On the Global Convergence of Fitted Q-Iteration with Two-layer Neural Network Parametrization

Mudit Gaur [1] **Vaneet Aggarwal** [2 3] **Mridul Agarwal** [4]

Abstract

Deep Q-learning based algorithms have been applied successfully in many decision making problems, while their theoretical foundations are not as well understood. In this paper, we study a Fitted Q-Iteration with two-layer ReLU neural network parameterization, and find the sample complexity guarantees for the algorithm. Our approach estimates the Q-function in each iteration using a convex optimization problem. We show that this approach achieves a sample complexity of $\tilde{\mathcal{O}}(1/\epsilon^2)$, which is order-optimal. This result holds for a countable state-spaces and does not require any assumptions such as a linear or low rank structure on the MDP.

1. Introduction

Reinforcement learning aims to maximize the cumulative rewards wherein an agent interacts with the system in a sequential manner, and has been used in many applications including games (Silver et al., 2017; Vinyals et al., 2017; Bonjour et al., 2022), robotics (Maes et al., 2003), autonomous driving (Kiran et al., 2022), ride-sharing (Al-Abbasi et al., 2019), networking (Geng et al., 2020), recommender systems (Warlop et al., 2018), etc. One of the key class of algorithms used in reinforcement learning are those based on Q-learning. Due to large state space (possibly infinite), such algorithms are parameterized (Van Hasselt et al., 2016). However, with general parametrization (e.g., neural network parametrization), sample complexity guarantees to achieve an ϵ gap to the optimal Q-values have not been fully understood. In this paper, we study this problem to provide the first sample complexity results for Q-learning based algorithm with neural network parametrization.

We note that most of the theoretical results for Q-learning based algorithms have been in tabular setups which assume finite states such as in (Li et al., 2020; Jin et al., 2018), or assume a linear function approximation (Carvalho et al., 2020; Wang et al., 2021a; Chen et al., 2022; Zhou et al., 2021) where the linear representation is assumed to be known. Although practically large state spaces are handled using deep neural networks due to the universal approximation theorem, the theoretical analysis with neural network parametrization of Q-learning is challenging as the training of neural network involves solving a non-convex optimization problem. Recently, the authors of (Fan et al., 2020) studied the problem of neural network parametrization for Q-learning based algorithms, and provided the asymptotic global convergence guarantees when the parameters of the neural network used to approximate the Q function are large. In this paper, we study the first results on sample complexity to achieve the global convergence of Q-learning based algorithm with neural network parametrization.

In this paper, we propose a fitted Q-Iteration based algorithm, wherein the Q-function is parametrized by a two-layer ReLU network. The key for the two-layer ReLU network is that finding the optimal parameters for estimating the Q-function could be converted to a convex problem. In order to reduce the problem from non-convex to convex, one can replace the ReLU function with an equivalent diagonal matrix. Further, the minima of both the non-convex ReLU optimization and the equivalent convex optimization are same (Wang et al., 2021b; Pilanci and Ergen, 2020).

We find the sample complexity for fitted Q-Iteration based algorithm with neural network parametrization. The gap between the learnt Q-function and the optimal Q-function arises from the parametrization error due to the 2-layer neural network, the error incurred due to imperfect reformulation of the neural network as a convex optimization problem as well as the error due to the random nature of the underlying MDP. This error reduces with increasing the number of iterations of the fitted Q-iteration algorithm, the number of iterations of the convex optimization step, the number of data points sampled at each step. We achieve

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

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

an overall sample complexity of $\tilde{\mathcal{O}}\left(\frac{1}{(1-\gamma)^6\epsilon^2}\right)$. Our proof consists of expressing the error in estimation of the Q-function at a fixed iteration in terms of the Bellman error incurred at each successive iteration upto that point. We then express the Bellman error in terms of the previously mentioned components for the statistical error. We then upper bound the expectation of these error components with respect to a distribution of the state action space.

2. Related Works

2.1. Fitted Q-Iteration

The analysis of Q-learning based algorithms has been studied since the early 1990's (Watkins and Dayan, 1992; Singh and Yee, 1994), where it was shown that small errors in the approximation of a task's optimal value function cannot produce arbitrarily bad performance when actions are selected by a greedy policy. Value Iteration algorithms, and its analogue Approximate Value iteration for large state spaces has been shown to have finite error bounds as (Puterman, 2014). Similar analysis for finite state space has been studied in (Bertsekas and Tsitsiklis, 1996) and (Munos, 2007).

When the state space is large (possibly infinite), the value function can not be updated at each state action pair at every iteration. Thus, approximate value iteration algorithms are used that obtains samples from state action space at each iteration and estimates the action value function by minimizing a loss which is a function of the sampled state action pairs. This is the basis of the Fitted Q-Iteration, first explained in (Boyan and Moore, 1994). Instead of updating the Q-function at each step of the trajectory, it collects a batch of sample transitions and updates the Q-functions based on the collected samples and the existing estimate of the Q-function. The obtained samples could be using a generative model (Munos, 2003; Ghavamzadeh et al., 2008), or using the buffer memory (Kozakowski et al., 2022; Wang et al., 2020).

2.2. Deep Neural Networks in Reinforcement Learning

Parametrization of Q-network is required for scaling the Q-learning algorithms to a large state space. Neural networks have been used previously to parameterize the Q-function (Tesauro et al., 1995; Xu and Gu, 2020; Fujimoto et al., 2019). This approach, also called Deep Q-learning has been widely used in many reinforcement learning applications (Yang et al., 2020; Damjanović et al., 2022; Gao et al., 2020). However, fundamental guarantees of Q-learning with such function approximation are limited, because the non-convexity of neural network makes the parameter optimization problem non-convex.

Even though the sample complexity of Q-learning based algorithms have not been widely studied for general parametrization, we note that such results have been studied widely for policy gradient based approaches (Agarwal et al., 2021; Wang et al., 2019; Zhang et al., 2022). The policy gradient approaches directly optimize the policy, while still having the challenge that the parametrization makes the optimization of parameters non-convex. This is resolved by assuming that the class of parametrization (e.g., neural networks) is rich in the sense that arbitrary policy could be approximated by the parametrization. However, note that for systems with infinite state spaces, results for policy gradient methods consist of upper bounds on the gradient of the estimate of the average reward function, such as in (Yuan et al., 2022). For upper bounds on the error of estimation of the value function, linear structure on the MDP has to be assumed, as is done in (Chen and Zhao, 2022).

In the case of analysis of Fitted Q-Iteration (FQI) algorithms such as (Fan et al., 2020), upper bounds on the error of estimation of the Q-function are obtained by considering sparse Neural networks with ReLU functions and Holder smooth assumptions on the Neural networks. At each iteration of the FQI algorithm, an estimate of the Q-function is obtained by optimizing a square loss error, which is non-convex in the parameters of the neural network used to represent the Q-function. Due to this non-convexity, the upper bounds are asymptotic in terms of the parameters of the neural network and certain unknown constants. (Xu and Gu, 2020) improves upon this result by demonstrating finite time error bounds for Q-learning with neural network parametrization. However the error bounds obtained can go unbounded as the number of iterations increase (See Appendix A), hence they do not give any sample complexity results. Our result establishes the first sample complexity results for a (possibly) infinite state space without the need for a linear structure on the MDP.

2.3. Neural Networks Parameter Estimation

The global optimization of neural networks has shown to be NP hard (Blum and Rivest, 1992). Even though Stochastic Gradient Descent algorithms can be tuned to give highly accurate results as in (Bengio, 2012), convergence analysis of such methods requires assumptions such as infinite width limit such as in (Zhu and Xu, 2021). Recently, it has been shown that the parameter optimization for two-layer ReLU neural networks can be converted to an equivalent convex program which is exactly solvable and computationally tractable (Pilanci and Ergen, 2020). Convex formulations for convolutions and deeper models have also been studied (Sahiner et al., 2020a;b). In this paper, we will use these approaches for estimating the parameterized Q-function.

3. Problem Setup

We study a discounted Markov Decision Process (MDP), which is described 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 finite set of actions, we represent set of state action pairs as $[0,1]^d$ (a d dimensional tuple with all elements in [0,1]), where d is a positive integer greater than 1. $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the probability transition kernel

$R : \mathcal{S} \times \mathcal{A} \to \mathcal{P}([0, R_{\max}])$ is the reward kernel on the state action space with $R_{\max}$ being the absolute value of the maximum reward and $0 < \gamma < 1$ is the discount factor. A policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ is a mapping that maps state to a probability distribution over the action space. Here, we denote by $\mathcal{P}(\mathcal{S}), \mathcal{P}(\mathcal{A}), \mathcal{P}([a,b])$, the set of all probability distributions over the state space, the action space, and a closed interval $[a,b]$, respectively. We define the action value function for a given policy π respectively as follows.

$$Q^\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r'(s_t, a_t) | s_0 = s, a_0 = a\right], \quad (1)$$

where $r'(s_t, a_t) \sim R(\cdot|s_t, a_t)$, $a_{t+1} \sim \pi(\cdot|s_{t+1})$ 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 follows:

$$Q^*(s,a) = \sup_\pi Q^\pi(s,a) \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}, \quad (2)$$

A policy that achieves the optimal action value functions is known as the *optimal policy* and is denoted as π^*. It can be shown that π^* is the greedy policy with respect to Q^* (Bertsekas and Shreve, 2007). Hence finding Q^* is sufficient to obtain the optimal policy. We define the Bellman operator for a policy π as follows

$$(T^\pi Q)(s,a) = r(s,a) + \gamma \int Q^\pi(s', \pi(s')) P(ds'|s,a), \quad (3)$$

where $r(s,a) = \mathbb{E}(r'(s,a)|(s,a))$ Similarly we define the Bellman Optimality Operator as

$$(TQ)(s,a) = \left(r(s,a) + \max_{a' \in \mathcal{A}} \gamma \int Q(s',a') P(ds'|s,a)\right), \quad (4)$$

Further, operator P^π is defined as

$$P^\pi Q(s,a) = \mathbb{E}[Q(s',a')|s' \sim P(\cdot|s,a), a' \sim \pi(\cdot|s')], \quad (5)$$

which 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, $P^{\pi_1} P^{\pi_2} \cdots P^{\pi_m}$ is the m-step Markov transition operator following policy π_t at steps $1 \le t \le m$. It defines a distribution on the state action space after m transitions from the initial state. We have the relation

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

Which defines P^* as

$$P^* Q(s,a) = \max_{a' \in \mathcal{A}} \mathbb{E}[Q(s',a')|s' \sim P(\cdot|s,a)], \quad (7)$$

in other words, P^* is the one step Markov transition operator with respect to the greedy policy of the function on which it is acting.

This gives us the relation

$$(TQ)(s,a) = r(s,a) + \gamma P^* Q(s,a), \quad (8)$$

For any measurable function $f : \mathcal{S} \times \mathcal{A} :\to \mathbb{R}$, we also define

$$\mathbb{E}(f)_\nu = \int_{\mathcal{S} \times \mathcal{A}} f d\nu, \quad (9)$$

for any distribution $\nu \in \mathcal{P}(\mathcal{S} \times \mathcal{A})$.

We define $\rho^\pi_{s_0}$ as the stationary state distribution induced by the policy π starting at state s_0 and $\zeta^\pi_{s_0}(s,a)$ is the corresponding stationary state action distribution defined as $\zeta^\pi_{s_0}(s,a) = \rho^\pi_{s_0}(s).\pi(a|s)$.

Similarly, we define $\rho^\pi_\nu(s)$ as the stationary state distribution induced by the policy π starting at state $s_0 \sim \nu$ and $\zeta^\pi_\nu(s,a) = \rho^\pi_\nu(s).\pi(a|s)$ as the corresponding stationary state action distribution.

For representing the action value function, we will use a 2 layer ReLU neural network. A 2-layer ReLU Neural Network with input $x \in \mathbb{R}^d$ is defined as

$$f(x) = \sum_{i=1}^m \sigma'(x^T u_i) \alpha_i, \quad (10)$$

where $m \ge 1$ is the number of neurons in the neural network, the parameter space is $\Theta_m = \mathbb{R}^{d \times m} \times \mathbb{R}^m$ and $\theta = (U, \alpha)$ is an element of the parameter space, where u_i is the ith column of U, and α_i is the ith coefficient of α. The function $\sigma' : \mathbb{R} \to \mathbb{R}_{\ge 0}$ is the ReLU or restricted linear function defined as $\sigma'(x) \triangleq \max(x, 0)$. In order to obtain parameter θ for a given set of data $X \in \mathbb{R}^{n \times d}$ and the corresponding response values $y \in \mathbb{R}^{n \times 1}$, we desire the parameter that minimizes the squared loss , given by

$$\mathcal{L}(\theta) = \arg\min_\theta ||\sum_{i=1}^m \sigma(Xu_i)\alpha_i - y||_2^2 + \beta \sum_{i=1}^m ||u_i||_2^2 \quad (11)$$

Here, we have the term $\sigma(Xu_i)$ which is a vector $\{\sigma'((x_j)^T u_i)\}_{j\in\{1,\cdots,n\}}$ where x_j is the jth row of X. It is the ReLU function applied to each element of the vector Xu_i. We note that the optimization in Equation (11) is non-convex in θ due to the presence of the ReLU activation function. In (Wang et al., 2021b), it is shown that this optimization problem has an equivalent convex form, provided that the number of neurons m goes above a certain threshold value. This convex problem is obtained by replacing the ReLU functions in the optimization problem with equivalent diagonal operators. The convex problem is given as

$$\mathcal{L}_\beta^{'}(p) \quad := \quad (\arg\min_p || \sum_{D_i \in D_X} D_i(Xp_i) - y||_2^2 \quad (12)$$

where $p \in \mathbb{R}^{d\times|D_X|}$.

D_X is the set of diagonal matrices D_i which depend on the dataset X. Except for cases of X being low rank it is not computationally feasible to obtain the set D_X. We instead use $\tilde{D} \in D_X$ to solve the convex problem

$$\mathcal{L}_\beta^{'}(p) \quad := \quad (\arg\min_p || \sum_{D_i \in \tilde{D}} D_i(Xp_i) - y||_2^2 \quad (13)$$

where $p \in \mathbb{R}^{d\times|\tilde{D}|}$.

The relevant details of the formulation and the definition of the diagonal matrices D_i are provided in Appendix B. For a set of parameters $\theta = (u, \alpha) \in \Theta$, we denote neural network represented by these parameters as

$$Q_\theta(x) = \sum_{i=1}^{m} \sigma'(x^T u_i)\alpha_i \quad (14)$$

4. Proposed Algorithm

In this section, we describe our Neural Network Fitted Q-iteration algorithm. The key in the algorithm is the use of convex optimization for the update of parameters of the Q-function. The algorithm, at each iteration k, updates the estimate of the Q function, here denoted as Q_k. The update at each step in the ideal case is to be done by applying the Bellman optimality operator defined in Equation (3). However, there are two issues which prevent us from doing that. First, we do not know the transition kernel P. Second for the case of an infinite state space, we cannot update the estimate at each iteration of the state space. Therefore, we apply an approximation of the Bellman optimality operator defined as

$$\hat{T}Q(s,a) = \left(r'(s,a) + \max_{a'\in\mathcal{A}} \gamma Q(s',a')\right), \quad (15)$$

Algorithm 1 Iterative algorithm to estimate Q function

Input: $\mathcal{S}$, $\mathcal{A}, \gamma$, Time Horizon K $\in \mathcal{Z}$, sampling distribution ν, one step transition operator κ, $T_{k:k\in\{1,\cdots,K\}}$

Initialize: $\tilde{Q}(s,a) = 0\ \forall(s,a) \in \mathcal{S}\times\mathcal{A}$

1: **for** $k \in \{1,\cdots,K\}$ **do**
2: Generate n_k state-action pairs by (i) sampling from the initial state-action distribution ν and (ii) applying the greedy policy based on the previously estimated Q_{k-1} values, denoted as π_k
3: Obtain $\{s_i', r_i'\}$ from $\kappa(s_i, a_i) = \{s_i', r_i'\}$
4: Set $y_i = r_i' + \gamma \max_{a'\in\mathcal{A}} \tilde{Q}(s_i', a')$, where $i \in \{1,\cdots,n\}$
5: Set X_k, Y_k as the matrix of the sampled state action pairs and vector of estimated Q values respectively
6: **Call Algorithm** 2 with input ($X = X_k$, $y = Y_k$, $T = T_k$) and return parameter θ
7: $\tilde{Q} = Q_\theta$
8: **end for**
Define π_{K+1} as the greedy policy with respect to $\tilde{Q}$
Output: An estimator $\tilde{Q}$ of Q^* and π_k as it's greedy policy

where $r'(s,a) \sim R(.|s,a)$. Since we cannot perform even this approximated optimality operator for all state action pairs due to the possibly infinite state space, we instead update our estimate of the Q function at iteration k as the 2 layer ReLU Neural Network which minimizes the following loss function

$$\frac{1}{n}\sum_{i=1}^{n}\left(Q_\theta(s_i,a_i) - \left(r'(s_i,a_i) + \max_{a'\in\mathcal{A}} \gamma Q_{k-1}(s',a')\right)\right)^2, \quad (16)$$

Here Q_{k-1} is the estimate of the Q function from iteration $k-1$ and the state action pairs have been sampled from some distribution, ν, over the state action pairs. Note that this is a problem of the form given in Equation (11) with $y_i = (r'(s_i,a_i) + \max_{a'\in\mathcal{A}} \gamma Q_{k-1}(s',a'))$ where $i \in (1,\cdots,n)$ and Q_θ represented as in Equation (14).

We define the Bellman error at iteration k as

$$\epsilon_k = TQ_{k-1} - Q_k, \quad (17)$$

The main algorithm, Algorithm 1, iteratively samples from the state action space at every iteration, and the corresponding reward is observed. For each of the sampled state action pairs, the approximate Bellman optimality operator given in Equation (15) is applied to obtain the corresponding output value y. We have access to these values under Assumption 3. The sampled set of state action pairs and the corre-

sponding y values are then passed to Algorithm 2, which returns the estimates of the neural network which minimizes the loss given in Equation (16). The algorithm updates the estimate of the action value function as the neural network corresponding to the estimated parameters.

Algorithm 2 Neural Network Parameter Estimation

1: **Input:** data (X, y, T)
2: **Sample:** $\tilde{D} = diag(1(Xg_i > 0)) : g_i \sim \mathcal{N}(0, I), i \in [|\tilde{D}|]$
3: **Initialize** $y^1 = 0, u^1 = 0$
Initialize $g(u) = || \sum_{D_i \in \tilde{D}} D_i X u_i - y||_2^2$
4: **for** $k \in \{0, \cdots, T\}$ **do**
5: $u^{k+1} = y_k - \alpha_k \nabla g(y_k)$
6: $y^{k+1} = \arg\min_{y:|y|_1 \leq \frac{R_{max}}{1-\gamma}} ||u_{k+1} - y||_2^2$
7: **end for**
8: Set $u^{T+1} = u^*$
9: Solve Cone Decomposition:
$\bar{v}, \bar{w} \in u_i^* = v_i - w_i, i \in [d]\}$ such that $v_i, w_i \in \mathcal{K}_i$ and at-least one v_i, w_i is zero.
10: Construct $(\theta = \{u_i, \alpha_i\})$ using the transformation

$$\psi(v_i, w_i) = \begin{cases} (v_i, 1), & \text{if } w_i = 0 \\ (w_i, -1), & \text{if } v_i = 0 \\ (0, 0), & \text{if } v_i = w_i = 0 \end{cases} \quad (18)$$

for all $i \in \{1, \cdots, m\}$
11: Return θ

Algorithm 2 optimizes the parameters for the neural network at each step of Algorithm 1. This is performed by reducing the problem to an equivalent convex problem as described in Appendix B. The algorithm first samples a set of diagonal matrices denoted by $\tilde{D}$ in line 2 of Algorithm 2. The elements of $\tilde{D}$ act as the diagonal matrix replacement of the ReLU function. Algorithm 2 then solves an optimization of the form given in Equation (13). This convex optimization is solved in Algorithm 2 using the projected gradient descent algorithm. After obtaining the optima for this convex program, denoted by $u^* = \{u_i^*\}_{i \in \{1, \cdots, |\tilde{D}|\}}$, in line 10, we transform them to the parameters of a neural network of the form given in Equation (14) which are then passed back to Algorithm 1. The procedure is described in detail along with the relevant definitions in Appendix B.

5. Error Characterization

We now characterize the errors which can result in the gap between the point of convergence and the optimal Q-function. To define the errors, we first define the various possible Q-functions which we can approximate in decreasing order of the accuracy.

We start by defining the best possible Q-function, Q_{k_1} for episode $k > 1$. Q_{k_1} is the best approximation of the function TQ_{k-1} possible from the class of two layer ReLU neural networks, with respect to the expected square from the true ground truth TQ_{k-1}.

Definition 1. *For a given iteration k of Algorithm 1, we define*

$$Q_{k_1} = \arg\min_{Q_\theta, \theta \in \Theta} \mathbb{E}(Q_\theta - TQ_{k-1})_\nu^2, \quad (19)$$

The expectation is with respect to the sampling distribution of the state action pairs denoted by ν. TQ_{k-1} is the bellman optimality operator applied to Q_{k-1}.

Note that we do not have access to the transition probability kernel P, hence we cannot calculate TQ_{k-1}. To alleviate this, we use the observed next states to estimate the Q-value function. Using this, we define Q_{k_2} as,.

Definition 2. *For a given function $Q : \mathcal{S} \times \mathcal{A} \to \left[0, \frac{R_{max}}{1-\gamma}\right]$, we define*

$$Q_{k_2} = \arg\min_{Q_\theta, \theta \in \Theta} \mathbb{E}_{(s,a) \sim \nu, s' \sim P(s'|s,a), r'(\cdot|s,a) \sim R(\cdot|s,a)} (Q_\theta(s,a) - (r'(s,a) + \gamma \max_{a'} Q_{k-1}(s', a'))^2, \quad (20)$$

Compared to Q_{k_1}, in Q_{k_2}, we are minimizing the expected square loss from target function $\big(r'(s,a) + \gamma \max_{a'} Q(s', a')\big)$ or the expected loss function.

To obtain Q_{k_2}, we still need to compute the true expected value in Equation 20. However, we still do not know the transition function P. To remove this limitation, we use sampling. Consider a set, $\mathcal{X}$, of state-action pairs sampled using distribution ν. We now define Q_{k_3} as,

Definition 3. *For a given set of state action pairs $\mathcal{X}$ and a given function $Q : \mathcal{S} \times \mathcal{A} \to \left[0, \frac{R_{max}}{1-\gamma}\right]$ we define*

$$Q_{k_3} = \arg\min_{Q_\theta, \theta \in \Theta} \frac{1}{|\mathcal{X}|} \sum_{(s_i, a_i) \in \mathcal{X}} \Big(Q_\theta(s_i, a_i) - \big(r'(s_i, a_i) + \gamma \max_{a' \in \mathcal{A}} Q_{k-1}(s_i', a')\big)\Big)^2, \quad (21)$$

where $r'(s_i, a_i)$, and s_i' are the observes reward and the observed next state for state action pair s_i, a_i respectively.

Q_{k_3} is the best possible approximation for Q-value function which minimizes the sample average of the square loss functions with the target values as $\big(r'(s_i, a_i) + \gamma \max_{a' \in \mathcal{A}} Q(s', a')\big)^2$ or the empirical loss function.

After defining the possible solutions for the Q-values using different loss functions, we define the errors.

We first define approximation error which represents the difference between TQ_{k-1} and its best approximation possible from the class of 2 layer ReLU neural networks. We have

Definition 4 (Approximation Error)**.** *For a given iteration k of Algorithm 1, $\epsilon_{k_1} = TQ_{k-1} - Q_{k_1}$, where Q_{k-1} is the estimate of the Q function at the iteration $k-1$.*

We also define Estimation Error which denotes the error between the best approximation of TQ_{k-1} possible from a 2 layer ReLU neural network and Q_{k_2}. We demonstrate that these two terms are the same and this error is zero.

Definition 5 (Estimation Error)**.** *For a given iteration k of Algorithm 1, $\epsilon_{k_2} = Q_{k_1} - Q_{k_2}$.*

We now define Sampling error which denotes the difference between the minimizer of expected loss function Q_{k_2} and the minimizer of the empirical loss function using samples, Q_{k_3}. We will use Rademacher complexity results to upper bound this error.

Definition 6 (Sampling Error)**.** *For a given iteration k of Algorithm 1, $\epsilon_{k_3} = Q_{k_2} - Q_{k_3}$. Here X_k is the set of state action pairs sampled at the kth iteration of Algorithm 1.*

Lastly, we define optimization error which denotes the difference between the minimizer of the empirical square loss function, Q_{k_3}, and our estimate of this minimizer that is obtained from the projected gradient descent algorithm.

Definition 7 (Optimization Error)**.** *For a given iteration k of Algorithm 1, $\epsilon_{k_4} = Q_{k_3} - Q_k$. Here Q_k is our estimate of the Q function at iteration k of Algorithm 1.*

6. Assumptions

In this section, we formally describe the assumptions that will be used in the results.

Assumption 1. *Let $\theta^* \triangleq \arg\min_{\theta\in\Theta} \mathcal{L}(\theta)$, where $\mathcal{L}(\theta)$ is defined in* (11) *and we denote $Q_{\theta^*}(\cdot)$ as $Q_\theta(\cdot)$ as defined in* (14) *for $\theta = \theta^*$. Also, let $\theta^*_{\tilde{D}} \triangleq \arg\min_{\theta\in\Theta} \mathcal{L}_{|\tilde{D}|}(\theta)$, where $\mathcal{L}_{\tilde{D}}(\theta)$ is defined in* (52)*. Further, we denote $Q_{\theta^*_{|\tilde{D}|}}(\cdot)$ as $Q_\theta(\cdot)$ as defined in* (14) *for $\theta = \theta^*_{|\tilde{D}|}$. Then we assume*

$$\mathbb{E}(|Q_{\theta^*} - Q_{\theta^*_{|\tilde{D}|}}|)_\nu \leq \epsilon_{|\tilde{D}|}, \tag{22}$$

for any $\nu \in \mathcal{P}(\mathcal{S}\times\mathcal{A})$

$\mathcal{L}_{\tilde{D}}(\theta)$ is the non-convex problem equivalent to the convex problem in (13). Thus, $\epsilon_{|\tilde{D}|}$ is a measure of the error incurred due to taking a sample of diagonal matrices $\tilde{D}$ and not the full set D_X. In practice, setting $|\tilde{D}|$ to be the same order of magnitude as d (dimension of the data) gives us a sufficient number of diagonal matrices to get a reformulation of the non convex optimization problem which performs comparably or better than existing gradient descent algorithms, therefore $\epsilon_{|\tilde{D}|}$ is only included for theoretical completeness and will be negligible in practice. This has been practically demonstrated in (Mishkin et al., 2022; Bartan and Pilanci, 2022; Sahiner et al., 2022). Refer to Appendix B for details of D_X, $\tilde{D}$ and $\mathcal{L}_{|\tilde{D}|}(\theta)$.

Assumption 2. *We assume that for all functions $Q : \mathcal{S}\times\mathcal{A} \to \left[0, \left(\frac{R_{\max}}{1-\gamma}\right)\right]$, there exists a function Q_θ where $\theta \in \Theta$ such that*

$$\mathbb{E}(Q_\theta - Q)^2_\nu \leq \epsilon_{bias}, \tag{23}$$

for any $\nu \in \mathcal{P}(\mathcal{S}\times\mathcal{A})$.

ϵ_{bias} reflects the error that is incurred due to the inherent lack of expressiveness of the neural network function class. In the analysis of (Fan et al., 2020), this error is assumed to be zero. We account for this error with an assumption similar to the one used in (Liu et al., 2020).

Assumption 3. *For any policy π on the MDP $\mathcal{M}$ with the starting state s_0, let $\zeta^\pi_{s_0}$ be the corresponding stationary state action distribution of the induced Markov chain and $(s,a)_\tau$ denote the state action pair obtained at step $\tau \in \mathbb{Z}$ by following policy π. We assume that there exists a positive integer m and $\rho \in (0,1)$ such that*

$$\sup_{s_0} d_{TV}\left(\mathbb{P}((s,a)_\tau \in \cdot|(s_0)), \zeta^\pi_{s_0}(\cdot)\right) \leq m\rho^\tau, \tag{24}$$

This assumption implies that the Markov chain for any policy is geometrically mixing. Such an assumption is widely used both in analysis of stochastic gradient descent literature such as (Doan, 2022; Sun et al., 2018), as well as finite time analysis of reinforcement learning algorithms such as (Wu et al., 2020; Xu et al., 2020).

Assumption 4. *Let ν_1 be a probability measure on $\mathcal{S}\times\mathcal{A}$ which is absolutely continuous with respect to the Lebesgue measure. Let $\{\pi_t\}$ be a sequence of policies and suppose that the state action pair has an initial distribution of ν_1. Then we assume that for all $\nu_1, \nu_2 \in \mathcal{P}(\mathcal{S}\times\mathcal{A})$ there exists a constant $\phi_{\nu_1,\nu_2} \leq \infty$ such that*

$$\sup_{\pi_1,\pi_2,\cdots,\pi_m} \left|\left|\frac{d(P^{\pi_1}P^{\pi_2}\cdots P^{\pi_m}\nu_2)}{d\nu_1}\right|\right|_\infty \leq \phi_{\nu_1,\nu_2} \tag{25}$$

for all $m \in \{0,\cdots,\infty\}$, where $\frac{d(P^{\pi_1}P^{\pi_2}\cdots P^{\pi_m}\nu_2)}{d\nu_1}$ denotes the Radon Nikodym derivative of the state action distribution $P^{\pi_1}P^{\pi_2}\cdots P^{\pi_m}\nu_2$ with respect to the distribution ν_1.

This assumption puts an upper bound on the difference between the state action distribution ν_1 and the state action distribution induced by sampling a state action pair from the distribution μ_2 followed by any possible policy for the next m steps for any finite value of m. Similar assumptions have been made in (Fan et al., 2020; Lazaric et al., 2016).

7. Supporting Lemmas

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

Lemma 1. *For any given iteration $k \in \{1, \cdots, K\}$ for the approximation error denoted by ϵ_{k_1} in Definition 4, we have*

$$\mathbb{E}\left(|\epsilon_{k_1}|\right)_\nu \leq \sqrt{\epsilon_{bias}}, \tag{26}$$

Proof Sketch: We use Assumption 2 and the definition of the variance of a random variable to obtain the required result. The detailed proof is given in Appendix E.1.

Lemma 2. *For any given iteration $k \in \{1, \cdots, K\}$, $Q_{k_1} = Q_{k_2}$, or equivalently $\epsilon_{k_2} = 0$*

Proof Sketch: We use Lemma 6 in Appendix C and use the definitions of Q_{k_1} and Q_{k_2} to prove this result. The detailed proof is given in Appendix E.2.

Lemma 3. *For any given iteration $k \in \{1, \cdots, K\}$, if the number of samples of the state action pairs sampled by Algorithm 1 at iteration k, denoted by n_k, satisfies*

$$n_k \quad \geq \quad \tilde{\mathcal{O}}\left((1-\gamma)^{-2}\epsilon^{-2}\right) \tag{27}$$

then the error ϵ_{k_3} defined in Definition 6 is upper bounded as

$$\mathbb{E}\left(|\epsilon_{k_3}|\right)_{\zeta_\nu^{\pi_k}} \leq \epsilon, \tag{28}$$

where π_k is the greedy policy with respect to Q_{k-1}.

Proof Sketch: First we note that for a fixed iteration k of Algorithm 1, $\mathbb{E}(R_{X,Q_{k-1}}(\theta)) = L_{Q_{k-1}}(\theta)$ where $R_{X,Q_{k-1}}(\theta)$ and $L_{Q_{k-1}}(\theta)$ are defined in Appendix E.3. We use this to get a probabilistic bound on the expected value of $|(Q_{k_2}) - (Q_{k_3})|$ using Rademacher complexity theory. The detailed proof is given in Appendix E.3.

Lemma 4. *For any given iteration $k \in \{1, \cdots, K\}$ of Algorithm 1, let the number of steps of the projected gradient descent performed by Algorithm 2, denoted by T_k, and the gradient descent step size α_k satisfy*

$$T_k \quad \geq \quad \left(\frac{C_k^{'}\epsilon}{l_k}\right)^{-2} L_k^2 ||u_k^*||_2^2 - 1 \tag{29}$$

$$\alpha_k \quad = \quad \frac{||u_k^*||_2}{L_k\sqrt{T_k+1}} \tag{30}$$

where $\epsilon < C_k^{'}$, for some constants $C_k^{'}, l_k$, L_k and $||\left(u_k^*\right)||_2$. Then the error ϵ_{k_4} defined in Definition 7 is upper bounded as

$$\mathbb{E}(|\epsilon_{k_4}|)_\nu \leq \epsilon + \epsilon_{|\tilde{D}|}, \tag{31}$$

Proof Sketch: We use the number of iterations T_k required to get an ϵ bound on the difference between the minimum objective value and the objective value corresponding to the estimated parameter at iteration T_k. We use the convexity of the objective and the Lipschitz property of the neural network to get a bound on the Q functions corresponding to the estimated parameters. The detailed proof is given in Appendix E.4.

8. Main Result

In this section, we provide the guarantees for the proposed algorithm, which is given in the following theorem.

Theorem 1. *Suppose Assumptions 1-4 hold. Let Algorithm 1 run for K iterations with n_k state-action pairs sampled at iteration $k \in \{1, \cdots, K\}$, T_k be the number of steps used and step size α_k in the projected gradient descent in Algorithm 2 at iteration $k \in \{1, \cdots, K\}$ and $|\tilde{D}|$ be the number of diagonal matrices sampled in Algorithm 2 for all iterations. Let $\nu \in \mathcal{P}(\mathcal{S} \times \mathcal{A})$ be the state action distribution used to sample the state action pairs in Algorithm 1. Further, let $\epsilon \in (0,1)$, C_k, $C_k^{'}$, l_k, ,L_k, $\phi_{\nu,\mu}, \beta_k, \eta, (||u_k^*||_2)$ be constants. If we have,*

$$K \quad \geq \quad \frac{1}{\log\left(\frac{1}{\gamma}\right)} \log\left(\frac{6\phi_{\nu,\mu}R_{max}}{\epsilon(1-\gamma)^2}\right) - 1 \tag{32}$$

$$n_k \quad \geq \quad \tilde{\mathcal{O}}(1-\gamma)^{-6}\epsilon^{-2} \tag{33}$$

$$T_k \quad \geq \quad \left(\frac{C_k^{'}\epsilon(1-\gamma)^2}{6l_k\phi_{\nu,\mu}\gamma}\right)^{-2} L_k^2||u_k^*||_2^2 - 1 \tag{34}$$

$$\alpha_k \quad = \quad \frac{||u_k^*||_2}{L_k\sqrt{T_k+1}} \quad \forall k \in \{1, \cdots, K\} \tag{35}$$

and $\epsilon < (C_k^{'})$ for all $k \in (1, \cdots, K)$, then we obtain

$$E(Q^* - Q^{\pi_K})_\mu \leq \epsilon + \frac{2\phi_{\nu,\mu}\gamma}{(1-\gamma)^2}(\sqrt{\epsilon_{bias}} + \epsilon_{|\tilde{D}|}),$$

where $\mu \in \mathcal{P}(\mathcal{S} \times \mathcal{A})$. Thus we get an overall sample complexity given by $\sum_{k=1}^{K} n_k = \tilde{\mathcal{O}}\left(\epsilon^{-2}(1-\gamma)^{-6}\right)$.

The above algorithm achieves a sample complexity of $\tilde{\mathcal{O}}(1/\epsilon^2)$, which is the first such result for general parametrized large state space reinforcement learning. Further, we note that $\tilde{\mathcal{O}}(1/\epsilon^2)$ is the best order-optimal result even in tabular setup (Zhang et al., 2020). In the following, we provide an outline of the proof, with the detailed proof provided in the Appendix D.

9. Proof of Theorem 1 Outline

The expectation of the difference between our estimated Q function denoted by Q^{π_K} and the optimal Q function denoted by Q^* (where π_k is the policy obtained at the final step K of algorithm 1) is first expressed as a function of the sum of the errors incurred by the actor and critic steps. The critic error is the Bellman error defined as in (17). This is then further split into different components which are analysed separately. The actor error is the error incurred due to the stochastic gradient descent steps. In the first stage, we demonstrate how the expectation of the error of estimation $(Q^{\pi_K} - Q^*)$ of the Q function is upper bounded by a function of the Bellman errors incurred till the final step K. The second part is to upper bound the expectation of the Bellman error.

Upper Bounding Q Error In Terms Of Bellman Error: Since we only have access to the approximate Bellman optimality operator defined in Equation (15), we will rely upon the analysis laid out in (Farahmand et al., 2010) and instead of the iteration of the value functions, we will apply a similar analysis to the action value function to get the desired result. We recreate the result for the value function from Lemmas 2 and 3 of (Munos, 2003) for the action value function Q to obtain

$$\begin{aligned} Q^* - Q_K &\leq \sum_{k=1}^{K-1} \gamma^{K-k-1} (P^{\pi^*})^{K-k-1} \epsilon_k, \quad (36) \\ &\quad + \gamma^K (P^{\pi^*})^K (Q^* - Q_0) \end{aligned}$$

and

$$\begin{aligned} Q^* - Q^{\pi_K} &\leq (I - \gamma P^{\pi_K})^{-1} \left(P^{\pi^*} - P^{\pi_K} \right) \\ &\quad (Q^* - Q_K) \quad (37) \end{aligned}$$

where $\epsilon_k = TQ_{k-1} - Q_k$.

We use the results in Equation (36) and (37) to obtain

$$\begin{aligned} \mathbb{E}(Q^* - Q^{\pi_K})_\mu &\leq \frac{2\gamma}{(1-\gamma)} \left[\sum_{k=1}^{K-1} \gamma^{K-k} \mathbb{E}(|\epsilon_k|)_\mu \right] + \\ &\quad \frac{2R_{\max}\gamma^{K+1}}{(1-\gamma)^2}, \quad (38) \end{aligned}$$

The first term on the right hand side is called as the algorithmic error, which depends on how good our approximation of the Bellman error is. 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 and depends only on the parameters of the MDP as well as the number of iterations of the FQI algorithm.

The expectation of the error of estimation is taken with respect to any arbitrary distribution on the state action space denoted as μ. The dependence of the expected value of the error of estimation on this distribution is expressed through the constant $\phi_{\nu,\mu}$, which is a measure of the similarity between the distributions ν and μ.

Upper Bounding Expectation of Bellman Error: The upper bound on $\mathbb{E}(Q^* - Q^{\pi_K})_\mu$ in Equation (38) is in terms of $\mathbb{E}(|\epsilon_k|)_\mu$, where $\epsilon_k = TQ_{k-1} - Q_k$ is a measure of how closely our estimate of the Q function at iteration k approximates the function obtained by applying the bellman optimality operator applied to the estimate of the Q function at iteration $k-1$. Intuitively, this error depends on how much data is collected at each iteration, how efficient our solution to the optimization step is to the true solution, and how well our function class can approximate the true Bellman optimality operator applied to the estimate at the end of the previous iteration. Building upon this intuition, we split ϵ_k into four different components as follows.

$$\begin{aligned} \epsilon_k &= TQ_{k-1} - Q_k \\ &= \underbrace{TQ_{k-1} - Q_{k1}}_{\epsilon_{k1}} + \underbrace{Q_{k1} - Q_{k2}}_{\epsilon_{k2}} + \underbrace{Q_{k2} - Q_{k3}}_{\epsilon_{k3}} \\ &\quad + \underbrace{Q_{k3} - Q_k}_{\epsilon_{k4}} \\ &= \epsilon_{k_1} + \epsilon_{k_2} + \epsilon_{k_3} + \epsilon_{k_4}, \quad (39) \end{aligned}$$

We use the Lemmas 1, 2, 3, and 4 to bound the error terms in Equation (39). Before plugging these in Equation (38), we replace ϵ in the Lemma results with $\epsilon' = \epsilon \frac{(1-\gamma)^3}{6\gamma}$ as these error terms are in a summation in Equation (38). We also bound the last term on the right hand side of Equation (38) by solving for a value of K that makes the term smaller than $\frac{\epsilon}{3}$.

10. Conclusion and Future Work

In this paper, we study a Fitted Q-Iteration with two-layer ReLU neural network parametrization, and find the sample complexity guarantees for the algorithm. Using the convex approach for estimating the Q-function, we show that our approach achieves a sample complexity of $\tilde{\mathcal{O}}(1/\epsilon^2)$, which is order-optimal. This demonstrates the first approach for achieving sample complexity beyond linear MDP assumptions for large state space.

This study raises multiple future problems. First is whether we can remove the assumption on the generative model, while estimating Q-function by efficient sampling of past samples. Further, whether the convexity in training that was needed for the results could be extended to more than two layers. Finally, efficient analysis for the error incurred when a sample of cones are chosen rather than the complete set of cones and how to efficiently choose this subset will help with a complete analysis by which Assumption 1 can

be relaxed.

11. Acknowledgements

This work was supported in part by research award from Cisco.

References

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. (2021). On the theory of policy gradient methods: Optimality, approximation, and distribution shift. In *The Journal of Machine Learning Research*, volume 22, pages 4431–4506. JMLRORG.

Al-Abbasi, A. O., Ghosh, A., and Aggarwal, V. (2019). Deeppool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 20(12):4714–4727.

Bartan, B. and Pilanci, M. (2022). Neural Fisher discriminant analysis: Optimal neural network embeddings in polynomial time. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 1647–1663. PMLR.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer.

Bertail, P. and Portier, F. (2019). Rademacher complexity for markov chains: Applications to kernel smoothing and metropolis–hastings. *Bernoulli*, 25:3912–3938.

Bertsekas, D. P. and Shreve, S. E. (2007). *Stochastic Optimal Control: The Discrete-Time Case*. Athena Scientific.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-dynamic programming.*, volume 3 of *Optimization and neural computation series*. Athena Scientific.

Blum, A. and Rivest, R. (1992). Training a 3-node neural network is NP-complete. *Neural Networks*, 5:117–127.

Bonjour, T., Haliem, M., Alsalem, A., Thomas, S., Li, H., Aggarwal, V., Kejriwal, M., and Bhargava, B. (2022). Decision making in monopoly using a hybrid deep reinforcement learning approach. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Boyan, J. and Moore, A. (1994). Generalization in reinforcement learning: Safely approximating the value function. In *Advances in neural information processing systems*, volume 7.

Carvalho, D., Melo, F. S., and Santos, P. (2020). A new convergent variant of Q-learning with linear function approximation. In *Adv. Neural Information Proc. Systems 33*, pages 19412–19421.

Chen, X. and Zhao, L. (2022). Finite-time analysis of single-timescale actor-critic. *arXiv preprint arXiv:2210.09921*.

Chen, Y., He, J., and Gu, Q. (2022). On the sample complexity of learning infinite-horizon discounted linear kernel mdps. In *International Conference on Machine Learning*, pages 3149–3183. PMLR.

Damjanović, I., Pavić, I., Puljiz, M., and Brcic, M. (2022). Deep reinforcement learning-based approach for autonomous power flow control using only topology changes. *Energies*, 15(19).

Doan, T. T. (2022). Finite-time analysis of markov gradient descent. *IEEE Transactions on Automatic Control*, pages 1–1.

Fan, J., Wang, Z., Xie, Y., and Yang, Z. (2020). A theoretical analysis of deep q-learning. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 486–489. PMLR.

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

Fujimoto, S., Meger, D., and Precup, D. (2019). Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR.

Gao, Z., Gao, Y., Hu, Y., Jiang, Z., and Su, J. (2020). Application of deep q-network in portfolio management. In *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*, pages 268–275.

Geng, N., Lan, T., Aggarwal, V., Yang, Y., and Xu, M. (2020). A multi-agent reinforcement learning perspective on distributed traffic engineering. In *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, pages 1–11. IEEE.

Ghavamzadeh, M., Szepesvári, C., Mannor, S., et al. (2008). Regularized fitted q-iteration: Application to planning. In *European Workshop on Reinforcement Learning*, pages 55–68. Springer.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. (2018). Is q-learning provably efficient? In *Advances in neural information processing systems*, volume 31.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A. A. A., Yogamani, S., and Pérez, P. (2022). Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926.

Kozakowski, P., Kaiser, L., Michalewski, H., Mohiuddin, A., and Kańska, K. (2022). Q-value weighted regression: Reinforcement learning with limited data. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Lazaric, A., Ghavamzadeh, M., and Munos, R. (2016). Analysis of classification-based policy iteration algorithms. *Journal of Machine Learning Research*, 17(19):1–30.

Li, G., Wei, Y., Chi, Y., Gu, Y., and Chen, Y. (2020). Sample complexity of asynchronous q-learning: Sharper analysis and variance reduction. In *Advances in Neural Information Processing Systems*, volume 33, pages 7031–7043. Curran Associates, Inc.

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

Maes, S., Tuyls, K., and Manderick, B. (2003). Reinforcement learning in large state spaces: Simulated robotic soccer as a testbed.

Mishkin, A., Sahiner, A., and Pilanci, M. (2022). Fast convex optimization for two-layer relu networks: Equivalent model classes and cone decompositions. In *International Conference on Machine Learning*, pages 15770–15816. PMLR.

Munos, R. (2003). Error bounds for approximate policy iteration. In *ICML*, volume 3, pages 560–567.

Munos, R. (2007). Performance Bounds in L_p norm for Approximate Value Iteration. *SIAM Journal on Control and Optimization*, 46(2):541–561.

Pilanci, M. and Ergen, T. (2020). Neural networks are convex regularizers: Exact polynomial-time convex optimization formulations for two-layer networks. In *International Conference on Machine Learning*, pages 7695–7705. PMLR.

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

Sahiner, A., Ergen, T., Ozturkler, B., Pauly, J., Mardani, M., and Pilanci, M. (2022). Unraveling attention via convex duality: Analysis and interpretations of vision transformers. In *International Conference on Machine Learning*, pages 19050–19088. PMLR.

Sahiner, A., Ergen, T., Pauly, J., and Pilanci, M. (2020a). Vector-output relu neural network problems are copositive programs: Convex analysis of two layer networks and polynomial-time algorithms. *arXiv preprint arXiv:2012.13329*.

Sahiner, A., Mardani, M., Ozturkler, B., Pilanci, M., and Pauly, J. (2020b). Convex regularization behind neural reconstruction. *arXiv preprint arXiv:2012.05169*.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359.

Singh, S. P. and Yee, R. C. (1994). An upper bound on the loss from approximate optimal-value functions. *Mach. Learn.*, 16(3):227–233.

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

Tesauro, G. et al. (1995). Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68.

Van Hasselt, H., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Vinyals, O., Ewalds, T., Bartunov, S., Georgiev, P., Vezhnevets, A. S., Yeo, M., Makhzani, A., Küttler, H., Agapiou, J., Schrittwieser, J., et al. (2017). Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*.

Virmaux, A. and Scaman, K. (2018). Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems*, volume 31.

Wang, C., Zhou, T., Chen, C., Hu, T., and Chen, G. (2020). Off-policy recommendation system without exploration. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 16–27. Springer.

Wang, L., Cai, Q., Yang, Z., and Wang, Z. (2019). Neural policy gradient methods: Global optimality and rates of convergence. *arXiv preprint arXiv:1909.01150*.

Wang, T., Zhou, D., and Gu, Q. (2021a). Provably efficient reinforcement learning with linear function approximation under adaptivity constraints. In *Advances in Neural Information Processing Systems*, volume 34, pages 13524–13536.

Wang, Y., Lacotte, J., and Pilanci, M. (2021b). The hidden convex optimization landscape of regularized two-layer relu networks: an exact characterization of optimal solutions. In *International Conference on Learning Representations*.

Warlop, R., Lazaric, A., and Mary, J. (2018). Fighting boredom in recommender systems with linear reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 31.

Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3):279–292.

Wu, Y. F., Zhang, W., Xu, P., and Gu, Q. (2020). A finite-time analysis of two time-scale actor-critic methods. *Advances in Neural Information Processing Systems*, 33:17617–17628.

Xu, P. and Gu, Q. (2020). A finite-time analysis of q-learning with neural network function approximation. In *International Conference on Machine Learning*, pages 10555–10565. PMLR.

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

Yang, T., Zhao, L., Li, W., and Zomaya, A. Y. (2020). Reinforcement learning in sustainable energy and electric systems: a survey. *Annu. Rev. Control.*, 49:145–163.

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

Zhang, M. S., Erdogdu, M. A., and Garg, A. (2022). Convergence and optimality of policy gradient methods in weakly smooth settings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9066–9073.

Zhang, Z., Zhou, Y., and Ji, X. (2020). Almost optimal model-free reinforcement learningvia reference-advantage decomposition. *Advances in Neural Information Processing Systems*, 33:15198–15207.

Zhou, D., He, J., and Gu, Q. (2021). Provably efficient reinforcement learning for discounted mdps with feature mapping. In *International Conference on Machine Learning*, pages 12793–12802. PMLR.

Zhu, H. and Xu, J. (2021). One-pass stochastic gradient descent in overparametrized two-layer neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 3673–3681. PMLR.

A. Comparison of Result with (Xu and Gu, 2020)

Analysis of Q learning algorithms with neural network function approximation was carried out in (Xu and Gu, 2020), where finite time error bounds were studied for estimation of the Q function for MDP's with countable state spaces. The final result in (Xu and Gu, 2020) is of the form

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}(Q(s,a,\theta_t)-Q^*(s,a)) \leq \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{T}}\right)+\tilde{\mathcal{O}}(\sqrt{\log(T)}) \tag{40}$$

Here, $Q(s,a,\theta_t)$ is the estimate of the Q function obtained using the neural network with the parameters obtained at step t of the Q learning algorithm represented by θ_t and $Q^*(s,a)$ is the optimal Q function. T is the total number of iterations of the Q learning algorithm.

The first term on the right hand side of (40) is monotonically decreasing with respect to T. The second term on the right hand side monotonically increases with respect to T. This term is present due to the error incurred by the linear approximation of the neural network representing the Q function. Our approach does not require this approximation due to the convex representation of the neural network. Due to the increasing function of T, it is not possible to obtain the number of iteration of the Q learning required to make the error of estimation smaller than some fixed value. Thus, this result cannot be said to be a sample complexity result. However, in our results, this is not the case.

B. Convex Reformulation with Two-Layer Neural Networks

In order to understand the convex reformulation of the squared loss optimization problem, consider the vector $\sigma(Xu_i)$

$$\sigma(Xu_i)=\begin{bmatrix}\{\sigma^{'}((x_1)^Tu_i)\}\\ \{\sigma^{'}((x_2)^Tu_i)\}\\ \vdots\\ \{\sigma^{'}((x_n)^Tu_i)\}\end{bmatrix} \tag{41}$$

Now for a fixed $X\in\mathbb{R}^{n\times d}$, different $u_i\in\mathbb{R}^{d\times 1}$ will have different components of $\sigma(Xu_i)$ that are non zero. For example, if we take the set of all u_i such that only the first element of $\sigma(Xu_i)$ are non zero (i.e, only $(x_1)^Tu_i\geq 0$ and $(x_j)^Tu_i<0$ $\forall j\in[2,\cdots,n]$) and denote it by the set $\mathcal{K}_1$, then we have

$$\sigma(Xu_i)=D_1(Xu_i)\quad \forall u_i\in\mathcal{K}_1,$$

where D_1 is the $n\times n$ diagonal matrix with only the first diagonal element equal to 1 and the rest 0. Similarly, there exist a set of $u's$ which result in $\sigma(Xu)$ having certain components to be non-zero and the rest zero. For each such combination of zero and non-zero components, we will have a corresponding set of $u_i's$ and a corresponding $n\times n$ Diagonal matrix D_i. We define the possible set of such diagonal matrices possible for a given matrix X as

$$D_X=\{D=diag(\mathbf{1}(Xu\geq 0)):u\in\mathbb{R}^d, D\in\mathbb{R}^{n\times n}\}, \tag{42}$$

where $diag(\mathbf{1}(Xu\geq 0))$ represents a matrix given by

$$D_{k,j}=\begin{cases}\mathbf{1}(x_j^Tu), & \text{for } k=j\\ 0 & \text{for } k\neq j\end{cases}, \tag{43}$$

where $\mathbf{1}(x)=1$ if $x>0$ and $\mathbf{1}(x)=0$ if $x\leq 0$. Corresponding to each such matrix D_i, there exists a set of u_i given by

$$\mathcal{K}_i=\{u\in\mathbb{R}^d:\sigma(Xu_i)=D_iXu_i, D_i\in D_X\} \tag{44}$$

where I is the $n \times n$ identity matrix. The number of these matrices D_i is upper bounded by 2^n. From (Wang et al., 2021b) the upper bound is $\mathcal{O}\left(r\left(\frac{n}{r}\right)^r\right)$ where $r = rank(X)$. Also, note that the sets $\mathcal{K}_i$ form a partition of the space $\mathbb{R}^{d\times 1}$. Using these definitions, we define the equivalent convex problem to the one in Equation (11) as

$$\mathcal{L}_\beta(v,w) := \arg\min_{v,w}(||\sum_{D_i\in D_X} D_i(X(v_i-w_i)) - y||_2^2 + \beta\sum_{D_i\in D_X}||v_i||_2 + ||w_i||_2) \tag{45}$$

where $v = \{v_i\}_{i\in 1,\cdots,|D_X|}$, $w = \{w_i\}_{i\in 1,\cdots,|D_X|}$, $v_i, w_i \in \mathcal{K}_i$, note that by definition, for any fixed $i \in \{1,\cdots,|D_X|\}$ at-least one of v_i or w_i are zero. If v^*, w^* are the optimal solutions to Equation (45), the number of neurons m of the original problem in Equation (11) should be greater than the number of elements of v^*, w^*, which have at-least one of v_i^* or w_i^* non-zero. We denote this value as $m^*_{X,y}$, with the subscript X denoting that this quantity depends upon the data matrix X and response y.

We convert v^*, w^* to optimal values of Equation (11), denoted by $\theta^* = (U^*, \alpha^*)$, using a function $\psi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d \times \mathbb{R}$ defined as follows

$$\psi(v_i, w_i) = \begin{cases} (v_i, 1), & \text{if } w_i = 0 \\ (w_i, -1), & \text{if } v_i = 0 \\ (0,0), & \text{if } v_i = w_i = 0 \end{cases} \tag{46}$$

where according to (Pilanci and Ergen, 2020) we have $(u_i^*, \alpha_i^*) = \psi(v_i^*, w_i^*)$, for all $i \in \{1,\cdots,|D_X|\}$ where u_i^*, α_i^* are the elements of θ^*. Note that restriction of α_i to $\{1,-1,0\}$ is shown to be valid in (Mishkin et al., 2022). For $i \in \{|D_X|+1,\cdots,m\}$ we set $(u_i^*, \alpha_i^*) = (0,0)$.

Since D_X is hard to obtain computationally unless X is of low rank, we can construct a subset $\tilde{D} \in D_X$ and perform the optimization in Equation (45) by replacing D_X with $\tilde{D}$ to get

$$\mathcal{L}_\beta(v,w) := \arg\min_{v,w}(||\sum_{D_i\in \tilde{D}} D_i(X(v_i-w_i)) - y||_2^2 + \beta\sum_{D_i\in \tilde{D}}||v_i||_2 + ||w_i||_2) \tag{47}$$

where $v = \{v_i\}_{i\in 1,\cdots,|\tilde{D}|}$, $w = \{w_i\}_{i\in 1,\cdots,|\tilde{D}|}$, $v_i, w_i \in \mathcal{K}_i$, by definition, for any fixed $i \in \{1,\cdots,|\tilde{D}|\}$ at-least one of v_i or w_i are zero.

The required condition for $\tilde{D}$ to be a sufficient replacement for D_X is as follows. Suppose $(v,w) = (\bar{v}_i, \bar{w}_i)_{i\in(1,\cdots,|\tilde{D}|)}$ denote the optimal solutions of Equation (47). Then we require

$$m \geq \sum_{D_i\in\tilde{D}} |\{\bar{v}_i : \bar{v}_i \neq 0\} \cup \{\bar{w}_i : \bar{w}_i \neq 0\}| \tag{48}$$

Or, the number of neurons in the neural network are greater than the number of indices i for which at-least one of v_i^* or w_i^* is non-zero. Further,

$$diag(Xu_i^* \geq 0 : i \in [m]) \in \tilde{D} \tag{49}$$

In other words, the diagonal matrices induced by the optimal u_i^*'s of Equation (11) must be included in our sample of diagonal matrices. This is proved in Theorem 2.1 of (Mishkin et al., 2022).

A computationally efficient method for obtaining $\tilde{D}$ and obtaining the optimal values of the Equation (11), is laid out in (Mishkin et al., 2022). In this method we first get our sample of diagonal matrices $\tilde{D}$ by first sampling a fixed number of vectors from a d dimensional standard multivariate distribution, multiplying the vectors with the data matrix X and then forming the diagonal matrices based of which co-ordinates are positive. Then we solve an optimization similar to the one in Equation (45), without the constraints, that its parameters belong to sets of the form $\mathcal{K}_i$ as follows.

$$\mathcal{L}_\beta^{'}(p) := \arg\min_p (|| \sum_{D_i \in \tilde{D}} D_i(Xp_i) - y||_2^2 + \beta \sum_{D_i \in \tilde{D}} ||p_i||_2), \tag{50}$$

where $p \in \mathbb{R}^{d \times |\tilde{D}|}$. In order to satisfy the constraints of the form given in Equation (45), this step is followed by a cone decomposition step. This is implemented through a function $\{\psi_i^{'}\}_{i \in \{1, \cdots, |\tilde{D}|\}}$. Let $p^* = \{p_i^*\}_{i \in \{1, \cdots, |\tilde{D}|\}}$ be the optimal solution of Equation (50). For each i we define a function $\psi_i^{'} : \mathbb{R}^d \to \mathbb{R}^d \times \mathbb{R}^d$ as

$$\begin{aligned} \psi_i^{'}(p_i) &= (v_i, w_i) \\ \textit{such that } p &= v_i - w_i, \textit{and } v_i, w_i \in \mathcal{K}_i \end{aligned} \tag{51}$$

Then we obtain $\psi(p_i^*) = (\bar{v}_i, \bar{w}_i)$. As before, at-least one of v_i, w_i is 0. Note that in practice we do not know if the conditions in Equation (48) and (49) are satisfied for a given sampled $\tilde{D}$. We express this as follows. If $\tilde{D}$ was the full set of Diagonal matrices then we would have $(\bar{v}_i, \bar{w}_i) = v_i^*, w_i^*$ and $\psi(\bar{v}_i, \bar{w}_i) = (u_i^*, \alpha_i^*)$ for all $i \in (1, \cdots, |D_X|)$. However, since that is not the case and $\tilde{D} \in D_X$, this means that $\{\psi(\bar{v}_i, \bar{w}_i)\}_{i \in (1, \cdots, |\tilde{D}|)}$ is an optimal solution of a non-convex optimization different from the one in Equation (11). We denote this non-convex optimization as $\mathcal{L}_{|\tilde{D}|}(\theta)$ defined as

$$\mathcal{L}_{|\tilde{D}|}(\theta) = \arg\min_\theta || \sum_{i=1}^{m^{'}} \sigma(Xu_i)\alpha_i - y||_2^2 + \beta \sum_{i=1}^{m} ||u_i||_2^2 + \beta \sum_{i=1}^{m} |\alpha_i|^2, \tag{52}$$

where $m^{'} = |\tilde{D}|$ or the size of the sampled diagonal matrix set. In order to quantify the error incurred due to taking a subset of D_X, we assume that the expectation of the absolute value of the difference between the neural networks corresponding to the optimal solutions of the non-convex optimizations given in Equations (52) and (11) is upper bounded by a constant depending on the size of $\tilde{D}$. The formal assumption and its justification is given in Assumption 1.

C. Supplementary lemmas and Definitions

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 $Z \in \mathbb{R}^n$, we define the Rademacher complexity of the set Z as*

$$Rad(Z) = \mathbb{E}\left(\sup_{z \in Z} \frac{1}{n} \sum_{i=1}^{d} \Omega_i z_i\right) \tag{53}$$

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 5. *Consider a set of observed data denoted by $z = \{z_1, z_2, \cdots z_n\} \in \mathbb{R}^n$, a parameter space Θ, a loss function $\{l : \mathbb{R} \times \Theta \to \mathbb{R}\}$ where $0 \leq l(\theta, z) \leq 1 \; \forall(\theta, z) \in \Theta \times \mathbb{R}$. 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 $\tilde{z}_i$ is a co-ordinate of $\tilde{z}$ sampled from some distribution over Z.*

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

$$\mathcal{L} = \{z \in Z \rightarrow l(\theta, z) \in \mathbb{R} : \theta \in \Theta\} \tag{54}$$

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

$$\mathcal{L} \circ z \ = \{(l(\theta, z_1), l(\theta, z_2), \cdots, l(\theta, z_n)) \in \mathbb{R}^n : \theta \in \Theta\} \tag{55}$$

Then, we have

$$\mathbb{E} \sup_{\theta \in \Theta} |\{r(\theta) - R(\theta)\}| \leq 2\mathbb{E}\left(Rad(\mathcal{L} \circ z)\right) \tag{56}$$

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* $a_\theta : \Theta \times X \rightarrow \mathbb{R}$*, then we have*

$$\mathbb{E} \sup_{\theta \in \Theta} |\{r(\theta) - R(\theta)\}| \leq 2L\mathbb{E}\left(Rad(\mathcal{A} \circ \{x_1, x_2, x_3, \cdots, x_n\})\right) \tag{57}$$

where

$$\mathcal{A} \circ \{x_1, x_2, \cdots, x_n\} \ = \{(a(\theta, x_1), a(\theta, x_2), \cdots, a(\theta, x_n)) \in \mathbb{R}^n : \theta \in \Theta\} \tag{58}$$

The detailed proof of the above statement is given in (Rebeschini, P. (2022). Algorithmic Foundations of Learning [Lecture Notes]. https://www.stats.ox.ac.uk/ rebeschi/teaching/AFoL/20/material/. 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 6. *Consider two 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

$$\arg\min_{f_\theta} \mathbb{E}_{x,y} \left(f_\theta(x) - g(x, y)\right)^2 = \arg\min_{f_\theta} \left(\mathbb{E}_{x,y} \left(f_\theta(x) - \mathbb{E}_{y^{'}|x}(g(x, y^{'})|x)\right)^2\right) \tag{59}$$

Proof. Denote the left hand side of Equation (59) as $\mathbb{X}_\theta$, then add and subtract $\mathbb{E}_{y|x}(g(x, y)|x)$ to it to get

$$\begin{aligned}
\mathbb{X}_\theta &= \arg\min_{f_\theta} \left(\mathbb{E}_{x,y} \left(f_\theta(x) - \mathbb{E}_{y^{'}|x}(g(x, y^{'})|x) + \mathbb{E}_{y^{'}|x}(g(x, y^{'})|x) - g(x, y)\right)^2\right) && (60) \\
&= \arg\min_{f_\theta} \left(\mathbb{E}_{x,y} \left(f_\theta(x) - \mathbb{E}_{y^{'}|x}(g(x, y^{'})|x)\right)^2 + \mathbb{E}_{x,y} \left(y - \mathbb{E}_{y^{'}|x}(g(x, y^{'})|x)\right)^2 - 2\mathbb{E}_{x,y}\Big(f_\theta(x) - \right. \\
&\quad \left. \mathbb{E}_{y^{'}|x}(g(x, y^{'})|x)\right) \left(g(x, y) - \mathbb{E}_{y^{'}|x}(g(x, y^{'})|x)\right)\Big) && (61)
\end{aligned}$$

Consider the third term on the right hand side of Equation (61)

$$
\begin{aligned}
2\mathbb{E}_{x,y}\left(f_\theta(x)-\mathbb{E}_{y'|x}(g(x,y')|x)\right)\left(g(x,y)-\mathbb{E}_{y'|x}(g(x,y')|x)\right) &= 2\mathbb{E}_x\mathbb{E}_{y|x}\left(f_\theta(x)-\mathbb{E}_{y'|x}(g(x,y')|x)\right) \\
&\quad \left(g(x,y)-\mathbb{E}_{y'|x}(g(x,y')|x)\right) && (62)\\
&= 2\mathbb{E}_x\left(f_\theta(x)-\mathbb{E}_{y'|x}(g(x,y')|x)\right) \\
&\quad \mathbb{E}_{y|x}\left(g(x,y)-\mathbb{E}_{y'|x}(g(x,y')|x)\right) && (63)\\
&= 2\mathbb{E}_x\left(f_\theta(x)-\mathbb{E}_{y'|x}(g(x,y')|x)\right) \\
&\quad \left(\mathbb{E}_{y|x}(g(x,y))-\mathbb{E}_{y|x}\left(\mathbb{E}_{y'|x}(g(x,y')|x)\right)\right) && (64)\\
&= 2\mathbb{E}_x\left(f_\theta(x)-\mathbb{E}(y|x)\right). && (65)\\
&\quad \left(\mathbb{E}_{y|x}(g(x,y))-\mathbb{E}_{y'|x}(g(x,y')|x)\right) && (66)\\
&= 0 && (67)
\end{aligned}
$$

Equation (62) is obtained by writing $\mathbb{E}_{x,y} = \mathbb{E}_x\mathbb{E}_{y|x}$ from the law of total expectation. Equation (63) is obtained from (62) as the term $f_\theta(x) - \mathbb{E}_{y'|x}(g(x,y')|x)$ is not a function of y. Equation (64) is obtained from (63) as $\mathbb{E}_{y|x}\left(\mathbb{E}_{y'|x}(g(x,y')|x)\right) = \mathbb{E}_{y'|x}(g(x,y')|x)$ because $\mathbb{E}_{y'|x}(g(x,y')|x)$ is not a function of y hence is constant with respect to the expectation operator $\mathbb{E}_{y|x}$.

Thus plugging in value of $2\mathbb{E}_{x,y}\left(f_\theta(x)-\mathbb{E}_{y'|x}(g(x,y')|x)\right)\left(g(x,y)-\mathbb{E}_{y'|x}(g(x,y')|x)\right)$ in Equation (61) we get

$$
\arg\min_{f_\theta}\mathbb{E}_{x,y}\left(f_\theta(x)-g(x,y)\right)^2 = \arg\min_{f_\theta}\left(\mathbb{E}_{x,y}\left(f_\theta(x)-\mathbb{E}_{x,y'}(g(x,y')|x)\right)^2+\mathbb{E}_{x,y}\left(g(x,y)-\mathbb{E}_{y'|x}(g(x,y')|x)\right)^2\right) \tag{68}
$$

Note that the second term on the right hand side of Equation (68) des not depend on $f_\theta(x)$ therefore we can write Equation (68) as

$$
\arg\min_{f_\theta}\mathbb{E}_{x,y}\left(f_\theta(x)-g(x,y)\right)^2 = \arg\min_{f_\theta}\left(\mathbb{E}_{x,y}\left(f_\theta(x)-\mathbb{E}_{y'|x}(g(x,y')|x)\right)^2\right) \tag{69}
$$

Since the right hand side of Equation (69) is not a function of y we can replace $\mathbb{E}_{x,y}$ with $\mathbb{E}_x$ to get

$$
\arg\min_{f_\theta}\mathbb{E}_{x,y}\left(f_\theta(x)-g(x,y)\right)^2 = \arg\min_{f_\theta}\left(\mathbb{E}_{x}\left(f_\theta(x)-\mathbb{E}_{y'|x}(g(x,y')|x)\right)^2\right) \tag{70}
$$

□

Lemma 7. *Consider an optimization of the form given in Equation* (47) *with the regularization term* $\beta = 0$ *denoted by* $\mathcal{L}_{|\tilde{D}|}$ *and it's convex equivalent denoted by* $\mathcal{L}_0$*. Then the value of these two loss functions evaluated at* $(v,w) = (v_i,w_i)_{i\in\{1,\cdots,|\tilde{D}|\}}$ *and* $\theta = \psi(v_i,w_i)_{i\in\{1,\cdots,|\tilde{D}|\}}$ *respectively are equal and thus we have*

$$
\mathcal{L}_{|\tilde{D}|}(\psi(v_i,w_i)_{i\in\{1,\cdots,|\tilde{D}|\}}) = \mathcal{L}_0((v_i,w_i)_{i\in\{1,\cdots,|\tilde{D}|\}}) \tag{71}
$$

Proof. Consider the loss functions in Equations (45), (50) with $\beta = 0$ are as follows

$$\mathcal{L}_0((v_i, w_i)_{i\in\{1,\cdots,|\tilde{D}|\}}) = ||\sum_{D_i\in\tilde{D}} D_i(X(v_i - w_i)) - y||_2^2 \tag{72}$$

$$\mathcal{L}_{|\tilde{D}|}(\psi(v_i, w_i)_{i\in\{1,\cdots,|\tilde{D}|\}}) = ||\sum_{i=1}^{|\tilde{D}|} \sigma(X\psi(v_i, w_i)_1)\psi(v_i, w_i)_2 - y||_2^2, \tag{73}$$

where $\psi(v_i, w_i)_1$, $\psi(v_i, w_i)_2$ represent the first and second coordinates of $\psi(v_i, w_i)$ respectively.

For any fixed $i \in \{1, \cdots, |\tilde{D}|\}$ consider the two terms

$$D_i(X(v_i - w_i)) \tag{74}$$
$$\sigma(X\psi(v_i, w_i)_1)\psi(v_i, w_i)_2 \tag{75}$$

For a fixed i either v_i or w_i is zero. In case both are zero, both of the terms in Equations (74) and (75) are zero as $\psi(0, 0) = (0, 0)$. Assume that for a given i $w_i = 0$. Then we have $\psi(v_i, w_i) = (v_i, 1)$. Then equations (74), (75) are.

$$D_i(X(v_i) \tag{76}$$
$$\sigma(X(v_i)) \tag{77}$$

But by definition of v_i we have $D_i(X(v_i) = \sigma(X(v_i))$, therefore Equations (76), (77) are equal. Alternatively if for a given i $v_i = 0$, then $\psi(v_i, w_i) = (w_i, -1)$, then the terms in (74), (75) become.

$$-D_i(X(w_i) \tag{78}$$
$$-\sigma(X(w_i)) \tag{79}$$

By definition of w_i we have $D_i(X(w_i) = \sigma(X(w_i))$, then the terms in (78), (78) are equal. Since this is true for all i, we have

$$\mathcal{L}_{|\tilde{D}|}(\psi(v_i, w_i)_{i\in\{1,\cdots,|\tilde{D}|\}}) = \mathcal{L}_0((v_i, w_i)_{i\in\{1,\cdots,|\tilde{D}|\}}) \tag{80}$$

□

Lemma 8. *The function $Q_\theta(x)$ defined in equation* (14) *is Lipschitz continuous in θ, where θ is considered a vector in $\mathbb{R}^{(d+1)m}$ with the assumption that the set of all possible θ belong to the set $\mathcal{B} = \{\theta : |\theta^* - \theta|_1 < 1\}$, where θ^* is some fixed value.*

Proof. First we show that for all $\theta_1 = \{u_i, \alpha_i\}, \theta_2 = \{u_i^{'}, \alpha_i^{'}\} \in \mathcal{B}$ we have $\alpha_i = \alpha_i^{'}$ for all $i \in (1, \cdots, m)$

Note that

$$|\theta_1 - \theta_2|_1 = \sum_{i=1}^{m} |u_i - u_i^{'}|_1 + \sum_{i=1}^{m} |\alpha_i - \alpha_i^{'}|, \tag{81}$$

where $|u_i - u_i^{'}|_1 = \sum_{j=1}^{d} |u_{i_j} - u_{i_j}^{'}|$ with $u_{i_j}, u_{i_j}^{'}$ denote the jth component of $u_i, u_i^{'}$ respectively.

By construction $\alpha_i, \alpha_i^{'}$ can only be 1, -1 or 0. Therefore if $\alpha_i \neq \alpha_i^{'}$ then $|\alpha_i - \alpha_i^{'}| = 2$ if both non zero or $|\alpha_i - \alpha_i^{'}| = 1$ if one is zero. Therefore $|\theta_1 - \theta_2|_1 \geq 1$. Which leads to a contradiction.

Therefore $\alpha_i = \alpha_i^{'}$ for all i and we also have

$$|\theta_1 - \theta_2|_1 = \sum_{i=1}^{m} |u_i - u_i^{'}|_1 \tag{82}$$

$Q_\theta(x)$ is defined as

$$Q_\theta(x) = \sum_{i=1}^{m} \sigma^{'}(x^T u_i)\alpha_i \tag{83}$$

From Proposition 1 in (Virmaux and Scaman, 2018) the function $Q_\theta(x)$ is Lipschitz continuous in x, therefore there exist $l > 0$ such that

$$|Q_\theta(x) - Q_\theta(y)| \leq l|x - y|_1 \tag{84}$$

$$|\sum_{i=1}^{m} \sigma^{'}(x^T u_i)\alpha_i - \sum_{i=1}^{m} \sigma^{'}(y^T u_i)\alpha_i| \leq l|x - y|_1 \tag{85}$$

If we consider a single neuron of Q_θ, for example $i = 1$, we have $l_1 > 0$ such that

$$|\sigma^{'}(x^T u_1)\alpha_i - \sigma^{'}(y^T u_1)\alpha_i| \leq l_1|x - y|_1 \tag{86}$$

Now consider Equation (86), but instead of considering the left hand side a a function of x, y consider it a function of u where we consider the difference between $\sigma^{'}(x^T u)\alpha_i$ evaluated at u_1 and $u_1^{'}$ such that

$$|\sigma^{'}(x^T u_1)\alpha_i - \sigma^{'}(x^T u_1^{'})\alpha_i| \leq l_1^x|u_1 - u_1^{'}|_1 \tag{87}$$

for some $l_1^x > 0$.

Similarly, for all other i if we change u_i to $u_i^{'}$ to be unchanged we have

$$|\sigma^{'}(x^T u_i)\alpha_i - \sigma^{'}(x^T u_i^{'})\alpha_i| \leq l_i^x|u_i - u_i^{'}|_1 \tag{88}$$

for all x if both $\theta_1, \theta_2 \in \mathcal{B}$.

Therefore we obtain

$$|\sum_{i=1}^{m} \sigma^{'}(x^T u_i)\alpha_i - \sum_{i=1}^{m} \sigma^{'}(x^T u_i^{'})\alpha_i| \leq \sum_{i=1}^{m} |\sigma^{'}(x^T u_i)\alpha_i - (x^T u_i^{'})\alpha_i| \tag{89}$$

$$\leq \sum_{i=1}^{m} l_i^x|u_i - u_i^{'}|_1 \tag{90}$$

$$\leq (\sup_i l_i^x) \sum_{i=1}^{m} |u_i - u_i^{'}|_1 \tag{91}$$

$$\leq (\sup_i l_i^x)|\theta_1 - \theta_2| \tag{92}$$

This result for a fixed x. If we take the supremum over x on both sides we get

$$\sup_x |\sum_{i=1}^{m} \sigma^{'}(x^T u_i)\alpha_i - \sum_{i=1}^{m} \sigma^{'}(x^T u_i^{'})\alpha_i| \quad \leq \quad (\sup_{i,x} l_i^x)|\theta_1 - \theta_2| \tag{93}$$

Denoting $(\sup_{i,x} l_i^x) = l$, we get

$$|\sum_{i=1}^{m} \sigma^{'}(x^T u_i)\alpha_i - \sum_{i=1}^{m} \sigma^{'}(x^T u_i^{'})\alpha_i| \quad \leq \quad l|\theta_1 - \theta_2|_1 \tag{94}$$

$$\forall x \in \mathbb{R}^d \tag{95}$$

□

D. Proof of Theorem 1

Proof. For ease of notations, let Q_1, Q_2 be two real valued functions on the state action space. The expression $Q_1 \geq Q_2$ implies $Q_1(s,a) \geq Q_2(s,a)\ \forall (s,a) \in \mathcal{S} \times \mathcal{A}$.

Q_k denotes our estimate of the action value function at step k of Algorithm 1 and Q^{π_k} denotes the action value function induced by the policy π_k which is the greedy policy with respect to Q_k.

Consider $\epsilon_{k+1} = TQ_k - Q_{k+1}$.

$$TQ_k \geq T^{\pi^*} Q_k \tag{96}$$

This follows from the definition of T^{π^*} and T in Equation (3) and (4), respectively.

Thus we get,

$$\begin{aligned}
Q^* - Q_{k+1} &= T^{\pi^*} Q^* - Q_{k+1} && (97)\\
&= T^{\pi^*} Q^* - T^{\pi^*} Q_k + T^{\pi^*} Q_k - TQ_k + TQ_k - Q_{k+1} && (98)\\
&= r(s,a) + \gamma P^{\pi^*} Q^* - (r(s,a) + \gamma P^{\pi^*} Q_k) + (r(s,a) + \gamma P^{\pi^*} Q_k) - (r(s,a) + \gamma P^* Q_k) + \epsilon_{k+1} \\
&= \gamma P^{\pi^*}(Q^* - Q_k) + \gamma P^{\pi^*} Q_k - \gamma P^* Q_k + \epsilon_{k+1} && (99)\\
&\leq \gamma(P^{\pi^*}(Q^* - Q_k)) + \epsilon_{k+1} && (100)
\end{aligned}$$

Right hand side of Equation (97) is obtained by writing $Q^* = T^{\pi^*} Q^*$. This is because the function Q^* is a stationary point with respect to the operator T^{π^*}. Equation (98) is obtained from (97) by adding and subtracting $T^{\pi^*} Q_k$. Equation (100) is obtained from (99) as $P^{\pi^*} Q_k \leq P^* Q_k$ and P^* is the operator with respect to the greedy policy of Q_k.

By recursion on k, we get,

$$Q^* - Q_K \leq \sum_{k=0}^{K-1} \gamma^{K-k-1} (P^{\pi^*})^{K-k-1} \epsilon_k + \gamma^K (P^{\pi^*})^K (Q^* - Q_0) \tag{101}$$

using $TQ_K \geq T^{\pi^*} Q_K$ (from definition of T^{π^*}) and $TQ_K = T^{\pi_K} Q_K$ as π_k is the greedy policy with respect to Q_k hence T^{π_K} acts on it the same way T does.

Similarly we write,

$$
\begin{aligned}
Q^* - Q^{\pi_K} &= T^{\pi^*}Q^* - T^{\pi_K}Q^{\pi_K} && (102)\\
&= T^{\pi^*}Q^* - T^{\pi^*}Q_K + T^{\pi^*}Q_K - TQ_K + TQ_k - T^{\pi_K}Q^{\pi_K} && (103)\\
&\leq T^{\pi^*}Q^* - T^{\pi^*}Q_K + TQ_k - T^{\pi_K}Q^{\pi_K} && (104)\\
&\leq r(s,a) + \gamma P^{\pi^*}Q^* - (r(s,a) + \gamma P^{\pi^*}Q_K) + (r(s,a) + \gamma P^*Q_k) - (r(s,a) + \gamma P^{\pi_K}Q^{\pi_K}) && (105)\\
&= \gamma P^{\pi^*}(Q^* - Q_K) + \gamma P^{\pi_K}(Q_K - Q^{\pi_K}) && (106)\\
&= \gamma P^{\pi^*}(Q^* - Q_K) + \gamma P^{\pi_K}(Q_K - Q^* + Q^* - Q^{\pi_K}) && (107)
\end{aligned}
$$

The right hand side of Equation (103) is obtained by adding and subtracting $T^{\pi^*}Q_K$ and TQ_K to the right hand side of Equation (102). Equation (104) is obtained from (103) by noting that the term $T^{\pi^*}Q_K - TQ_K$ is non-positive for all (s,a) as T is the greedy operator on a Q function and results in a higher or equal value than any other operator. Equation (106) is obtained from (105) by writing $P^{\pi_k}Q^{\pi_k} = P^*Q^{\pi_k}$. This is true as π_k is the greedy policy with respect to Q^{π_k}, hence the operator P^* acts on Q^{π_k} in the same way as P^{π_k}. Equation (107) is obtained from (106) by adding and subtracting Q^* to the second term on the right hand side.

By rearranging the terms in Equation (107) we get

$$
\begin{aligned}
Q^* - Q^{\pi_K} &\leq \gamma P^{\pi^*}(Q^* - Q_K) + \gamma P^{\pi_K}(Q_K - Q^* + Q^* - Q^{\pi_K}) && (108)\\
&\leq \gamma(P^{\pi^*} - P^{\pi_K})(Q^* - Q_K) + \gamma P^{\pi_K}(+Q^* - Q^{\pi_K}) && (109)
\end{aligned}
$$

Thus, we have

$$
(I - \gamma P^{\pi_K})(Q^* - Q^{\pi_K}) \leq \gamma(P^{\pi^*} - P^{\pi_K})(Q^* - Q_K) \qquad (110)
$$

This implies

$$
(Q^* - Q^{\pi_K}) \leq \gamma(I - \gamma P^{\pi_K})^{-1}\left(P^{\pi^*} - P^{\pi_K}\right)(Q^* - Q_K) \qquad (111)
$$

Equation (111) is obtained from (110) by taking the inverse of the operator $(I - \gamma P^{\pi_K})$ on both sides of (110)

Plugging in value of $(Q^* - Q_K)$ from Equation (101) in Equation (111), we obtain

$$
\begin{aligned}
Q^* - Q^{\pi_K} &\leq \gamma(I - \gamma P^{\pi_K})^{-1}\left(P^{\pi^*} - P^{\pi_K}\right)\left[\sum_{k=1}^{K-1}\gamma^{K-k-1}(P^{\pi^*})^{K-k-1}\epsilon_k + \gamma^K(P^{\pi^*})^K(Q^* - Q_0)\right] && (112)\\
&\leq \gamma(I - \gamma P^{\pi_K})^{-1}\left(P^{\pi^*} - P^{\pi_K}\right)\left[\sum_{k=1}^{K-1}\gamma^{K-k-1}(P^{\pi^*})^{K-k-1}|\epsilon_k| + \gamma^K(P^{\pi^*})^K(Q^* - Q_0)\right] && (113)\\
&\leq \gamma(I - \gamma P^{\pi_K})^{-1}\left(P^{\pi^*} + P^{\pi_K}\right)\left[\sum_{k=1}^{K-1}\gamma^{K-k-1}(P^{\pi^*})^{K-k-1}|\epsilon_k| + \gamma^K(P^{\pi^*})^K(Q^* - Q_0)\right] && (114)\\
&\leq \gamma(I - \gamma P^{\pi_K})^{-1}\left(P^{\pi^*} + P^{\pi_K}\right)\left[\sum_{k=1}^{K-1}\gamma^{K-k-1}(P^{\pi^*})^{K-k-1}|\epsilon_k| + \gamma^K(P^{\pi^*})^K(Q^* - Q_0)\right] && (115)\\
&\leq \gamma(I - \gamma P^{\pi_K})^{-1}\Bigg[\sum_{k=1}^{K-1}\gamma^{K-k-1}\left((P^{\pi^*})^{K-k} + P^{\pi_k}(P^{\pi^*})^{K-k-1}\right)|\epsilon_k| + \\
&\qquad \gamma^K\left((P^{\pi^*})^{K+1} + P^{\pi_k}(P^{\pi^*})^K\right)(Q^* - Q_0)\Bigg] && (116)
\end{aligned}
$$

Equation (114) is obtained from Equation (113) by noting that the operator $\left(P^{\pi^*} + P^{\pi_K}\right)$ acting on $|\epsilon_k|$ will produce a larger value than $\left(P^{\pi^*} - P^{\pi_K}\right)$ acting on $|\epsilon_k|$.

$$Q^* - Q^{\pi_K} \leq (I - \gamma P^{\pi_K})^{-1} \Bigg[\sum_{k=1}^{K-1} \gamma^{K-k-1} \left((P^{\pi^*})^{K-k} + P^{\pi_k} (P^{\pi^*})^{K-k-1} \right) |\epsilon_k| + \gamma^K \left((P^{\pi^*})^{K+1} + P^{\pi_k} (P^{\pi^*})^K \right) (Q^* - Q_0) \Bigg] \tag{117}$$

We know $Q^* \leq \frac{R_{\max}}{1-\gamma}$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ and $Q_0 = 0$ by initialization. Therefore, we have

$$Q^* - Q^{\pi_K} \leq (I - \gamma P^{\pi_K})^{-1} \Bigg[\sum_{k=1}^{K-1} \gamma^{K-k-1} \left((P^{\pi^*})^{K-k} + P^{\pi_k} (P^{\pi^*})^{K-k-1} \right) |\epsilon_k| + \gamma^K \left((P^{\pi^*})^{K+1} + P^{\pi_k} (P^{\pi^*})^K \right) \frac{R_{\max}}{(1-\gamma)} \Bigg] \tag{118}$$

For simplicity, in stating our results, we also define

$$\alpha_k \triangleq \begin{cases} \frac{(1-\gamma)\gamma^{K-k-1}}{1-\gamma^{K+1}} & \text{if } 0 \leq k < K, \\ \frac{(1-\gamma)\gamma^K}{1-\gamma^{K+1}} & \text{if } k = K \end{cases} \tag{119}$$

$$A_k \triangleq \begin{cases} \frac{(1-\gamma)}{2} (I - \gamma P^{\pi_K})^{-1} \left((P^{\pi^*})^{K-k} + P^{\pi_K} (P^{\pi^*})^{K-k-1} \right) & \text{if } 0 \leq k < K, \\ \frac{(1-\gamma)}{2} (I - \gamma P^{\pi_K})^{-1} \left((P^{\pi^*})^{K+1} + P^{\pi_K} (P^{\pi^*})^K \right) & \text{if } k = K \end{cases} \tag{120}$$

We then substitute the value α_k and A_k from Equation (119) and (120) respectively to get,

$$Q^* - Q^{\pi_K} \leq \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2} \left[\sum_{k=1}^{K-1} \alpha_k A_k |\epsilon_k| + \alpha_K A_K \frac{R_{\max}}{1-\gamma} \right] \tag{121}$$

Taking an expectation on both sides with respect to $\mu \in \mathcal{P}(\mathcal{S} \times \mathcal{A})$ we get

$$\mathbb{E}(Q^* - Q^{\pi_K})_\mu \leq \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2} \left[\sum_{k=1}^{K-1} \underbrace{\mathbb{E}(\alpha_k A_k |\epsilon_k|)_\mu}_{A} + \underbrace{\alpha_K A_K \frac{R_{\max}}{1-\gamma}}_{B} \right] \tag{122}$$

Consider the term A in Equation (122) Plugging in the value of $\alpha_k A_k$ in the term $\mathbb{E}(\alpha_k A_k(\epsilon_k))_\nu$, we get

$$\begin{aligned} A &= \mathbb{E}\left(\frac{(1-\gamma)^2 \gamma^{K-k}}{2(1-\gamma^{K+1})} (I - \gamma P^{\pi_k})^{-1} \left((P^{\pi^*})^{K-k} + (P^{\pi^*})^{K-k-1} P^{\pi_k} \right) |\epsilon_k| \right)_\mu && (123) \\ &= \mathbb{E}\left(\frac{(1-\gamma)^2 \gamma^{K-k}}{2(1-\gamma^{K+1})} \sum_{m=1}^{\infty} (\gamma P^{\pi_k})^m \left((P^{\pi^*})^{K-k} + (P^{\pi^*})^{K-k-1} P^{\pi_k} \right) |\epsilon_k| \right)_\mu && (124) \\ &= \mathbb{E}\left(\frac{(1-\gamma)^2 \gamma^{K-k}}{2(1-\gamma^{K+1})} \sum_{m=1}^{\infty} \left((\gamma P^{\pi_k})^m (P^{\pi^*})^{K-k} + (\gamma P^{\pi_k})^{m+1} (P^{\pi^*})^{K-k-1} \right) |\epsilon_k| \right)_\mu && (125) \\ &= \frac{(1-\gamma)^2 \gamma^{K-k}}{2(1-\gamma^{K+1})} \sum_{m=1}^{\infty} \left(\underbrace{\mathbb{E}\left((\gamma P^{\pi_k})^m (P^{\pi^*})^{K-k} |\epsilon_k| \right)_\mu}_{A1} + \underbrace{\mathbb{E}\left((\gamma P^{\pi_k})^{m+1} (P^{\pi^*})^{K-k-1} |\epsilon_k| \right)_\mu}_{A2} \right) && (126) \\ & && (127) \end{aligned}$$

Equation (124) is obtained from (123) by writing $(I-\gamma P^{\pi_k})^{-1} = \sum_{m=1}^{\infty}(\gamma P^{\pi_k})^m$ using the binomial expansion formula.

Consider the term $A1$ in Equation (125), We have

$$\begin{aligned} A1 &= \mathbb{E}\left((\gamma P^{\pi_k})^m (P^{\pi^*})^{K-k}|\epsilon_k|\right)_\mu && (128)\\ &= \int_{\mathcal{S}\times\mathcal{A}} (\gamma P^{\pi_k})^m (P^{\pi^*})^{K-k}|\epsilon_k| d\mu(s,a) && (129)\\ &\leq \int_{\mathcal{S}\times\mathcal{A}} \frac{d\left((\gamma P^{\pi_k})^m (P^{\pi^*})^{K-k}\mu\right)}{d\nu}|\epsilon_k| d\nu(s,a) && (130)\\ &\leq \gamma^m \phi_{\nu,\mu}\left(\int_{\mathcal{S}\times\mathcal{A}} |\epsilon_k| d\nu(s,a)\right) && (131)\\ &\leq \gamma^m \phi_{\nu,\mu}\mathbb{E}(|\epsilon_k|)_\nu && (132) \end{aligned}$$

Here, Equation (131) is obtained from (130) from Assumption 4.

In the same manner, we have

$$A2 \leq \gamma^m \phi_{\nu,\mu}\mathbb{E}(|\epsilon_k|)_\nu \tag{133}$$

Denote the term in Equation (129) as

$$\int_{\mathcal{S}\times\mathcal{A}} (\gamma P^{\pi_k})^m (P^{\pi^*})^{K-k}(|\epsilon_k|) d\mu(s,a) = \int_{\mathcal{S}\times\mathcal{A}} (|\epsilon_k|) d\tilde{\nu} \tag{134}$$

Where $(P^{\pi_k})^m (P^{\pi^*})^{K-k}\mu$ is the marginal distribution of the state action pair at step $m+K-k+1$ denoted by $\tilde{\nu}$ for notational simplicity. It is the state action distribution at step $m+K-k+1$ obtained by starting the state action pair sampled form μ and the following the policies π^* for $K-k$ steps and then π_k for m steps. We then get Equation (130) from (129) from the definition of the Radon Nikodym derivative as follows.

$$\int_{\mathcal{S}\times\mathcal{A}} |\epsilon_k| d\tilde{\nu} = \int_{\mathcal{S}\times\mathcal{A}} \frac{d\tilde{\nu}}{d\nu}|\epsilon_k| d\nu \tag{135}$$

Plugging upper bound on $A1$ and $A2$ into Equation (126), we obtain

$$\begin{aligned} A &\leq \frac{(1-\gamma)^2\gamma^{K-k}}{2(1-\gamma^{K+1})}\sum_{m=1}^{\infty}(2\phi_{\nu,\mu}\gamma^m\mathbb{E}(|\epsilon_k|)_\nu) && (136)\\ &\leq \frac{(1-\gamma)^2\gamma^{K-k}}{2(1-\gamma^{K+1})}\left(\frac{2\phi_{\nu,\mu}\mathbb{E}(|\epsilon_k|)_\nu}{1-\gamma}\right) && (137)\\ &\leq \frac{\phi_{\nu,\mu}(1-\gamma)\gamma^{K-k}}{(1-\gamma^{K+1})}\mathbb{E}(|\epsilon_k|)_\nu && (138)\\ &\leq \frac{\phi_{\nu,\mu}\gamma^{K-k}}{(1-\gamma^{K+1})}\mathbb{E}(|\epsilon_k|)_\nu && (139) \end{aligned}$$

Now consider B in Equation (122). We have

$$
\begin{aligned}
B &= \alpha_K A_k \frac{R_{max}}{1-\gamma} && (140)\\
&= \alpha_K (1-\gamma) \frac{R_{max}}{1-\gamma} && (141)\\
&= \frac{\gamma^K R_{max}}{(1-\gamma^{K+1})} && (142)
\end{aligned}
$$

Equation (141) is obtained from the fact that the transition operator A_k acting on the constant $\frac{R_{max}}{1-\gamma}$ is equivalent to multiplying it by $1-\gamma$. Equation (142) is obtained by plugging in the value of α_K from Equation (119). Plugging upper bound on A and value of B into Equation (122), we obtain

$$
\mathbb{E}(Q^* - Q^{\pi_K})_\mu \leq \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2} \left[\sum_{k=1}^{K-1} \frac{\phi_{\nu,\mu}\gamma^{K-k}}{(1-\gamma^{K+1})} \mathbb{E}(|\epsilon_k|)_\nu + \frac{\gamma^K R_{max}}{2(1-\gamma^{K+1})} \right] \tag{143}
$$

$$
\mathbb{E}(Q^* - Q^{\pi_K})_\mu \leq \frac{2\phi_{\nu,\mu}\gamma}{(1-\gamma)} \left[\sum_{k=1}^{K-1} \gamma^{K-k} \mathbb{E}(|\epsilon_k|)_\nu \right] + \frac{2R_{\max}\gamma^{K+1}}{(1-\gamma)^2} \tag{144}
$$

From Equation (39) we have that

$$
\begin{aligned}
\epsilon_k &= TQ_{k-1} - Q_k \\
&= \epsilon_{k_1} + \epsilon_{k_2} + \epsilon_{k_3} + \epsilon_{k_4} && (145)
\end{aligned}
$$

Thus, equation (144) becomes

$$
\begin{aligned}
E(Q^* - Q^{\pi_K})_\mu &\leq \frac{2\gamma}{(1-\gamma)} \left[\sum_{k=1}^{K-1} \gamma^{K-k} \mathbb{E}(|\epsilon_{k_1} + \epsilon_{k_2} + \epsilon_{k_3} + \epsilon_{k_4}|)_\mu \right] + \frac{2R_{\max}\gamma^{K+1}}{(1-\gamma)^3} && (146)\\
&\leq \underbrace{\frac{2\gamma}{(1-\gamma)} \left[\sum_{k=1}^{K-1} \gamma^{K-k} \mathbb{E}(|\epsilon_{k_1}|)_\mu \right]}_{T1} + \underbrace{\frac{2\gamma}{(1-\gamma)} \left[\sum_{k=1}^{K-1} \gamma^{K-k} \mathbb{E}(|\epsilon_{k_2}|)_\mu \right]}_{T2} \\
&\quad + \underbrace{\frac{2\gamma}{(1-\gamma)} \left[\sum_{k=1}^{K-1} \gamma^{K-k} \mathbb{E}(|\epsilon_{k_3}|)_\mu \right]}_{T3} + \underbrace{\frac{2\gamma}{(1-\gamma)} \left[\sum_{k=1}^{K-1} \gamma^{K-k} \mathbb{E}(|\epsilon_{k_4}|)_\mu \right]}_{T4} \\
&\quad + \underbrace{\frac{2R_{\max}\gamma^{K+1}}{(1-\gamma)^2}}_{T5} && (147)
\end{aligned}
$$

Consider $T1$, from Lemma 1, we have $\mathbb{E}(|\epsilon_{k_1}|)_\nu \leq \sqrt{\epsilon_{bias}}$. Thus, for term $T1$, we obtain

$$
\begin{aligned}
T1 &= \frac{2\phi_{\nu,\mu}\gamma}{(1-\gamma)}\left[\sum_{k=1}^{K-1}\gamma^{K-k}\mathbb{E}(|\epsilon_{k_1}|)_\nu\right] && (148)\\
&\leq \frac{2\phi_{\nu,\mu}\gamma}{(1-\gamma)}\left[\sum_{k=1}^{K-1}\gamma^{K-k}\sqrt{\epsilon_{bias}}\right] && (149)\\
&\leq \frac{2\phi_{\nu,\mu}\sqrt{\epsilon_{bias}}\gamma}{(1-\gamma)}\frac{1}{1-\gamma} && (150)\\
T1 &\leq \frac{2\phi_{\nu,\mu}\gamma\sqrt{\epsilon_{bias}}}{(1-\gamma)^2} && (151)
\end{aligned}
$$

Equation (150) is obtained from (149) by using the inequality $\sum_{k=1}^{K-1}\gamma^{K-k} \leq \frac{1}{1-\gamma}$.

Consider $T2$ we have from Lemma 2 $\epsilon_{k2} = 0$.

Therefore

$$T2 = 0 \tag{152}$$

Consider $T3$, from Lemma 3, we have that if the number of samples of state action pairs at iteration k denoted by n_k satisfy

$$n_k \geq \tilde{\mathcal{O}}(\epsilon^{-2}(1-\gamma)^{-2}) \tag{153}$$

then we have

$$\mathbb{E}\left(\epsilon_{k_3}\right)_{\zeta_\nu^{\pi_k}} \leq \epsilon \tag{154}$$

Further we have

$$\mathbb{E}\left(|\epsilon_{k_3}|\right)_\nu \leq (\phi_{\zeta_\nu^{\pi_k},\nu})\mathbb{E}\left(|\epsilon_{k_3}|\right)_{\zeta_\nu^{\pi_k}} \leq (\phi_{\zeta_\nu^{\pi_k},\nu})\epsilon \tag{155}$$

Plugging in $\epsilon = \frac{\epsilon}{6\phi_{\zeta_\nu^{\pi_k},\nu}\phi_{\nu,\mu}\gamma}(1-\gamma)^2$ in Equation (155), we get that if number of samples of state action pairs at iteration k denoted by n_k satisfy

$$
\begin{aligned}
n_k \quad &\geq \quad \tilde{\mathcal{O}}(\epsilon^{-2}(1-\gamma)^{-6}) \\
&\qquad \forall k \in \{1, \cdots, K\}
\end{aligned} \tag{156}
$$

then we have

$$
\begin{aligned}
T3 &= \frac{2\phi_{\nu,\mu}\gamma}{(1-\gamma)}\left[\sum_{k=1}^{K-1}\gamma^{K-k}\mathbb{E}(|\epsilon_{k_3}|)_\nu\right] && (157)\\
&\leq \frac{2\phi_{\nu,\mu}\gamma}{(1-\gamma)}\left[\sum_{k=1}^{K-1}\gamma^{K-k}\epsilon\frac{(1-\gamma)^2}{6\phi_{\nu,\mu}\gamma}\right] && (158)\\
&\leq \frac{2\gamma\epsilon}{(1-\gamma)}\left[\sum_{k=1}^{K-1}\gamma^{K-k}\frac{(1-\gamma)^2}{6\gamma}\right] && (159)\\
&\leq \frac{\epsilon}{3(1-\gamma)}\left[\sum_{k=1}^{K-1}\gamma^{K-k}(1-\gamma)^2\right] && (160)\\
&\leq \frac{\epsilon}{3(1-\gamma)}\left[\frac{1}{1-\gamma}(1-\gamma)^2\right] && (161)\\
&\leq \frac{\epsilon}{3}(1-\gamma^K) && (162)\\
T3 &\leq \frac{\epsilon}{3} && (163)
\end{aligned}
$$

Equation (161) is obtained from (160) by using the inequality $\sum_{k=1}^{K-1}\gamma^{K-k} \leq \frac{1}{1-\gamma}$.

Consider $T4$. From Lemma 4, we have that if the number of iterations of the projected gradient descent algorithm at iteration k denoted by T_k and the step size denoted by α_k satisfy

$$
\begin{aligned}
T_k &\geq \left(\frac{C_k^{'}\epsilon}{l_k}\right)^{-2} L_k^2||u_k^*||_2^2 - 1 && (164)\\
\alpha_k &= \frac{||u_k^*||_2}{L_k\sqrt{T_k+1}} && (165)\\
& && (166)
\end{aligned}
$$

and $\epsilon < C_k^{'}$, then we have

$$
\mathbb{E}(|\epsilon_{k4}|)_\nu \leq \epsilon + \epsilon_{|\tilde{D}|} \tag{167}
$$

Plugging in $\epsilon = \epsilon\frac{(1-\gamma)^2}{6\phi_{\nu,\mu}\gamma}$ in Equation (164) we get that if the number of iterations of the projected gradient descent algorithm at iteration k denoted by T_k and the step size denoted by α_k satisfy

$$
\begin{aligned}
T_k &\geq \left(\frac{C_k^{'}\epsilon(1-\gamma)^2}{6l\phi_{\nu,\mu}\gamma}\right)^{-2} L_k^2||u_k^*||_2^2 - 1 && (168)\\
\alpha_k &= \frac{||u_k^*||_2}{L_k\sqrt{T_k+1}} && (169)\\
& \forall k \in \{1,\cdots,K\}
\end{aligned}
$$

and if $\epsilon < C_k^{'}$ for all $k \in (1,\cdots,K)$, then we have

$$
\begin{aligned}
T4 &= \frac{2\phi_{\nu,\mu}\gamma}{(1-\gamma)}\left[\sum_{k=1}^{K-1}\gamma^{K-k}\mathbb{E}(|\epsilon_{k_4}|)_\nu\right] && (170)\\
&\leq \frac{2\phi_{\nu,\mu}\gamma}{(1-\gamma)}\left[\sum_{k=1}^{K-1}\gamma^{K-k}\epsilon\frac{(1-\gamma)^2}{6\phi_{\nu,\mu}\gamma}\right]+\frac{2\phi_{\nu,\mu}\gamma}{(1-\gamma)}\left[\sum_{k=1}^{K-1}\gamma^{K-k}\epsilon_{|\tilde{D}|}\right] && (171)\\
&\leq \frac{2\gamma\epsilon}{(1-\gamma)}\left[\sum_{k=1}^{K-1}\gamma^{K-k}\frac{(1-\gamma)^2}{6\gamma}\right]+\frac{2\phi_{\nu,\mu}\epsilon_{|\tilde{D}|}\gamma}{(1-\gamma)}\frac{1}{1-\gamma} && (172)\\
&\leq \frac{\epsilon}{3(1-\gamma)}\left[\sum_{k=1}^{K-1}\gamma^{K-k}(1-\gamma)^2\right]+\frac{2\phi_{\nu,\mu}\epsilon_{|\tilde{D}|}}{(1-\gamma)^2} && (173)\\
&\leq \frac{\epsilon}{3(1-\gamma)}\left[\frac{1}{1-\gamma}(1-\gamma)^2\right]+\frac{2\phi_{\nu,\mu}\epsilon_{|\tilde{D}|}}{(1-\gamma)^2} && (174)\\
&\leq \frac{\epsilon}{3}(1-\gamma^K)+\frac{2\phi_{\nu,\mu}\epsilon_{|\tilde{D}|}}{(1-\gamma)^2} && (175)\\
&\leq \frac{\epsilon}{3}+\frac{2\phi_{\nu,\mu}\epsilon_{|\tilde{D}|}}{(1-\gamma)^2} && (176)
\end{aligned}
$$

Equation (171), (174) are obtained from (170), (173), respectively, by using the inequality $\sum_{k=1}^{K-1}\gamma^{K-k}\leq\frac{1}{1-\gamma}$.

Consider $T5$. Assume K is large enough such that

$$
\begin{aligned}
K &\geq \frac{1}{\log\left(\frac{1}{\gamma}\right)}\log\left(\frac{6R_{max}}{\epsilon(1-\gamma)^2}\right)-1 && (177)\\
(K+1)\log\left(\frac{1}{\gamma}\right) &\geq \log\left(\frac{6R_{max}}{\epsilon(1-\gamma)^2}\right) && (178)\\
(K+1)\log(\gamma) &\leq \log\left(\frac{\epsilon(1-\gamma)^2}{6R_{max}}\right) && (179)\\
\gamma^{K+1} &\leq \frac{\epsilon(1-\gamma)^2}{6R_{max}} && (180)\\
\frac{2R_{max}\gamma^{K+1}}{(1-\gamma)^2} &\leq \frac{\epsilon}{3} && (181)
\end{aligned}
$$

Equation (179) is obtained from (178) by multiplying on -1 on both sides and noting that $\log(x)=-\log\left(\frac{1}{x}\right)$

Thus, we obtain that if the number of iterations of Algorithm 1, number of state actions pairs sampled at iteration k denoted by n_k and the number of iterations of the projected gradient descent on iteration k denoted by T_k and the step size denoted by α_k satisfy

$$K \geq \frac{1}{\log\left(\frac{1}{\gamma}\right)} \log\left(\frac{6R_{max}}{\epsilon(1-\gamma)^2}\right) - 1 \tag{182}$$

$$n_k \geq \tilde{\mathcal{O}}(\epsilon^{-2}(1-\gamma)^{-6}) \tag{183}$$

$$T_k \geq \left(\frac{C_k^{'}\epsilon(1-\gamma)^2}{6l_k\phi_{\nu,\mu}\gamma}\right)^{-2} L_k^2||u_k^*||_2^2 - 1 \tag{184}$$

$$\alpha_k = \frac{||u_k^*||_2}{L_k\sqrt{T_k+1}} \quad \forall k \in \{1, \cdots, K\} \tag{185}$$

and if $\epsilon < (C_k^{'})$ for all $k \in (1, \cdots, K)$,

We have

$$E(Q^* - Q^{\pi_K})_\mu \leq \epsilon + \frac{2\gamma}{(1-\gamma)^2}(\sqrt{\epsilon_{bias}} + \epsilon_{\tilde{D}}) \tag{186}$$

□

E. Proof of Supporting Lemmas

E.1. Proof Of Lemma 1

Proof. Using Assumption 2 and the definition of Q_{k1} for some iteration k of Algorithm 1 we have

$$\mathbb{E}(TQ_{k-1} - Q_{k1})_\nu^2 \leq \epsilon_{bias} \tag{187}$$

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

$$\mathbb{E}(|TQ_{k-1} - Q_{k1}|)_\nu^2 \leq \epsilon_{bias} \tag{188}$$

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 $|TQ_{k-1} - Q_{k1}|$ we get

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

$$\mathbb{E}(|TQ_{k-1} - Q_{k1}|)_\nu = \sqrt{\mathbb{E}(|TQ_{k-1} - Q_{k1}|)_\nu^2 - Var(|TQ_{k-1} - Q_{k1}|)_\nu} \tag{189}$$

Therefore we get

$$\mathbb{E}(|TQ_{k-1} - Q_{k1}|)_\nu \leq \sqrt{\epsilon_{bias}} \tag{190}$$

Since $\epsilon_{k_1} = TQ_{k-1} - Q_{k1}$ we have

$$\mathbb{E}(|\epsilon_{k_1}|)_\nu \leq \sqrt{\epsilon_{bias}} \tag{191}$$

■ □

E.2. Proof Of Lemma 2

Proof. From Lemma 6, we have

$$\arg\min_{f_\theta} \mathbb{E}_{x,y}\left(f_\theta(x) - g(x,y)\right)^2 = \arg\min_{f_\theta}\left(\mathbb{E}_{x,y}\left(f_\theta(x) - \mathbb{E}(g(y^{'},x)|x)\right)^2\right) \tag{192}$$

The function $f_\theta(x)$ to be $Q_\theta(s,a)$ and $g(x,y)$ to be the function $r^{'}(s,a) + \max_{a' \in \mathcal{A}} \gamma Q_{k-1}(s^{'}, a^{'})$.

We also have y as the two dimensional random variable $(r^{'}(s,a), s^{'})$. We now have $(s,a) \sim \nu$ and $s^{'}|(s,a) \sim P(.|(s,a))$ and $r^{'}(s,a) \sim R(.|s,a)$.

Then the loss function in (59) becomes

$$\mathbb{E}_{(s,a)\sim\nu, s'\sim P(s'|s,a), r(s,a)\sim\mathcal{R}(.|s,a)}(Q_\theta(s,a) - (r^{'}(s,a) + \max_{a'} \gamma Q_{k-1}(s^{'}, a^{'})))^2 \tag{193}$$

Therefore by Lemma 6, we have that the function $Q_\theta(s,a)$ which minimizes Equation (193) it will be minimizing

$$\mathbb{E}_{(s,a)\sim\nu}(Q_\theta(s,a) - \mathbb{E}_{s'\sim P(s'|s,a), r\sim\mathcal{R}(.|s,a))}(r^{'}(s,a) + \max_{a'} \gamma Q_{k-1}(s^{'}, a^{'})|s,a))^2 \tag{194}$$

But we have from Equation (4) that

$$\mathbb{E}_{s'\sim P(s'|s,a), r\sim R(.|s,a))}(r^{'}(s,a) + \max_{a'} \gamma Q_{k-1}(s^{'}, a^{'})|s,a) = TQ_{k-1} \tag{195}$$

Combining Equation (193) and (195) we get

$$\arg\min_{Q_\theta} \mathbb{E}_{(s,a)\sim\nu, s'\sim P(s'|s,a), r\sim R(.|s,a))}(Q_\theta(s,a) - (r(s,a) + \max_{a'} \gamma Q_{k-1}(s^{'}, a^{'})))^2 = \arg\min_{Q_\theta} \mathbb{E}_{(s,a)\sim\nu}(Q_\theta(s,a) - TQ_{k-1})^2 \tag{196}$$

The left hand side of Equation (196) is Q_{k_2} as defined in Definition 2 and the right hand side is Q_{k_1} as defined in Definition 1, which gives us

$$Q_{k_2} = Q_{k_1} \tag{197}$$

□

E.3. Proof Of Lemma 3

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

$$R_{X,Q_{k-1}}(\theta) = \frac{1}{|X|} \sum_{(s_i,a_i)\in X} \left(Q_\theta(s_i,a_i) - \left(r^{'}(s_i,a_i) + \gamma \max_{a'\in\mathcal{A}} Q_{k-1}(s_i^{'}, a^{'}) \right) \right)^2,$$

Here, $X = \{s_i, a_i\}_{i=\{1,\cdots,|X|\}}$, where s_i, a_i are drawn from the Markov chain whose stationary state action distribution is $(s_i,a_i) \sim \zeta_\nu^{\pi_k}$, $r(s_i,a_i) \sim R(.|s_i,a_i)$ and $s_i^{'} \sim P(.|s_i,a_i)$. $\theta \in \Theta$, Q_θ is as defined in Equation (14) and Q_{k-1} is the estimate of the Q function obtained at iteration $k-1$ of Algorithm 1.

We also define the term

$$L_{Q_{k-1}}(\theta) \triangleq \mathbb{E}_{(s,a)\sim\zeta_\nu^{\pi_k}, s'\sim P(s'|s,a), r'(\cdot|s,a)\sim R(\cdot|s,a)}(Q_\theta(s,a) - (r'(s,a) + \gamma \max_{a'} Q_{k-1}(s',a'))^2$$

We denote by $\theta_{k_2}, \theta_{k_3}$ the parameters of the neural networks Q_{k_2}, Q_{k_3} respectively. Q_{k_2}, Q_{k_3} are defined in Definition 2 and 3 respectively.

We then obtain,

$$\begin{aligned} R_{X,Q_{k-1}}(\theta_{k_2}) - R_{X,Q_{k-1}}(\theta_{k_3}) &\leq R_{X,Q_{k-1}}(\theta_{k_2}) - R_{X,Q_{k-1}}(\theta_{k_3}) + L_{Q_{k-1}}(\theta_{k_3}) - L_{Q_{k-1}}(\theta_{k_2}) && (198)\\ &= R_{X,Q_{k-1}}(\theta_{k_2}) - L_{Q_{k-1}}(\theta_{k_2}) + L_{Q_{k-1}}(\theta_{k_3}) - R_{X,Q_{k-1}}(\theta_{k_3}) && (199)\\ &\leq \underbrace{|R_{X,Q_{k-1}}(\theta_{k_2}) - L_{Q_{k-1}}(\theta_{k_2})|}_{I} + \underbrace{|R_{X,Q_{k-1}}(\theta_{k_3}) - L_{Q_{k-1}}(\theta_{k_3})|}_{II} && (200) \end{aligned}$$

We get the inequality in Equation (198) because $L_{Q_{k-1}}(\theta_{k_3}) - L_{Q_{k-1}}(\theta_{k_2}) > 0$ as Q_{k_2} is the minimizer of the loss function $L_{Q_{k-1}}(Q_\theta)$.

We take the Expectation with respect to $(s_0, a_0) \sim \zeta_\nu^{\pi_k}$, $r(s_0, a_0) \sim R(.|s_0, a_0), s_1 \sim P(.|s_0, a_0)$ on both sides of Equation (200). Note that thus the expectation is only with respect s_0, a_0, s_1, r_0.

$$\mathbb{E}(R_{X,Q_{k-1}}(\theta_{k_2}) - R_{X,Q_{k-1}}(\theta_{k_3})) \leq \mathbb{E}|R_{X,Q_{k-1}}(\theta_{k_2}) - L_{Q_{k-1}}(\theta_{k_2})| + \mathbb{E}|L_{Q_{k-1}}(\theta_{k_3}) - R_{X,Q_{k-1}}(\theta_{k_3})| \quad (201)$$

Now for notational convenience, we define for a fixed i consider the term $f_i(Q)$ as.

$$\begin{aligned} f(s_i, a_i, s_{i+1}, r_i) &= (Q_{k_2}(s_i, a_i) - (r(s_i, a_i) + \mathbb{E}_{a' \sim \pi^{\lambda_k}} Q^2_{k-1}(s_{i+1}, a'))))^2 \\ &\quad -(Q_{k_3}(s_i, a_i) - (r(s_i, a_i) + \mathbb{E}_{a' \sim \pi^{\lambda_k}} Q^3_{k-1}(s_{i+1}, a'))))^2 && (202) \end{aligned}$$

We can write the right hand side of Equation (200) as

$$R_{X,Q_{k-1}}(\theta_{k_2}) - R_{X,Q_{k-1}}(\theta_{k_3}) = \frac{1}{n_k} \sum_i^{n_k} f(s_i, a_i, s_{i+1}, r_i) \quad (203)$$

For a fixed i, consider the term

$$\left| \int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i, a_i, s_{i+1}, r_i) d\sigma_i - \int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i, a_i, s_{i+1}, r_i) d\sigma_1 \right| \quad (204)$$

Here, σ_i is the measure on s_i, a_i, s_{i+1}, r_i at the ith step of the Markov chain induced by the policy π_k and σ_1 is the measure corresponding to $(s_0, a_0) \sim \zeta_\nu^{\pi_k}$, $r(s_1, a_1) \sim R(.|s_1, a_1), s_1 \sim P(.|s_1, a_1)$. We then consider the term

$$\begin{aligned} &\left| \int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i, a_i, s_{i+1}, r_i) d\sigma_i - \int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i, a_i, s_{i+1}, r_i) d\sigma_1 \right| \\ &\qquad \leq \left| \sup_{s,a,s',r} f(s, a, s', r) \right| \int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} |\Sigma_i - \Sigma_1| d\sigma && (205) \end{aligned}$$

Here, $\Sigma_i = \frac{d\sigma_i}{d\sigma}$ and $\Sigma_1 = \frac{d\sigma_1}{d\sigma}$ and σ is a measure on the space $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \mathbb{R}$. We have from the definition of total variation between two probability measures and Assumption 3 that

$$\int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} |\Sigma_i - \Sigma_1| d\sigma = d_{TV}(\sigma_i, \sigma_1) \leq m\rho^i \quad (206)$$

Note that Assumption 3 is formally on the Markov chain s_i, a_i, however, we extend the assumption to the Markov chain $\{s_i, a_i, s_{i+1}, r_i\}_{i=\{1,\cdots,n_k\}}$ as $s_{i+1} \sim \int P^{'}(s_i, a_i) dP(.|s_i, a_i)$, where $P^{'}$ denotes the distribution of s_i, a_i. Therefore the total variation between the distribution of s_{i+1} in the Markov chain and the corresponding stationary state distribution is a function of the difference in corresponding distributions of s_i, a_i which are themselves bounded by Assumption 3. A similar argument can be applied to r_i. Also for $i \geq 2$, $f(s_i, a_i, s_{i+1}, r_i)$ is constant with respect to σ_1 but the above equation holds regardless.

Now plugging Equation (206) in Equation (205) and noting that $\sup_{s,a,s^{'},r} f(s, a, s^{'}, r) \leq \left(\frac{R_{max}}{1-\gamma}\right)^2$ we obtain

$$\left|\int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i, a_i, s_{i+1}, r_i) d\sigma_i - \int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i, a_i, s_{i+1}, r_i) d\sigma_1\right| \leq \left(\frac{R_{max}}{1-\gamma}\right)^2 m\rho^i \tag{207}$$

Which implies that

$$\int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i, a_i, s_{i+1}, r_i) d\sigma_1 \geq \int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i, a_i, s_{i+1}, r_i) d\sigma_i + \left(\frac{R_{max}}{1-\gamma}\right)^2 m\rho^i \tag{208}$$

Now consider the term

$$\left|\int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i, a_i, s_{i+1}, r_i) d\sigma_i - \int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i{}', a_i{}', s_i{}'', r_i{}') d\sigma_i^{'}\right| \tag{209}$$

Here σ_i is the same measure as defined for Equation (204) and $\sigma_i^{'}$ is the measure corresponding to $(s_i{}', a_i{}') \sim \zeta_\nu^{\pi_k}$, $r_i{}' \sim R(.|s_i{}', a_i{}'), s_i{}'' \sim P(.|s_i{}', a_i{}')$. Similar to Equation (207) we have

$$\int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i, a_i, s_{i+1}, r_i) d\sigma_i \geq \int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i{}', a_i{}', s_i{}'', r_i{}') d\sigma_i^{'} - \left(\frac{R_{max}}{1-\gamma}\right)^2 m\rho^i \tag{210}$$

Combining Equation (207) and Equation (210) we get

$$\int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i, a_i, s_{i+1}, r_i) d\sigma_1 \geq \int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i{}', a_i{}', s_i{}'', r_i{}') d\sigma_i^{'} - 2\left(\frac{R_{max}}{1-\gamma}\right)^2 m\rho^i \tag{211}$$

Now summing over i and dividing by n_k we get.

$$\begin{aligned}\frac{1}{n_k}\sum_i^{n_k}\int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i, a_i, s_{i+1}, r_i) d\sigma_1 &\geq \frac{1}{n_k}\sum_i^{n_k}\int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i{}', a_i{}', s_i{}'', r_i{}') d\sigma_i^{'} - \frac{2}{n_k}\sum_i^{n_k}\left(\frac{R_{max}}{1-\gamma}\right)^2 m\rho^i \\ &\geq \frac{1}{n_k}\sum_i^{n_k}\int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f(s_i{}', a_i{}', s_i{}'', r_i{}') d\sigma_i^{'} - \frac{2}{n_k}\left(\frac{R_{max}}{1-\gamma}\right)^2\left(\frac{m}{1-\rho}\right)\end{aligned} \tag{212}$$

Note that the left hand side of Equation (212) is $\mathbb{E}(R_{X,Q_{k-1}}(\theta_{k_2}) - R_{X,Q_{k-1}}(\theta_{k_3}))$, where the expectation is with respect to $(s_0, a_0) \sim \zeta_\nu^{\pi_k}$, $r(s_i, a_i) \sim R(.|s_i, a_i)$. Thus, we have

$$\mathbb{E}(R_{X,Q_{k-1}}(\theta_{k_2}) - R_{X,Q_{k-1}}(\theta_{k_3})) \quad \geq \quad \frac{1}{n_k}\sum_{i}^{n_k}\int_{\mathcal{S}\times\mathcal{A}\times\mathcal{S}\times\mathbb{R}} f({s_i}', {a_i}', {s_i}'', {r_i}')d\sigma_i^{'} - \frac{2}{n_k}\left(\frac{R_{max}}{1-\gamma}\right)^2\left(\frac{m}{1-\rho}\right) \tag{213}$$

Now, from Theorem 5 of (Bertail and Portier, 2019) we have that

$$\mathbb{E}|\sup_{\theta\in\Theta}(R_{X,Q_{k-1}}(\theta_{k_2})) - L_{Q_{k-1}}(\theta_{k_2})| \leq \frac{1}{n_k}\left(4\left(Rad(\mathcal{L}\circ z)\right) + 4\left(\sup_{\theta\in\Theta} L_{Q_{k-1}}(\theta)\right)\sqrt{n_k}C_k^1 + C_k^2\right) \tag{214}$$

where $z = (s, a, s^{'}, r)$, $\mathcal{L} = \{z \in Z \rightarrow l(\theta, z), \theta \in \Theta\}$, l is the function given by $\{l : \mathcal{Z} \rightarrow \mathbb{R} : l = (Q_\theta(s,a) - (r'(s,a) + \gamma \max_{a'} Q_{k-1}(s', a'))^2\}$, and $n_k = |X|$. C_k^1, C_k^2 are constants that depends on the Markov chain $(s_i, a_i, s_{i+1}, r_i)_{i\in\{1,\cdots,n_k\}}$ and the expectation is with respect to $(s,a) \sim \nu, s' \sim P(s'|s,a), r'(\cdot|s,a) \sim R(\cdot|s,a)$ or the initial distribution of the Markov chain.

Now from Proposition 11 of (Bertail and Portier, 2019) we have

$$Rad(\mathcal{L}\circ z) \leq C\left(\frac{R_{max}}{1-\gamma}\right)^2 \log(\log(n_k))\sqrt{n_k} \tag{215}$$

where C is a universal constant. Plugging Equation (215) into Equation (214) we have

$$\mathbb{E}|\sup_{\theta\in\Theta}(R_{X,Q_{k-1}}(\theta)) - L_{Q_{k-1}}(\theta)| \leq \mathcal{O}\left(\frac{\log\log(n_k)}{\sqrt{n_k}(1-\gamma)^2}\right) \tag{216}$$

Here, the expectation is with respect to $(s,a) \sim \nu, s' \sim P(s'|s,a), r'(\cdot|s,a) \sim R(\cdot|s,a)$ or the initial distribution of the Markov chain. This implies that

$$\begin{aligned}
&\mathbb{E}_{(s,a)\sim\zeta_\nu^{\pi_k}, s'\sim P(s'|s,a), r'(\cdot|s,a)\sim R(\cdot|s,a)}|\sup_{\theta\in\Theta}(R_{X,Q_{k-1}}(\theta)) - L_{Q_{k-1}}(\theta)| \\
&\leq \phi_k \mathbb{E}_{(s,a)\sim\nu, s'\sim P(s'|s,a), r'(\cdot|s,a)\sim R(\cdot|s,a)}|\sup_{\theta\in\Theta}(R_{X,Q_{k-1}}(\theta)) - L_{Q_{k-1}}(\theta)| \\
&\leq \mathcal{O}\left(\frac{\log\log(n_k)}{\sqrt{n_k}(1-\gamma)^2}\right)
\end{aligned} \tag{217}$$

where $\phi_k = \left|\left|\frac{d\nu}{d\zeta_\nu^{\pi_k}}\right|\right|_\infty$. Thus we get

$$\mathbb{E}|R_{X,Q_{k-1}}(\theta_{k_2}) - L_{Q_{k-1}}(\theta_{k_2})| \quad \leq \quad \mathcal{O}\left(\frac{\log\log(n_k)}{\sqrt{n_k}(1-\gamma)^2}\right) \tag{218}$$

$$\mathbb{E}|L_{Q_{k-1}}(\theta_{k_3}) - R_{X,Q_{k-1}}(\theta_{k_3})| \quad \leq \quad \mathcal{O}\left(\frac{\log\log(n_k)}{\sqrt{n_k}(1-\gamma)^2}\right) \tag{219}$$

where the expectation is with respect to $(s,a) \sim \zeta_\nu^{\pi_k}, s' \sim P(s'|s,a), r'(\cdot|s,a) \sim R(\cdot|s,a)$.

Now plugging Equation (218), (219) into the right hand side of Equation (201), we obtain

$$\mathbb{E}\left(R_{X,Q_{k-1}}(\theta_{k_2}) - R_{X,Q_{k-1}}(\theta_{k_3})\right) \quad \leq \quad \mathcal{O}\left(\frac{\log\log(n_k)}{\sqrt{n_k}(1-\gamma)^2}\right) \tag{220}$$

Further using Equation (213), we have

$$\frac{1}{n_k}\sum_{i=1}^{n_k}\mathbb{E}f({s_i}',{a_i}',{s_i}'',{r_i}') - 2\frac{1}{n_k}\left(\frac{R_{max}}{1-\gamma}\right)^2\left(\frac{m}{1-\rho}\right) \quad \leq \quad \mathcal{O}\left(\frac{\log\log(n_k)}{\sqrt{n_k}(1-\gamma)^2}\right) \tag{221}$$

which given us

$$\frac{1}{n_k}\sum_{i=1}^{n_k}\mathbb{E}f({s_i}',{a_i}',{s_i}'',{r_i}') \quad \leq \quad \mathcal{O}\left(\frac{\log\log(n_k)}{\sqrt{n_k}(1-\gamma)^2}\right) + \frac{2}{n_k}\left(\frac{R_{max}}{1-\gamma}\right)^2\left(\frac{m}{1-\rho}\right) \tag{222}$$

$$\leq \quad \mathcal{O}\left(\frac{\log\log(n_k)}{\sqrt{n_k}(1-\gamma)^2}\right) \tag{223}$$

Therefore, if $n_k \geq \tilde{\mathcal{O}}(\epsilon^{-2}(1-\gamma)^{-2})$, we have

$$\frac{1}{n_k}\sum_{i=1}^{n_k}\mathbb{E}f({s_i}',{a_i}',{s_i}'',{r_i}') \quad \leq \quad \epsilon \tag{224}$$

Noting that all terms inside the summation of the right hand side of Equation (224) are identical, we can re-write Equation (224) to get

$$\mathbb{E}\underbrace{(Q_{k_2}(s,a) - Q_{k_3}(s,a))}_{A1}\underbrace{(Q_{k_2}(s,a) + Q_{k_3}(s,a) - 2(r(s,a)) + \gamma\max_{a\in\mathcal{A}}Q_{k-1}(s^{'},a))}_{A2} \quad \leq \quad \epsilon \tag{225}$$

where the expectation is now over a single $(s,a) \sim \zeta_{\nu}^{\pi_k}, r(s,a) \sim R(.|s,a)$ and $s^{'} \sim P(.|s,a)$. We re-write Equation (225) as

$$\int\underbrace{(Q_{k_2}(s,a) - Q_{k_3}(s,a))}_{A1}\underbrace{(Q_{k_2}(s,a) + Q_{k_3}(s,a) - 2(r(s,a)) + \gamma\max_{a\in\mathcal{A}}Q_{k-1}(s^{'},a))}_{A2}d\mu_1(s,a)d\mu_2(r)d\mu_3(s^{'}) \quad \leq \quad \epsilon \tag{226}$$

where μ_1, μ_2, μ_3 are the measures with respect to (s,a), $r^{'}$ and $s^{'}$ respectively

Now for the integral in Equation (226) we split the integral into four different integrals. Each integral is over the set of $(s,a), r^{'}, s^{'}$ corresponding to the 4 different combinations of signs of $A1, A2$.

$$\int_{\{(s,a),r',s'\}:A1\geq 0,A2\geq 0}(A1)(A2)d\nu(s,a)d\mu_2(r)d\nu_3(s^{'}) + \int_{\{(s,a),r',s'\}:A1<0,A2<0}(A1)(A2)d\mu_1(s,a)d\mu_2(r)d\mu_3(s^{'}) + \int_{\{(s,a),r',s'\}:A1\geq 0,A2<0}(A1)(A2)d\mu_1(s,a)d\mu_2(r)d\nu_3(s^{'}) + \int_{\{(s,a),r',s'\}:A1<0,A2\geq 0}(A1)(A2)d\mu_1(s,a)d\mu_2(r)d\mu_3(s^{'}) \leq \epsilon \tag{227}$$

Now note that the first 2 terms are non-negative and the last two terms are non-positive. We then write the first two terms as

$$\int_{\{(s,a),r',s'\}:A1\geq 0,A2\geq 0}(A1)(A2)d(s,a)d\mu_1(s,a)d\mu_2(r)d\mu_3(s^{'}) = C_{k_1}\int |Q_{k_2}-Q_{k_3}|d\mu_1(s,a) \\ = C_{k_1}\mathbb{E}(|Q_{k_2}-Q_{k_3}|)_{\zeta_\nu^{\pi_k}} \tag{228}$$

$$\int_{\{(s,a),r',s'\}:A1<0,A2<0}(A1)(A2)d(s,a)d\mu_1(s,a)d\mu_2(r)d\mu_3(s^{'}) = C_{k_2}\int |Q_{k_2}-Q_{k_3}|d\mu_1(s,a) \\ = C_{k_2}\mathbb{E}(|Q_{k_2}-Q_{k_3}|)_{\zeta_\nu^{\pi_k}} \tag{229}$$

We write the last two terms as

$$\int_{\{(s,a),r',s'\}:A1\geq 0,A2<0}(A1)(A2)d\mu_1(s,a)d\mu_2(r)d\mu_3(s^{'}) = C_{k_3}\epsilon \tag{230}$$

$$\int_{\{(s,a),r',s'\}:A1<0,A2\geq 0}(A1)(A2)d\nu(s,a)d\mu_2(r)d\mu_3(s^{'}) = C_{k_4}\epsilon \tag{231}$$

Here $C_{k_1}, C_{k_2}, C_{k_4}$ and C_{k_4} are positive constants. Plugging Equations (228), (229), (230), (231) into Equation (225).

$$(C_{k_1}+C_{k_2})\mathbb{E}(|Q_{k_2}-Q_{k_3}|)_{\zeta_\nu^{\pi_k}} - (C_{k_3}+C_{k_4})\epsilon \quad \leq \quad \epsilon \tag{232}$$

$$\tag{233}$$

which implies

$$\mathbb{E}(|Q_{k_2}-Q_{k_3}|)_{\zeta_\nu^{\pi_k}} \quad \leq \quad \left(\frac{1+C_{k_3}+C_{k_4}}{C_{k_1}+C_{k_2}}\right)\epsilon \tag{234}$$

$$\tag{235}$$

Now define $\left(\frac{1+C_{k_3}+C_{k_4}}{C_{k_1}+C_{k_2}}\right) = C_k$ to get

$$\mathbb{E}(|Q_{k_2}-Q_{k_3}|)_{\zeta_\nu^{\pi_k}} \quad \leq \quad C_k\epsilon \tag{236}$$

$$\tag{237}$$

Thus we have that if $n_k \geq \tilde{\mathcal{O}}(\epsilon^{-2}(1-\gamma)^{-2})$ then we have

$$\mathbb{E}(|Q_{k_2}-Q_{k_3}|)_{\zeta_\nu^{\pi_k}} \quad \leq \quad \epsilon \tag{238}$$

$$\tag{239}$$

□

E.4. Proof Of Lemma 4

Proof. For a given iteration k of Algorithm 1 the optimization problem to be solved in Algorithm 2 is the following

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(Q_\theta(s_i,a_i) - \left(r(s_i,a_i) + \gamma \max_{a^{'}\in\mathcal{A}} \gamma Q_{k-1}(s^{'},a^{'})\right)\right)^2 \tag{240}$$

Here, Q_{k-1} is the estimate of the Q function from the iteration $k-1$ and the state action pairs $(s_i,a_i)_{i=\{1,\cdots,n\}}$ have been sampled from a distribution over the state action pairs denoted by ν. Since $\min_\theta \mathcal{L}(\theta)$ is a non convex optimization problem we instead solve the equivalent convex problem given by

$$u_k^* = \arg\min_u g_k(u) = \arg\min_u ||\sum_{D_i\in\tilde{D}} D_i X_k u_i - y_k||_2^2 \tag{241}$$

$$\textit{subject to} |u|_1 \leq \frac{R_{\max}}{1-\gamma} \tag{242}$$

Here, $X_k \in \mathbb{R}^{n_k\times d}$ is the matrix of sampled state action pairs at iteration k, $y_k \in \mathbb{R}^{n_k\times 1}$ is the vector of target values at iteration k. $\tilde{D}$ is the set of diagonal matrices obtained from line 2 of Algorithm 2 and $u \in \mathbb{R}^{|\tilde{D}d|\times 1}$ (Note that we are treating u as a vector here for notational convenience instead of a matrix as was done in Section 4).

The constraint in Equation (242) ensures that the all the co-ordinates of the vector $\sum_{D_i\in\tilde{D}} D_i X_k u_i$ are upper bounded by $\frac{R_{max}}{1-\gamma}$ (since all elements of X_k are between 0 and 1). This ensures that the corresponding neural network represented by Equation (14) is also upper bounded by $\frac{R_{max}}{1-\gamma}$. We use the a projected gradient descent to solve the constrained convex optimization problem which can be written as.

$$u_k^* = \arg\min_{u:|u|_1\leq\frac{R_{\max}}{1-\gamma}} g_k(u) = \arg\min_{u:|u|_1\leq\frac{R_{\max}}{1-\gamma}} ||\sum_{D_i\in\tilde{D}} D_i X_k u_i - y_k||_2^2 \tag{243}$$

From Ang, Andersen(2017). "Continuous Optimization" [Notes]. https://angms.science/doc/CVX we have that if the step size $\alpha_k = \frac{||u_k^*||_2}{L_k\sqrt{T_k+1}}$, after T_k iterations of the projected gradient descent algorithm we obtain

$$(g_k(u_{T_k}) - g_k(u^*)) \leq L_k \frac{||u_k^*||_2}{\sqrt{T_k+1}} \tag{244}$$

Where L_k is the lipschitz constant of $g_k(u)$ and u_{T_k} is the parameter estimate at step T_k.

Therefore if the number of iteration of the projected gradient descent algorithm T_k and the step-size α satisfy

$$T_k \geq L_k^2 ||u_k^*||_2^2 \epsilon^{-2} - 1, \tag{245}$$

$$\alpha_k = \frac{||u_k^*||_2}{L_k\sqrt{T_k+1}}, \tag{246}$$

we have

$$(g_k(u_{T_k}) - g_k(u^*)) \leq \epsilon \tag{247}$$

Let $(v_i^*, w_i^*)_{i\in(1,\cdots,|\tilde{D}|)}$, $(v_i^{T_k}, w_i^{T_k})_{i\in(1,\cdots,|\tilde{D}|)}$ be defined as

$$(v_i^*, w_i^*)_{i\in(1,\cdots,|\tilde{D}|)} = \psi_i^{'}(u_i^*)_{i\in(1,\cdots,|\tilde{D}|)} \tag{248}$$

$$(v_i^{T_k}, w_i^{T_k})_{i\in(1,\cdots,|\tilde{D}|)} = \psi_i^{'}(u_i^{T_k})_{i\in(1,\cdots,|\tilde{D}|)} \tag{249}$$

where $\psi^{'}$ is defined in Equation (51).

Further, we define $\theta^*_{|\tilde{D}|}$ and θ^{T_k} as

$$\theta^*_{|\tilde{D}|} = \psi(v_i^*, w_i^*)_{i\in(1,\cdots,|\tilde{D}|)} \tag{250}$$

$$\theta^{T_k} = \psi(v_i^{T_k}, w_i^{T_k})_{i\in(1,\cdots,|\tilde{D}|)} \tag{251}$$

where ψ is defined in Equation (46), $\theta^*_{|\tilde{D}|} = \arg\min_\theta \mathcal{L}_{|\tilde{D}|}(\theta)$ for $\mathcal{L}_{|\tilde{D}|}(\theta)$ defined in Appendix B.

Since $(g(u_{T_k}) - g(u^*)) \leq \epsilon$, then by Lemma 7, we have

$$\mathcal{L}_{|\tilde{D}|}(\theta^{T_k}) - \mathcal{L}_{|\tilde{D}|}(\theta^*_{|\tilde{D}|}) \leq \epsilon \tag{252}$$

Note that $\mathcal{L}_{|\tilde{D}|}(\theta^{T_k}) - \mathcal{L}_{|\tilde{D}|}(\theta^*_{|\tilde{D}|})$ is a constant value. Thus we can always find constant $C_k^{'}$ such that

$$C_k^{'}|\theta^{T_k} - \theta^*_{|\tilde{D}|}|_1 \leq \mathcal{L}_{|\tilde{D}|}(\theta^{T_k}) - \mathcal{L}_{|\tilde{D}|}(\theta^*_{|\tilde{D}|}) \tag{253}$$

$$|\theta^{T_k} - \theta^*_{|\tilde{D}|}|_1 \leq \frac{\mathcal{L}(\theta^{T_k}) - \mathcal{L}(\theta^*)}{C_k^{'}} \tag{254}$$

Therefore if we have

$$T_k \geq L_k^2||u_k^*||_2^2\epsilon^{-2} - 1, \tag{255}$$

$$\alpha_k = \frac{||u_k^*||_2}{L_k\sqrt{T_k+1}}, \tag{256}$$

then we have

$$|\theta^{T_k} - \theta^*|_1 \leq \frac{\epsilon}{C_k^{'}} \tag{257}$$

which according to Equation (254) implies that

$$C_k^{'}|\theta^{T_k} - \theta^*_{|\tilde{D}|}|_1 \leq \mathcal{L}_{|\tilde{D}|}(\theta^{T_k}) - \mathcal{L}_{|\tilde{D}|}(\theta^*_{|\tilde{D}|}) \leq \epsilon \tag{258}$$

Dividing Equation (258) by $C_k^{'}$ we get

$$|\theta^{T_k} - \theta^*_{|\tilde{D}|}|_1 \leq \frac{\mathcal{L}_{|\tilde{D}|}(\theta^{T_k}) - \mathcal{L}_{|\tilde{D}|}(\theta^*_{|\tilde{D}|})}{C_k^{'}} \leq \frac{\epsilon}{C_k^{'}} \tag{259}$$

Which implies

$$|\theta^{T_k} - \theta^*_{|\tilde{D}|}|_1 \leq \frac{\epsilon}{C_k^{'}} \tag{260}$$

Assuming ϵ is small enough such that $\frac{\epsilon}{C_k^{'}} < 1$ from lemma 8, this implies that there exists an $l_k > 0$ such that

$$|Q_{\theta^{T_k}}(s,a) - Q_{\theta^*_{|\tilde{D}|}}(s,a)| \leq l_k|\theta^{T_k} - \theta^*_{|\tilde{D}|}|_1 \tag{261}$$

$$\leq \frac{l_k\epsilon}{C_k^{'}} \tag{262}$$

for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. Equation (262) implies that if

$$T_k \geq L_k^2||u_k^*||_2^2\epsilon^{-2} - 1, \tag{263}$$

$$\alpha = \frac{||u_k^*||_2}{L_k\sqrt{T_k+1}}, \tag{264}$$

then we have

$$\mathbb{E}(|Q_{\theta^{T_k}}(s,a) - Q_{\theta^*_{|\tilde{D}|}}(s,a)|)_\nu \leq \frac{l_k\epsilon}{C_k^{'}} \tag{265}$$

By definition in section D Q_k is our estimate of the Q function at the kth iteration of Algorithm 1 and thus we have $Q_{\theta^{T_k}} = Q_k$ which implies that

$$\mathbb{E}(|Q_k(s,a) - Q_{\theta^*_{\tilde{D}}}(s,a)|)_\nu \leq \frac{l_k\epsilon}{C_k^{'}} \tag{266}$$

If we replace ϵ by $\frac{C_k^{'}\epsilon}{l_k}$ in Equation (265), we get that if

$$T_k \geq \left(\frac{C_k^{'}\epsilon}{l_k}\right)^{-2} L_k^2||u_k^*||_2^2 - 1, \tag{267}$$

$$\alpha = \frac{||u_k^*||_2}{L_k\sqrt{T_k+1}}, \tag{268}$$

we have

$$\mathbb{E}(|Q_k(s,a) - Q_{\theta^*_{\tilde{D}}}(s,a)|)_\nu \leq \epsilon \tag{269}$$

From Assumption 1, we have that

$$\mathbb{E}(|Q_{\theta^*}(s,a) - Q_{\theta^*_{\tilde{D}}}(s,a)|)_\nu \leq \epsilon_{|\tilde{D}|} \tag{270}$$

where $\theta^* = \arg\min_{\theta\in\Theta} \mathcal{L}(\theta)$ and by definition of Q_{k_3} in Definition 6, we have that $Q_{k_3} = Q_{\theta^*}$. Therefore if we have

$$T_k \geq \left(\frac{C_k^{'}\epsilon}{l_k}\right)^{-2} L_k^2||u_k^*||_2^2 - 1, \tag{271}$$

$$\alpha = \frac{||u_k^*||_2}{L_k\sqrt{T_k+1}}, \tag{272}$$

we have

$$\begin{aligned}
\mathbb{E}(|Q_k(s,a) - Q_{k_3}(s,a)|)_\nu &\leq \mathbb{E}(|Q_k(s,a) - Q_{\theta^*_{\tilde{D}}}(s,a)|)_\nu + \mathbb{E}(|Q_{k_3}(s,a) - Q_{\theta^*_{\tilde{D}}}(s,a)|)_\nu && (273)\\
&\leq \epsilon + \epsilon_{|\tilde{D}|} && (274)
\end{aligned}$$

□