An Improved Finite-time Analysis of Temporal Difference Learning with Deep Neural Networks

Zhifa Ke [1] **Zaiwen Wen** [2] **Junyu Zhang** [3]

Abstract

Temporal difference (TD) learning algorithms with neural network function parameterization have well-established empirical success in many practical large-scale reinforcement learning tasks. However, theoretical understanding of these algorithms remains challenging due to the nonlinearity of the action-value approximation. In this paper, we develop an improved non-asymptotic analysis of the neural TD method with a general L-layer neural network. New proof techniques are developed and an improved new $\tilde{\mathcal{O}}(\epsilon^{-1})$ sample complexity is derived. To our best knowledge, this is the first finite-time analysis of neural TD that achieves an $\tilde{\mathcal{O}}(\epsilon^{-1})$ complexity under the Markovian sampling, as opposed to the best known $\tilde{\mathcal{O}}(\epsilon^{-2})$ complexity in the existing literature.

1. Introduction

The temporal difference (TD) learning method, firstly designed for policy evaluation (Sutton, 1988), is a fundamental building block of many popular Reinforcement Learning (RL) algorithms. In standard TD learning algorithms for tabular MDP, based on the Bellman operator, the agent iteratively obtains a state-action-reward-transition tuple and then updates the Q values by a weighted average of the current value and the TD target. Once the algorithm converges, the Q function is considered to be the final return obtained by executing the target policy given some initial action-state pair.

[1]Center for Data Science, Peking University, China. [2]Beijing International Center for Mathematical Research, Center for Machine Learning Research and Changsha Institute for Computing and Digital Economy, Beijing, China [3]Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore. Correspondence to: Zhifa Ke <kezhifa@stu.pku.edu.cn>, Zaiwen Wen <wenzw@pku.edu.cn>, Junyu Zhang <junyuz@nus.edu.sg>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

For large-scale reinforcement learning (RL) problems, appropriate parameterization of the Q function is crucial for better scalability of the TD algorithms. Common examples include linear (Tesauro et al., 1995), general smooth nonlinear (Maei et al., 2009), and neural network (Mnih et al., 2013) function approximations. However, it is well known that the naive extension of TD learning and Q-learning algorithms can diverge under the general function approximation Tsitsiklis & Van Roy (1996). To encourage convergence, numerous variants of TD and Q-learning have been proposed, including Least-squares TD (LSTD) (Bradtke & Barto, 1996; Boyan, 2002) and gradient TD (GTD) (Sutton et al., 2009a;b), to name a few.

The applications of neural network function approximation have witnessed huge empirical success in many real-world tasks, including Deep Q-network (DQN) algorithms (Mnih et al., 2013; Van Hasselt et al., 2016), policy improvement method (Sutton et al., 1999), trust region policy optimization (Schulman et al., 2015) and the actor-critic algorithms (Konda & Tsitsiklis, 1999; Lillicrap et al., 2015; Fujimoto et al., 2018), etc. However, due to the analysis difficulties brought by the function approximation, a significant gap exists between the empirical success and the theoretical understanding of these algorithms. Hence analyzing the convergence and sample complexity of TD learning and Q-learning under various Q function parameterizations has always been an active topic in the RL community during the past decades.

Early works focus on the asymptotic convergence of the algorithms with tabular or linear function approximation. For the tabular (stochastic) TD or Q learning method, Jaakkola et al. (1993) established the asymptotic convergence for the first time. Later on, the asymptotic convergence of algorithms with linear function approximation has been extensively discussed using ODE-based methods, see e.g. Tsitsiklis & Van Roy (1996); Perkins & Pendrith (2002); Borkar (2009). Meanwhile, in contrast to the convergent results for RL algorithms under the tabular or linear settings, TD with nonlinear function approximation is known to diverge in general Tsitsiklis & Van Roy (1996); Brandfonbrener & Bruna (2019). To overcome this issue, Maei et al. (2009) proposed to optimize the Mean Squared Projected Bellman

Error (MSPBE) via a gradient-based algorithm. Due to the problem nonconvexity, only asymptotic convergence to stationary points can be guaranteed.

More recently, benefiting from the improved techniques for analyzing stochastic optimization algorithms, there has been a growing number of research on providing finite-time analysis for TD and Q-learning algorithms with function approximations.

For linear function approximation, the non-asymptotic results of TD learning and its variants are relatively well-understood, including TD Bhandari et al. (2018); Dalal et al. (2018); Zou et al. (2019), gradient TD Dalal et al. (2018); Touati et al. (2018); Liu et al. (2020a), and Least-Squares TD Lazaric et al. (2010); Prashanth et al. (2014); Tagorti & Scherrer (2015), etc. In particular, Bhandari et al. (2018) established the first finite-time analysis of linear Q-learning under both i.i.d. sampling and Markovian sampling settings.

For neural network function approximation, which is directly related to this paper, we provide a more detailed discussion. Based on the recent advances in the understanding of optimizing ReLU network Jacot et al. (2018); Du et al. (2018); Allen-Zhu et al. (2019a;b); Cao & Gu (2019; 2020), a few recent works have successfully developed the finite-time analysis of the neural TD and neural Q-learning algorithms, as long as the Q network is sufficiently wide. Let Q^* be the true action-value function and let $Q(s,a;\boldsymbol{\theta})$ denote the action-value function parameterized by a neural network with weights $\boldsymbol{\theta}$, at any state action pair (s,a). Then we aim to find some ϵ-optimal parameter $\bar{\boldsymbol{\theta}}$ such that $\mathbb{E}\big[(Q(s,a;\bar{\boldsymbol{\theta}})-Q^*(s,a))^2\big] \leq \epsilon + \epsilon_{\mathcal{F}}$, where the expectation is taken over the possible randomness in the output $\bar{\theta}$ as well as the distribution over the state-action pairs (s,a), and $\epsilon_{\mathcal{F}}$ is the optimal approximation error of the parameterization function class. In (Xu & Gu, 2020), a neural Q-learning algorithm with a general L-layer ReLU network is analyzed, and an $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity is guaranteed given that the network is sufficiently wide. In (Cai et al., 2023), the authors studied both the neural TD learning and neural Q-learning algorithms for minimizing the MSPBE for policy evaluation and policy optimization, respectively. For policy evaluation, the Q^* in the definition of an ϵ-optimal solution is defaulted as Q^π with π being the policy to be evaluated. For both cases, an $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity is guaranteed for wide two-layer ReLU networks. In (Sun et al., 2022), an $\tilde{O}(\epsilon^{-\frac{2}{2-\alpha}})$ complexity has been achieved by an adaptive neural TD algorithm with multi-layer ReLU networks, where $\alpha \in (0,1]$ is a constant that characterizes the sparsity and decay rate of the stochastic semi-gradients. However, without additional assumption, only an $\tilde{O}(\epsilon^{-2})$ complexity with $\alpha = 1$ can be theoretically guaranteed. Finally, for policy evaluation problems, there are also several works that aim at reducing the width of the over-parameterized Q networks in the existing works (Tian et al., 2022; Cayci et al., 2023). In terms of complexity, both of them requires $\tilde{\mathcal{O}}(\epsilon^{-2})$ samples to obtain an ϵ-optimal solution.

Despite the fact that existing analysis of the neural TD or neural Q-learning algorithms merely provides the $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity under various settings, an $\tilde{\mathcal{O}}(\epsilon^{-1})$ sample complexity should be expected. In fact, a double-loop fitted Q-iteration (FQI) method (Fan et al., 2020) and its single-loop Gauss-Newton variant (Ke et al., 2023) can achieve an $\tilde{\mathcal{O}}(\epsilon^{-1})$ sample complexity is obtained for two-layer Q networks. Let $\mathcal{T}$ be the Bellman (optimality) operator, then the FQI method repeatedly solves a nonlinear least square subproblem to obtain the next iteration: $\boldsymbol{\theta}_{k+1} \approx \operatorname{argmin}_{\boldsymbol{\theta}\in\Theta} \mathbb{E}\big[(Q(s,a;\boldsymbol{\theta}) - \mathcal{T}Q(s,a;\boldsymbol{\theta}_k))^2\big]$. Compared to the single-loop neural TD or neural Q-learning method that takes only one sample (or a mini-batch) to update the weights of Q networks, the update scheme of FQI requires repeatedly solving a subproblem to sufficiently high accuracy to enable convergence, which makes it inefficient and less favorable in practice. Therefore, we would like to raise a question:

> *Can we improve the existing analysis of the neural temporal difference learning algorithm and obtain an $\tilde{\mathcal{O}}(\epsilon^{-1})$ sample complexity under general multi-layer Q neural networks?*

To answer this question, we revisit the convergence analysis of the neural TD learning or Q-learning algorithms under the non-i.i.d. Markovian observations where a general L-layer neural network is used for Q function parameterization. By proposing a new subspace analysis technique, under suitable conditions, we derive a brand new $\tilde{\mathcal{O}}(\epsilon^{-1})$ sample complexity for neural TD learning or Q-learning, improving the state-of-the-art $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity in the existing works. Our contributions are summarized as follows.

- Under the non-i.i.d. Markovian sampling setting, we derive an $\tilde{\mathcal{O}}(\epsilon^{-1})$ sample complexity for both neural TD learning and Q-learning methods under the multi-layer network approximation for Q functions. Our result also improves the best known $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity in the existing works.
- Based on our newly developed techniques, we further provide a finite-sample analysis for a minimax neural Q-learning algorithm that solves two-player zero-sum Markov games. An $\tilde{\mathcal{O}}(\epsilon^{-1})$ sample complexity is obtained under the non-i.i.d. Markovian sampling setting.

Technically, the subspace analysis approach that we propose to establish the $\tilde{O}(\epsilon^{-1})$ sample complexity is by itself of independent interest. We believe this technique can potentially

	Neural Approximation	Network Depth	Network Width	Activation	Sample Complexity
(Bhandari et al., 2018)	No	NA	NA	NA	$\mathcal{O}(1/\epsilon)$
(Cai et al., 2023)	Yes	2	$\Omega(1/\epsilon^4)$	ReLu	$\mathcal{O}(1/\epsilon^2)$
(Xu & Gu, 2020)	Yes	L	$\Omega(1/\epsilon^6)$	ReLu	$\mathcal{O}(1/\epsilon^2)$
(Sun et al., 2022)	Yes	L	$\Omega(1/\epsilon^6)$	ReLu	$\mathcal{O}(1/\epsilon^{\frac{2}{2-\alpha}}), \alpha \in (0,1]$
(Tian et al., 2022)	Yes	L	$\Omega(1/\epsilon^2)$	ELU, GeLU	$\mathcal{O}(1/\epsilon^2)$
Ours	Yes	L	$\Omega(1/\epsilon^2)$	ELU, GeLU	$\mathcal{O}(1/\epsilon)$

Table 1. Sample complexity for parameterized Q learning to find some $\bar{\boldsymbol{\theta}}$ such that $\mathbb{E}\left[\|Q(s,a;\bar{\boldsymbol{\theta}}) - Q^*(s,a)\|_\mu^2\right] \leq \varepsilon$, where $\|f\|_\mu^2 := \int |f|^2 d\mu$ and $Q^*(s,a)$ satisfies the Bellman optimality equation $Q^*(s,a) = \mathcal{T}Q^*(s,a)$.

be applied to linear Q-learning algorithms and linear Actor-Critic algorithms without requiring the positive definiteness assumption of the feature covariance matrix (Bhandari et al., 2018; Zou et al., 2019; Barakat et al., 2022), while maintaining the $\tilde{O}(\epsilon^{-1})$ complexity.

In summary, we provide a comprehensive comparison between our work and the most related works in their respective settings and sample complexity in Table 1. Our work establishes an optimal sample complexity analysis within a broader contextual framework.

2. Preliminaries

We consider the infinite-horizon discounted Markov decision process (MDP), which is denoted as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbb{P}, r, \gamma)$. We consider a general state space $\mathcal{S}$ and a finite action space $\mathcal{A}$. At any state $s \in \mathcal{S}$, if the agent takes an action $a \in \mathcal{A}$, it will receive a reward $r(s,a) \in [-R_{\max}, R_{\max}]$ and transition to the next state $s' \in \mathcal{S}$ with probability $\mathbb{P}(s'|s,a)$. We call r the reward function and $\mathbb{P}$ the transition kernel. Let $\gamma \in (0,1)$ be a discount factor, then an MDP aims to find a sequence of actions $\{a_t\}_{t\geq 0}$ to maximize the expected and discounted cumulative reward $\mathbb{E}\big[\sum_{t=0}^{\infty} \gamma^t \cdot r(s_t,a_t)|s_0 \sim \mu\big]$, where μ is the distribution of the initial state s_0.

Let $\Delta_{\mathcal{A}}$ denote the set of all probability distributions over the action space $\mathcal{A}$, and let a policy $\pi : \mathcal{S} \mapsto \Delta_{\mathcal{A}}$ be a mapping that returns a probability distribution $\pi(\cdot|s) \in \Delta_{\mathcal{A}}$ given any state $s \in \mathcal{S}$. If an agent follows a policy π, then at any state s_t, it will act by sampling an action $a_t \sim \pi(\cdot|s_t)$. Therefore, the action-value function (Q-function) under the policy π is

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

for $\forall (s,a) \in \mathcal{S}\times\mathcal{A}$, where all actions except a_0 are sampled according to π. For any mapping $Q : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$, let the Bellman operator $\mathcal{T}^\pi$ be

$$\mathcal{T}^\pi Q(s,a) := r(s,a) + \gamma \mathbb{E}\left[Q(s',a') \mid s' \sim \mathbb{P}(\cdot \mid s,a), \; a' \sim \pi(\cdot \mid s')\right], \; \forall s,a.$$

Then $\mathcal{T}^\pi$ is a γ-contraction under the infinity norm and Q^π is the unique solution to the fixed-point equation $Q = \mathcal{T}^\pi Q$ (Bertsekas, 2012). If the Q function is parameterized by some function $Q(s,a;\boldsymbol{\theta})$ to gain better scalability for large-scale RL problems, popular approaches for finding a good $\boldsymbol{\theta}$ include minimizing the the Mean-Squared Bellman Error (MSBE):

$$\min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{(s,a)\sim\mu}\Big[\left(Q(s,a;\boldsymbol{\theta}) - \mathcal{T}^\pi Q(s,a;\boldsymbol{\theta})\right)^2\Big], \quad (1)$$

and minimizing the Mean-Squared Projected Bellman Error (MSPBE):

$$\min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{(s,a)\sim\mu}\left[\left(Q(s,a;\boldsymbol{\theta}) - \Pi_{\mathcal{F}}\mathcal{T}^\pi Q(s,a;\boldsymbol{\theta})\right)^2\right], \quad (2)$$

where Θ is a feasible domain of the parameter $\boldsymbol{\theta}$, μ is some distribution over state action pairs, and $\Pi_{\mathcal{F}}$ is the projection onto some function class $\mathcal{F}$. Typical choices of $\mathcal{F}$ includes the Q function parameterization class itself $\mathcal{F} := \{Q(\cdot;\boldsymbol{\theta}) : \theta \in \Theta\}$ (Maei et al., 2009), and some local linearization of the parameterization function class (Cai et al., 2023).

In this paper, we study the neural temporal difference learning method where the action-value function is parameterized by some multi-layer neural network. Let us define a feed-forward neural network by the following recursion:

$$\boldsymbol{x}^{(l)} = \frac{1}{\sqrt{m}}\sigma\left(\boldsymbol{W}_l \boldsymbol{x}^{(l-1)}\right), \quad l \in \{1,2,\cdots,L\}, \quad (3)$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{m\times d}$, $\boldsymbol{W}_l \in \mathbb{R}^{m\times m}$ for $2 \leq l \leq L$ are the weight matrices of the network, $\sigma(\cdot)$ is an activation function, and the input is a feature map $\boldsymbol{x}^{(0)} = \phi(s,a) \in \mathbb{R}^d$ for any state action pair (s,a). For simplicity of notation, we write $\boldsymbol{x} = \boldsymbol{x}^{(0)}$, then $Q(s,a)$ is parameterized by

$$Q(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{\sqrt{m}}\boldsymbol{b}^\top \boldsymbol{x}^{(L)}, \quad (4)$$

where the parameter $\boldsymbol{\theta} = (\mathrm{Vec}(\boldsymbol{W}_1); \cdots; \mathrm{Vec}(\boldsymbol{W}_L))$ denotes the collection of all weight matrices, and $\boldsymbol{b}$ is given by a random initialization. $\mathrm{Vec}(\cdot)$ stands for the vetorization operator that reshapes a matrix to a column vector by stacking its columns one by one and the ";" separator in $\boldsymbol{\theta}$ stands

for the vertical stacking of the elements. That is, we reshape $\boldsymbol{\theta}$ to a long column vector for the notational convenience in later discussion.

Assumption 2.1. The activation function $\sigma(\cdot)$ is L_1-Lipschitz and L_2-smooth, i.e. , for $\forall y_1, y_2 \in \mathbb{R}$:

$$|\sigma(y_1) - \sigma(y_2)| \leq L_1|y_1 - y_2|$$

and

$$|\sigma'(y_1) - \sigma'(y_2)| \leq L_2|y_1 - y_2|.$$

Assumption 2.1 indicates that our results below are not based on the popular ReLU activation function. However, we primarily focus on some twice-differentiable activation functions (such as Sigmoid, ELU, GeLU, etc.), which are smooth approximations of the ReLU function and are frequently utilized in practical problems (Devlin et al., 2018; Godfrey, 2019). Such a setup aligns with (Liu et al., 2020b), and provides a $\mathcal{O}(m^{-\frac{1}{2}})$-smooth property for the neural Q-function.

Let $\boldsymbol{\theta}^0 = \left(\text{Vec}(\boldsymbol{W}_1^0); \cdots; \text{Vec}(\boldsymbol{W}_L^0)\right)$ be the initial solution. For each l, we initialize the weights of $\boldsymbol{W}_l^0$ elementwise from a normal distribution $\mathcal{N}(0,1)$ and each element of $\boldsymbol{b}$ is drawn uniformly from $\{-1,+1\}$. The parameter $\boldsymbol{b}$ will not be optimized during training. For regularity purpose, we would like to restrict the iterations to a bounded set around $\boldsymbol{\theta}^0$, which is defined as

$$S_\omega := \{\boldsymbol{\theta} = (\text{Vec}(\boldsymbol{W}_1); \cdots; \text{Vec}(\boldsymbol{W}_L)) : \|\boldsymbol{\theta} - \boldsymbol{\theta}^0\|_2 \leq \omega, 1 \leq l \leq L\}.$$

In each iteration t, the neural Q-learning algorithm obtains a sample of state-action-reward-transition tuple $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$ and computes the TD error by

$$\Delta_t = Q(\boldsymbol{x}_t; \boldsymbol{\theta}^t) - \left(r_t + \gamma Q(\boldsymbol{x}_{t+1}; \boldsymbol{\theta}^t)\right) \tag{5}$$

with $\boldsymbol{x}_t = \phi(s_t, a_t), \boldsymbol{x}_{t+1} = \phi(s_{t+1}, a_{t+1})$. Then a projected stochastic semi-gradient step is performed to update the weight matrices:

$$\boldsymbol{\theta}^{t+1} = \Pi_{S_\omega}\left(\boldsymbol{\theta}^t - \eta_t \boldsymbol{g}\left(\boldsymbol{\theta}^t\right)\right) \tag{6}$$

with

$$\boldsymbol{g}(\boldsymbol{\theta}^t) = \Delta_t \cdot \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^t).$$

We formally describe the neural TD learning method in Algorithm 1.

Algorithm 1 Neural Temporal Difference Learning with Markovian Sampling

Input: A learning policy π, a discount factor $\gamma \in (0,1)$, a sequence of learning rates $\{\eta_t\}_{t\geq 0}$, a maximum iteration number T, a projection radius $\omega > 0$, a Q network with architecture (4).
Initialization: Generate each entry of $\boldsymbol{W}_l^0$ independently from $\mathcal{N}(0,1)$, for $l = 1, 2, \cdots, L$, and each entry of $\boldsymbol{b}$ independently from Unif$\{-1,+1\}$. Generate $s_0 \sim \mu, a_0 \sim \pi(\cdot|s_0)$.
for $t = 0, 1, \cdots, T-1$ **do**
  Sample $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$ from the learning policy π with $a_{t+1} \sim \pi(\cdot|s_{t+1})$.
  Compute the TD error Δ_t by (5).
  Update $\boldsymbol{\theta}^{t+1}$ by the projected stochastic semi-gradient step (6).
end for
Output: $\boldsymbol{\theta}^T$.

One remark is that, under the non-i.i.d. Markovian sampling setting, the agent is only able to generate a trajectory of samples following some given learning policy π, which is very common in the offline RL (Wu et al., 2019; Levine et al., 2020; Kostrikov et al., 2021) where the data trajectories are generated by some learning policy.

In later sections, we will revisit the Algorithm 1 and design a novel subspace analysis technique for this method and achieve an improved sample complexity of $\tilde{\mathcal{O}}(\epsilon^{-1})$. Moreover, by replacing the TD error induced by the Bellman operator (5) with TD error induced by the Bellman optimality operator: $\Delta_t = Q(\boldsymbol{x}_t; \boldsymbol{\theta}^t) - \left(r_t + \gamma \max_{b\in\mathcal{A}} Q(s', b; \boldsymbol{\theta}^t)\right)$, Algorithm 1 can be reduced to the neural Q-learning method for finding optimal state-action value Q^*. Our analysis for neural TD learning can be extended to the neural Q-learning analogously and obtain the same $\tilde{\mathcal{O}}(\epsilon^{-1})$ sample complexity.

3. Convergence of Neural Temporal Difference Learning

3.1. Basic Settings and Assumptions

To analyze Algorithm 1, let us first define the local linearization function class of the multi-layer Q network (4) at the random initialization $\boldsymbol{\theta}^0$:

$$\mathcal{F}_{\omega,m} := \left\{\widehat{Q}(\cdot\,; \boldsymbol{\theta}) = Q(\cdot\,; \boldsymbol{\theta}^0) + \left\langle \nabla_{\boldsymbol{\theta}} Q(\cdot\,; \boldsymbol{\theta}^0), \boldsymbol{\theta} - \boldsymbol{\theta}^0 \right\rangle\right\} \tag{7}$$

for any $\boldsymbol{\theta} \in S_\omega$. Consider the MSPBE minimization problem:

$$\min_{\boldsymbol{\theta}\in S_\omega} \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\left(Q(\boldsymbol{x}; \boldsymbol{\theta}) - \Pi_{\mathcal{F}_{\omega,m}} \mathcal{T}^\pi Q(\boldsymbol{x}; \boldsymbol{\theta})\right)^2\right], \tag{8}$$

where μ is the initial state distribution, π is the learning policy, and $\mathbb{P}$ is the transition kernel, the expectation $\mathbb{E}_{\mu,\pi,\mathbb{P}}[\cdot]$ is taken over $s \sim \mu$, $a \sim \pi(\cdot|s)$, and $s' \sim \mathbb{P}(\cdot|s,a), a' \sim \pi(\cdot|s')$ in $\mathcal{T}^\pi$. Define the set Ξ_β as

$$\Xi_\beta := \{\boldsymbol{\theta} \in S_\beta : \widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}) = \Pi_{\mathcal{F}_{\omega,m}} \mathcal{T}^\pi \widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}), \quad \forall \boldsymbol{x} = \phi(s,a)\}. \tag{9}$$

Then the set Ξ_ω consists of the points $\boldsymbol{\theta}$ with which $\widehat{Q}(\cdot\,;\boldsymbol{\theta})$ forms a fixed point of the projected Bellman operator $\Pi_{\mathcal{F}_{\omega,m}} \mathcal{T}^\pi$ for the problem (8). By Section 4.1 in (Cai et al., 2023), the fixed point of $\Pi_{\mathcal{F}_{\omega,m}} \mathcal{T}^\pi$ is unique for $\boldsymbol{\theta} \in S_\omega$. Therefore, the following relationship holds

$$\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}) = \widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}'). \tag{10}$$

for $\forall \boldsymbol{x} = \phi(s,a),\ \forall (s,a) \in \mathcal{S}\times\mathcal{A},\ \forall \boldsymbol{\theta},\boldsymbol{\theta}' \in \Xi_\beta, \forall \beta \geq \omega$. Moreover, it is also shown that a point $\boldsymbol{\theta}^* \in \Xi_\omega$ if and only if it satisfies the stationarity condition:

$$\mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\widehat{\Delta}\left(\boldsymbol{x},\boldsymbol{x}';\boldsymbol{\theta}^*\right)\left\langle \nabla_{\boldsymbol{\theta}} \widehat{Q}\left(\boldsymbol{x};\boldsymbol{\theta}^*\right), \boldsymbol{\theta}-\boldsymbol{\theta}^*\right\rangle\right] \geq 0, \tag{11}$$

where $\widehat{Q}(\cdot\,;\boldsymbol{\theta}^*) \in \mathcal{F}_{\omega,m}$ is a local linearization provided by (7) and $\widehat{\Delta}$ is defined as

$$\widehat{\Delta}\left(\boldsymbol{x},\boldsymbol{x}';\boldsymbol{\theta}^*\right) = \widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*) - \left(r(s,a) + \gamma \widehat{Q}\left(\boldsymbol{x}';\boldsymbol{\theta}^*\right)\right).$$

Hence people may analyze the gap between $Q^\pi(\cdot)$ and $Q(\cdot,;\boldsymbol{\theta}^T)$ by first connecting it to $\widehat{Q}(\cdot\,;\boldsymbol{\theta}^*)$. Based on this, Cai et al. (2023) derived an $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity for the neural TD method. Now we define

$$\Sigma_\pi \quad = \quad \mathbb{E}_{\mu,\pi}\left[\nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x};\boldsymbol{\theta}^0) \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x};\boldsymbol{\theta}^0)^\top\right]. \tag{12}$$

It is worth noting that the matrix Σ_π only depends on π and $\boldsymbol{\theta}^0$. In the original assumption about (12), (Zou et al., 2019; Xu & Gu, 2020) in fact assumed positive definiteness ($\succ 0$) of Σ_π, which can be viewed as a generalized version of the positive definite feature covariance matrix assumption in the analysis of linear TD and linear Q-learning, see e.g. (Zou et al., 2019). However, in this paper we adopt the following weaker regularity assumption.

Assumption 3.1. Let $\overline{\sigma_{\min}}(\Sigma_\pi)$ denote the minimum non-zero singular value of the matrix Σ_π, then there exist constants $\lambda_0, m^* > 0$ such that $\overline{\sigma_{\min}}(\Sigma_\pi) \geq \lambda_0$ as long as the Q network width $m \geq m^*$.

For neural Q function approximation, a sufficient but not necessary condition for Assumption 3.1 can be obtained by exploiting the theory of over-parameterized neural networks. Roughly speaking, for a finite MDP with an L-layer ReLU Q network, if the feature map satisfies $\phi(s,a) \nparallel \phi(s',a')$ for $\forall (s,a) \neq (s',a')$, the results of (Jacot et al., 2018; Allen-Zhu et al., 2019a;b; Cao & Gu, 2019; 2020) suggest that there exist $\lambda', m^* > 0$ such that with high probability $\mathrm{Gram}(\boldsymbol{\theta}_0) \succ \lambda' \cdot \mathbf{I}$ for networks with width $m \geq m^*$. Here $\mathrm{Gram}(\boldsymbol{\theta}_0)$ stands for the Gram matrix of the network at the initialization $\boldsymbol{\theta}_0$. A lower bound on $\overline{\sigma_{\min}}(\Sigma_\pi)$ can then be constructed with λ', refer to Remark D.6 in Appendix D.

Finally, to facilitate the sample complexity analysis under the non-i.i.d. Markovian sampling setting, let us make the following assumption on the fast mixing rate of the MDP sample trajectories, which is widely adopted in the related analysis (Zou et al., 2019; Xu & Gu, 2020; Cai et al., 2023).

Assumption 3.2. We assume that the Markov chain $\{s_t\}_{t=0,1,\ldots}$ induced by the learning policy π and the transition kernel $\mathbb{P}$ is uniformly ergodic with its invariant measure $\mathbb{P}^\pi$. Furthermore, we assume that there are constants $\kappa > 0, \rho \in (0,1)$ such that

$$\sup_{s\in\mathcal{S}} d_{TV}\left(\mathbb{P}\left(s_t \in \cdot \mid s_0 = s\right), \mathbb{P}^\pi\right) \leq \kappa \rho^t$$

for all $t \geq 0$.

Without loss of generality, we also make the following technical assumption, which is not fundamental as opposed to Assumption 3.1, and 3.2.

Assumption 3.3. We assume the initial state distribution μ to be the stationary state distribution under policy π.

This assumption is in fact very natural. Concerning the stationarity of μ, it can always be guaranteed by abandoning the first $\tilde{\mathcal{O}}(t_{\mathrm{mix}})$ samples while Assumption 3.2 indicates that the mixing time $t_{\mathrm{mix}} = \tilde{\mathcal{O}}(1)$. This assumption guarantees that the operator $\mathcal{T}^\pi$ is γ-contractive w.r.t. $\|\cdot\|_\mu$ in policy evaluation. Similar assumptions are included in (Bhandari et al., 2018; Cai et al., 2023).

3.2. An Improved Complexity of Neural TD Learning

To derive the $\tilde{\mathcal{O}}(\epsilon^{-1})$ sample complexity, we rely on the following key observation on subspace decomposition, which is beyond the existing analysis framework.

Proposition 3.4. *Let $\mathcal{R}(\Sigma_\pi)$ and $\mathcal{K}(\Sigma_\pi)$ denote the range space and kernel space of the matrix Σ_π, respectively. Then for any parameter $\boldsymbol{\theta} \in S_\omega$, there exists $\boldsymbol{\theta}_*$ such that*

$$\boldsymbol{\theta}_* \in \Xi_{2\omega} \qquad \text{and} \qquad \boldsymbol{\theta} - \boldsymbol{\theta}_* \in \mathcal{R}(\Sigma_\pi),$$

which also implies that the projections of $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_*$ onto the subspace $\mathcal{K}(\Sigma_\pi)$ are identical.

Based on this argument, for the iteration sequence $\{\boldsymbol{\theta}^t\}_{t\geq 0}$ generated by Algorithm 1, there exists a sequence $\{\boldsymbol{\theta}_*^t\}_{t\geq 0} \subseteq \Xi_{2\omega}$ such that $\{\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\}_{t\geq 0} \subseteq \mathcal{R}(\Sigma_\pi)$. Therefore, unlike the existing works that analyze $\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^*\|^2$ for some $\boldsymbol{\theta}^* \in \Xi_\omega$, c.f. (Cai et al., 2023; Xu & Gu, 2020), we will prove a much faster convergence in $\|\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\|^2$. Combined with (10), this further indicates the improved sample

complexity in this paper. The proof of this proposition is presented as follows.

Proof. For the ease of discussion, let us denote the dimension of weight parameter $\boldsymbol{\theta}$ as n. Then we may denote $\Sigma_\pi \in \mathbb{R}^{n\times n}$ and $\boldsymbol{\theta} \in \mathbb{R}^n$. First of all let us fix an arbitrary $\bar{\boldsymbol{\theta}} \in \Xi_\omega$, then we may decompose it into two orthogonal components:

$$\bar{\boldsymbol{\theta}} = \bar{\boldsymbol{\theta}}_{\|} + \bar{\boldsymbol{\theta}}_{\perp} \quad \text{s.t.} \quad \bar{\boldsymbol{\theta}}_{\|} \in \mathcal{R}(\Sigma_\pi) \text{ and } \bar{\boldsymbol{\theta}}_{\perp} \in \mathcal{K}(\Sigma_\pi).$$

Similarly, we can decompose the currently considered vector $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta} = \boldsymbol{\theta}_{\|} + \boldsymbol{\theta}_{\perp} \quad \text{s.t.} \quad \boldsymbol{\theta}_{\|} \in \mathcal{R}(\Sigma_\pi) \text{ and } \boldsymbol{\theta}_{\perp} \in \mathcal{K}(\Sigma_\pi).$$

Note that having an arbitrary vector $\boldsymbol{v} \in \mathbb{R}^n$ in the kernel space of Σ_π means that $\Sigma_\pi \boldsymbol{v} = 0$, which further indicates that

$$\begin{aligned} 0 &= \boldsymbol{v}^\top \Sigma_\pi \boldsymbol{v} \\ &= \boldsymbol{v}^\top \mathbb{E}_{\mu,\pi}\left[\nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x};\boldsymbol{\theta}^0) \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x};\boldsymbol{\theta}^0)^\top\right] \boldsymbol{v} \\ &= \mathbb{E}_{\mu,\pi}\left[\left\langle \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x};\boldsymbol{\theta}^0), \boldsymbol{v}\right\rangle^2\right]. \end{aligned}$$

Therefore, under the measure $(s,a) \sim \mu \times \pi$, we have

$$\boldsymbol{v} \in \mathcal{K}(\Sigma_\pi) \quad \Longrightarrow \quad \left\langle \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x};\boldsymbol{\theta}^0), \boldsymbol{v}\right\rangle = 0 \quad a.s. \quad (13)$$

where $a.s.$ stands for almost surely. Therefore, define $\boldsymbol{\theta}_* = \bar{\boldsymbol{\theta}}_{\|} + \boldsymbol{\theta}_{\perp}$, we can check the stationarity condition (11) for $\boldsymbol{\theta}_*$ by establishing:

$$\begin{aligned} &\mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\widehat{\Delta}\left(\boldsymbol{x},\boldsymbol{x}';\boldsymbol{\theta}_*\right) \cdot \left\langle \nabla_{\boldsymbol{\theta}} \widehat{Q}\left(\boldsymbol{x};\boldsymbol{\theta}_*\right), \boldsymbol{\theta}' - \boldsymbol{\theta}_*\right\rangle\right] \quad (14) \\ &= \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\widehat{\Delta}\left(\boldsymbol{x},\boldsymbol{x}';\bar{\boldsymbol{\theta}}\right) \cdot \left\langle \nabla_{\boldsymbol{\theta}} \widehat{Q}\left(\boldsymbol{x};\boldsymbol{\theta}_0\right), \boldsymbol{\theta}' - \bar{\boldsymbol{\theta}}\right\rangle\right] \geq 0 \end{aligned}$$

The proof of (14) is lengthy and is thus moved to Appendix A.1 for succinctness. As a result we have $\boldsymbol{\theta}_* \in \Xi_{2\omega}$ and $\boldsymbol{\theta} - \boldsymbol{\theta}_* = \bar{\boldsymbol{\theta}}_{\|} \in \mathcal{K}(\Sigma_\pi)$. Note that for any $\boldsymbol{\theta} \in S_\omega$,

$$\begin{aligned} \|\boldsymbol{\theta}_* - \boldsymbol{\theta}^0\| &\leq \|\bar{\boldsymbol{\theta}}_{\|} - \boldsymbol{\theta}^0_{\|}\| + \|\boldsymbol{\theta}_{\perp} - \boldsymbol{\theta}^0_{\perp}\| \\ &\leq \|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}^0\| + \|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}^0\| \leq 2\omega, \end{aligned}$$

which completes the proof. □

Following basic linear algebra analysis, we also have the following proposition.

Proposition 3.5. *Under Assumption 3.1, suppose the adopted Q network is sufficiently wide so that $m \geq m^*$, then for any $\boldsymbol{\theta} \in \mathcal{R}(\Sigma_\pi)$, we have $\boldsymbol{\theta}^\top \Sigma_\pi \boldsymbol{\theta} \geq \lambda_0 \|\boldsymbol{\theta}\|_2^2$.*

Proposition 3.4 indicates that the variations in the local linearization of Q-function values solely depend on the variations in parameters within the subspace $\mathcal{R}(\Sigma_\pi)$. In the mean while, Proposition 3.5 indicates that such local linearization is non-singular within $\mathcal{R}(\Sigma_\pi)$. Based on these observations, we can first provide a fast convergence of $\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^t_*\|^2 = \mathcal{O}(1/T)$ and then show that $\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^T) - \widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*)\right)^2\right] = \mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^T) - \widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^T_*)\right)^2\right] \leq \mathcal{O}(1/T)$ for any $\boldsymbol{\theta}^* \in \Xi_\omega$. We summarize this result in Theorem 3.6 while presenting its proof in Appendix A.2.

Theorem 3.6. *Suppose Assumptions 3.1, 3.2 and 3.3 hold. We set $\omega = \widetilde{C}_1$ and the learning rate $\eta_t = \frac{1}{2(1-\gamma)\lambda_0(t+1)}$. If the feature map $\|\phi(s,a)\| = 1$ for each state-action pair (s,a) and the network width $m \geq m^*$, then the output $\boldsymbol{\theta}^T$ of Algorithm 1 satisfies*

$$\begin{aligned} &\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^T) - \widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*)\right)^2 \mid \boldsymbol{\theta}^0\right] \\ \leq\ & \frac{\widetilde{C}_3(\log T + 1)}{(1-\gamma)^2\lambda_0^2 T} + \frac{\widetilde{C}_4 m^{-1/2}}{(1-\gamma)\lambda_0} \cdot \sqrt{\log(T/\delta)} \\ +\ & \frac{\widetilde{C}_5 \tau^* \left(\log(T/\delta) + 1\right) \log T}{(1-\gamma)^2\lambda_0^2 T}, \end{aligned}$$

with probability at least $1 - 2\delta - 2L\exp\left(-\widetilde{C}_2 m\right)$, where τ^* is the mixing time of Markov chain in Assumption 3.2, and $\widetilde{C}_1, \cdots, \widetilde{C}_5 > 0$ are universal constants.

Let Q^π be the true state-action value function that satisfies the Bellman equation $Q^\pi = \mathcal{T}^\pi Q^\pi$. Then based on the convergence of the local linearization in Theorem 3.6, we establish the global convergence of neural temporal difference learning as Theorem 3.7.

Theorem 3.7. *Suppose the conditions in Theorem 3.6 hold. Then the output of Algorithm 1 satisfies*

$$\begin{aligned} &\mathbb{E}\Big[\left(Q(\phi(s,a);\boldsymbol{\theta}^T) - Q^\pi(s,a)\right)^2 \mid \boldsymbol{\theta}^0\Big] \\ \leq\ & \frac{3\mathbb{E}\left[\left(Q^\pi(s,a) - \Pi_{\mathcal{F}_{\omega,m}} Q^\pi(s,a)\right)^2\right]}{(1-\gamma)^2} + \widetilde{C}_6 m^{-1} \\ &+ \frac{\widetilde{C}_7(\log T + 1)}{(1-\gamma)^2\lambda_0^2 T} + \frac{\widetilde{C}_8 m^{-1/2}}{(1-\gamma)\lambda_0}\sqrt{\log(T/\delta)} \quad (15) \\ &+ \frac{\widetilde{C}_9 \tau^* \left(\log(T/\delta) + 1\right) \log T}{(1-\gamma)^2\lambda_0^2 T} \end{aligned}$$

w.p. $1 - 2\delta - 2L\exp\left(-\widetilde{C}_2 m\right)$, where $\widetilde{C}_6, \cdots, \widetilde{C}_9 > 0$ are universal constants.

Let $\epsilon_{\mathcal{F}} := \frac{3}{(1-\gamma)^2}\mathbb{E}\left[\left(Q^\pi(s,a) - \Pi_{\mathcal{F}_{\omega,m}} Q^\pi(s,a)\right)^2\right]$ be the optimal approximation error of the function class $\mathcal{F}_{\omega,m}$. Then Theorem 3.7 demonstrates that under suitable parameter choices, neural TD learning method identify an approximation error bound of $\mathcal{O}(\epsilon_{\mathcal{F}} + \epsilon + m^{-\frac{1}{2}})$ within $\tilde{\mathcal{O}}(\epsilon^{-1})$ samples. Existing works include Cai et al. (2023); Xu &

Gu (2020); Tian et al. (2022); Cayci et al. (2023) achieve $\tilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity, and (Sun et al., 2022) achieves $\mathcal{O}\big(\epsilon^{-\frac{2}{2-a}}\big), a \in (0,1]$ with additional assumptions.

Following a similar analysis while adopting an additional regularity assumption on the matrix Σ_π, one can further extend the above analysis to the neural Q-learning by substituting the Bellman operator with the Bellman optimality operator. A similar $\mathcal{O}(\epsilon^{-1})$ sample complexity can still be achieved, which is relegated to Appendix for succinctness.

4. Convergence of Minimax Neural Q-Learning

A two-player zero-sum Markov game (Littman, 1994; Bowling & Veloso, 2001; Perolat et al., 2018), as a simple variant of MDP, is defined as a six-tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}_1, \mathcal{A}_2, \mathbb{P}, r, \gamma)$. Here $\mathcal{S}$ is state space, $\mathcal{A}_1$ and $\mathcal{A}_2$ are the action space of the first and second player, respectively, $\mathbb{P} : \mathcal{A}_1 \times \mathcal{A}_2 \to \mathcal{P}(\mathcal{S})$ is the transition probability, $r : \mathcal{S} \times \mathcal{A}_1 \times \mathcal{A}_2 \to \mathbb{R}$ is the reward function and γ is the discounted factor. At time t, player 1 and player 2 take actions ($a_t^1 \in \mathcal{A}_1$ and $a_t^2 \in \mathcal{A}_2$) simultaneously. Player 1 obtains the reward $r(s_t, a_t^1, a_t^2)$. while player 2 obtains $-r(s_t, a_t^1, a_t^2)$. The goal of the two players is to maximize their cumulative rewards respectively. For a policy pair (π_1, π_2), we can define the state-action value function as follows:

$$Q^{\pi_1,\pi_2}(s,a^1,a^2) = \mathbb{E}_{\pi_1,\pi_2}\left[\sum_{t=0}^{\infty}\gamma^t \cdot r(s_t,a_t^1,a_t^2) \mid s_0 = s, a_0^1 = a^1, a_0^2 = a^2\right], \ \forall s, a^1, a^2.$$

The optimal state-action value function Q^* is defined as

$$\begin{aligned} Q^*(s,a^1,a^2) &= \max_{\pi_1}\min_{\pi_2} Q^{\pi_1,\pi_2}(s,a^1,a^2) \\ &= \min_{\pi_2}\max_{\pi_1} Q^{\pi_1,\pi_2}(s,a^1,a^2). \end{aligned}$$

We denote the optimal policy pair $\pi^* = \{\pi_1^*, \pi_2^*\}$ if $Q^*(s,a^1,a^2) = Q^{\pi_1^*,\pi_2^*}$. Moreover, the Minimax Bellman operator $\mathcal{H}$ for the Markov game is defined as

$$\mathcal{H}Q(s,a^1,a^2) = r(s,a^1,a^2) + \gamma\mathbb{E}\left[\min_{b^1}\max_{b^2} Q(s',b^1,b^2) \mid s' \sim \mathbb{P}(\cdot \mid s,a^1,a^2)\right], \ \forall s,a^1,a^2.$$

Thus $\mathcal{H}Q^* = Q^*$. Let the feature map $\boldsymbol{x} = \phi(s,a^1,a^2)$ and $\pi = \{\pi^1, \pi^2\}$ be a given learning policy for players 1 and 2. Assume that $\{s_t, a_t^1, a_t^2, r_t\}_{t=0}^T$ is a sampled trajectory of states, actions and rewards obtained from the environment using policy π. Let us recall the definition of the local linearization function class $\mathcal{F}_{\omega,m}$ introduced in (7). Consider the MSPBE minimization problem with multi-layer neural network approximation:

$$\min_{\boldsymbol{\theta}\in S_\omega} \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\big(Q(\boldsymbol{x};\boldsymbol{\theta}) - \Pi_{\mathcal{F}_{\omega,m}}\mathcal{H}Q(\boldsymbol{x};\boldsymbol{\theta})\big)^2\right].$$

To solve this problem, we still adopt the projected stochastic semi-gradient iteration method is provided described by (6), that is,

$$\boldsymbol{\theta}^{t+1} = \Pi_{S_\omega}\Big(\boldsymbol{\theta}^t - \eta_t \boldsymbol{g}\left(\boldsymbol{\theta}^t\right)\Big), \tag{16}$$

while redefining the stochastic semi-gradient estimator $\boldsymbol{g}(\boldsymbol{\theta}^t)$ as

$$\boldsymbol{g}(\boldsymbol{\theta}^t) = \Delta\left(s_t, a_t^1, a_t^2, s_{t+1}; \boldsymbol{\theta}^t\right) \cdot \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^t),$$

where $\boldsymbol{x}_t := \phi(s_t, a_t^1, a_t^2)$ and

$$\Delta\left(s_t, a_t^1, a_t^2, s_{t+1}; \boldsymbol{\theta}^t\right) = Q(\boldsymbol{x}_t;\boldsymbol{\theta}^t) - \Big(r(s_t,a_t^1,a_t^2) + \gamma \max_{b^1\in\mathcal{A}_1}\min_{b^2\in\mathcal{A}_2} Q(\phi(s_{t+1},b^1,b^2);\boldsymbol{\theta}^t)\Big). \tag{17}$$

Now we redefine the function class $\mathcal{F}_{\omega,m}$ as a collection of all local linearization of $Q(\boldsymbol{x};\boldsymbol{\theta})$ at the initial point $\boldsymbol{\theta}^0$:

$$\mathcal{F}_{\omega,m} = \Big\{\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}) = Q(\boldsymbol{x};\boldsymbol{\theta}^0) + \big\langle \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x};\boldsymbol{\theta}^0), \boldsymbol{\theta} - \boldsymbol{\theta}^0\big\rangle, \ \boldsymbol{\theta} \in S_\omega\Big\}.$$

To analyze this method, for any $\beta > 0$, we redefine the set Ξ_β introduced in (9) by replacing the Bellman operator $\mathcal{T}^\pi$ with the Minimax Bellman operator $\mathcal{H}$. Similar to (10), we still have a point $\boldsymbol{\theta}^* \in \Xi_\omega$ if and only if

$$\mathbb{E}_{\mu,\pi,\mathbb{P}}\Big[\widehat{\Delta}\left(s,a^1,a^2,s';\boldsymbol{\theta}^*\right)\big\langle\nabla_{\boldsymbol{\theta}}\widehat{Q}\left(\phi(s,a^1,a^2);\boldsymbol{\theta}^*\right), \boldsymbol{\theta} - \boldsymbol{\theta}^*\big\rangle\Big] \geq 0,$$

where $\widehat{Q}(\cdot;\boldsymbol{\theta}^*) \in \mathcal{F}_{\omega,m}$, and $\widehat{\Delta}\left(s,a^1,a^2,s';\boldsymbol{\theta}\right)$ has the same structure as $\Delta\left(s,a^1,a^2,s';\boldsymbol{\theta}\right)$ expect that the function $Q(\cdot;\boldsymbol{\theta})$ is replaced by $\widehat{Q}(\cdot;\boldsymbol{\theta})$.

Unlike the neural temporal difference learning method that aims at evaluating the state-action values of a fixed learning policy. The Minimax Bellman operator significantly sophisticates the analysis. Let us redefine the feature covariance matrix Σ_π with respect to the learning policy $\pi = \{\pi^1, \pi^2\}$, that is

$$\Sigma_\pi = \mathbb{E}_\pi\left[\nabla_{\boldsymbol{\theta}} Q(s,a^1,a^2;\boldsymbol{\theta}^0)\nabla_{\boldsymbol{\theta}} Q(s,a^1,a^2;\boldsymbol{\theta}^0)^\top\right].$$

Let the actions $(a_{\boldsymbol{\theta}}^1, a_{\boldsymbol{\theta}}^2)$ satisfies $\big\langle\nabla_{\boldsymbol{\theta}} Q(s,a^1,a^2;\boldsymbol{\theta}^0),\boldsymbol{\theta}\big\rangle = \max_{a^1\in\mathcal{A}_1}\min_{a^2\in\mathcal{A}_2}\big\langle\nabla_{\boldsymbol{\theta}} Q(s,a^1,a^2;\boldsymbol{\theta}^0),\boldsymbol{\theta}\big\rangle$. For each parameter pair $\boldsymbol{\theta}_s = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$, we define the action pair $(a_{\boldsymbol{\theta}_s}^1, a_{\boldsymbol{\theta}_s}^2)$ that satisfies

$$(a_{\boldsymbol{\theta}_s}^1, a_{\boldsymbol{\theta}_s}^2) = \arg\max_{(a^1,a^2)\in\left\{\left(a_{\boldsymbol{\theta}_1}^1,a_{\boldsymbol{\theta}_2}^2\right),\left(a_{\boldsymbol{\theta}_2}^1,a_{\boldsymbol{\theta}_1}^2\right)\right\}} \Big\{\big|\big\langle\nabla_{\boldsymbol{\theta}} Q(s,a^1,a^2;\boldsymbol{\theta}^0),\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\big\rangle\big|\Big\}.$$

Then for any $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$, the minimax feature covariance matrix is defined as follows:

$$\Sigma_\pi^*(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2) = \mathbb{E}_\pi\left[\nabla_{\boldsymbol{\theta}} Q(s,a_{\boldsymbol{\theta}_s}^1,a_{\boldsymbol{\theta}_s}^2;\boldsymbol{\theta}^0)\nabla_{\boldsymbol{\theta}} Q(s,a_{\boldsymbol{\theta}_s}^1,a_{\boldsymbol{\theta}_s}^2;\boldsymbol{\theta}^0)^\top\right].$$

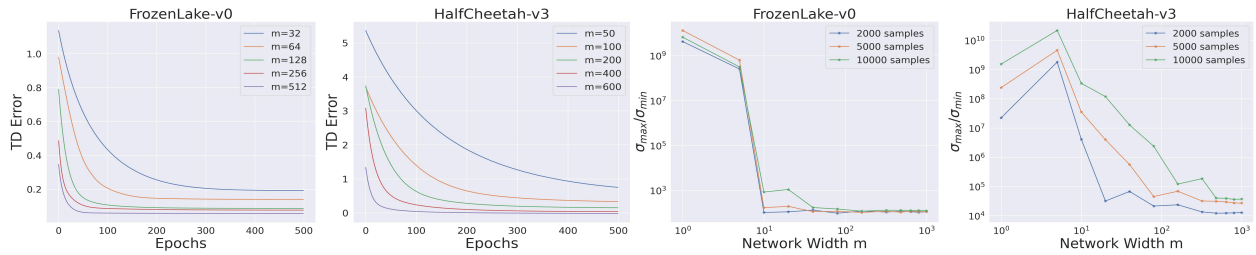

Figure 1. Training curves and the ratio of the largest and smallest non-zero singular values of Σ_π over different network widths m.

Assumption 4.1. For any $\boldsymbol{\theta}^1, \boldsymbol{\theta}^2$, there exists a constant $\nu \in (0,1)$ such that $(1-\nu)^2\Sigma_\pi - \gamma^2\Sigma_\pi^*(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \succeq 0$.

Note the original version of this assumption in (Zou et al., 2019) in fact requires a strict positive definite condition: $((1-\nu)^2\Sigma_\pi - \gamma^2\Sigma_\pi^*(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \succ 0$. Under this additional assumption, (Zou et al., 2019) obtained an $\tilde{\mathcal{O}}(\epsilon^{-1})$ sample complexity for minimax Q-learning with linear function approximation. With the help of our subspace analysis technique, in this paper, we relax it to the positive semi-definiteness ($\succeq 0$). Now we are ready to state our result for minimax neural Q-learning.

Theorem 4.2. *Suppose Assumptions 3.1, 3.2 and 4.1 hold. We set $\omega = \widetilde{C}_1$ and the learning rate $\eta_t = \frac{1}{2\nu\lambda_0(t+1)}$. If the feature map $\|\phi(s, a^1, a^2)\| = 1$ for each state-action pair (s, a^1, a^2) and the network width $m \geq m^*$, then the output $\boldsymbol{\theta}^T$ of neural minimax Q-learning Algorithm 3 satisfies*

$$\begin{aligned}
&\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^T) - \widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*)\right)^2 \mid \boldsymbol{\theta}^0\right] \\
\leq\ & \frac{\widetilde{C}_3(\log T + 1)}{\nu^2\lambda_0^2 T} + \frac{\widetilde{C}_4 m^{-1/2}}{\nu\lambda_0}\cdot\sqrt{\log(T/\delta)} \\
&+ \frac{\widetilde{C}_5\tau^*\left(\log(T/\delta)+1\right)\log T}{\nu^2\lambda_0^2 T},
\end{aligned}$$

with probability at least $1 - 2\delta - 2L\exp\left(-\widetilde{C}_2 m\right)$, where τ^* is the mixing time of Markov chain in Assumption 3.2, and $\left\{\widetilde{C}_i > 0\right\}_{i=1,\ldots,5}$ are universal constants.

Theorem 4.2 establishes a finite-time analysis of $\tilde{\mathcal{O}}(\epsilon^{-1})$-sample complexity for minimax neural Q-learning in terms of the function class $\mathcal{F}_{\omega,m}$. For a more specific description and theorem proof, see Appendix C. To the best of our knowledge, this is the first analysis of minimax Q-learning with neural network function approximation, characterized by a complexity bound of $\tilde{\mathcal{O}}(\epsilon^{-1})$.

5. Experiments

Finally, we construct several experiments over the OpenAI Gym (Brockman et al., 2016) tasks and validate our theoretical findings. We consider a two-layer neural network, as follows:

$$Q(s, a; \boldsymbol{\theta}) := \frac{1}{\sqrt{m}}\sum_{r=1}^{m} b_r\sigma(\boldsymbol{\theta}_r^\top\phi(s,a)),$$

where $\sigma(\cdot)$ is ELU activation in this section. Furthermore, details regarding the initialization and iteration methods for the parameters can be found in Section 2. For all experiments, we generate samples based on a prescribed ϵ-greedy policy with $\epsilon = 0.1$. To prevent redundancy in the features $\phi(s,a)$, we employ one-hot encoding for discrete action-state spaces and implement a fixed grid discretization for continuous spaces. when both $\phi(s,a)$ and $\phi(s',a')$ belong to the same one-hot encoding or grid, we treat them as the same sample point. Our investigation into the impact of network width on the TD learning algorithm will be conducted from two perspectives: (i) examining whether the network width m is correlated with the TD error, and (ii) exploring the existence of constants m^* and λ_0 that satisfy Assumption 3.1.

The four subfigures in Figure 1 represent two types of environments: one with a discrete state space and the other with a continuous state space. The first two subfigures depict the convergence performance of the TD algorithm at different network widths. We generate 2,000 sample points and run for 500 epochs. Notably, as the parameter m increases, the TD algorithm demonstrates faster convergence, resulting in smaller final TD errors. The latter two subfigures illustrate the existence of m^* and λ_0. Specifically, we compute the largest non-zero singular value $\sigma_{\max}$ and smallest non-zero singular value $\sigma_{\min}$ of the matrix Σ_π. To mitigate the absolute magnitude of $\sigma_{\min}$, we introduce the ratio $r = \sigma_{\max}/\sigma_{\min}$ as a metric to validate Assumption 3.1. It can be observed that the value of r approaches a constant as m increases for all cases, providing empirical support for the validity of the assumption.

6. Conclusion

We study the finite-time analysis of the TD and Q learning methods with neural network approximation, where the

state-action pairs are generated by a given policy under the Markovian sampling. Besides the convergence to the true action-value function except for an inevitable function approximation error, an improved analysis technique is introduced to establish an $\tilde{\mathcal{O}}(\epsilon^{-1})$ complexity for the neural TD and Q learning methods, which improves the existing $\tilde{\mathcal{O}}(\epsilon^{-2})$ complexity. For future work, it is also interesting to investigate if the proposed technique can improve the current complexity estimate of the actor-critic methods, which are partially built upon the neural TD methods.

7. Acknowledgements

Dr. Zaiwen Wen is supported in part by the NSFC grant 12331010. Dr. Junyu Zhang is supported in part by the MOE AcRF grant A-0009530-05-00.

References

Allen-Zhu, Z., Li, Y., and Liang, Y. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 32, 2019a.

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pp. 242–252. PMLR, 2019b.

Barakat, A., Bianchi, P., and Lehmann, J. Analysis of a target-based actor-critic algorithm with linear function approximation. In *International Conference on Artificial Intelligence and Statistics*, pp. 991–1040. PMLR, 2022.

Bertsekas, D. *Dynamic programming and optimal control: Volume I*, volume 1. Athena scientific, 2012.

Bhandari, J., Russo, D., and Singal, R. A finite time analysis of temporal difference learning with linear function approximation. In *Conference on learning theory*, pp. 1691–1692. PMLR, 2018.

Borkar, V. S. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

Bowling, M. and Veloso, M. Rational and convergent learning in stochastic games. In *International joint conference on artificial intelligence*, volume 17, pp. 1021–1026. Citeseer, 2001.

Boyan, J. A. Technical update: Least-squares temporal difference learning. *Machine learning*, 49(2):233–246, 2002.

Bradtke, S. J. and Barto, A. G. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1):33–57, 1996.

Brandfonbrener, D. and Bruna, J. Geometric insights into the convergence of nonlinear td learning. *arXiv preprint arXiv:1905.12185*, 2019.

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

Cai, Q., Yang, Z., Lee, J. D., and Wang, Z. Neural temporal difference and q learning provably converge to global optima. *Mathematics of Operations Research*, 2023.

Cao, Y. and Gu, Q. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Cao, Y. and Gu, Q. Generalization error bounds of gradient descent for learning over-parameterized deep relu networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3349–3356, 2020.

Cayci, S., Satpathi, S., He, N., and Srikant, R. Sample complexity and overparameterization bounds for temporal difference learning with neural network approximation. *IEEE Transactions on Automatic Control*, 2023.

Dalal, G., Szörényi, B., Thoppe, G., and Mannor, S. Finite sample analyses for td (0) with function approximation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pp. 1675–1685. PMLR, 2019.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

Fan, J., Wang, Z., Xie, Y., and Yang, Z. A theoretical analysis of deep q-learning. In *Learning for dynamics and control*, pp. 486–489. PMLR, 2020.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Godfrey, L. B. An evaluation of parametric activation functions for deep learning. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 3006–3011. IEEE, 2019.

Jaakkola, T., Jordan, M., and Singh, S. Convergence of stochastic iterative dynamic programming algorithms. *Advances in neural information processing systems*, 6, 1993.

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

Ke, Z., Wen, Z., and Zhang, J. Provably efficient gauss-newton temporal difference learning method with function approximation. *arXiv preprint arXiv:2302.13087*, 2023.

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

Kostrikov, I., Fergus, R., Tompson, J., and Nachum, O. Offline reinforcement learning with fisher divergence critic regularization. In *International Conference on Machine Learning*, pp. 5774–5783. PMLR, 2021.

Lazaric, A., Ghavamzadeh, M., and Munos, R. Finite-sample analysis of lstd. In *ICML-27th International Conference on Machine Learning*, pp. 615–622, 2010.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

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

Littman, M. L. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pp. 157–163. Elsevier, 1994.

Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S., and Petrik, M. Finite-sample analysis of proximal gradient td algorithms. *arXiv preprint arXiv:2006.14364*, 2020a.

Liu, C., Zhu, L., and Belkin, M. On the linearity of large non-linear models: when and why the tangent kernel is constant. *Advances in Neural Information Processing Systems*, 33:15954–15964, 2020b.

Maei, H., Szepesvari, C., Bhatnagar, S., Precup, D., Silver, D., and Sutton, R. S. Convergent temporal-difference learning with arbitrary smooth function approximation. *Advances in neural information processing systems*, 22, 2009.

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

Perkins, T. J. and Pendrith, M. D. On the existence of fixed points for q-learning and sarsa in partially observable domains. In *ICML*, pp. 490–497, 2002.

Perolat, J., Piot, B., and Pietquin, O. Actor-critic fictitious play in simultaneous move multistage games. In *International Conference on Artificial Intelligence and Statistics*, pp. 919–928. PMLR, 2018.

Prashanth, L., Korda, N., and Munos, R. Fast lstd using stochastic approximation: Finite time analysis and application to traffic control. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 66–81. Springer, 2014.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

Sun, T., Li, D., and Wang, B. Finite-time analysis of adaptive temporal difference learning with deep neural networks. *Advances in Neural Information Processing Systems*, 35:19592–19604, 2022.

Sutton, R. S. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

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

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C., and Wiewiora, E. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th annual international conference on machine learning*, pp. 993–1000, 2009a.

Sutton, R. S., Szepesvári, C., and Maei, H. R. A convergent o (n) algorithm for off-policy temporal-difference learning with linear function approximation. *Advances in neural information processing systems*, 21(21):1609–1616, 2009b.

Tagorti, M. and Scherrer, B. On the rate of convergence and error bounds for lstd (λ). In *International Conference on Machine Learning*, pp. 1521–1529. PMLR, 2015.

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

Tian, H., Paschalidis, I., and Olshevsky, A. On the performance of temporal difference learning with neural networks. In *The Eleventh International Conference on Learning Representations*, 2022.

Touati, A., Bacon, P.-L., Precup, D., and Vincent, P. Convergent tree backup and retrace with function approximation. In *International Conference on Machine Learning*, pp. 4955–4964. PMLR, 2018.

Tsitsiklis, J. and Van Roy, B. Analysis of temporal-diffference learning with function approximation. *Advances in neural information processing systems*, 9, 1996.

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

Wu, Y., Tucker, G., and Nachum, O. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

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

Zou, S., Xu, T., and Liang, Y. Finite-sample analysis for sarsa with linear function approximation. *Advances in neural information processing systems*, 32, 2019.

A. Details of Section 3

A.1. Proof of (14)

$$
\begin{aligned}
& \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\widehat{\Delta}\left(\boldsymbol{x},\boldsymbol{x}';\boldsymbol{\theta}_*\right)\cdot\left\langle\nabla_{\boldsymbol{\theta}}\widehat{Q}\left(\boldsymbol{x};\boldsymbol{\theta}_*\right),\boldsymbol{\theta}'-\boldsymbol{\theta}_*\right\rangle\right] \\
\overset{(i)}{=}\ & \mathbb{E}_{\mu,\pi}\left[\mathbb{E}_{\mathbb{P}}\left[\widehat{\Delta}\left(\boldsymbol{x},\boldsymbol{x}';\boldsymbol{\theta}_*\right)\right]\cdot\left\langle\nabla_{\boldsymbol{\theta}}\widehat{Q}\left(\boldsymbol{x};\boldsymbol{\theta}_0\right),\boldsymbol{\theta}'-\bar{\boldsymbol{\theta}}\right\rangle\right] \\
\overset{(ii)}{=}\ & \mathbb{E}_{\mu,\pi}\left[\mathbb{E}_{\mathbb{P}}\left[\widehat{\Delta}\left(\boldsymbol{x},\boldsymbol{x}';\bar{\boldsymbol{\theta}}\right)\right]\cdot\left\langle\nabla_{\boldsymbol{\theta}}\widehat{Q}\left(\boldsymbol{x};\boldsymbol{\theta}_0\right),\boldsymbol{\theta}'-\bar{\boldsymbol{\theta}}\right\rangle\right] \\
=\ & \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\widehat{\Delta}\left(\boldsymbol{x},\boldsymbol{x}';\bar{\boldsymbol{\theta}}\right)\cdot\left\langle\nabla_{\boldsymbol{\theta}}\widehat{Q}\left(\boldsymbol{x};\boldsymbol{\theta}_0\right),\boldsymbol{\theta}'-\bar{\boldsymbol{\theta}}\right\rangle\right]
\end{aligned}
$$

where (i) is because $\nabla_{\boldsymbol{\theta}}\widehat{Q}\left(\cdot\,;\boldsymbol{\theta}_*\right)=\widehat{Q}\left(\cdot\,;\boldsymbol{\theta}_0\right)$, the decomposition

$$
\boldsymbol{\theta}'-\boldsymbol{\theta}_*=\boldsymbol{\theta}'-\bar{\boldsymbol{\theta}}+\bar{\boldsymbol{\theta}}-\boldsymbol{\theta}_*=\boldsymbol{\theta}'-\bar{\boldsymbol{\theta}}+\left(\bar{\boldsymbol{\theta}}_\perp-\boldsymbol{\theta}_\perp\right)
$$

the fact that $\left(\bar{\boldsymbol{\theta}}_\perp-\boldsymbol{\theta}_\perp\right)\in\mathcal{K}(\Sigma_\pi)$, and (13), (ii) is because $\left(\bar{\boldsymbol{\theta}}-\boldsymbol{\theta}^*\right)\in\mathcal{K}(\Sigma_\pi)$.

A.2. Proof of Theorem 3.6

Proof. Recall the definition of the semi-gradient in Section (6). We denote $\bar{\mathbf{g}}(\boldsymbol{\theta})$ as its expectation. Let $\bar{\mathbf{m}}(\boldsymbol{\theta})$ and $\mathbf{m}(\boldsymbol{\theta})$ also be the corresponding semi-gradients based on the linearized function $\widehat{Q}(\cdot;\boldsymbol{\theta})$, that is,

$$
\begin{aligned}
\mathbf{g}(\boldsymbol{\theta}^t) &= \Delta(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\boldsymbol{\theta}^t)\cdot\nabla_{\boldsymbol{\theta}}Q(\boldsymbol{x}_t;\boldsymbol{\theta}^t), \quad \bar{\mathbf{g}}(\boldsymbol{\theta}^t) = \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\mathbf{g}(\boldsymbol{\theta}^t)\right] \\
\mathbf{m}(\boldsymbol{\theta}^t) &= \widehat{\Delta}(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\boldsymbol{\theta}^t)\cdot\nabla_{\boldsymbol{\theta}}Q(\boldsymbol{x}_t;\boldsymbol{\theta}^0), \quad \bar{\mathbf{m}}(\boldsymbol{\theta}^t) = \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\mathbf{m}(\boldsymbol{\theta}^t)\right],
\end{aligned}
$$

where

$$
\begin{aligned}
\Delta(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\boldsymbol{\theta}^t) &= Q(\boldsymbol{x}_t;\boldsymbol{\theta}^t)-\left(r(s_t,a_t)+\gamma\cdot Q(\boldsymbol{x}_{t+1};\boldsymbol{\theta}^t)\right), \\
\widehat{\Delta}(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\boldsymbol{\theta}^t) &= \widehat{Q}(\boldsymbol{x}_t;\boldsymbol{\theta}^t)-\left(r(s_t,a_t)+\gamma\cdot\widehat{Q}(\boldsymbol{x}_{t+1};\boldsymbol{\theta}^t)\right).
\end{aligned}
$$

To simplify the notation, let $\Delta_t:=\Delta(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\boldsymbol{\theta}^t)$ and $\widehat{\Delta}_t:=\widehat{\Delta}(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\boldsymbol{\theta}^t)$. Recall the definition of the range space $\mathcal{R}(\Sigma_\pi)$ and the kernel space $\mathcal{K}(\Sigma_\pi)$. By Proposition 3.4, we know that $\boldsymbol{v}_1^\top\boldsymbol{v}_2=0$ for any vector $\boldsymbol{v}_1\in\mathcal{R}(\Sigma_\pi),\boldsymbol{v}_2\in\mathcal{K}(\Sigma_\pi)$ thus $\left\langle\nabla_{\boldsymbol{\theta}}Q(\boldsymbol{x};\boldsymbol{\theta}^0),\boldsymbol{\theta}_\perp\right\rangle=0$ for any feature map $\boldsymbol{x}$ and parameter $\boldsymbol{\theta}_\perp\in\mathcal{K}(\Sigma_\pi)$. Then we can decompose $\|\boldsymbol{\theta}^{t+1}-\boldsymbol{\theta}_*^{t+1}\|^2$ as

$$
\begin{aligned}
\|\boldsymbol{\theta}^{t+1}-\boldsymbol{\theta}_*^{t+1}\|^2 &= \left\|\Pi_{S_{2\omega}}\left(\boldsymbol{\theta}^t-\eta_t\mathbf{g}(\boldsymbol{\theta}^t)\right)-\Pi_{S_{2\omega}}\left(\boldsymbol{\theta}_*^{t+1}\right)\right\|^2 \\
&\le \|\boldsymbol{\theta}^t-\eta_t\mathbf{g}(\boldsymbol{\theta}^t)-\boldsymbol{\theta}_*^{t+1}\|^2 \\
&= \|\boldsymbol{\theta}^t-\eta_t\mathbf{g}(\boldsymbol{\theta}^t)-\boldsymbol{\theta}_*^t+\boldsymbol{\theta}_*^t-\boldsymbol{\theta}_*^{t+1}\|^2 \\
&= \|\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t\|^2+\eta_t^2\|\mathbf{g}(\boldsymbol{\theta}^t)\|^2+\|\boldsymbol{\theta}_*^t-\boldsymbol{\theta}_*^{t+1}\|^2-2\eta_t\left\langle\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t,\mathbf{g}(\boldsymbol{\theta}^t)\right\rangle \\
&\quad -2\eta_t\left\langle\boldsymbol{\theta}_*^t-\boldsymbol{\theta}_*^{t+1},\mathbf{g}(\boldsymbol{\theta}^t)\right\rangle+2\eta_t\left\langle\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t,\boldsymbol{\theta}_*^t-\boldsymbol{\theta}_*^{t+1}\right\rangle \\
&\overset{(i)}{=} \|\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t\|^2-2\eta_t\left\langle\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t,\mathbf{g}(\boldsymbol{\theta}^t)\right\rangle \\
&\quad -2\eta_t\left\langle\boldsymbol{\theta}_*^t-\boldsymbol{\theta}_*^{t+1},\mathbf{g}(\boldsymbol{\theta}^t)-\mathbf{m}(\boldsymbol{\theta}^t)\right\rangle+\eta_t^2\|\mathbf{g}(\boldsymbol{\theta}^t)\|^2,
\end{aligned}
$$

where (i) follows

$$
\left\langle\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t,\boldsymbol{\theta}_*^t-\boldsymbol{\theta}_*^{t+1}\right\rangle=\left\langle\boldsymbol{\theta}_\|^t-\boldsymbol{\theta}_\|^*,\boldsymbol{\theta}_\perp^t-\boldsymbol{\theta}_\perp^{t+1}\right\rangle=0,
$$

and

$$
\begin{aligned}
\left\langle\boldsymbol{\theta}_*^t-\boldsymbol{\theta}_*^{t+1},\mathbf{m}(\boldsymbol{\theta}^t)\right\rangle &= \left\langle\boldsymbol{\theta}_\perp^t-\boldsymbol{\theta}_\perp^{t+1},\mathbf{m}(\boldsymbol{\theta}^t)\right\rangle \\
&= \widehat{\Delta}_t\cdot\left\langle\boldsymbol{\theta}_\perp^t-\boldsymbol{\theta}_\perp^{t+1},\nabla Q(\boldsymbol{x}_t;\boldsymbol{\theta}^0)\right\rangle = 0.
\end{aligned}
$$

Recall the stationarity condition (11), for any $t \in \{1, 2, \cdots, T\}$,

$$\begin{aligned} 0 &\leq \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\widehat{\Delta}\left(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}; \boldsymbol{\theta}^*\right)\left\langle \nabla_{\boldsymbol{\theta}} \widehat{Q}\left(\boldsymbol{x}_t; \boldsymbol{\theta}^*\right), \boldsymbol{\theta}^t - \boldsymbol{\theta}^*\right\rangle\right] \\ &= \left\langle \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\widehat{\Delta}\left(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}; \boldsymbol{\theta}_*^t\right) \nabla_{\boldsymbol{\theta}} \widehat{Q}\left(\boldsymbol{x}_t; \boldsymbol{\theta}_*^t\right)\right], \boldsymbol{\theta}^t - \boldsymbol{\theta}^*\right\rangle \\ &\overset{(i)}{=} \left\langle \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\widehat{\Delta}\left(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}; \boldsymbol{\theta}_*^t\right) \nabla_{\boldsymbol{\theta}} \widehat{Q}\left(\boldsymbol{x}_t; \boldsymbol{\theta}_*^t\right)\right], \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\right\rangle = \left\langle \bar{\mathbf{m}}(\boldsymbol{\theta}_*^t), \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\right\rangle, \end{aligned}$$

where (i) is the same as the proof in Section A.1. Therefore,

$$\begin{aligned} & \|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}_*^{t+1}\|^2 \\ \leq\ & \|\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\|^2 - 2\eta_t\left\langle \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t, \mathbf{g}(\boldsymbol{\theta}^t)\right\rangle - 2\eta_t\left\langle \boldsymbol{\theta}_*^t - \boldsymbol{\theta}_*^{t+1}, \mathbf{g}(\boldsymbol{\theta}^t) - \mathbf{m}(\boldsymbol{\theta}^t)\right\rangle + \eta_t^2\|\mathbf{g}(\boldsymbol{\theta}^t)\|^2 \\ =\ & \|\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\|^2 - 2\eta_t\left\langle \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t, \mathbf{g}(\boldsymbol{\theta}^t) - \mathbf{m}(\boldsymbol{\theta}^t)\right\rangle - 2\eta_t\left\langle \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t, \mathbf{m}(\boldsymbol{\theta}^t) - \bar{\mathbf{m}}(\boldsymbol{\theta}^t)\right\rangle \\ & -2\eta_t\left\langle \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t, \bar{\mathbf{m}}(\boldsymbol{\theta}^t)\right\rangle - 2\eta_t\left\langle \boldsymbol{\theta}_*^t - \boldsymbol{\theta}_*^{t+1}, \mathbf{g}(\boldsymbol{\theta}^t) - \mathbf{m}(\boldsymbol{\theta}^t)\right\rangle + \eta_t^2\|\mathbf{g}(\boldsymbol{\theta}^t)\|^2 \\ \overset{(i)}{\leq}\ & \|\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\|^2 + \eta_t^2 \underbrace{\|\mathbf{g}(\boldsymbol{\theta}^t)\|^2}_{\mathrm{I}_1\text{: Gradient Bound}} \underbrace{-2\eta_t\left\langle \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^{t+1}, \mathbf{g}(\boldsymbol{\theta}^t) - \mathbf{m}(\boldsymbol{\theta}^t)\right\rangle}_{\mathrm{I}_2\text{: Gradient Gap}} \\ & \underbrace{-2\eta_t\left\langle \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t, \mathbf{m}(\boldsymbol{\theta}^t) - \bar{\mathbf{m}}(\boldsymbol{\theta}^t)\right\rangle}_{\mathrm{I}_3\text{: Markov Sampling Error}} \underbrace{-2\eta_t\left\langle \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t, \bar{\mathbf{m}}(\boldsymbol{\theta}^t) - \bar{\mathbf{m}}(\boldsymbol{\theta}_*^t)\right\rangle}_{\mathrm{I}_4\text{: Gradient Decent}}, \end{aligned} \tag{18}$$

where (i) follows $\left\langle \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t, \bar{\mathbf{m}}(\boldsymbol{\theta}_*^t)\right\rangle \geq 0$ for any $0 \leq t \leq T-1$.

Next, we analyze the upper bounds of $\mathbf{I}_1$, $\mathbf{I}_2$, $\mathbf{I}_3$ and $\mathbf{I}_4$ item by item. To simplify the notation, let $\{C_i > 0\}_{i=1,\ldots,7}$ be universal constants in this section. We set $\omega = C_1$ and $\delta \in (0,1)$. By Lemma D.5, we have

$$\|\boldsymbol{g}(\boldsymbol{\theta}^t)\|^2 \leq C_2\sqrt{\log(T/\delta)} \tag{19}$$

and

$$\begin{aligned} \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\left|\left\langle \mathbf{g}\left(\boldsymbol{\theta}^t\right) - \mathbf{m}\left(\boldsymbol{\theta}^t\right), \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^{t+1}\right\rangle\right| \mid \boldsymbol{\theta}^0\right] &\leq \left(C_3\omega m^{-\frac{1}{2}}\sqrt{\log(T/\delta)} + C_4 m^{-\frac{1}{2}}\right)\left\|\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^{t+1}\right\| \\ &\overset{(i)}{\leq} C_5 m^{-1/2}\sqrt{\log(T/\delta)}, \end{aligned} \tag{20}$$

with probability at least $1 - 2\delta - 2\exp\left(-C_6 m\right)$, where (i) follows $\omega = C_1$ and

$$\begin{aligned} \|\theta^t - \theta_*^{t+1}\| &\leq \|\theta^t - \theta^0\| + \|\theta^0 - \theta_*^0\| + \|\theta_*^0 - \theta_*^{t+1}\| \\ &= \|\theta^t - \theta^0\| + \|\theta^0 - \theta_*^0\| + \|\theta_\perp^0 - \theta_\perp^{t+1}\| \\ &\leq \|\theta^t - \theta^0\| + \|\theta^0 - \theta_*^0\| + \|\theta^0 - \theta^{t+1}\| \leq 3\omega. \end{aligned}$$

Thus (19) and (20) provide upper bounds on $\mathbf{I}_1$ and $\mathbf{I}_2$, respectively. The next lemma provides an estimate of the Markov sampling error.

Lemma A.1. *Suppose the learning rate sequence $\{\eta_0, \eta_1, \ldots, \eta_T\}$ is non-increasing. Under Assumption 3.2, it holds that*

$$\mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\left\langle \mathbf{m}\left(\boldsymbol{\theta}^t\right) - \overline{\mathbf{m}}\left(\boldsymbol{\theta}^t\right), \boldsymbol{\theta}^t - \boldsymbol{\theta}^*\right\rangle \mid \boldsymbol{\theta}^0\right] \leq C_7\left(\log(T/\delta) + C_1^2\right)\tau^*\eta_{\max\{0, t-\tau^*\}}, \tag{21}$$

for any fixed $t \leq T$, where

$$\tau^* = \min\left\{t = 0, 1, 2, \ldots \mid \kappa\rho^t \leq \eta_T\right\}$$

is the mixing time of the Markov chain $\{s_t, a_t\}_{t=0,1,\ldots}$.

Proof. We adopt the proof framework outlined in Lemma 6.2 of Xu & Gu (2020). However, variations in the neural network settings lead to differences in the norms of gradients and parameters, thereby resulting in slight variations in the results. Thereby we have

$$\mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\left\langle \mathbf{m}\left(\boldsymbol{\theta}^t\right)-\overline{\mathbf{m}}\left(\boldsymbol{\theta}^t\right),\boldsymbol{\theta}^t-\boldsymbol{\theta}^*\right\rangle \mid \boldsymbol{\theta}^0\right]\leq C_7\left(\log(T/\delta)+\omega^2\right)\tau^*\eta_{\max\{0,t-\tau^*\}}.$$

□

Looking back at the definitions of λ_0 and Σ_π, and the discussion in Section 3.2, we derive Lemmas A.2 and A.3 to estimate $\mathbf{I}_4$.

Lemma A.2. *Let λ_0 as the minimum nonzero singular value of Σ_π. For any $\boldsymbol{\theta}\in\mathcal{R}(\Sigma_\pi)$, we have*

$$\boldsymbol{\theta}^\top\Sigma_\pi\boldsymbol{\theta}\geq\lambda_0\|\boldsymbol{\theta}\|_2^2.$$

Lemma A.3. *Under Assumption 3.3, we have that*

$$\mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\left\langle\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t,\bar{\mathbf{m}}(\boldsymbol{\theta}^t)-\bar{\mathbf{m}}(\boldsymbol{\theta}_*^t)\right\rangle\mid\boldsymbol{\theta}^0\right]\geq(1-\gamma)\lambda_0\cdot\|\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t\|^2. \tag{22}$$

Proof. Define $d\sim\mu\times\pi$. To begin with, the Bellman operator $\mathcal{T}^\pi$ is a γ-contraction with ℓ_2-norm since d is the stationary distribution of (s,a) corresponding to the policy π. In details, consider

$$\begin{aligned}\mathbb{E}_{(s,a)\sim d}\left[\left(\mathcal{T}^\pi Q_1(\boldsymbol{x})-\mathcal{T}^\pi Q_2(\boldsymbol{x})\right)^2\right]&=\gamma^2\mathbb{E}_{(s,a)\sim d}\left[\mathbb{E}\left[\left(Q_1(\boldsymbol{x}')-Q_2(\boldsymbol{x}')\right)^2\mid s'\sim\mathbb{P}(\cdot|s,a),a'\sim\pi(\cdot|s')\right]\right]\\&\overset{(i)}{\leq}\gamma^2\mathbb{E}_{(s,a)\sim d}\left[\left(Q_1(\boldsymbol{x})-Q_2(\boldsymbol{x})\right)^2\right],\end{aligned} \tag{23}$$

where (i) follows that $\boldsymbol{x}$ and $\boldsymbol{x}'$ have the same stationary distribution. To simplify the notation, we denote $\mathbb{E}[\cdot]$ as $\mathbb{E}_{\mu,\pi,\mathbb{P}}[\cdot]$ in the proof of this lemma. Then we compute

$$\begin{aligned}&\mathbb{E}\left[\left\langle\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t,\bar{\mathbf{m}}(\boldsymbol{\theta}^t)-\bar{\mathbf{m}}(\boldsymbol{\theta}_*^t)\right\rangle\mid\boldsymbol{\theta}^0\right]\\=\;&\mathbb{E}\left[\left(\widehat{\Delta}(\boldsymbol{x},\boldsymbol{x}';\boldsymbol{\theta}^t)-\widehat{\Delta}(\boldsymbol{x},\boldsymbol{x}';\boldsymbol{\theta}_*^t)\right)\left\langle\nabla Q(\boldsymbol{x};\boldsymbol{\theta}^0),\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t\right\rangle\mid\boldsymbol{\theta}^0\right]\\=\;&\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^t)-\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}_*^t)\right)\left\langle\nabla Q(\boldsymbol{x};\boldsymbol{\theta}^0),\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t\right\rangle\mid\boldsymbol{\theta}^0\right]\\&-\gamma\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x}';\boldsymbol{\theta}^t)-\widehat{Q}(\boldsymbol{x}';\boldsymbol{\theta}_*^t)\right)\left\langle\nabla Q(\boldsymbol{x};\boldsymbol{\theta}^0),\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t\right\rangle\mid\boldsymbol{\theta}^0\right]\\=\;&\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}^t)-\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}_*^t)\right)^2\right]-\gamma\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}^t)-\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}_*^t)\right)\cdot\left(\widehat{Q}(\boldsymbol{x}';\boldsymbol{\theta}^t)-\widehat{Q}(\boldsymbol{x}';\boldsymbol{\theta}_*^t)\right)\right]\\\overset{(i)}{\geq}\;&\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}^t)-\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}_*^t)^2\right)\right]-\gamma\sqrt{\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}^t)-\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}_*^t)\right)^2\right]}\cdot\sqrt{\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x}';\boldsymbol{\theta}^t)-\widehat{Q}(\boldsymbol{x}';\boldsymbol{\theta}_*^t)\right)^2\right]}\\\overset{(i)}{\geq}\;&(1-\gamma)\,\mathbb{E}\left[\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}^t)-\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}_*^t)^2\right]\\=\;&(1-\gamma)(\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t)^\top\Sigma_\pi(\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t)\\\overset{(ii)}{\geq}\;&(1-\gamma)\lambda_0\cdot\|\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t\|^2,\end{aligned} \tag{24}$$

where (i) follows the Cauchy-Schwarz inequality, (ii) follows (23), and (iii) follows $\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t=\boldsymbol{\theta}_{\|}^t-\bar{\boldsymbol{\theta}}_{\|}\in\mathcal{R}(\Sigma_\pi)$ and Lemma A.2, which provides λ_0-strong convexity. Thus we complete the proof of Lemma A.3. □

Given $\boldsymbol{\theta}^0$, taking the expectation on both sides of (18) and plugging (19)∼(22) into (18) yields that

$$\begin{aligned}\mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\|\boldsymbol{\theta}^{t+1}-\boldsymbol{\theta}_*^{t+1}\|^2\mid\boldsymbol{\theta}^0\right]\leq\;&(1-2\eta_t(1-\gamma)\lambda_0)\mathbb{E}\left[\|\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t\|^2\mid\boldsymbol{\theta}^0\right]+C_2\eta_t^2\\&+2\eta_tC_5m^{-1/2}\sqrt{\log(T/\delta)}+2\eta_tC_7\left(\log(T/\delta)+C_1^2\right)\tau^*\eta_{\max\{0,t-\tau^*\}}.\end{aligned}$$

We choose $\eta_t = \frac{1}{2(1-\gamma)\lambda_0(t+1)}$ and have that

$$(1-\gamma)\lambda_0(t+1)\mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}_*^{t+1}\|^2 \mid \boldsymbol{\theta}^0\right] \leq (1-\gamma)\lambda_0 t\, \mathbb{E}\left[\|\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\|^2 \mid \boldsymbol{\theta}^0\right] + C_2\eta_t + C_5 m^{-1/2}\sqrt{\log(T/\delta)} + C_7\left(\log(T/\delta) + C_1^2\right)\tau^*\eta_{\max\{0,t-\tau^*\}}.$$

Summing (25) from $t = 0, 1, \cdots, T-1$ yields that

$$\begin{aligned}\mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\|\boldsymbol{\theta}^T - \boldsymbol{\theta}_*^T\|^2 \mid \boldsymbol{\theta}^0\right] &\leq \frac{1}{(1-\gamma)\lambda_0 T}\sum_{t=0}^{T-1}\Big(C_2\eta_t + C_5 m^{-1/2}\sqrt{\log(T/\delta)} \\ &\quad + C_7\left(\log(T/\delta) + C_1^2\right)\tau^*\eta_{\max\{0,t-\tau^*\}}\Big) \\ &\leq \frac{C_2(\log T + 1)}{2(1-\gamma)^2\lambda_0^2 T} + \frac{C_5 m^{-1/2}\sqrt{\log(T/\delta)}}{(1-\gamma)\lambda_0} + \frac{C_8\tau^*\left(\log(T/\delta)+1\right)\log T}{2(1-\gamma)^2\lambda_0^2 T}.\end{aligned}$$

Therefore, according to the gradient bound (19), we have

$$\begin{aligned}\mathbb{E}_{\mu,\pi}\left[\left(\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^T) - \widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*)\right)^2 \mid \boldsymbol{\theta}^0\right] &= \mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^T) - \widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}_*^T)\right)^2 \mid \boldsymbol{\theta}^0\right] \\ &\leq C_3^2 m\mathbb{E}\left[\|\boldsymbol{\theta}^T - \boldsymbol{\theta}_*^T\|^2 \mid \boldsymbol{\theta}^0\right] \\ &\leq \frac{C_2^3(\log T + 1)}{2(1-\gamma)^2\lambda_0^2 T} + \frac{C_2^2 C_5 m^{-1/2}\sqrt{\log(T/\delta)}}{(1-\gamma)\lambda_0} \\ &\quad + \frac{C_2^2 C_8\tau^*\left(\log(T/\delta)+1\right)\log T}{2(1-\gamma)^2\lambda_0^2 T}\end{aligned}$$

with probability at least $1 - 2\delta - 2L\exp\left(-C_6 m\right)$. Let $\widetilde{C}_1 = \max\{1, C_1\}, \widetilde{C}_2 = C_6, \widetilde{C}_3 = \frac{C_2^3}{2}, \widetilde{C}_4 = \frac{C_2^2 C_5}{2}$, and $\widetilde{C}_5 = \frac{C_2^2 C_8}{2}$, and we complete the proof. □

A.3. Proof of Theorem 3.7

Proof. Let $(s,a) \sim \mu \times \pi =: d$. To simplify the notation, we denote $\mathbb{E}[\cdot]$ as $\mathbb{E}_{(s,a)\sim d}[\cdot]$ in this subsection. Note that

$$\begin{aligned}\mathbb{E}\left[\left(Q(\boldsymbol{x};\boldsymbol{\theta}^T) - Q^*(s,a)\right)^2 \mid \boldsymbol{\theta}^0\right] &\leq 3\mathbb{E}\left[\left(Q(\boldsymbol{x};\boldsymbol{\theta}^T) - \widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^T)\right)^2 \mid \boldsymbol{\theta}^0\right] \\ &\quad + 3\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^T) - \widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*)\right)^2 \mid \boldsymbol{\theta}^0\right] + 3\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*) - Q^*(s,a)\right)^2 \mid \boldsymbol{\theta}^0\right].\end{aligned}$$

By Lemma D.4, we have

$$\mathbb{E}\left[\left(Q(\boldsymbol{x};\boldsymbol{\theta}^T) - \widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^T)\right)^2 \mid \boldsymbol{\theta}^0\right] \leq C_8 m^{-1} \tag{25}$$

with probability at least $1-\delta$. Recall that $\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*)$ is the fixed point of $\Pi_{\mathcal{F}_{\omega,m}}\mathcal{T}$ and $Q^*(s,a)$ is the fixed point of $\mathcal{T}$. We define the ℓ_2-norm $\|f(s,a)\|_d^2 = \mathbb{E}_{(s,a)\sim d}\left[f(s,a)^2\right]$. Thus

$$\begin{aligned}\left\|\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*) - Q^*(s,a)\right\|_d &= \left\|\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*) - \Pi_{\mathcal{F}_{\omega,m}}Q^*(s,a) + \Pi_{\mathcal{F}_{\omega,m}}Q^*(s,a) - Q^*(s,a)\right\|_d \\ &\overset{(i)}{=} \left\|\Pi_{\mathcal{F}_{\omega,m}}\mathcal{T}\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*) - \Pi_{\mathcal{F}_{\omega,m}}\mathcal{T}Q^*(s,a) + \Pi_{\mathcal{F}_{\omega,m}}Q^*(s,a) - Q^*(s,a)\right\|_d \\ &\leq \left\|\Pi_{\mathcal{F}_{\omega,m}}\mathcal{T}\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*) - \Pi_{\mathcal{F}_{\omega,m}}\mathcal{T}Q^*(s,a)\right\|_d + \left\|\Pi_{\mathcal{F}_{\omega,m}}Q^*(s,a) - Q^*(s,a)\right\|_d \\ &\overset{(ii)}{\leq} \gamma\left\|\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*) - Q^*(s,a)\right\|_d + \left\|\Pi_{\mathcal{F}_{\omega,m}}Q^*(s,a) - Q^*(s,a)\right\|_d,\end{aligned}$$

where (i) is due to the properties of the fixed point, and (ii) is due to $\Pi_{\mathcal{F}_{\omega,m}}\mathcal{T}$ is γ-contractive on the ∞-norm. This further means that

$$\left\|\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*) - Q^*(s,a)\right\|_d^2 \leq \frac{1}{(1-\gamma)^2}\left\|\Pi_{\mathcal{F}_{\omega,m}}Q^*(s,a) - Q^*(s,a)\right\|_d^2. \tag{26}$$

Plugging (25) and (26) into (25) and using Theorem 3.6, we complete the proof. □

B. Convergence Results of Neural Q-learning

B.1. Neural Q-Learning Algorithm

For neural Q-learning, let us redefine some of the above notations. Let the optimal Q-function be $Q^*(s,a) = \sup_\pi Q^\pi(s,a)$ for all state action pairs (s,a), then the optimal sequence of actions that maximizes the expected cumulative reward will follow $a_t = \text{argmax}_{a' \in \mathcal{A}} Q^*(s_t, a'), t \geq 0$. Therefore, to obtain a near-optimal policy, it is sufficient to find some $\hat{Q}$ that approximates Q^* well. Define the Bellman optimality operator $\mathcal{T}$ as

$$\mathcal{T}Q(s,a) := r(s,a) + \gamma \mathbb{E}\left[\max_{a'} Q(s',a') \mid s' \sim \mathbb{P}(\cdot \mid s,a)\right],$$

for any (s,a). Let us remain the definition of the local linearization function class $\mathcal{F}_{\omega,m}$ introduced in (7). Consider the MSPBE minimization problem with multi-layer neural network approximation:

$$\min_{\boldsymbol{\theta} \in S_\omega} \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\left(Q(\boldsymbol{x};\boldsymbol{\theta}) - \Pi_{\mathcal{F}_{\omega,m}} \mathcal{T}Q(\boldsymbol{x};\boldsymbol{\theta})\right)^2\right].$$

Then the projected neural Q-learning algorithm can be written as follows:

$$\boldsymbol{\theta}^{t+1} = \Pi_{S_\omega}\left(\boldsymbol{\theta}^t - \eta_t \boldsymbol{g}\left(\boldsymbol{\theta}^t\right)\right), \quad \text{with} \quad \boldsymbol{g}(\boldsymbol{\theta}^t) = \Delta\left(s_t, a_t, s_{t+1}; \boldsymbol{\theta}^t\right) \cdot \nabla_{\boldsymbol{\theta}} Q(\phi(s_t,a_t);\boldsymbol{\theta}^t) \tag{27}$$

where

$$\Delta\left(s,a,s';\boldsymbol{\theta}^t\right) = Q(\phi(s,a);\boldsymbol{\theta}^t) - \left(r(s,a) + \gamma \max_{b \in \mathcal{A}} Q\left(\phi\left(s',b\right);\boldsymbol{\theta}^t\right)\right). \tag{28}$$

The algorithm details can be described by Algorithm 2 as follows.

Algorithm 2 Neural Q-Learning with Markovian Sampling

Input: A learning policy π, a discount factor $\gamma \in (0,1)$, a sequence of learning rates $\{\eta_t\}_{t \geq 0}$, a maximum iteration number T, a projection radius $\omega > 0$, a Q network with architecture (4).

Initialization: Generate each entry of $\boldsymbol{W}_l^0$ independently from $\mathcal{N}(0,1)$, for $l = 1, 2, \cdots, L$, and each entry of $\boldsymbol{b}$ independently from Unif$\{-1,+1\}$.

for $t = 0, 1, \cdots, T-1$ **do**

Sample (s_t, a_t, r_t, s_{t+1}) from the learning policy π with $a_t \sim \pi(\cdot|s_t)$.

Compute the TD error Δ_t by (28).

Update $\boldsymbol{\theta}^{t+1}$ by the projected stochastic semi-gradient step (27).

end for

Output: $\boldsymbol{\theta}^T$.

B.2. Global Convergence

Similar to Section 3, we define the function class $\mathcal{F}_{\omega,m}$ as a collection of all local linearization of $Q(\boldsymbol{x};\boldsymbol{\theta})$ at the initial point $\boldsymbol{\theta}^0$:

$$\mathcal{F}_{\omega,m} := \left\{\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}) = Q(\boldsymbol{x};\boldsymbol{\theta}^0) + \left\langle \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x};\boldsymbol{\theta}^0), \boldsymbol{\theta} - \boldsymbol{\theta}^0 \right\rangle,\ \boldsymbol{\theta} \in S_\omega\right\}.$$

Let $\widehat{Q}(\cdot;\boldsymbol{\theta}^*) \in \mathcal{F}_{\omega,m}$, and $\widehat{\Delta}\left(s,a,s';\boldsymbol{\theta}\right)$ has the same structure as $\Delta\left(s,a,s';\boldsymbol{\theta}\right)$ expect that the function $Q(\cdot;\boldsymbol{\theta})$ is replaced by $\widehat{Q}(\cdot;\boldsymbol{\theta})$. The stationary point $\boldsymbol{\theta}^*$ satisfies $\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*) = \Pi_{\mathcal{F}_{\omega,m}} \mathcal{T}\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*)$ for neural Q-learning. We redefine Ξ_β by replacing the Bellman operator $\mathcal{T}^\pi$ in Section 3 with the Bellman optimality operator $\mathcal{T}$. A point $\boldsymbol{\theta}^* \in \Xi_\omega$ if and only if

$$\mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\widehat{\Delta}\left(s,a,s';\boldsymbol{\theta}^*\right)\left\langle \nabla_{\boldsymbol{\theta}} \widehat{Q}\left(\phi(s,a);\boldsymbol{\theta}^*\right), \boldsymbol{\theta} - \boldsymbol{\theta}^*\right\rangle\right] \geq 0.$$

The maximum operator introduced by the Bellman optimality operator significantly sophisticates the analysis. Let us remain the definition of Σ_π in (12), and we define $\Sigma_\pi^*(\boldsymbol{\theta})$ as follows:

$$\mathbb{E}_{\mu,\pi}\left[\nabla_{\boldsymbol{\theta}} Q(\phi(s, a_{\max}^{\boldsymbol{\theta}});\boldsymbol{\theta}^0) \nabla_{\boldsymbol{\theta}} Q(\phi(s, a_{\max}^{\boldsymbol{\theta}});\boldsymbol{\theta}^0)^\top\right], \tag{29}$$

where $a_{\max}^{\boldsymbol{\theta}} = \arg\max_{a \in \mathcal{A}} \left|\left\langle \nabla_{\boldsymbol{\theta}} Q(s,a;\boldsymbol{\theta}^0), \boldsymbol{\theta}\right\rangle\right|$. To facilitate the analysis of neural Q-learning, we further assume the following regularity condition introduced by (Xu & Gu, 2020).

Assumption B.1. $\exists \nu \in (0,1)$ such that $(1-\nu)^2\Sigma_\pi - \gamma^2\Sigma_\pi^*(\boldsymbol{\theta}) \succeq 0$ for any $\boldsymbol{\theta}^0$ and $\boldsymbol{\theta} \in S_\omega$.

The original version of this assumption comes from (Xu & Gu, 2020), which requires a strict positive definite condition: $(1-\nu)^2\Sigma_\pi - \gamma^2\Sigma_\pi^*(\boldsymbol{\theta}) \succ 0$. Under this additional assumption, (Xu & Gu, 2020) obtained an $\widetilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity for neural Q-learning. A similar complexity result was also derived in (Cai et al., 2023) under a similar regularity condition on the learning policy π. At this time, we relax it to the positive semi-definiteness ($\succeq 0$) and provide a convergence result of neural Q-learning. See Theorem B.2.

Theorem B.2. *Suppose Assumptions 3.1, 3.2 and B.1 hold. We set $\omega = \widetilde{C}_1$ and the learning rate $\eta_t = \frac{1}{2\nu\lambda_0(t+1)}$. If the feature map $\|\phi(s,a)\| = 1$ for each state-action pair (s,a) and the network width $m \geq m^*$, then the output $\boldsymbol{\theta}^T$ of neural Q-learning algorithm (i.e. (27)) satisfies*

$$\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^T) - \widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^*)\right)^2 \mid \boldsymbol{\theta}^0\right] \leq \frac{\widetilde{C}_3(\log T + 1)}{\nu^2\lambda_0^2 T} + \frac{\widetilde{C}_4 m^{-1/2}}{\nu\lambda_0} \cdot \sqrt{\log(T/\delta)} + \frac{\widetilde{C}_5\tau^*\left(\log(T/\delta)+1\right)\log T}{\nu^2\lambda_0^2 T},$$

with probability at least $1 - 2\delta - 2L\exp\big(-\widetilde{C}_2 m\big)$, where τ^* is the mixing time of Markov chain in Assumption 3.2, and $\widetilde{C}_1, \cdots, \widetilde{C}_5 > 0$ are universal constants.

Proof. For a little notation abuse, we redefine

$$\begin{aligned}
\mathbf{g}(\boldsymbol{\theta}^t) &= \Delta(s_t, a_t, s_t', \boldsymbol{\theta}^t)\cdot\nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t;\boldsymbol{\theta}^t), \quad \bar{\mathbf{g}}(\boldsymbol{\theta}^t) = \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\mathbf{g}(\boldsymbol{\theta}^t)\right] \\
\mathbf{m}(\boldsymbol{\theta}^t) &= \widehat{\Delta}(s_t, a_t, s_t', \boldsymbol{\theta}^t)\cdot\nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t;\boldsymbol{\theta}^0), \quad \bar{\mathbf{m}}(\boldsymbol{\theta}^t) = \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\mathbf{m}(\boldsymbol{\theta}^t)\right],
\end{aligned}$$

where

$$\begin{aligned}
\Delta(s_t, a_t, s_t', \boldsymbol{\theta}^t) &= Q(\boldsymbol{x}_t;\boldsymbol{\theta}^t) - \left(r(s_t,a_t) + \gamma\max_{b\in\mathcal{A}} Q(\phi(s_{t+1},b);\boldsymbol{\theta}^t)\right), \\
\widehat{\Delta}(s_t, a_t, s_t', \boldsymbol{\theta}^t) &= \widehat{Q}(\boldsymbol{x}_t;\boldsymbol{\theta}^t) - \left(r(s_t,a_t) + \gamma\max_{b\in\mathcal{A}} \widehat{Q}(\phi(s_{t+1},b);\boldsymbol{\theta}^t)\right).
\end{aligned}$$

Let $\Delta_t = \Delta(s_t, a_t, s_t', \boldsymbol{\theta}^t)$ and $\widehat{\Delta}_t = \widehat{\Delta}(s_t, a_t, s_t', \boldsymbol{\theta}^t)$. Similarly, (18) can be derived in neural Q-learning. To estimate the terms $\mathbf{I}_1 \sim \mathbf{I}_4$, we can apply Lemmas D.5 and A.1. However, due to the utilization of the Bellman optimality operator in neural Q-learning, some modifications based on Lemma A.3 are required.

Lemma B.3. *Under Assumption B.1, we have that*

$$\mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\left\langle \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t, \bar{\mathbf{m}}(\boldsymbol{\theta}^t) - \bar{\mathbf{m}}(\boldsymbol{\theta}_*^t)\right\rangle \mid \boldsymbol{\theta}^0\right] \geq \nu\lambda_0 \cdot \|\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\|^2. \tag{30}$$

Proof. To simplify the notation, we denote $\mathbb{E}[\cdot]$ as $\mathbb{E}_{\mu,\pi,\mathbb{P}}[\cdot]$ in the proof of this lemma. Define $\widehat{Q}^{\#}(s;\boldsymbol{\theta}) := \max_{a\in\mathcal{A}}\widehat{Q}(\phi(s,a);\boldsymbol{\theta})$. Then we have

$$\begin{aligned}
&\mathbb{E}\left[\left\langle \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t, \bar{\mathbf{m}}(\boldsymbol{\theta}^t) - \bar{\mathbf{m}}(\boldsymbol{\theta}_*^t)\right\rangle \mid \boldsymbol{\theta}^0\right] \\
=\ &\mathbb{E}\left[\left(\widehat{\Delta}(s_t,a_t,s_{;t},\boldsymbol{\theta}^t) - \widehat{\Delta}(s_t,a_t,s_{;t},\boldsymbol{\theta}_*^t)\right)\left\langle\nabla Q(\boldsymbol{x};\boldsymbol{\theta}^0), \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\right\rangle \mid \boldsymbol{\theta}^0\right] \\
=\ &\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}^t) - \widehat{Q}(\boldsymbol{x};\boldsymbol{\theta}_*^t)\right)\left\langle\nabla Q(\boldsymbol{x};\boldsymbol{\theta}^0), \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\right\rangle \mid \boldsymbol{\theta}^0\right] \\
&-\gamma\mathbb{E}\left[\left(\widehat{Q}^{\#}(s;\boldsymbol{\theta}^t) - \widehat{Q}^{\#}(s;\boldsymbol{\theta}_*^t)\right)\left\langle\nabla Q(\boldsymbol{x};\boldsymbol{\theta}^0), \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\right\rangle \mid \boldsymbol{\theta}^0\right] \qquad (31) \\
=\ &\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}^t) - \widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}_*^t)\right)^2\right] - \gamma\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}^t) - \widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}_*^t)\right)\cdot\left(\widehat{Q}^{\#}(s;\boldsymbol{\theta}^t) - \widehat{Q}^{\#}(s;\boldsymbol{\theta}_*^t)\right)\right].
\end{aligned}$$

For the second term of (31), we consider

$$\begin{aligned}
\mathbb{E}\left[\left(\widehat{Q}^{\#}(s;\boldsymbol{\theta}^t)-\widehat{Q}^{\#}(s;\boldsymbol{\theta}_*^t)\right)^2\right] &\leq \mathbb{E}\left[\max_{a\in\mathcal{A}}\left|\widehat{Q}(s,a;\boldsymbol{\theta}^t)-\widehat{Q}(s,a;\boldsymbol{\theta}_*^t)\right|^2\right] \\
&\overset{(i)}{=} \mathbb{E}\left[\max_{a\in\mathcal{A}}\left|\widehat{Q}(s,a;\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t)\right|^2\right] \\
&= (\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t)^\top\Sigma_\pi^*(\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t) \\
&\overset{(ii)}{\leq} \frac{(1-\nu)^2}{\gamma^2}(\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t)^\top\Sigma_\pi\cdot(\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t) \\
&\overset{(iii)}{=} \frac{(1-\nu)^2}{\gamma^2}\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}^t)-\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}_*^t)\right)^2\right], \qquad (32)
\end{aligned}$$

where (i) and (iii) follow that $\widehat{Q}(\boldsymbol{x};\cdot)$ is linear, and (ii) follows Assumption B.1. Therefore,

$$\begin{aligned}
&\mathbb{E}\left[\left\langle\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t,\bar{\mathbf{m}}(\boldsymbol{\theta}^t)-\bar{\mathbf{m}}(\boldsymbol{\theta}_*^t)\right\rangle\mid\boldsymbol{\theta}^0\right] \\
=\;& \mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}^t)-\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}_*^t)\right)^2\right]-\gamma\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}^t)-\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}_*^t)\right)\cdot\left(\widehat{Q}^{\#}(s;\boldsymbol{\theta}^t)-\widehat{Q}^{\#}(s;\boldsymbol{\theta}_*^t)\right)\right] \\
\geq\;& \mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}^t)-\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}_*^t)^2\right)\right]-\gamma\sqrt{\mathbb{E}\left[\left(\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}^t)-\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}_*^t)\right)^2\right]}\cdot\sqrt{\mathbb{E}\left[\left(\widehat{Q}^{\#}(s;\boldsymbol{\theta}^t)-\widehat{Q}^{\#}(s;\boldsymbol{\theta}_*^t)\right)^2\right]} \\
\overset{(i)}{\geq}\;& \left(1-\gamma\cdot\frac{1-\nu}{\gamma}\right)\mathbb{E}\left[\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}^t)-\widehat{Q}(\boldsymbol{x},\boldsymbol{\theta}_*^t)^2\right] \\
=\;& \nu\cdot(\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t)^\top\Sigma_\pi(\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t) \\
\overset{(ii)}{\geq}\;& \nu\lambda_0\cdot\|\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t\|^2,
\end{aligned}$$

where (i) follows (32), and (ii) follows $\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t=\boldsymbol{\theta}_{\|}^t-\boldsymbol{\theta}^*\in\mathcal{R}(\boldsymbol{\Sigma}_\pi)$ and Lemma A.2, which provides λ_0-strong convexity. □

Now given $\boldsymbol{\theta}^0$, we can deduce that

$$\begin{aligned}
\mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\|\boldsymbol{\theta}^{t+1}-\boldsymbol{\theta}_*^{t+1}\|^2\mid\boldsymbol{\theta}^0\right] &\leq (1-2\eta_t\nu\lambda_0)\mathbb{E}\left[\|\boldsymbol{\theta}^t-\boldsymbol{\theta}_*^t\|^2\mid\boldsymbol{\theta}^0\right]+C_1\eta_t^2 \\
&\quad+2\eta_tC_2m^{-1/2}\sqrt{\log(T/\delta)}+2\eta_tC_3\left(\log(T/\delta)+C_1^2\right)\tau^*\eta_{\max\{0,t-\tau^*\}},
\end{aligned}$$

with probability at least $1-2\delta-2L\exp(-C_4m)$, where $\{C_i>0\}_{i=1,\ldots,4}$ are universal constants in this subsection. Choosing $\eta_t=\frac{1}{2\nu\lambda_0(t+1)}$ can derive the similar results as (25). This suggests that we can utilize the techniques outlined in Section A.2 to finalize the remaining proof of Theorem B.2. As a result, we conclude the proof.

□

C. Details of Section 4

We formally describe the minimax neural Q-learning method in Algorithm 3.

Algorithm 3 Minimax Neural Q-Learning with Gaussian Initialization

Input: A learning policy pair $\pi = (\pi^1, \pi^2)$, a discount factor $\gamma \in (0,1)$, a sequence of learning rates $\{\eta_t\}_{t\geq 0}$, a maximum iteration number T, a projection radius $\omega > 0$, a Q network with architecture (4).

Initialization: Generate each entry of $\boldsymbol{W}_l^0$ independently from $\mathcal{N}(0,1)$, for $l = 1, 2, \cdots, L$, and each entry of $\boldsymbol{b}$ independently from Unif$\{-1, +1\}$.

for $t = 0, 1, \cdots, T-1$ **do**

Sample $(s_t, a_t^1, a_t^2, r_t, s_{t+1})$ from the learning policy pair π with $a_t^1 \sim \pi^1(\cdot|s_t), a_t^2 \sim \pi^2(\cdot|s_t)$.

Compute the TD error Δ_t by (17).

Update $\boldsymbol{\theta}^{t+1}$ by the projected stochastic semi-gradient step (16).

end for

Output: $\boldsymbol{\theta}^T$.

C.1. Proof of Theorem 4.2

The proof of Theorem 4.2 is similar to Sections A.2 and B.2. However, due to the difference in Bellman operators, we still need to make some modifications to Lemma A.3 or Lemma B.3. See Lemma C.1.

Lemma C.1. *Under Assumption 4.1, we have that*

$$\mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\left\langle \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t, \bar{\mathbf{m}}(\boldsymbol{\theta}^t) - \bar{\mathbf{m}}(\boldsymbol{\theta}_*^t)\right\rangle \mid \boldsymbol{\theta}^0\right] \geq \nu\lambda_0 \cdot \|\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\|^2. \tag{33}$$

Proof. To simplify the notation, we denote $\mathbb{E}[\cdot]$ as $\mathbb{E}_{\mu,\pi,\mathbb{P}}[\cdot]$ in the proof of this lemma. Define $\widehat{Q}^{\#}(s;\boldsymbol{\theta}) := \max_{a^1\in\mathcal{A}} \min_{a^2\in\mathcal{A}} \widehat{Q}(\phi(s,a^1,a^2);\boldsymbol{\theta})$. Define the sets $\mathcal{S}_+ = \{s : \widehat{Q}^{\#}(s;\boldsymbol{\theta}^t) > \widehat{Q}^{\#}(s;\boldsymbol{\theta}_*^t)\}$ and $\mathcal{S}_- = \mathcal{S}/\mathcal{S}_+$. For each $s \in \mathcal{S}_+$,

$$\begin{aligned}
\widehat{Q}^{\#}(s;\boldsymbol{\theta}^t) - \widehat{Q}^{\#}(s;\boldsymbol{\theta}_*^t) &= \left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}^t},a^2_{\boldsymbol{\theta}^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}^t\right\rangle - \left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}_*^t},a^2_{\boldsymbol{\theta}_*^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}_*^t\right\rangle \\
&= \left(\left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}^t},a^2_{\boldsymbol{\theta}^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}^t\right\rangle - \left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}^t},a^2_{\boldsymbol{\theta}_*^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}^t\right\rangle\right) - \\
&\quad \left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}^t},a^2_{\boldsymbol{\theta}_*^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\right\rangle - \\
&\quad \left(\left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}^t},a^2_{\boldsymbol{\theta}_*^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}_*^t\right\rangle - \left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}_*^t},a^2_{\boldsymbol{\theta}_*^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}_*^t\right\rangle\right) \\
&\leq \left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}^t},a^2_{\boldsymbol{\theta}_*^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\right\rangle
\end{aligned}$$

and

$$\begin{aligned}
\widehat{Q}^{\#}(s;\boldsymbol{\theta}^t) - \widehat{Q}^{\#}(s;\boldsymbol{\theta}_*^t) &= \left(\left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}^t},a^2_{\boldsymbol{\theta}^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}^t\right\rangle - \left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}_*^t},a^2_{\boldsymbol{\theta}^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}^t\right\rangle\right) - \\
&\quad \left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}_*^t},a^2_{\boldsymbol{\theta}^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\right\rangle - \\
&\quad \left(\left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}_*^t},a^2_{\boldsymbol{\theta}^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}_*^t\right\rangle - \left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}_*^t},a^2_{\boldsymbol{\theta}_*^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}_*^t\right\rangle\right) \\
&\geq \left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}_*^t},a^2_{\boldsymbol{\theta}^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\right\rangle.
\end{aligned}$$

In the same way, for each $s \in \mathcal{S}_-$,

$$\begin{aligned}
\left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}^t},a^2_{\boldsymbol{\theta}_*^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\right\rangle &\leq \widehat{Q}^{\#}(s;\boldsymbol{\theta}^t) - \widehat{Q}^{\#}(s;\boldsymbol{\theta}_*^t) \\
&\leq \left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}_*^t},a^2_{\boldsymbol{\theta}^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\right\rangle.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\left|\widehat{Q}^{\#}(s;\boldsymbol{\theta}^t) - \widehat{Q}^{\#}(s;\boldsymbol{\theta}_*^t)\right| \leq \max\Big\{&\left|\left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}^t},a^2_{\boldsymbol{\theta}_*^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\right\rangle\right|, \\
&\left|\left\langle \nabla_{\boldsymbol{\theta}} Q\left(\phi(s,a^1_{\boldsymbol{\theta}_*^t},a^2_{\boldsymbol{\theta}^t});\boldsymbol{\theta}^0\right),\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^t\right\rangle\right|\Big\}.
\end{aligned} \tag{34}$$

By Assumption 4.1, we compute

$$\begin{aligned}\mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\left|\widehat{Q}^{\#}(s;\boldsymbol{\theta}^t)-\widehat{Q}^{\#}(s;\boldsymbol{\theta}^t_*)\right|^2\right] &\overset{(i)}{\leq} \left(\boldsymbol{\theta}^t-\boldsymbol{\theta}^t_*\right)^\top \Sigma^*_\pi(\boldsymbol{\theta}^t,\boldsymbol{\theta}^t_*)\left(\boldsymbol{\theta}^t-\boldsymbol{\theta}^t_*\right)\\ &\overset{(ii)}{\leq} \frac{(1-\nu)^2}{\gamma^2}\cdot\left(\boldsymbol{\theta}^t-\boldsymbol{\theta}^t_*\right)^\top\Sigma_\pi\left(\boldsymbol{\theta}^t-\boldsymbol{\theta}^t_*\right) \qquad (35)\\ &= \frac{(1-\nu)^2}{\gamma^2}\cdot\mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\left|\widehat{Q}(s,a^1,a^2;\boldsymbol{\theta}^t)-\widehat{Q}(s,a^1,a^2;\boldsymbol{\theta}^t_*)\right|^2\right],\end{aligned}$$

where (i) is due to (34), and (ii) is due to Assumption 4.1. Similar to Lemma B.3, we can also obtain (31) in this lemma. By substituting (35) into (31), the proof can be completed.

□

Now we are ready to prove Theorem 4.2. For a little notation abuse, we redefine

$$\begin{aligned}\mathbf{g}(\boldsymbol{\theta}^t) &= \Delta(s_t,a_t^1,a_t^2,s_t',\boldsymbol{\theta}^t)\cdot\nabla_{\boldsymbol{\theta}}Q(\boldsymbol{x}_t;\boldsymbol{\theta}^t), \quad \bar{\mathbf{g}}(\boldsymbol{\theta}^t) = \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\mathbf{g}(\boldsymbol{\theta}^t)\right]\\ \mathbf{m}(\boldsymbol{\theta}^t) &= \widehat{\Delta}(s_t,a_t^1,a_t^2,s_t',\boldsymbol{\theta}^t)\cdot\nabla_{\boldsymbol{\theta}}Q(\boldsymbol{x}_t;\boldsymbol{\theta}^0), \quad \bar{\mathbf{m}}(\boldsymbol{\theta}^t) = \mathbb{E}_{\mu,\pi,\mathbb{P}}\left[\mathbf{m}(\boldsymbol{\theta}^t)\right],\end{aligned}$$

where

$$\begin{aligned}\Delta(s_t,a_t^1,a_t^2,s_t',\boldsymbol{\theta}^t) &= Q(\boldsymbol{x}_t;\boldsymbol{\theta}^t)-\left(r(s_t,a_t)+\gamma\max_{b^1\in\mathcal{A}}\min_{b^2\in\mathcal{A}}Q(\phi(s_{t+1},b^1,b^2);\boldsymbol{\theta}^t)\right),\\ \widehat{\Delta}(s_t,a_t^1,a_t^2,s_t',\boldsymbol{\theta}^t) &= \widehat{Q}(\boldsymbol{x}_t;\boldsymbol{\theta}^t)-\left(r(s_t,a_t)+\gamma\max_{b^1\in\mathcal{A}}\min_{b^2\in\mathcal{A}}\widehat{Q}(\phi(s_{t+1},b^1,b^2);\boldsymbol{\theta}^t)\right).\end{aligned}$$

Let $\Delta_t=\Delta(s_t,a_t^1,a_t^2,s_t',\boldsymbol{\theta}^t)$ and $\widehat{\Delta}_t=\widehat{\Delta}(s_t,a_t^1,a_t^2,s_t',\boldsymbol{\theta}^t)$. After redefining the corresponding notation, we can similarly derive (18) and adopt the associated lemmas. Due to the introduction of the additional Assumption 4.1, we provide Lemma C.1, ensuring that terms $\mathbf{I}_1\sim\mathbf{I}_4$ can be estimated. The remainder of the proof is entirely analogous to Sections A.2 and B.2. Thus, we conclude the proof.

D. Supporting Lemmas for Multi-layer Neural Network

Recalling the definition of the parameterized Q-function, we present the following lemmas related to neural network functions, which play a crucial role in illustrating the main results of our paper. $\{\tau_i>0\}_{i=1,\ldots,10}$ mentioned below are universal constants.

Lemma D.1. *For any $t\in\{1,2,\cdots,T\}$, we have*

$$\|\boldsymbol{\theta}^t\|\leq\tau_1\sqrt{m},\quad w.p.1-L\exp\left(-\tau_2 m\right).$$

Proof. By Lemma G.2 in (Du et al., 2019), $\|\boldsymbol{\theta}^0\|\leq\mathcal{O}\left(\sqrt{m}\right)$ with probability at least $1-L\exp\left(-\tau_2 m\right)$. Therefore,

$$\begin{aligned}\|\boldsymbol{\theta}^t\|_2 &\leq \|\boldsymbol{\theta}^t-\boldsymbol{\theta}^0\|+\|\boldsymbol{\theta}^0\|\\ &\leq \omega+\|\boldsymbol{\theta}^0\|\\ &\leq \mathcal{O}\left(\sqrt{m}\right).\end{aligned}$$

□

Lemma D.2. *For any $l\in\{1,2,\cdots,L\}$, we have*

$$\|\boldsymbol{x}^{(l)}\|\leq\tau_3\sqrt{m},\quad \textit{and}\quad \|\nabla_{\boldsymbol{\theta}}Q(\boldsymbol{x};\boldsymbol{\theta})\|\leq\tau_4,\quad w.p.\ 1-L\exp\left(-\tau_2 m\right).$$

Lemma D.2 has been proved by (Tian et al., 2022) (Lemmas A.6∼A.10).

Lemma D.3. *For any $t \in \{1, 2, \cdots, T\}$ and $\boldsymbol{\theta} \in S_\omega$, we have*

$$|Q(\boldsymbol{x}_t; \boldsymbol{\theta})| \leq \tau_5 \sqrt{\log(T/\delta)}, \quad w.p.\ 1 - \delta - L\exp(-\tau_2 m).$$

Proof. By Lemma D.2, we have $\frac{1}{\sqrt{m}}\|\boldsymbol{x}^{(L)}\| \leq \tau_3,\ w.p.\ 1 - L\exp(-\tau_2 m)$. Recall the definition of the parameterized Q-function:

$$Q(\boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{\sqrt{m}} \boldsymbol{b}^\top \boldsymbol{x}^{(L)},$$

where each element of $\boldsymbol{b}$ is generated from a uniform distribution over $\{-1, +1\}$. For each $\boldsymbol{x}$, by Hoeffding inequality, we have

$$\mathbb{P}\left(\left|\left\langle \frac{1}{m}\boldsymbol{x}^{(L)}, \boldsymbol{b}\right\rangle\right| \geq t\right) \leq 2\exp\left(-\frac{2t^2}{4\|\frac{1}{m}\boldsymbol{x}^{(L)}\|^2}\right) \overset{(i)}{\leq} 2\exp\left(-\frac{t^2}{2\tau_3^2}\right),$$

where (i) follows Lemma D.2. Substituting $\frac{\delta}{T} = 2\exp\left(-\frac{t^2}{2\tau_3^2}\right)$, we get

$$\mathbb{P}\left(\left|\left\langle \frac{1}{m}\boldsymbol{x}^{(L)}, \boldsymbol{b}\right\rangle\right| \geq \tau_3\sqrt{2\log(T/\delta)}\right) \leq \frac{\delta}{T}.$$

Now, by the union bound, if we set $\tau_5 = \tau_3\sqrt{2}$, then

$$\mathbb{P}\left(\max_{t\in[T]} |Q(\boldsymbol{x}_t; \boldsymbol{\theta})| \geq \tau_5\sqrt{\log(T/\delta)}\right) \leq \sum_{t=1}^{T} \mathbb{P}\left(|Q(\boldsymbol{x}_t; \boldsymbol{\theta})| \geq \tau_5\sqrt{\log(T/\delta)}\right) \leq \delta,$$

which completes the proof. □

Lemma D.4. *Denote $\nabla^2_{\boldsymbol{\theta}} Q(\boldsymbol{x}; \boldsymbol{\theta})$ as the Hessian matrix of $Q(\boldsymbol{x}; \boldsymbol{\theta})$. Then for all $\boldsymbol{x}, \boldsymbol{\theta} \in S_\omega$, we have $w.p.\ 1 - \delta$ that*

$$\|\nabla^2_{\boldsymbol{\theta}} Q(\boldsymbol{x}; \boldsymbol{\theta})\|_2 \leq \tau_6 m^{-\frac{1}{2}}$$

and

$$\left|Q(\boldsymbol{x}; \boldsymbol{\theta}) - \widehat{Q}(\boldsymbol{x}; \boldsymbol{\theta})\right| \leq \tau_7 m^{-\frac{1}{2}}.$$

Proof. The first inequality in Lemma D.4 has been proved by (Liu et al., 2020b) (Theorem 3.2), which implies that $Q(\boldsymbol{x}; \boldsymbol{\theta})$ is $\mathcal{O}(m^{-\frac{1}{2}})$-smoothness w.r.t. θ. Therefore,

$$\left|Q(\boldsymbol{x}; \boldsymbol{\theta}) - \widehat{Q}(\boldsymbol{x}; \boldsymbol{\theta})\right| = \left|Q(\boldsymbol{x}; \boldsymbol{\theta}) - Q(\boldsymbol{x}; \boldsymbol{\theta}^0) - \left\langle \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}; \boldsymbol{\theta}^0), \boldsymbol{\theta}^0 - \boldsymbol{\theta}\right\rangle\right| = \mathcal{O}(m^{-\frac{1}{2}}).$$

□

Lemma D.5. *Let $\boldsymbol{\theta} \in S_\omega$ with the radius satisfying $\omega = \mathcal{O}(1)$. Then for all $\|\boldsymbol{x}\|_2 = 1$ and $\boldsymbol{\theta}^t_* \in S_{2\omega}$ in the neural temporal difference learning algorithm 1, it holds that*

$$\left|\left\langle \mathbf{g}\left(\boldsymbol{\theta}^t\right) - \mathbf{m}\left(\boldsymbol{\theta}^t\right), \boldsymbol{\theta}^t - \boldsymbol{\theta}^{t+1}_*\right\rangle\right| \qu\leq \quad \left(\tau_8\tau_9\omega m^{-\frac{1}{2}}\sqrt{\log(T/\delta)} + 2\tau_4\tau_8 m^{-\frac{1}{2}}\right)\left\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^{t+1}_*\right\|$$

with probability at least $1 - 2\delta - 2L\exp(-\tau_2 m)$ over the randomness of the initial point, and $\left\|\mathbf{g}\left(\boldsymbol{\theta}^t\right)\right\|_2 \leq \tau_{10}\sqrt{\log(T/\delta)}$ holds with probability at least $1 - \delta - 2L\exp(-\tau_2 m)$.

Proof. Note that

$$\begin{aligned}
\left\|\mathbf{g}\left(\boldsymbol{\theta}^t\right) - \mathbf{m}\left(\boldsymbol{\theta}^t\right)\right\| &= \left\|\Delta(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \boldsymbol{\theta}^t)\cdot\nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^t) - \widehat{\Delta}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \boldsymbol{\theta}^t)\cdot\nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^0)\right\| \\
&\leq \left\|\Delta(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \boldsymbol{\theta}^t)\cdot\left(\nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^t) - \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^0)\right)\right\| \\
&\quad + \left\|\left(\Delta(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \boldsymbol{\theta}^t) - \widehat{\Delta}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \boldsymbol{\theta}^t)\right)\cdot\nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^0)\right\|,
\end{aligned} \tag{36}$$

where

$$\begin{aligned}\Delta(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \boldsymbol{\theta}^t) &= Q(\boldsymbol{x}_t; \boldsymbol{\theta}^t) - \left(r(s_t, a_t) + \gamma Q(\boldsymbol{x}_{t+1}; \boldsymbol{\theta}^t)\right), \\ \widehat{\Delta}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \boldsymbol{\theta}^t) &= \widehat{Q}(\boldsymbol{x}_t; \boldsymbol{\theta}^t) - \left(r(s_t, a_t) + \gamma \widehat{Q}(\boldsymbol{x}_{t+1}; \boldsymbol{\theta}^t)\right).\end{aligned}$$

For the first term in (36), by Lemma D.3, we have $w.p.\ 1-\delta-L\exp\left(-\tau_2 m\right)$ that

$$\left|\Delta(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \boldsymbol{\theta}^t)\right| \leq \tau_8 \sqrt{\log(T/\delta)}.$$

By Lemma D.4, we get that

$$\left\|\nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^t) - \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^0)\right\| \leq \|\nabla^2_{\boldsymbol{\theta}} Q(\boldsymbol{x_t}; \boldsymbol{\theta_t})\|_2 \cdot \|\boldsymbol{\theta}^t - \boldsymbol{\theta}^0\| \leq \tau_9 \omega m^{-\frac{1}{2}}.$$

Therefore,

$$\left\|\Delta(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \boldsymbol{\theta}^t) \cdot \left(\nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^t) - \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^0)\right)\right\| \leq \tau_8 \tau_9 \omega m^{-\frac{1}{2}} \sqrt{\log(T/\delta)}. \tag{37}$$

For the second term in (36), we decompose it into

$$\begin{aligned}&\left\|\left(\Delta(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \boldsymbol{\theta}^t) - \widehat{\Delta}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \boldsymbol{\theta}^t)\right) \cdot \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^0)\right\| \\ \leq\ &\left\|\left(Q(\boldsymbol{x}, \boldsymbol{\theta}^t) - \widehat{Q}(\boldsymbol{x}, \boldsymbol{\theta}^t)\right) \cdot \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^0)\right\| + \left\|\left(Q(\boldsymbol{x}_{t+1}; \boldsymbol{\theta}^t) - \widehat{Q}(\boldsymbol{x}_{t+1}; \boldsymbol{\theta}^t)\right) \cdot \nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^0)\right\| \\ \leq\ &\left|Q(\boldsymbol{x}, \boldsymbol{\theta}^t) - \widehat{Q}(\boldsymbol{x}, \boldsymbol{\theta}^t)\right| \cdot \left\|\nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^0)\right\| + \left|Q(\boldsymbol{x}_{t+1}; \boldsymbol{\theta}^t) - \widehat{Q}(\boldsymbol{x}_{t+1}; \boldsymbol{\theta}^t)\right| \cdot \left\|\nabla_{\boldsymbol{\theta}} Q(\boldsymbol{x}_t; \boldsymbol{\theta}^0)\right\| \\ \overset{(i)}{\leq}\ &2\tau_4 \tau_8 m^{-\frac{1}{2}},\end{aligned} \tag{38}$$

with probability at least $w.p.\ 1-\delta-L\exp\left(-\tau_2 m\right)$, where (i) is due to Lemmas D.3 and D.4. Plugging (37) and (38) into (36) yields that

$$\begin{aligned}\left|\left\langle \mathbf{g}\left(\boldsymbol{\theta}^t\right) - \mathbf{m}\left(\boldsymbol{\theta}^t\right), \boldsymbol{\theta}^t - \boldsymbol{\theta}_*^{t+1}\right\rangle\right| &\leq \left\|\mathbf{g}\left(\boldsymbol{\theta}^t\right) - \mathbf{m}\left(\boldsymbol{\theta}^t\right)\right\| \cdot \left\|\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^{t+1}\right\| \\ &\leq \left(\tau_8 \tau_9 \omega m^{-\frac{1}{2}} \sqrt{\log(T/\delta)} + 2\tau_4 \tau_8 m^{-\frac{1}{2}}\right) \left\|\boldsymbol{\theta}^t - \boldsymbol{\theta}_*^{t+1}\right\|.\end{aligned}$$

Thus we complete the proof. □

Lemma D.5 provides the upper bounds on $\mathbf{I}_1$ and $\mathbf{I}_2$ in Section A.2. As discussed in Section 3.1, for a finite MDP, the Gram matrix of the L-layer neural network function is positive definite and has a minimum eigenvalue of $\mathcal{O}(1)$ when the network width is sufficiently large. This, in fact, serves as an upper bound for m^* in Assumption 3.1. Further details are provided in Remark D.6.

Remark D.6. In this special case, we assume that both state space and action space are finite. Let $|\mathcal{S}|$ and $|\mathcal{A}|$ represent the dimensions of the state space and action space, respectively. For simplicity of notation, we view $\mathbf{Q}(\boldsymbol{\theta})$ as an $|\mathcal{S}||\mathcal{A}| \times 1$ column vector, with (s, a) being a multi-index arranged in the lexicographical order. Let $d \sim \mu \times \pi$ and $\mathbf{D} = \mathrm{diag}(d)$ be an $|\mathcal{S}||\mathcal{A}|$-dimensional diagonal matrix, whose (s, a)-th diagonal entry is $d(s, a)$, and the order of (s, a) in $\mathbf{D}$ is the same as $\mathbf{Q}(\boldsymbol{\theta})$. Denote $\mathbf{J}$ as the Jacobian matrix of $\mathbf{Q}(\boldsymbol{\theta}^0)$ and $\mathbf{J_D} = \mathbf{D}^{\frac{1}{2}}\mathbf{J}$. Thus we can rewrite $\Sigma_\pi = \mathbf{J}_\mathbf{D}^\top \mathbf{J_D}$. Notice that Σ_π is different from the Gram matrix Jacot et al. (2018); Du et al. (2018; 2019); Cao & Gu (2019); Allen-Zhu et al. (2019b) in deep neural network. To derive the μ-weighted Gram matrix $\mathrm{Gram}(\boldsymbol{\theta}^0)$, we provide the following definition.

Definition D.7. (Du et al. (2019); Cao & Gu (2019); Allen-Zhu et al. (2019b;a), Neural Tangent Kernel Matrix). For any $i, j \in [|\mathcal{S}||\mathcal{A}|]$, define

$$\begin{aligned}&\widetilde{\boldsymbol{\Theta}}_{i,j}^{(1)} = \boldsymbol{\Sigma}_{i,j}^{(1)} = \langle \widehat{\boldsymbol{x}}_i, \widehat{\boldsymbol{x}}_j \rangle, \quad \mathbf{A}_{ij}^{(l)} = \begin{pmatrix} \boldsymbol{\Sigma}_{i,i}^{(l)} & \boldsymbol{\Sigma}_{i,j}^{(l)} \\ \boldsymbol{\Sigma}_{i,j}^{(l)} & \boldsymbol{\Sigma}_{j,j}^{(l)} \end{pmatrix}, \\ &\boldsymbol{\Sigma}_{i,j}^{(l+1)} = 2 \cdot \mathbb{E}_{(u,v)\sim N\left(\mathbf{0}, \mathbf{A}_{ij}^{(l)}\right)}[\sigma(u)\sigma(v)], \\ &\widetilde{\boldsymbol{\Theta}}_{i,j}^{(l+1)} = \widetilde{\boldsymbol{\Theta}}_{i,j}^{(l)} \cdot 2 \cdot \mathbb{E}_{(u,v)\sim N\left(\mathbf{0}, \mathbf{A}_{ij}^{(l)}\right)}[\sigma'(u)\sigma'(v)] + \boldsymbol{\Sigma}_{i,j}^{(l+1)},\end{aligned}$$

where $\widehat{\boldsymbol{x}} = \sqrt{d(s,a)}\phi(s,a)$. Then we call $\boldsymbol{\Theta}^{(L)} = \left[\left(\widetilde{\boldsymbol{\Theta}}_{i,j}^{(L)} + \boldsymbol{\Sigma}_{i,j}^{(L)}\right)/2\right]_{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}||\mathcal{A}|}$ the μ-weighted neural tangent kernel matrix of an L-layer ReLU network on μ-weighted state-action pairs $\widehat{\boldsymbol{x}}_1, \ldots, \widehat{\boldsymbol{x}}_{|\mathcal{S}||\mathcal{A}|}$.

There is a large body of work Jacot et al. (2018); Du et al. (2018; 2019); Cao & Gu (2019); Allen-Zhu et al. (2019b) exploring the positive definiteness of Gram($\boldsymbol{\theta}^0$) in the literature. Suppose that for all pairs $(s,a),(s',a')$, $\|\phi(s,a)\|=1, d(s,a)>0$ and $\phi(s,a)\nparallel\phi(s',a')$. The results of Theorem 1 and Proposition 2 in Jacot et al. (2018) shows that for an L-layer neural network with Gaussian initialization parameters,

$$\left\langle \nabla_{\boldsymbol{\theta}} Q(\widehat{\boldsymbol{x}}_i;\boldsymbol{\theta}^0), \nabla_{\boldsymbol{\theta}} Q(\widehat{\boldsymbol{x}}_j,\boldsymbol{\theta}^0)\right\rangle \to \boldsymbol{\Theta}^{(L)}_{i,j}, \quad \text{as } m\to\infty.$$

That is, under the NTK regime, the μ-weighted Gram matrix Gram($\boldsymbol{\theta}^0$) $=\mathbf{J_D}\mathbf{J_D}^\top$ converges to $\boldsymbol{\Theta}^{(L)}$ when m is sufficiently large. Let $\lambda_{\min}\left(\boldsymbol{\Theta}^{(L)}\right)=2\lambda'=\mathcal{O}(1)$. For any $\delta\in(0,1)$, there exists $m^*=\text{Poly}(|\mathcal{S}||\mathcal{A}|,L,\delta,\lambda')$ such that if $m\geq m^*$, we have

$$\lambda_{\min}\left(\text{Gram}(\boldsymbol{\theta}^0)\right)\geq\lambda' \quad w.p.\ 1-\delta.$$

This signifies that if the network width $m\geq m^*$, then we have $\lambda_0\geq\lambda_{\min}\left(\text{Gram}(\boldsymbol{\theta}^0)\right)\geq\lambda'$, thereby substantiating our claim.

E. Additional Notes on the Experiments in Section 5

In this section, we further discuss the experimental setup introduced in Section 5. As mentioned in Section 5, our experiments mainly test the following two aspects: (1) how does the network width m affects the final error of the algorithm (first two subfigures in Figure 1); (2) the minimum nonzero singular value in Assumption 3.3 (latter two subfigures in Figure 1).

For point (1), we first generate 2000 samples according to a given policy π to imitate the Markov process of Algorithm 1. A two-layer neural network with ELU activation is introduced, and the parameters are initialized using Algorithm 1. We set the initial learning rate at 0.001 with linear decay (per epoch) and a batch size of 100. Notably, as the parameter m increases, the TD algorithm demonstrates smaller final TD errors.

For point (2), our experiments are based on three main points. First, note that the norm of feature map and parameter random initialization will affect the scaling of the gradient norm w.r.t. $Q(s,a;\boldsymbol{\theta})$. Thus we employ the ratio $r=\sigma_{\max}/\sigma_{\min}$ to characterize the minimum non-zero singular value in order to eliminate the impact of numerical scaling. Second, we set varying network widths to verify the existence of λ_0. Finally, it's tough to directly obtain the joint distribution of (s,a) with a fixed learning policy. However, we have that $\left\|\frac{1}{N}\sum_{(s,a)\sim\mu\times\pi}\nabla_{\boldsymbol{\theta}}Q(s,a;\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}Q(s,a;\boldsymbol{\theta})^\top-\Sigma_\pi\right\|_2=\mathcal{O}(1/N)$. To avoid the effects of sampling randomness, we estimate Σ_π from different samples. The experiments demonstrate that the minimum non-zero singular value λ_0 converges to a constant as m increases.