \frac{k-1}{k} \qquad (63) \\
&\leq \frac{\alpha_k}{\alpha_{k-1}} \qquad (64)
\end{aligned}$$

Therefore we have

$$\begin{aligned}
A = \Pi_{k=2}^{k=t}\left(1 - \frac{\alpha_k \sqrt{\mu'}}{3}\right)\delta_{\lambda_2} &\leq \Pi_{k=2}^{k=t}\left(\frac{\alpha_k}{\alpha_{k-1}}\right)\delta_{\lambda_2} \qquad (65) \\
&\leq \frac{\alpha_t}{\alpha_1}\delta_{\lambda_2} = \frac{1}{t}\delta_{\lambda_2} \qquad (66)
\end{aligned}$$

Now let us consider the term B is equation (60)

$$B = \sum_{k=0}^{k=t-2}\left(\Pi_{i=0}^{k-1}\left(1 - \frac{\alpha_{(t-i)}\sqrt{\mu'}}{3}\right)\right)^{\mathbb{1}(k\geq 1)} \alpha_{t-k}(||\nabla J(\lambda_{t-k}) - d_{t-k}|| + \epsilon') \qquad (67)$$

Now if we consider the coefficients of $(||\nabla J(\lambda_{t-k}) - d_{t-k}|| + \epsilon^{'})$, we see as follows.

For $k = 0$ the product term is 1 because of the indicator function $\mathbb{1}(k \geq 1)$.

For $k = 1$, suppose the coefficient is $\alpha_k = \frac{\alpha}{k}$. Then we have

$$\left(1 - \frac{\alpha\sqrt{\mu^{'}}}{3t}\right)\frac{\alpha}{t-1} = \left(\frac{t - \frac{\alpha\sqrt{\mu^{'}}}{3}}{t-1}\right)\frac{\alpha}{t} \tag{68}$$

For $k = 2$ we have

$$\left(1 - \frac{\frac{\alpha\sqrt{\mu^{'}}}{3}}{t}\right)\left(1 - \frac{\frac{\alpha\sqrt{\mu^{'}}}{3}}{t-1}\right)\frac{\alpha}{t-2} = \left(\frac{t - \frac{\alpha\sqrt{\mu^{'}}}{3} - 1}{t-2}\right)\left(\frac{t - \frac{\alpha\sqrt{\mu^{'}}}{3}}{t-1}\right)\frac{\alpha}{t} \tag{69}$$

In general for a k this coefficient is thus

$$\Pi_{i=1}^{k}\left(\frac{t - (\frac{\alpha\sqrt{\mu^{'}}}{3} + i - 1)}{t-i}\right)\frac{\alpha}{t} \tag{70}$$

Now for $\alpha = \frac{7}{2\sqrt{\mu}}$ the numerator in all the product terms is less than the denominator hence product term is less than 1. Therefore, all the coefficients in B are upper bounded by α_t. Thus we have

$$B \quad \leq \quad \frac{\alpha}{t}\sum_{k=1}^{k=t-2}(||\nabla J(\lambda_k) - d_k||) + \epsilon^{'} \tag{71}$$

Now let us consider the term C is equation (60)

$$C \quad = \quad L_J\sum_{k=0}^{k=t-2}\left(\Pi_{i=0}^{i=k-1}\left(1 - \frac{\alpha_{(t-i)}\sqrt{\mu^{'}}}{3}\right)\right)^{\mathbb{1}(k\geq 1)}(\alpha_{t-k})^2 \tag{72}$$

Now similar to what was done for A consider the coefficients of ${\alpha_{t-k}}^2$.

For $k = 0$ the product term is 1 because of the indicator function $\mathbb{1}(k \geq 1)$.

for $k = 1$ if we have $\alpha = \frac{7}{2\sqrt{\mu}}$ then

$$\left(1 - \frac{\alpha\sqrt{\mu^{'}}}{3t}\right)\left(\frac{\alpha}{t-1}\right)^2 \quad \leq \quad \left(\frac{\alpha}{t-1}\right)^2 \tag{73}$$

for $k = 2$ if we have $\alpha = \frac{7}{2\sqrt{\mu}}$ then

$$\left(1 - \frac{\alpha\mu^{'}}{t}\right)\left(1 - \frac{\alpha\mu^{'}}{t-1}\right)\left(\frac{\alpha}{t-2}\right)^2 \quad = \quad \frac{\left(t - \frac{\alpha\sqrt{\mu^{'}}}{3}\right)}{t}\frac{\left(t - \frac{\alpha\sqrt{\mu^{'}}}{3} - 1\right)}{t-1}\left(\frac{\alpha}{t-2}\right)^2 \tag{74}$$

$$\leq \quad \left(\frac{\alpha}{t-2}\right)^2 \tag{75}$$

This is because both terms in the coefficient of $\left(\frac{\alpha}{t-2}\right)^2$ are less than 1.

In general for any k if we have $\alpha = \frac{7}{2\sqrt{\mu}}$ then we have

$$\Pi_{i=0}^{i=k-1}\left(1-\frac{\alpha\mu^{'}}{t-i}\right)\left(\frac{\alpha}{t-k}\right)^2 = \frac{\left(t-\frac{\alpha\sqrt{\mu^{'}}}{3}\right)}{t}\frac{\left(t-\frac{\alpha\sqrt{\mu^{'}}}{3}-1\right)}{t-1}\cdots\frac{\left(t-\frac{\alpha\sqrt{\mu^{'}}}{3}-k+1\right)}{t-k+1}\left(\frac{\alpha}{t-k}\right)^2 \tag{76}$$

$$\leq \left(\frac{\alpha}{t-k}\right)^2 \tag{77}$$

Therefore we have

$$C \leq L_J\sum_{k=2}^{k=t-2}\left(\frac{\alpha}{t-k}\right)^2 \tag{78}$$

$$\leq \frac{L_J\cdot\alpha^2}{t} \tag{79}$$

We get Equation (79) from (78) by using the fact that $\sum_{k=1}^{t}\frac{1}{k^2} \leq \frac{1}{t}$. Now plugging equation (66), (71) and (79) into equation (60) we get

$$\delta_{\lambda_t} \leq \left(\frac{1}{t}\right)\delta_{\lambda_2} + \frac{\alpha}{t}\underbrace{\sum_{k=0}^{k=t-2}(||\nabla J(\lambda_k)-d_t)||)}_{A^{'}} + \frac{L_J\cdot\alpha^2}{t} + \epsilon^{'} \tag{80}$$

Now consider the terms inside $A^{'}$, if we define $H_{k,J} = \mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}}(\nabla\log\pi_{\lambda_k}(a|s)Q_{k,J}(s,a))$

$$(||\nabla J(\lambda_k)-d_t+H_{k,J}-H_{k,J}||) \leq ||\nabla J(\lambda_k)-H_{k,J}||)+||d_t-H_{k,J}|| \tag{81}$$

$$\leq ||\nabla J(\lambda_k)-H_{k,J}||+\left|\left|\frac{1}{n}\sum_{i=1}^{n}(\nabla\log\pi_{\lambda_k}(a_i|s_i)Q_{k,J}(s_i,a_i)-(H_{k,J}))\right|\right| \tag{82}$$

$$\leq ||\nabla J(\lambda_k)-H_{k,J}||+ \sqrt{d\cdot\sum_{p=1}^{d}\left(\left(\sum_{i=1}^{n}\frac{1}{n}\nabla\log\pi_{\lambda_k}(a_i|s_i)Q_{k,J}(s_i,a_i)\right)_p-(H_{k,J})_p\right)^2} \tag{83}$$

$$\mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}}(||\nabla J(\lambda_k)-d_t||) \leq ||\nabla J(\lambda_k)-H_{k,J}||+ \sqrt{d\cdot\sum_{p=1}^{d}\mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}}\left(\left(\sum_{i=1}^{n}\frac{1}{n}\nabla\log\pi_{\lambda_k}(a_i|s_i)Q_{k,J}(s_i,a_i)\right)_p-(H_{k,J})_p\right)^2} \tag{84}$$

$$\leq ||\nabla J(\lambda_k)-H_{k,J}||+\frac{1}{\sqrt{n}}dM_gV_{max} \tag{85}$$

$$\leq M_g\mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}}|Q^{\pi_{\lambda_k}}(s,a)-Q_{k,J}(s,a)|+\frac{1}{\sqrt{n}}dM_gV_{max} \tag{86}$$

We obtain Equation (83) from Equation (82) by noting that l1 norm is upper bounded by the l2 norm multiplied by the square root of the dimensions. Here $(\nabla\log\pi_{\lambda_k}(a_i|s_i)Q_{k,J}(s_i,a_i))_p$ and $(H_{k,J})_p$ in Equation (83) are the pth co-ordinates

of the gradients. We obtain Equation (84) from Equation (83) by taking the expectation with respect to $(s,a) \sim d_\nu^{\pi_{\lambda_k}}$ on both sides of Equation (83) and applying Jensen's inequality on the final term on the right hand side. Note that the first term on the right hand side will remain unaffected as it is already an expectation term. We obtain Equation (85) from Equation (84) by noting that the variance of the random variable $\nabla \log \pi_{\lambda_k}(a|s) Q_{k,J}(s,a)$ is bounded when the expectation is over (s,a). We combine this with the fact that the variance of the mean is the variance divided by the number of samples, which is this case is n. We can assume the variance of $\nabla \log \pi_{\lambda_k}(a|s) Q_{k,J}(s,a)_p$ is bounded because from Assumption 1 we have $||\nabla \log \pi_{\lambda_k}(a|s)|| \leq M_g$ and from lemma F.4 of (Fu et al., 2021) and the fact that the state action space is bounded we have that $Q_{k,J}$ will be bounded. We obtain Equation (86) from Equation (85) by using the policy gradient identity which states that $\nabla J(\lambda_k) = \mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}} \nabla log \pi_{\lambda_k}(a|s) Q^{\pi_{\lambda_k}}(s,a)$ and by using Assumption 1 which states that $||\nabla \log \pi_{\lambda_k}(a|s)|| \leq M_g$ and

Now taking the expectation over $(s,a) \sim d_\nu^{\pi_{\lambda_k}}$ on both sides of Equation (80) and substituting into it the result from Equation (86) we get

$$\delta_{\lambda_t} \leq \left(\frac{1}{t}\right) \delta_{\lambda_2} + \frac{\alpha}{t} M_g \sum_{k=0}^{k=t-2} \mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}} |Q^{\pi_{\lambda_k}}(s,a) - Q_{k,J}(s,a)| + \frac{1}{\sqrt{n}} d M_g V_{max} + \frac{L_J \cdot \alpha^2}{t} + \epsilon' \tag{87}$$

We now want to bound the term $\mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}} |Q^{\pi_{\lambda_k}}(s,a) - Q_{k,J}(s,a)|$. Let $Q_{k,j}$ denotes our estimate of the action value function at iteration k of Algorithm 1 and iteration j of the first inner for loop of Algorithm 1. $Q^{\pi_{\lambda_k}}$ denotes the action value function induced by the policy π_{λ_k}.

Consider $\epsilon_{k,j+1} = T^{\pi_{\lambda_k}} Q_{k,j} - Q_{k,j+1}$.

Thus we get,

$$\begin{aligned}
Q^{\pi_{\lambda_k}} - Q_{k,j+1} =& T^{\pi_{\lambda_k}} Q^{\pi_{\lambda_k}} - Q_{k,j+1} && (88)\\
=& T^{\pi_{\lambda_k}} Q^{\pi_{\lambda_k}} - T^{\pi_{\lambda_k}} Q_{k,j} + T^{\pi_{\lambda_k}} Q_{k,j} \\
& - Q_{k,j+1} && (89)\\
=& \gamma (P^{\pi_{\lambda_k}} (Q^{\pi_{\lambda_k}} - Q_{k,j})) + \epsilon_{k,j+1} && (90)\\
|Q^{\pi_{\lambda_k}} - Q_{k,j+1}| \leq& \gamma (P^{\pi_{\lambda_k}} (|Q^{\pi_{\lambda_k}} - Q_{k,j}|)) + |\epsilon_{k,j+1}| && (91)
\end{aligned}$$

Right hand side of Equation (88) is obtained by writing $Q^{\pi_{\lambda_k}} = T^{\pi_{\lambda_k}} Q^{\pi_{\lambda_k}}$. This is because the function $Q^{\pi_{\lambda_k}}$ is a stationary point with respect to the operator $T^{\pi_{\lambda_k}}$. Equation (89) is obtained from (88) by adding and subtracting $T^{\pi^{\lambda_k}}$. We get (91) from (90) by taking the absolute value on both sides and applying the triangle inequality on the right hand side.

By recursion on j, backwards to $j = 0$, we get,

$$|Q^{\pi_{\lambda_k}} - Q_{k,J}| \leq \sum_{j=0}^{J-2} \gamma^{J-j-1} (P^{\pi_{\lambda_k}})^{J-j-1} |\epsilon_{k,j+1}| + \gamma^J (P^{\pi_{\lambda_k}})^J (|Q^{\pi_{\lambda_k}} - Q_0|) \tag{92}$$

From this we obtain

$$\begin{aligned}
\mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}} |Q^{\pi_{\lambda_k}} - Q_{k,J}| \leq& \sum_{k=0}^{J-2} \gamma^{J-j-1} \mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}} ((P^{\pi_{\lambda_k}})^{K-J-1} |\epsilon_{k,j+1}|) \\
& + \gamma^J \mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}} (P^{\pi_{\lambda_k}})^J (|Q^{\pi_{\lambda_k}} - Q_0|) && (93)
\end{aligned}$$

For a fixed j consider the term $\mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}} ((P^{\pi_{\lambda_k}})^{J-j-1} |\epsilon_{k,j+1}|)$. We then write

$$\mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}}\left((P^{\pi_{\lambda_k}})^{J-j-1}|\epsilon_{k,j+1}|\right) \leq \left\|\frac{d(P^{\pi_{\lambda_k}})^{J-j-1}\cdot\mu_k^{'}}{d\mu_k}\right\|_\infty \int |\epsilon_{k,j+1}|\, d\mu_k \tag{94}$$

$$\leq (\phi_{\mu_k^{'},\mu_j})\mathbb{E}_{(s,a)\sim\zeta_\nu^{\pi_{\lambda_k}}(s,a)}(|\epsilon_{k,j+1}|) \tag{95}$$

Here μ_k is the measure associated with the state action distribution given by sampling from $\zeta_\nu^{\pi_{\lambda_k}}(s,a)$ and $(P^{\pi_{\lambda_k}})^{J-j-1}\cdot\mu_k^{'}$, is the measure associated with applying the operator $P^{\pi_{\lambda_k}}$, $J-j-1$ times after sampling from the distribution $(s,a)\sim d_\nu^{\pi_{\lambda_k}}$. $(\phi_{\mu_k^{'},\mu_j})$ is the supremum of the Radon Nikodym derivative of $(P^{\pi_{\lambda_k}})^{J-j-1}\cdot\mu_k^{'}$ with respect to $\zeta_\nu^{\pi_{\lambda_k}}(s,a)$. Note that in works such as (Fu et al., 2021), the Radon Nikodym derivative of with respect to the optimal policy are assumed bounded. In our case we are assuming an upper bound on a smaller quantity since $(P^{\pi_{\lambda_k}})^{J-j-1}\cdot\mu_k^{'}$ is closer to it's corresponding stationary distribution than the optimal policy since we have assumed ergodicity. Thus Equation (93) becomes

$$\mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}}|Q^{\pi_{\lambda_k}}-Q_{k,J}| \leq \sum_{k=0}^{J-2}\gamma^{J-j-1}(\phi_{\mu_k^{'},\mu_j})\mathbb{E}_{(s,a)\sim\zeta_\nu^{\pi_{\lambda_k}}(s,a)}(|\epsilon_{k,j+1}|)| + \gamma^J Q_{max} \tag{96}$$

We get the second term on the right hand side by noting that from lemma F.4 of (Fu et al., 2021) combined with the bounded state action space we have that Q_0 is bounded and $Q^{\pi_{\lambda_k}} \leq \frac{1}{1-\gamma}$. Now splitting $\epsilon_{k,j+1}$ as was done in Equation (21) we obtain

$$\begin{aligned}\mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}}|Q^{\pi_{\lambda_k}}-Q_{k,J}| \leq \sum_{j=0}^{J-2}\gamma^{J-j-1}\big((\phi_{\mu_k^{'},\mu_j})\mathbb{E}|\epsilon_{k,j+1}^1| + (\phi_{\mu_k^{'},\mu_j})\mathbb{E}|\epsilon_{k,j+1}^2| \\ + (\phi_{\mu_k^{'},\mu_j})\mathbb{E}|\epsilon_{k,j+1}^3| + (\phi_{\mu_k^{'},\mu_j})\mathbb{E}|\epsilon_{k,j+1}^4|\big) + \tilde{\mathcal{O}}(\gamma^J)\end{aligned} \tag{97}$$

The expectation on the right hand side is with respect to $(s,a)\sim\zeta_\nu^{\pi_{\lambda_k}}(s,a)$. Now using Lemmas 4, 5, 6, 7 we have

$$\mathbb{E}_{(s,a)\sim d_\nu^{\pi_{\lambda_k}}}|Q^{\pi_{\lambda_k}}-Q_{k,j}| \leq \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{n}}\right) + \tilde{\mathcal{O}}\left(L^{-\frac{1}{4}}\right) + \tilde{\mathcal{O}}\left(m^{-\frac{1}{12}}D^{\frac{7}{2}}\right) + \tilde{\mathcal{O}}(\sqrt{\epsilon_{approx}}) + \tilde{\mathcal{O}}(\gamma^J) \tag{98}$$

Plugging (98) into (87) we get

$$\begin{aligned}\delta_{\lambda_t} &\leq \left(\frac{1}{t}\right)\delta_{\lambda_2} + \frac{\alpha}{t}\sum_{k=0}^{k=t-2}\left(\tilde{\mathcal{O}}\left(\frac{1}{\sqrt{n}}\right) + \tilde{\mathcal{O}}\left(\frac{1}{L^{-\frac{1}{4}}}\right) + \tilde{\mathcal{O}}\left(m^{-\frac{1}{12}}D^{\frac{7}{2}}\right) + \tilde{\mathcal{O}}(\sqrt{\epsilon_{approx}}) + \tilde{\mathcal{O}}(\gamma^J)\right) \\ &+ \frac{1}{\sqrt{n}}dM_gV_{max} + \frac{L_J\cdot\alpha^2}{t} + \epsilon^{'} \\ &\leq \tilde{\mathcal{O}}\left(\frac{1}{t}\right) + \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{n}}\right) + \tilde{\mathcal{O}}\left(\frac{1}{L^{-\frac{1}{4}}}\right) + \tilde{\mathcal{O}}\left(m^{-\frac{1}{12}}D^{\frac{7}{2}}\right) + \tilde{\mathcal{O}}(\gamma^J) + \tilde{\mathcal{O}}(\sqrt{\epsilon_{approx}}) + \tilde{\mathcal{O}}(\sqrt{\epsilon_{bias}})\end{aligned} \tag{99}$$

□

D. Proof of Supporting Lemmas

D.1. Proof Of Lemma 4

Proof. Using Assumption 4 and the definition of $Q_{k,j}^1$ for some iteration k of Algorithm 1 we have

$$\mathbb{E}_{s,a}(T^{\pi_{\lambda_k}}Q_{k,j-1} - Q_{k,j}^1)^2 \leq \epsilon_{approx} \tag{100}$$

where $(s,a) \sim \zeta_{\nu}^{\pi_{\lambda_k}}(s,a)$.

Since $|a|^2 = a^2$ we obtain

$$\mathbb{E}(|T^{\pi_{\lambda_k}} Q_{k,j-1} - Q^1_{k,j}|)^2 \leq \epsilon_{approx} \tag{101}$$

We have for a random variable x, $Var(x) = \mathbb{E}(x^2) - (\mathbb{E}(x))^2$ hence $\mathbb{E}(x) = \sqrt{\mathbb{E}(x^2) - Var(x)}$, Therefore replacing x with $|T^{\pi_{\lambda_k}} Q^{\pi_{\lambda_k}} - Q_{k1}|$ we get

using the definition of the variance of a random variable we get

$$\mathbb{E}(|T^{\pi_{\lambda_k}} Q_{k,j-1} - Q^1_{k,j}|) = \sqrt{\mathbb{E}(|T^{\pi_{\lambda_k}} Q_{k,j-1} - Q^1_{k,j}|)^2 - Var(|T^{\pi_{\lambda_k}} Q_{k,j-1} - Q^1_{k,j}|)} \tag{102}$$

$$\leq \sqrt{\mathbb{E}(|T^{\pi_{\lambda_k}} Q_{k,j-1} - Q^1_{k,j}|)^2} \tag{103}$$

Therefore by definition of $Q^1_{k,j}$ and assumption 4 we have

$$\mathbb{E}(T^{\pi_{\lambda_k}} Q_{k,j-1} - Q^1_{k,j}|) \leq \sqrt{\epsilon_{approx}} \tag{104}$$

Since $\epsilon^1_{k,j} = T^{\pi_{\lambda_k}} Q^{\pi_{\lambda_k}} - Q_{k1}$ we have

$$\mathbb{E}(|\epsilon^1_{k,j}|) \leq \sqrt{\epsilon_{approx}} \tag{105}$$

□

D.2. Proof Of Lemma 5

Proof. From Lemma 3, we have

$$\arg\min_{f_\theta} \mathbb{E}_{x,y} \left(f_\theta(x) - g(x,y)\right)^2 = \arg\min_{f_\theta} \left(\mathbb{E}_x \left(f_\theta(x) - \mathbb{E}(g(y^{'},x)|x)\right)^2\right) \tag{106}$$

We label x to be the state action pair (s,a), y is the next state action pair denoted by $(s^{'},a^{'})$. The function $f_\theta(x)$ to be $Q_\theta(s,a)$ and $g(x,y)$ to be the function $r(s,a) + \gamma Q_{k,j-1}(s^{'},a^{'})$

Then the loss function corresponding to the lest hand side of Equation (28) becomes

$$\mathbb{E}(Q_\theta(s,a) - (r(s,a) + \gamma Q_{k,j-1}(s^{'},a^{'})))^2 \tag{107}$$

where $(s,a) \sim \zeta_{\nu}^{\pi_{\lambda_k}}(s,a)$, $s^{'} \sim P(.|(s,a))$, $a^{'} \sim \pi^{\lambda_k}(.|s^{'})$.

Therefore by Lemma 3, we have that the function $Q_\theta(s,a)$ which minimizes Equation (107) it will be minimizing

$$\mathbb{E}_{(s,a)\sim\zeta_{\nu}^{\pi_{\lambda_k}}(s,a)}(Q_\theta(s,a) - \mathbb{E}_{s''\sim P(.|s,a),a''\sim\pi^{\lambda_k}(.|s')}(r(s,a) + \gamma Q_{k,j-1}(s^{''},a^{''})|s,a))^2 \tag{108}$$

But we have from Equation (6) that

$$\mathbb{E}_{s''\sim P(.|s,a),a''\sim\pi^{\lambda_k}(.|s')}(r(s,a) + \gamma Q_{k,j-1}(s^{''},a_i^{''})|s,a) = T^{\pi_{\lambda_k}} Q_{k,j-1} \tag{109}$$

Combining Equation (107) and (109) we get

$$\arg\min_{Q_\theta} \mathbb{E}(Q_\theta(s,a) - (r(s,a) + \gamma Q_{k,j-1}(s^{'},a_i^{'})))^2 = \arg\min_{Q_\theta} \mathbb{E}(Q_\theta(s,a) - T^{\pi_{\lambda_k}} Q_{k,j-1})^2 \tag{110}$$

The left hand side of Equation (110) is $Q^2_{k,j}$ as defined in Definition 2 and the right hand side is $Q^1_{k,j}$ as defined in Definition 1, which gives us

$$Q^2_{k,j} = Q^1_{k,j} \tag{111}$$

□

D.3. Proof Of Lemma 6

Proof. We define $R_{X_k,Q_{k,j-1}}(\theta)$ as

$$R_{X_k,Q_{k,j-1}}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(Q_\theta(s_i,a_i) - \left(r(s_i,a_i) + \gamma Q_{k,j-1}(s_i^{'},a_i^{'})\right)\right)^2,$$

Here, X_k denotes the set of tuples $(s,a,s^{'},a^{'})$ sampled at iteration k of algorithm 1. They are sampled from a Markov chain whose stationary state action distribution is, $(s,a) \sim \zeta_\nu^{\pi_{\lambda_k}}$. Q_θ is is the neural network corresponding to the parameter θ and $Q_{k,j-1}$ is the estimate of the Q function obtained at iteration k of the outer for loop and iteration $j-1$ of the first inner for loop of Algorithm 1.

We also define the term

$$L_{Q_{k,j-1}}(\theta) = \mathbb{E}(Q_\theta(s,a) - (r(s,a) + \gamma Q_{k,j-1}(s',a'))^2 \tag{112}$$

where $(s,a) \sim \zeta^{\nu}_{\pi_{\lambda_k}}, s^{'} \sim P(.|(s,a)), a^{'} \sim \pi_{\lambda_k}(.|s^{'})$

We denote by $\theta^2_{k,j}, \theta^3_{k,j}$ the parameters of the neural networks $Q^2_{k,j}, Q^3_{k,j}$ respectively for notational convenience. $Q^2_{k,j}, Q^3_{k,j}$ are defined in Definition 2 and 3 respectively.

We then obtain,

$$\begin{aligned}
R_{X_k,Q_{k,j-1}}(\theta^2_{k,j}) - R_{X_k,Q_{k,j-1}}(\theta^3_{k,j}) &\leq R_{X_k,Q_{k,j-1}}(\theta^2_{k,j}) - R_{X_k,Q_{k,j-1}}(\theta^3_{k,j}) \\
&\quad + L_{Q_{k,j-1}}(\theta^3_{k,j}) - L_{Q_{k,j-1}}(\theta^2_{k,j}) && (113)\\
&= R_{X_k,Q_{k,j-1}}(\theta^2_{k,j}) - L_{Q_{k,j-1}}(\theta^2_{k,j}) \\
&\quad - R_{X_k,Q_{k,j-1}}(\theta^3_{k,j}) + L_{Q_{k,j-1}}(\theta^2_{k,j}) && (114)\\
&\leq \underbrace{|R_{X_k,Q_{k,j-1}}(\theta^2_{k,j}) - L_{Q_{k,j-1}}(\theta^2_{k,j})|}_{I} \\
&\quad + \underbrace{|R_{X_k,Q_{k,j-1}}(\theta^3_{k,j}) - L_{Q_{k,j-1}}(\theta^3_{k,j})|}_{II} && (115)
\end{aligned}$$

We get the inequality in Equation (113) because $L_{Q_{k,j-1}}(\theta^3_{k,j}) - L_{Q_{k,j-1}}(\theta^2_{k,j}) > 0$ as $Q^2_{k,j}$ is the minimizer of the loss function $L_{Q_{k,j-1}}(Q_\theta)$. We take the absolute value on both sides of (115). We can do this and the sign will remain the same as the left hand side is positive since $R_{X_k,Q_{k,j-1}}(\theta^2_{k,j}) \geq R_{X_k,Q_{k,j-1}}(\theta^3_{k,j})$ since $\theta^3_{k,j}$ is the minimizer of $R_{X_k,Q_{k,j-1}}$. Thus we get

$$|R_{X_k,Q_{k,j-1}}(\theta^2_{k,j}) - R_{X_k,Q_{k,j-1}}(\theta^3_{k,j})| \leq \underbrace{|R_{X_k,Q_{k,j-1}}(\theta^2_{k,j}) - L_{Q_{k,j-1}}(\theta^2_{k,j})|}_{I} + \underbrace{|R_{X_k,Q_{k,j-1}}(\theta^3_{k,j}) - L_{Q_{k,j-1}}(\theta^3_{k,j})|}_{II} \tag{116}$$

Consider Lemma 2. The loss function $R_{X_k,Q_{k,j-1}}(\theta)$ can be written as the mean of loss functions of the form $l(a_\theta(s_i, a_i, s_i^{'}, a_i^{'}), y_i)$ where l is the square function. $a_\theta(s_i, a_i, s_i^{'}, a_i^{'}) = Q_\theta(s_i, a_i)$ and $y_i = \Big(r(s_i, a_i) + \gamma Q_{k,j-1}(s_i^{'}, a_i^{'})\Big)$. Thus we have

$$\mathbb{E}\sup_{\theta\in\Theta''} |R_{X_k,Q_{k,j-1}}(\theta) - L_{Q_{k,j-1}}(\theta)| \leq 2\eta^{'}\mathbb{E}\Big(Rad(\mathcal{A}\circ\{(s_1, a_1, s_1^{'}, a_1^{'}), (s_2, a_2, s_2^{'}, a_2^{'}), \cdots, (s_n, a_n, s_n^{'}, a_n^{'})\})\Big) \tag{117}$$

Note that the expectation is over all $(s_i, a_i, s_i^{'}, a_i^{'})$ and the parameter set is $\Theta'' = \{\theta^2_{k,j}, \theta^3_{k,j}\}$. We use this set because we only need this inequality to hold for $Q^2_{k,j}$ and $Q^3_{k,j}$. Here $n = |X_k|$, $(\mathcal{A}\circ\{(s_1, a_1, s_1^{'}, a_1^{'}), (s_2, a_2, s_2^{'}, a_2^{'}), \cdots, (s_n, a_n, s_n^{'}, a_n^{'}) : \theta\in\Theta''\} = \{Q_\theta(s_1, a_1), Q_\theta(s_2, a_2), \cdots, Q_\theta(s_n, a_n) : \theta\in\Theta''\}$ and $\eta_i^{'}$ is the Lipschitz constant for the square function over the state action space.

Now as is shown in Proposition 11 of (Bertail & Portier, 2019) we have that

$$\Big(Rad(\mathcal{A}\circ\{(s_1, a_1, s_1^{'}, a_1^{'}), (s_2, a_2, s_2^{'}, a_2^{'}), \cdots, (s_n, a_n, s_n^{'}, a_n^{'})\})\Big) \leq \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{n}}\right) \tag{118}$$

Note that in (Bertail & Portier, 2019) the term of the form $\sum_{i=1}^n (f(x_i) - \mathbb{E}f(x))$ has been upper bounded by a factor of $\tilde{\mathcal{O}}(\sqrt{n})$. The way we have defined $R_{X_k,Q_{k,j-1}}$ means we have a term of the form $\frac{1}{n}\sum_{i=1}^n (f(x_i) - \mathbb{E}f(x))$ on the left hand side of Equation (117) which implies that it is upper bounded by a term of the form $\tilde{\mathcal{O}}\left(\frac{1}{\sqrt{n}}\right)$.

We use this result as the state action pairs are drawn not from the stationary state of the policy π_{λ_k} but from a Markov chain with the same steady state distribution. Thus we have for $\theta = \theta^2_{k,j}$

$$\mathbb{E}|(R_{X_k,Q_{k,j-1}}(\theta^2_{k,j})) - L_{Q_{k,j-1}}(\theta^2_{k,j})| \leq \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{n}}\right) \tag{119}$$

The same argument can be applied for $\theta = \theta^3_{k,j}$ to get

$$\mathbb{E}|(R_{X_k,Q_{k,j-1}}(\theta^3_{k,j})) - L_{Q_{k,j-1}}(\theta^3_{k,j})| \leq \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{n}}\right) \tag{120}$$

Then, plugging Equation (119),(120) into Equation (116) we have

$$\mathbb{E}\left|R_{X_k,Q_{k,j-1}}(\theta^2_{k,j}) - R_{X_k,Q_{k,j-1}}(\theta^3_{k,j})\right| \leq \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{n}}\right) \tag{121}$$

Plugging in the definition of $R_{X_k,Q_{k,j-1}}(\theta^2_{k,j}), R_{X_k,Q_{k,j-1}}(\theta^3_{k,j})$ in equation (120), (121) into (116) and denoting $\tilde{\mathcal{O}}\left(\frac{1}{\sqrt{n}}\right)$ as ϵ we get

$$\frac{1}{n}\sum_{i=1}^n \Big(\mathbb{E}|(Q^2_{k,j}(s_i, a_i) - (r(s_i, a_i) + \gamma Q_{k,j-1}(s_i^{'}, a_i^{'})))^2 - (Q^3_{k,j}(s_i, a_i) - (r(s_i, a_i) + \gamma Q_{k,j-1}(s_i^{'}, a_i^{'})))^2|\Big) \leq \epsilon \tag{122}$$

Now for a fixed i consider the term α_i defined as.

$$\mathbb{E}|(Q^2_{k,j}(s_i,a_i) - (r(s_i,a_i) + \gamma Q_{k,j-1}(s_i^{'},a_i^{'})))^2 - (Q^3_{k,j}(s_i,a_i) - (r(s_i,a_i) + \gamma Q_{k,j-1}(s_i^{'},a_i^{'})))^2| \tag{123}$$

where $s_i, a_i, s_i^{'}, a_i^{'}$ are drawn from the state action distribution at the ith step of the Markov chain induced by following the policy π_{λ_k}.

Now for a fixed i consider the term β_i defined as.

$$\mathbb{E}|(Q^2_{k,j}(s_i,a_i) - (r(s_i,a_i) + \gamma Q_{k,j-1}(s_i^{'},a_i^{'})))^2 - (Q^3_{k,j}(s_i,a_i) - (r(s_i,a_i) + \gamma Q_{k,j-1}(s_i^{'},a_i^{'})))^2| \tag{124}$$

where s_i, a_i for all i are drawn from the steady state action distribution with $(s_i,a_i) \sim \zeta^{\nu}_{\pi_{\lambda_k}}$, $s_i^{'} \sim P(.|s,a)$ and $a_i^{'} \sim \pi^{\lambda_k}(.|s_i^{'})$. Note here that α_i and β_i are the same function with only the state action pairs being drawn from different distributions.

Using these definitions we obtain

$$\begin{aligned} |\mathbb{E}(\alpha_i) - \mathbb{E}(\beta_i)| &\leq \sup_{(s_i,a_i,s_i^{'},a_i^{'})} |2.\max(\alpha_i,\beta_i)|(\kappa_i) && (125)\\ &\leq R_{max}.p\rho^i && (126) \end{aligned}$$

We obtain Equation (125) by using the identity $|\int f d\mu - \int f d\nu| \leq |\max(f)|\int|(d\mu - d\nu)| \leq |\max(f)|\cdot d_{TV}(\mu,\nu)$, where μ and ν are two σ finite state action probability measures and f is a bounded measurable function. We have used κ_i to represent the total variation distance between the measures of $(s,a,s^{'},a^{'})$ induced by sampling form the steady state action distribution denoted by $(s,a) \sim \zeta^{\nu}_{\pi_{\lambda_k}}$ and the measures of $(s_i,a_i,s_i^{'},a_i^{'})$ induced at the ith step of the Markov chain induced by following the policy π^{λ_k}. We obtain Equation (126) from Equation (125) by using Assumption 2 and the fact from lemma F.4 of (Fu et al., 2021) combined with the bounded state action space we have that $\sup_{(s_i,a_i)} |2.\max(\alpha_i,\beta_i)|$ will be bounded.

From equation (126) we get

$$\mathbb{E}(\beta_i) \leq \mathbb{E}(\alpha_i) + R_{max}.p\rho^i \tag{127}$$

We get Equation (127) from Equation (126) using the fact that $|a-b| \leq c$ implies that $(-c \geq (b-a) \leq c)$ which in turn implies $b \leq a + c$.

Using Equation (127) in equation (122) we get

$$\begin{aligned} &\frac{1}{n}\sum_{i=1}^{n}\Big(\mathbb{E}|(Q^2_{k,j}(s_i,a_i) - (r(s_i,a_i) + \gamma Q_{k,j-1}(s_i^{'},a_i^{'})))^2 - (Q^3_{k,j}(s_i,a_i) - (r(s_i,a_i) + \gamma Q_{k,j-1}(s_i^{'},a_i^{'})))^2|\Big)\\ \leq\ & \epsilon + \frac{1}{n}\sum_{i=1}^{n} R_{max}.p\rho^i\\ \leq\ & \epsilon + \frac{R_{max}}{n}p\frac{1}{1-\rho} \end{aligned} \tag{128}$$

In Equation (128) $(s_i,a_i) \sim \zeta^{\nu}_{\pi_{\lambda_k}}, s^{'} \sim P(.|s,a), a \sim \pi^{\lambda_k}(.|s^{'})$ for all i.

Since now all terms in the summation on the left hand side of Equation (128) are i.i.d, Equation (128) is equivalent to,

$$\mathbb{E}|(Q^2_{k,j}(s,a) - (r(s,a) + \gamma Q_{k,j-1}(s^{'},a^{'})))^2 - (Q^3_{k,j}(s_i,a_i) - (r(s,a) + \gamma Q_{k,j-1}(s^{'},a^{'})))^2| \quad \leq \quad \epsilon \tag{129}$$

Now we apply an argument similar to the *state regularity assumption* of (Tian et al., 2023) wherein it is assumed that there exists a positive constant $\lambda^{'}$ such that $|Q_{\theta_1} - Q_{\theta_2}| \geq \lambda^{'}|\theta_1 - \theta_2|$ for any $\theta_1, \theta_2 \in \Theta^{'}$.

Using the same argument for the function $F_{\theta'}(\theta) = (Q_\theta(s,a) - (r(s,a) + \gamma Q_{\theta'}(s^{'},a^{'})))^2$, we assume there exists a positive constant $\lambda_{\theta'}$ such that $|F_{\theta'}(\theta_1) - F_{\theta'}(\theta_2)| \geq \lambda_{\theta'}|\theta_1 - \theta_2|$ for any $\theta_1, \theta_2 \in \Theta^{'}$. We then define $\lambda^{''} = \inf_{\theta' \in \Theta'} \lambda_{\theta'}$.

Further we use the lipschitz property of neural networks to obtain that $|Q_{\theta_1} - Q_{\theta_2}| \leq L_Q|\theta_1 - \theta_2|$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, where L_Q is the lipschitz parameter of the neural networks where $\theta \in \Theta^{'}$. Combining these two results we obtain that $L_Q.\lambda^{''}|Q_{\theta_1} - Q_{\theta_2}| \leq |F_{\theta'}(\theta_1) - F_{\theta'}(\theta_2)|$ for any $\theta^{'} \in \Theta^{'}$.

Applying this result to Equation (129) we get

$$\lambda^{''}.L_Q\mathbb{E}|Q^2_{k,j} - Q^3_{k,j}| \quad \leq \quad \epsilon \tag{130}$$

where now the expectation is only over $(s,a) \sim \zeta^{\nu}_{\pi_{\lambda_k}}$ since the left hand side is no longer a function of $s^{'}, a^{'}$. This is equivalent to

$$\mathbb{E}_{(s,a)\sim\zeta^{\nu}_{\pi_{\lambda_k}}}|Q^2_{k,j} - Q^3_{k,j}| \quad \leq \quad \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{n}}\right) \tag{131}$$

which is the required result

□

D.4. Proof Of Lemma 7

Proof. Consider the loss function in Definition 3 given by

$$\frac{1}{n}\sum_{i=1}^{n}\Big(Q_\theta(s_i,a_i) - \big(r(s,a) + \gamma Q_{k,j-1}(s_i^{'},a_i^{'})\big)\Big)^2 \tag{132}$$

We can consider the function in Equation (132) as an expectation of the loss function $\Big(Q_\theta(s,a) - \big(r(s,a) + \gamma Q_{k,j-1}(s^{'},a^{'})\big)\Big)^2$ where there is an equal probability $\frac{1}{n}$ on each of the observed tuples $(s_i, a_i, s_i^{'}, a_i^{'})$ obtained at the kth iteration of the outer for loop of Algorithm 1.

We define the local linearization of a function Q_θ as follows

$$\bar{Q}_\theta = Q_{\theta_0} + (\theta - \theta_0)^T \nabla Q_\theta \tag{133}$$

Following the analysis in Proposition C.4 of (Fu et al., 2021) we obtain from Equation G.49 that over the random initialization of θ_0 we have with probability at least at least $1 - \exp(-\Omega(m^{-\frac{2}{3}})D^{-\frac{2}{3}})$.

$$\mathbb{E}(\bar{Q}_{\theta'} - \bar{Q}_{\theta^*})^2 = \tilde{\mathcal{O}}(L^{-\frac{1}{2}}) + \tilde{\mathcal{O}}(m^{-\frac{1}{6}} D^7) \tag{134}$$

Here $\theta^{'} = \frac{1}{L}\sum_{i=1}^{L}\theta_i$ and θ^* is the minimizer of the loss function in Equation (133) and the expectation is over the finite measure where each tuples (s_i, a_i) has a probability mass of $\frac{1}{n}$.

Now we have from Jensen's Inequality

$$\mathbb{E}|\bar{Q}_{\theta'} - \bar{Q}_{\theta^*}| \leq \sqrt{\mathbb{E}(\bar{Q}_{\theta'} - \bar{Q}_{\theta^*})^2} = \tilde{\mathcal{O}}(L^{-\frac{1}{4}}) + \tilde{\mathcal{O}}(m^{-\frac{1}{12}} D^{\frac{7}{2}}) \tag{135}$$

Consider Lemma F.3 from (Fu et al., 2021), we have with probability at $1 - \exp(-\Omega(m^{-\frac{2}{3}})D^{-\frac{2}{3}})$ over the random initialization of θ_0 that for any $\theta \in \Theta^{'}$

$$|Q_\theta(s,a) - \bar{Q}_\theta(s,a)| \leq \tilde{\mathcal{O}}(m^{-\frac{1}{6}} D^{\frac{5}{2}}) \tag{136}$$

Thus we have

$$\begin{aligned}
\mathbb{E}|Q_{\theta'} - Q_{\theta^*}| &\leq \mathbb{E}|Q_{\theta'} - \bar{Q}_{\theta'} + \bar{Q}_{\theta'} - Q_{\theta^*} - \bar{Q}_{\theta^*} + \bar{Q}_{\theta^*}| && (137)\\
&\leq \mathbb{E}|\bar{Q}_{\theta'} - \bar{Q}_{\theta^*}| + |Q_{\theta'} - \bar{Q}_{\theta'}| + |Q_{\theta^\star} - \bar{Q}_{\theta^\star}| && (138)\\
&\leq \tilde{\mathcal{O}}(L^{-\frac{1}{4}}) + \tilde{\mathcal{O}}(m^{-\frac{1}{12}} D^{\frac{7}{2}}) && (139)
\end{aligned}$$

Now from theorem 6.10 of (Rudin, 1987) we have that there exists a positive function $f_j : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ such that

$$\mathbb{E}^{'}|Q_{\theta'} - Q_{\theta^*}| = \mathbb{E}|f_j \cdot (Q_{\theta'} - Q_{\theta^*})| \tag{140}$$

Here $\mathbb{E}^{'}$ is the expectation with respect to the state action pair sampled from $(s,a) \sim \zeta_\nu^{\pi_{\lambda_k}}$. Here the function f_j is bounded function because $Q_{\theta'}$ and Q_{θ^*} are bounded functions since there parameters are bounded as shown in lemma F.4 of (Fu et al., 2021) and the fact that the state action space is bounded.

Thus we can say

$$\mathbb{E}_{(s,a)\sim\zeta_\nu^{\pi_{\lambda_k}}}|Q_{\theta'} - Q_{\theta^*}| \leq (\sup f_j)\mathbb{E}|Q_{\theta'} - Q_{\theta^*}| \leq \tilde{\mathcal{O}}(L^{-\frac{1}{4}}) + \tilde{\mathcal{O}}(m^{-\frac{1}{12}} D^{\frac{7}{2}}) \tag{141}$$

Note that $\theta^{'}$ is the parameter we obtain at the end of critic step denoted by $\theta_{k,j}$ and θ^* is by definition $\theta^3_{k,j}$. Therefore we get the required result.

$$\mathbb{E}_{(s,a)\sim\zeta_\nu^{\pi_{\lambda_k}}}|Q_{\theta^3_{k,j}} - Q_{\theta_{k,j}}| \leq \tilde{\mathcal{O}}(L^{-\frac{1}{4}}) + \tilde{\mathcal{O}}(m^{-\frac{1}{12}} D^{\frac{7}{2}}) \tag{142}$$

□