A Theoretical Analysis of Deep Q-Learning

Zhuoran Yang* Yuchen Xie† Zhaoran Wang†

May 30, 2019

Abstract

Despite the great empirical success of deep reinforcement learning, its theoretical foundation is less well understood. In this work, we make the first attempt to theoretically understand the deep Q-network (DQN) algorithm (Mnih et al., 2015) from both algorithmic and statistical perspectives. In specific, we focus on a slight simplification of DQN that fully captures its key features. Under mild assumptions, we establish the algorithmic and statistical rates of convergence for the action-value functions of the iterative policy sequence obtained by DQN. In particular, the statistical error characterizes the bias and variance that arise from approximating the action-value function using deep neural network, while the algorithmic error converges to zero at a geometric rate. As a byproduct, our analysis provides justifications for the techniques of experience replay and target network, which are crucial to the empirical success of DQN. Furthermore, as a simple extension of DQN, we propose the Minimax-DQN algorithm for zero-sum Markov game with two players. Borrowing the analysis of DQN, we also quantify the difference between the policies obtained by Minimax-DQN and the Nash equilibrium of the Markov game in terms of both the algorithmic and statistical rates of convergence.

1 Introduction

Reinforcement learning (RL) attacks the multi-stage decision-making problems by interacting with the environment and learning from the experiences. With the breakthrough in deep learning, deep reinforcement learning (DRL) demonstrates tremendous success in solving highly challenging problems, such as the game of Go (Silver et al., 2016, 2017), robotics (Kober and Peters, 2012), and dialogue systems (Chen et al., 2017). In DRL, the value or policy functions are often represented as deep neural networks and the related deep learning techniques can be readily applied. For example, deep Q-network (DQN) (Mnih et al., 2015), asynchronous advantage actor-critic (A3C) (Mnih et al., 2016), and and trust region policy optimization (TRPO) (Schulman et al., 2015) demonstrate superhuman performance in various applications and become standard algorithms for artificial intelligence.

[*] Department of Operations Research and Financial Engineering, Princeton University

[†] Department of Industrial Engineering and Management Sciences, Northwestern University

Despite its great empirical success, there exists a gap between the theory and practice of DRL. In particular, most existing theoretical work on reinforcement learning focuses on the tabular case where the state and action spaces are finite, or the case where the value function is linear. Under these restrictive settings, the algorithmic and statistical perspectives of reinforcement learning are well-understood via the tools developed for convex optimization and linear regression. However, in presence of nonlinear function approximators such as deep neural network, the theoretical analysis of reinforcement learning becomes intractable as it involves solving a highly nonconvex statistical optimization problem.

To bridge such a gap in DRL, we make the first attempt to theoretically understand DQN, which can be cast as an extension of the classical Q-learning algorithm (Watkins and Dayan, 1992) that uses deep neural network to approximate the action-value function. Although the algorithmic and statistical properties of the classical Q-learning algorithm are well-studied, theoretical analysis of DQN is highly challenging due to its differences in the following two aspects.

First, in online gradient-based temporal-difference reinforcement learning algorithms, approximating the action-value function often leads to instability. Baird (1995) proves that this is the case even with linear function approximation. The key technique to achieve stability in DQN is experience replay (Lin, 1992; Mnih et al., 2015). In specific, a replay memory is used to store the trajectory of the Markov decision process (MDP). At each iteration of DQN, a mini-batch of states, actions, rewards, and next states are sampled from the replay memory as observations to train the Q-network, which approximates the action-value function. The intuition behind experience replay is to achieve stability by breaking the temporal dependency among the observations used in the training of the deep neural network.

Second, in addition to the aforementioned Q-network, DQN uses another neural network named target network to obtain an unbiased estimator of the mean-squared Bellman error used in training the Q-network. The target network is synchronized with the Q-network after each period of iterations, which leads to a coupling between the two networks. Moreover, even if we fix the target network and focus on updating the Q-network, the subproblem of training a neural network still remains less well-understood in theory.

In this paper, we focus on a slight simplification of DQN, which is amenable to theoretical analysis while fully capturing the above two aspects. In specific, we simplify the technique of experience replay with an independence assumption, and focus on deep neural networks with rectified linear units (ReLU) (Nair and Hinton, 2010) and large batch size. Under this setting, DQN is reduced to the neural fitted Q-iteration (FQI) algorithm (Riedmiller, 2005) and the technique of target network can be cast as the value iteration. More importantly, by adapting the approximation results for ReLU networks to the analysis of Bellman operator, we establish the algorithmic and statistical rates of convergence for the iterative policy sequence obtained by DQN. As shown in the main results in §3, the statistical error characterizes the bias and variance that arise from approximating the action-value function using neural network, while the algorithmic error geometrically decays to zero as the number of iteration goes to infinity.

Furthermore, we extend the DQN to two-player zero-sum Markov games (Shapley, 1953). The proposed algorithm, named Minimax-DQN, can be viewed as a combination of the Minimax-Q learning algorithm for tabular zero-sum Markov games (Littman, 1994) and deep neural networks for function approximation. Compared with DQN, the main difference lies in the approaches to compute the target values. In DQN, the target is computed via maximization over the action space. In contrast, the target obtained computed by solving the Nash equilibrium of a zero-sum matrix game in Minimax-DQN, which can be efficiently attained via linear programming. Despite such a difference, both these two methods can be viewed as applying Bellman operator to the Q-network. Thus, borrowing the analysis of DQN, we also establish theoretical results for Minimax-DQN. Specifically, we quantify the suboptimality of policy returned by the algorithm by the difference between the action-value functions associated with this policy and with the Nash equilibrium policy of the Markov game. For this notion of suboptimality, we establish the both algorithmic and statistical rates of convergence, which implies that the action-value function converges to the optimal counterpart up to an unimprovable statistical error in geometric rate.

Our contribution is three-fold. First, we establish the algorithmic and statistical errors of the neural FQI algorithm, which can be viewed as a slight simplification of DQN. Under mild assumptions, our results show that the proposed algorithm obtains a sequence of Q-networks that geometrically converges to the optimal action-value function up to an intrinsic statistical error induced by the approximation bias of ReLU network and finite sample size. Second, as a byproduct, our analysis justifies the techniques of experience replay and target network used in DQN, where the latter can be viewed as the value iteration. Third, we propose the Minimax-DQN algorithm that extends DQN to two-player zero-sum Markov games. Borrowing the analysis for DQN, we establish the algorithmic and statistical convergence rates of the action-value functions associated with the sequence of policies returned by the Minimax-DQN algorithm.

1.1 Related Work

There is a huge literature in deep reinforcement learning, where algorithms are based on Q-learning or policy gradient (Sutton et al., 2000). We refer the reader to Arulkumaran et al. (2017) for a survey of the recent developments of DRL. In addition, the DQN algorithm is first proposed in Mnih et al. (2015), which applies DQN to Artari 2600 games (Bellemare et al., 2013). The extensions of DQN include double DQN (van Hasselt et al., 2016), dueling DQN (Wang et al., 2016), deep recurrent Q-network (Hausknecht and Stone, 2015), and asynchronous DQN (Mnih et al., 2016). All of these algorithms are corroborated only by numerical experiments, without theoretical guarantees. Moreover, these algorithms not only inherit the tricks of experience replay and the target network proposed in the original DQN, but develop even more tricks to enhance the performance. Furthermore, recent work such as Schaul et al. (2016); Liu and Zou (2017); Zhang and Sutton (2017) study the effect of experience replay and propose various modifications.

In addition, our work is closely related to the literature on batch reinforcement learning (Lange et al., 2012), where the goal is to estimate the value function given transition data. These problems are usually formulated into least-squares regression, for which various algorithms are proposed with finite-sample analysis. However, most existing work focus on the settings where the value function are approximated by linear functions. See Bradtke and Barto (1996); Boyan (2002); Lagoudakis and Parr (2003); Lazaric et al. (2016); Farahmand et al. (2010); Lazaric et al. (2012);

Tagorti and Scherrer (2015) and the references therein for results of the least-squares policy iteration (LSPI) and Bellman residue minimization (BRM) algorithms. Beyond linear function approximation, a recent work Farahmand et al. (2016) study the performance of LSPI and BRM when the value function belongs to a reproducing kernel Hilbert space. However, we study the fitted Q-iteration algorithm, which is a batch RL counterpart of DQN. The fitted Q-iteration algorithm is proposed in Ernst et al. (2005), and Riedmiller (2005) propose the neural FQI algorithm. A finite-sample bound for FQI is established in Munos and Szepesvári (2008) for a large class of regressors. However, their results are not applicable to ReLU networks due to the huge capacity of deep neural networks. Furthermore, various extensions of FQI are studied in Antos et al. (2008a); Farahmand et al. (2009); Tosatto et al. (2017); Geist et al. (2019) to handle continuous actions space, ensemble learning, and entropy regularization.

Furthermore, our work is also related to works that apply reinforcement learning to zero-sum Markov games. The Minimax-Q learning is proposed by Littman (1994), which is an online algorithm that is an extension Q-learning. Subsequently, for Markov games, various online algorithms are also proposed with theoretical guarantees. These work consider either the tabular case or linear function approximation. See, e.g., Bowling (2001); Conitzer and Sandholm (2007); Prasad et al. (2015); Wei et al. (2017); Pérolat et al. (2018); Srinivasan et al. (2018); Wei et al. (2017) and the references therein. In addition, batch reinforcement learning is also applied to zero-sum Markov games by Lagoudakis and Parr (2002); Perolat et al. (2015); Pérolat et al. (2016a,b); Zhang et al. (2018), which are closely related to our work. All of these works consider either linear function approximation or a general function class with bounded pseudo-dimension (Anthony and Bartlett, 2009). However, there results cannot directly imply finite-sample bounds for Minimax-DQN due to the huge capacity of deep neural networks.

Finally, our work is also related a line of research on the model capacity of ReLU deep neural networks, which leads to understanding the generalization property of deep learning (Mohri et al., 2012; Kawaguchi et al., 2017). Specifically, Bartlett (1998); Neyshabur et al. (2015b,a); Bartlett et al. (2017a); Golowich et al. (2017); Liang et al. (2017) propose various norms computed from the networks parameters and establish capacity bounds based upon these norms. In addition, Maass (1994); Bartlett et al. (1999); Schmidt-Hieber (2017); Bartlett et al. (2017b); Klusowski and Barron (2016); Suzuki (2018); Bauer et al. (2019) study the Vapnik-Chervonenkis (VC) dimension of neural networks and Dziugaite and Roy (2017); Neyshabur et al. (2017b) establish the PAC-Bayes bounds for neural networks. Among these work, our work is more related to Schmidt-Hieber (2017); Suzuki (2018), which relate the VC dimension of the ReLU networks to a set of hyperparameters used to define the networks. Based on the VC dimension, they study the statistical error of nonparametric regression using ReLU networks. In sum, theoretical understanding of deep learning is pertinent to the study of DRL algorithms. See Kawaguchi et al. (2017); Neyshabur et al. (2017a); Fan et al. (2019) and the references therein for recent developments on theoretical analysis of the generalization property of deep learning.

1.2 Notation

For a measurable space with domain $\mathcal{S}$, we denote by $\mathcal{B}(\mathcal{S}, V)$ the set of measurable functions on $\mathcal{S}$ that are bounded by V in absolute value. Let $\mathcal{P}(\mathcal{S})$ be the set of all probability measures over $\mathcal{S}$. For any $\nu \in \mathcal{P}(\mathcal{S})$ and any measurable function $f\colon \mathcal{S} \to \mathbb{R}$, we denote by $\|f\|_{\nu,p}$ the ℓ_p-norm of f with respect to measure ν for $p \geq 1$. In addition, for simplicity, we write $\|f\|_\nu$ for $\|f\|_{2,\nu}$. In addition, let $\{f(n), g(n)\}_{n\geq 1}$ be two positive series. We write $f(n) \lesssim g(n)$ if there exists a constant C such that $f(n) \leq C \cdot g(n)$ for all n larger than some $n_0 \in \mathbb{N}$. In addition, we write $f(n) \asymp g(n)$ if $f(n) \lesssim g(n)$ and $g(n) \lesssim f(n)$.

2 Background

In this section, we introduce the background. We first lay out the formulation of the reinforcement learning problem, and then define the family of ReLU neural networks.

2.1 Reinforcement Learning

A discounted Markov decision process is defined by a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$. Here $\mathcal{S}$ is the set of all states, which can be countable or uncountable, $\mathcal{A}$ is the set of all actions, $P\colon \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the Markov transition kernel, $R\colon \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathbb{R})$ is the distribution of the immediate reward, and $\gamma \in (0, 1)$ is the discount factor. In specific, upon taking any action $a \in \mathcal{A}$ at any state $s \in \mathcal{S}$, $P(\cdot\,|s, a)$ defines the probability distribution of the next state and $R(\cdot\,|s, a)$ is the distribution of the immediate reward. Moreover, for regularity, we further assume that $\mathcal{S}$ is a compact subset of $\mathbb{R}^d$ which can be infinite, $\mathcal{A} = \{a_1, a_2, \ldots, a_M\}$ has finite cardinality M, and the rewards are uniformly bounded by $R_{\max}$, i.e., $R(\cdot\,|s, a)$ is supported on $[-R_{\max}, R_{\max}]$ for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$.

A policy $\pi\colon \mathcal{S} \to \mathcal{P}(\mathcal{A})$ for the MDP maps any state $s \in \mathcal{S}$ to a probability distribution $\pi(\cdot\,|\,s)$ over $\mathcal{A}$. For policy π, the corresponding value function $V^\pi\colon \mathcal{S} \to \mathbb{R}$ is defined as the cumulative discounted reward obtained by when the actions are executed according to π, that is,

$$V^\pi(s) = \mathbb{E}\biggl[\sum_{t=0}^{\infty} \gamma^t \cdot R_t \,\bigg|\, S_0 = s, A_t \sim \pi(\cdot\,|\,S_t), S_{t+1} \sim P(\cdot\,|\,S_t, A_t)\biggr]. \tag{2.1}$$

Similarly, the action-value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as

$$Q^\pi(s, a) = \mathbb{E}\biggl[\sum_{t=0}^{\infty} \gamma^t \cdot R_t \,\bigg|\, S_0 = s, A_0 = a, A_t \sim \pi(\cdot\,|S_t), S_{t+1} \sim P(\cdot\,|\,S_t, A_t)\biggr]. \tag{2.2}$$

For any given action-value function $Q\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, define the one-step greedy policy π_Q as any policy that selects the action with the largest Q-value, that is, for any $s \in \mathcal{S}$, $\pi_Q(\cdot\,|s)$ satisfies

$$\pi_Q(a\,|\,s) = 0 \ \text{ if } \quad Q(s, a) \neq \max_{a' \in \mathcal{A}} Q(s, a'). \tag{2.3}$$

Moreover, we define operator P^π by

$$(P^\pi Q)(s, a) = \mathbb{E}\bigl[Q(S', A')\,\big|\,S' \sim P(\cdot\,|\,s, a), A' \sim \pi(\cdot\,|\,S')\bigr], \tag{2.4}$$

and define the Bellman operator T^π by $(T^\pi Q)(s,a) = r(s,a) + \gamma \cdot (P^\pi Q)(s,a)$, where $r(s,a) = \int r R(\mathrm{d}r \,|\, s,a)$ is the expected reward obtained at state s when taking action a. Then it can be verified that Q^π is the unique fixed point of T^π.

The goal of reinforcement learning is to find the optimal policy, which achieves the largest cumulative reward. To characterize optimality, we define optimal action-value function Q^* as

$$Q^*(s,a) = \sup_\pi Q^\pi(s,a), \tag{2.5}$$

where the supremum is taken over all policies. Based on Q^*, we define the optimal policy π^* as any policy that is greedy with respect to Q^*. It can be shown that $Q^* = Q^{\pi^*}$. Finally, we define the Bellman optimality operator T via

$$(TQ)(s,a) = r(s,a) + \gamma \cdot \mathbb{E}\bigl[\max_{a' \in \mathcal{A}} Q(S',a') \,\big|\, S' \sim P(\cdot \,|\, s,a)\bigr]. \tag{2.6}$$

Then we have Bellman optimality equation $TQ^* = Q^*$.

2.2 Deep Neural Network

We study the performance of DQN with rectified linear unit (ReLU) activation function $\sigma(u) = \max(u,0)$. For any positive integer L and $\{d_j\}_{i=0}^{L+1} \subseteq \mathbb{N}$, a ReLU network $f\colon \mathbb{R}^{d_0} \to \mathbb{R}^{d_{L+1}}$ with L hidden layers and width $\{d_j\}_{i=0}^{L+1}$ is of form

$$f(x) = W_{L+1}\sigma(W_L\sigma(W_{L-1}\ldots\sigma(W_2\sigma(W_1x+v_1)+v_2)\ldots v_{L-1})+v_L), \tag{2.7}$$

where $W_\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ and $v_\ell \in \mathbb{R}^{d_\ell}$ are the weight matrix and the shift vector in the ℓ-th layer, respectively. Here we apply σ to to each entry of its argument in (2.7). In deep learning, the network structure is fixed, and the goal is to learn the network parameters (weights) $\{W_\ell, v_\ell\}_{\ell \in [L+1]}$ with the convention that $v_{L+1} = 0$. For deep neural networks, the number of parameters greatly exceeds the input dimension d_0. To restrict the model class, we focus on the class of ReLU networks where most parameters are zero.

Definition 2.1 (Sparse ReLU Network)**.** For any $L, s \in \mathbb{N}$, $\{d_j\}_{i=0}^{L+1} \subseteq \mathbb{N}$, and $V > 0$, the family of sparse ReLU networks bounded by V with L hidden layers, network width d, and weight sparsity s is defined as

$$\mathcal{F}(L, \{d_j\}_{i=0}^{L+1}, s, V) = \biggl\{f\colon \max_{\ell \in [L+1]} \|\widetilde{W}_\ell\|_\infty \le 1, \sum_{\ell=1}^{L+1} \|\widetilde{W}_\ell\|_0 \le s, \max_{j \in [d_{L+1}]} \|f_j\|_\infty \le V\biggr\}, \tag{2.8}$$

where we denote (W_ℓ, v_ℓ) by $\widetilde{W}_\ell$. Moreover, f in (2.8) is expressed as in (2.7), and f_j is the j-th component of f.

Here we focus on functions that are uniformly bounded because the value functions in (2.1) and (2.2) are always bounded by $V_{\max} = R_{\max}/(1-\gamma)$. In the sequel, we write $\mathcal{F}(L, \{d_j\}_{j=0}^{L+1}, s, V_{\max})$ as $\mathcal{F}(L, \{d_j\}_{j=0}^{L+1}, s)$ to simplify the notation. In addition, we restrict the networks weights to be sparse, i.e., s is much smaller compared with the total number of parameters. Such an assumption implies that the network has sparse connections, which are useful for applying deep learning in memory-constrained situations such as mobile devices (Han et al., 2015; Liu et al., 2015).

Moreover, we introduce the notion of Hölder smoothness as follows, which is a generalization of Lipschitz continuity, and is widely used to characterize the regularity of functions.

Definition 2.2 (Hölder Smooth Function)**.** Let $\mathcal{D}$ be a compact subset of $\mathbb{R}^r$, where $r \in \mathbb{N}$. We define the set of Hölder smooth functions on $\mathcal{D}$ as

$$\mathcal{C}_r(\mathcal{D}, \beta, H) = \left\{ f\colon \mathcal{D} \to \mathbb{R}\colon \sum_{\boldsymbol{\alpha}\colon |\boldsymbol{\alpha}| < \beta} \|\partial^{\alpha} f\|_\infty + \sum_{\boldsymbol{\alpha}\colon \|\boldsymbol{\alpha}\|_1 = \lfloor \beta \rfloor} \sup_{x,y \in \mathcal{D}, x \neq y} \frac{|\partial^{\boldsymbol{\alpha}} f(x) - \partial^{\boldsymbol{\alpha}}(y)|}{\|x - y\|_\infty^{\beta - \lfloor \beta \rfloor}} \leq H \right\},$$

where $\beta > 0$ and $H > 0$ are parameters and $\lfloor \beta \rfloor$ is the largest integer no greater than β. In addition, here we use the multi-index notation by letting $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_r)^\top \in \mathbb{N}^r$, and $\partial^{\boldsymbol{\alpha}} = \partial^{\alpha_1} \ldots \partial^{\alpha_r}$.

Finally, we conclude this section by defining functions that can be written as a composition of multiple Hölder functions, which captures complex mappings in real-world applications such as multi-level feature extraction.

Definition 2.3 (Composition of Hölder Functions)**.** Let $q \in \mathbb{N}$ and $\{p_j\}_{j \in [q]} \subseteq \mathbb{N}$ be integers, and let $\{a_j, b_j\}_{j \in [q]} \subseteq \mathbb{R}$ such that $a_j < b_j$ $j \in [q]$. Moreover, let $g_j\colon [a_j, b_j]^{p_j} \to [a_{j+1}, b_{j+1}]^{p_{j+1}}$ be a function, $\forall j \in [q]$. Let $(g_{jk})_{k \in [p_{j+1}]}$ be the components of g_j, and we assume that each g_{jk} is Hölder smooth, and depends on at most t_j of its input variables, where t_j could be much smaller than p_j, i.e., $g_{jk} \in \mathcal{C}_{t_j}([a_j, b_j]^{t_j}, \beta_j, H_j)$. Finally, we denote by $\mathcal{G}(\{p_j, t_j, \beta_j, H_j\}_{j \in [q]})$ the family of functions that can be written as compositions of $\{g_j\}_{j \in [q]}$, with the convention that $p_{q+1} = 1$. That is, for any $f \in \mathcal{G}(\{p_j, t_j, \beta_j, H_j\}_{j \in [q]})$, we can write

$$f = g_q \circ g_{q-1} \circ \ldots \circ g_2 \circ g_1,$$

with $g_{jk} \in \mathcal{C}_{t_j}([a_j, b_j]^{t_j}, \beta_j, H_j)$ for each $k \in [p_{j+1}]$ and $j \in [q]$.

3 Understanding Deep Q-Network

In the DQN algorithm, a deep neural network $Q_\theta\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is used to approximate Q^*, where θ is the parameter. For completeness, we state the DQN as Algorithm 3 in §A. As shown in the experiments in (Mnih et al., 2015), two tricks are pivotal for the success of DQN.

First, DQN use the trick of experience replay (Lin, 1992). Specifically, at each time t, we store the transition (S_t, A_t, R_t, S_{t+1}) into the replay memory $\mathcal{M}$, and then sample a minibatch of independent samples from $\mathcal{M}$ to train the neural network via stochastic gradient descent. Since the trajectory of MDP has strong temporal correlation, the goal of experience replay is to obtain uncorrelated samples, which yields accurate gradient estimation for the stochastic optimization problem.

Another trick is to use a target network $Q_{\theta^\star}$ with parameter $\theta^\star$. Specifically, with independent samples $\{(s_i, a_i, r_i, s'_i)\}_{i \in [n]}$ from the replay memory, to update the parameter θ of the Q-network, we compute the target $Y_i = r_i + \gamma \cdot \max_{a\in\mathcal{A}} Q_{\theta^\star}(s_i', a)$, and update θ by the gradient of

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n}[Y_i - Q_\theta(s_i, a_i)]^2. \tag{3.1}$$

Whereas parameter $\theta^\star$ is updated once every T_{target} steps by letting $\theta^\star = \theta$. That is, the target network is hold fixed for T_{target} steps, and is thus updated in a slower pace.

To demystify DQN, it is crucial to understand the role played by these two tricks. For experience replay, in practice, the replay memory size is usually very large. For example, the replay memory size is 10^6 in Mnih et al. (2015). Moreover, DQN use the ϵ-greedy policy, which enables exploration over $\mathcal{S}\times\mathcal{A}$. Thus, when the replay memory is large, experience replay is close to sampling independent transitions from an explorative policy. This reduces the variance of the $\nabla L(\theta)$, which is used to update θ.

Thus, experience replay stabilizes the training of DQN, which benefits the algorithm in terms of computation. To understand the statistical property of DQN, we replace the experience replay by sampling independent transitions from a fixed distribution $\sigma \in \mathcal{P}(\mathcal{S}\times\mathcal{A})$. That is, instead of sampling from the replay memory, we sample i.i.d. observations $\{(S_i, A_i)\}_{i\in[n]}$ from σ. Moreover, for any $i \in [n]$, let R_i and S_i' be the immediate reward and the next state when taking action A_i at state S_i. Under this setting, we have $\mathbb{E}(Y_i \,|\, S_i, A_i) = (TQ_{\theta^\star})(S_i, A_i)$, where $Q_{\theta^\star}$ is the target network, which, as we show as follows, is motivated from a statistical consideration.

Let us first neglect the target network and set $\theta^\star = \theta$. Using bias-variance decomposition, the the expected value of $L(\theta)$ in (3.1) is

$$\mathbb{E}\big[L(\theta)\big] = \|Q_\theta - TQ_\theta\|_\sigma^2 + \mathbb{E}\Big\{\big[Y_1 - (TQ_\theta)(S_1, A_1)\big]^2\Big\}. \tag{3.2}$$

Here the first term in (3.2) is known as the mean-squared Bellman error (MSBE), and the second term is the variance of Y_1. Whereas $L(\theta)$ can be viewed as the empirical version of the MSBE, which has bias $\mathbb{E}\{[Y_1 - (TQ_\theta)(S_1, A_1)]^2\}$ that also depends on θ. Thus, without the target network, minimizing $L(\theta)$ can be drastically different from minimizing the MSBE.

To resolve this problem, we use a target network in (3.1), which has expectation

$$\mathbb{E}\big[L(\theta)\big] = \|Q_\theta - TQ_{\theta^*}\|_\sigma^2 + \mathbb{E}\Big\{\big[Y_1 - (TQ_{\theta^*})(S_1, A_1)\big]^2\Big\},$$

where the variance of Y_1 does not depend on θ. Thus, minimizing $L(\theta)$ is close to solving

$$\underset{\theta\in\Theta}{\text{minimize}}\, \|Q_\theta - TQ_{\theta^\star}\|_\sigma^2, \tag{3.3}$$

where Θ is the parameter space. Note that in DQN we hold $\theta^\star$ still and update θ for T_{target} steps. When T_{target} is sufficiently large and we neglect the fact that the objective in (3.3) is nonconvex, we would update θ by the minimizer of (3.3) for fixed $\theta^\star$.

Therefore, in the ideal case, DQN aims to solve the minimization problem (3.3) with $\theta^\star$ fixed, and then update $\theta^\star$ by the minimizer θ. Interestingly, this view of DQN offers a statistical interpretation of the target network. In specific, if $\{Q_\theta \colon \theta \in \Theta\}$ is sufficiently large such that it contains $TQ_{\theta^\star}$, then

(3.3) has solution $Q_\theta = TQ_{\theta^\star}$, which can be viewed as one-step of value iteration (Sutton and Barto, 2011) for neural networks. In addition, in the sample setting, $Q_{\theta^\star}$ is used to construct $\{Y_i\}_{i\in[n]}$, which serve as the response in the regression problem defined in (3.1), with $(TQ_{\theta^\star})$ being the regression function.

Furthermore, turning the discussion above into a realizable algorithm, we obtain the neural fitted Q-iteration (FQI) algorithm, which generates a sequence of value functions. Specifically, let $\mathcal{F}$ be a class of function defined on $\mathcal{S} \times \mathcal{A}$. In the k-th iteration of FQI, let $\widetilde{Q}_k$ be current estimate of Q^*. Similar to (3.1) and (3.3), we define $Y_i = R_i + \gamma \cdot \max_{a\in\mathcal{A}} \widetilde{Q}_k(S_i', a)$, and update $\widetilde{Q}_k$ by

$$\widetilde{Q}_{k+1} = \operatorname*{argmin}_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\bigl[Y_i - f(S_i, A_i)\bigr]^2. \tag{3.4}$$

This gives the fitted-Q iteration algorithm, which is stated in Algorithm 1.

When $\mathcal{F}$ is the family of neural networks, Algorithm 1 is known as the neural FQI, which is proposed in Riedmiller (2005). Thus, we can view neural FQI as an modification of DQN, where we replace experience replay by sampling from a fixed distribution σ, so as to understand its the statistical property. As a byproduct, such a modification naturally justifies the trick of target network in DQN. In addition, note that the optimization problem in (3.4) appears in each iteration of FQI, which is nonconvex when neural networks are used. However, since we focus solely on the statistical aspect, we make the assumption that the global optima of (3.4) can be reached, which is also contained $\mathcal{F}$. Interestingly, a recent line of research on deep learning (Du et al., 2018b,a; Zou et al., 2018; Chizat and Bach, 2018; Allen-Zhu et al., 2018a,b; Jacot et al., 2018; Cao and Gu, 2019; Arora et al., 2019; Ma et al., 2019; Mei et al., 2019; Yehudai and Shamir, 2019) has established global convergence of gradient-based algorithms for empirical risk minimization when the neural networks are overparametrized. We provide more discussions on the computation aspect in §B. Furthermore, we make the i.i.d. assumption in Algorithm 1 to simplify the analysis. Antos et al. (2008b) study the performance of fitted value iteration with fixed data used in the regression subproblems repeatedly, where the data is sampled from a single trajectory based on a fixed policy such that the induced Markov chain satisfies certain conditions on the mixing time. Using similar analysis as in Antos et al. (2008b), our algorithm can also be extended to handled fixed data that is collected beforehand.

4 Theoretical Results

We establish statistical guarantees for DQN with ReLU networks. Specifically, let Q^{π_K} be the action-value function corresponding to π_K, which is returned by Algorithm 1. In the following, we obtain an upper bound for $\|Q^{\pi_K} - Q^*\|_{1,\mu}$, where $\mu \in \mathcal{P}(\mathcal{S}\times\mathcal{A})$ is allowed to be different from ν. In addition, we assume that the state space $\mathcal{S}$ is a compact subset in $\mathbb{R}^r$ and the action space $\mathcal{A}$ is finite. Without loss of generality, we let $\mathcal{S} = [0,1]^r$ hereafter, where r is a fixed integer. To begin with, we first specify the function class $\mathcal{F}$ in Algorithm 1.

Algorithm 1 Fitted Q-Iteration Algorithm

```
**Input:** MDP $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, function class $\mathcal{F}$, sampling distribution $\sigma$, number of iterations $K$, number of samples $n$, the initial estimator $\widetilde{Q}_0$.
**for** $k = 0, 1, 2, \ldots, K-1$ **do**
    Sample i.i.d. observations $\{(S_i, A_i), i \in [n]\}$ from $\sigma$, obtain $R_i \sim R(\cdot \ |S_i, A_i)$ and $S_i' \sim P(\cdot \ |S_i, A_i)$.
    Compute $Y_i = R_i + \gamma \cdot \max_{a \in \mathcal{A}} \widetilde{Q}_k(S_i', a)$.
    Update the action-value function:
        $\widetilde{Q}_{k+1} \leftarrow \operatorname*{argmin}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \big[Y_i - f(S_i, A_i)\big]^2.$
**end for**
Define policy $\pi_K$ as the greedy policy with respect to $\widetilde{Q}_K$.
**Output:** An estimator $\widetilde{Q}_K$ of $Q^*$ and policy $\pi_K$.
```

Definition 4.1 (Function Classes)**.** Following Definition 2.1, let $\mathcal{F}(L, \{d_j\}_{j=0}^{L+1}, s)$ be the family of sparse ReLU networks defined on $\mathcal{S}$ with $d_0 = r$ and $d_{L+1} = 1$. Then we define $\mathcal{F}_0$ by

$$\mathcal{F}_0 = \big\{ f \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R} \colon f(\cdot, a) \in \mathcal{F}(L, \{d_j\}_{i=0}^{L+1}, s) \text{ for any } a \in \mathcal{A} \big\}. \tag{4.1}$$

In addition, let $\mathcal{G}(\{p_j, t_j, \beta_j, H_j\}_{j \in [q]})$ be set of composition of Hölder smooth functions defined on $\mathcal{S} \subseteq \mathbb{R}^r$. Similar to $\mathcal{F}_0$, we define a function class $\mathcal{G}_0$ as

$$\mathcal{G}_0 = \big\{ f \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R} \colon f(\cdot, a) \in \mathcal{G}(\{p_j, t_j, \beta_j, H_j\}_{j \in [q]}) \text{ for any } a \in \mathcal{A} \big\}. \tag{4.2}$$

By this definition, for any function $f \in \mathcal{F}_0$ and any action $a \in \mathcal{A}$, $f(\cdot, a)$ is a ReLU network defined on $\mathcal{S}$, which is standard for Q-networks. Moreover, $\mathcal{G}_0$ contains a broad family of smooth functions on $\mathcal{S} \times \mathcal{A}$. In the following, we make a mild assumption on $\mathcal{F}_0$ and $\mathcal{G}_0$.

Assumption 4.2. We assume that for any $f \in \mathcal{F}_0$, we have $Tf \in \mathcal{G}_0$, where T is the Bellman optimality operator defined in (2.6). That is, for any $f \in \mathcal{F}$ and any $a \in \mathcal{A}$, $(Tf)(s, a)$ can be written as compositions of Hölder smooth functions as a function of $s \in \mathcal{S}$.

We remark that this assumption holds when the MDP satisfies some smoothness conditions. For any state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, let $P(\cdot \,|\, s, a)$ be the density of the next state. By the definition of the Bellman optimality operator in (2.6), we have

$$(Tf)(s, a) = r(s, a) + \gamma \cdot \int_{\mathcal{S}} \big[\max_{a' \in \mathcal{A}} f(s', a')\big] \cdot P(s' \,|\, s, a) \mathrm{d}s'. \tag{4.3}$$

For any $s' \in \mathcal{S}$ and $a \in \mathcal{A}$, we define functions g_1, g_2 by letting $g_1(s) = r(s, a)$ and $g_2(s) = P(s' \,|\, s, a)$. Suppose both g_1 and g_2 are Hölder smooth functions on $\mathcal{S} = [0, 1]^r$ with parameters β and H. Since $\|f\|_\infty \leq V_{\max}$, by changing the order of integration and differentiation with respect to s in (4.3), we obtain that function $s \to (Tf)(s, a)$ belongs to the Hölder class $\mathcal{C}_r(\mathcal{S}, \beta, H')$ with $H' = H(1 + V_{\max})$.

Furthermore, in the more general case, suppose for any fixed $a \in \mathcal{A}$, we can write $P(s' \,|\, s, a)$ as $h_1[h_2(s,a), h_3(s')]$, where $h_2 \colon \mathcal{S} \to \mathbb{R}^{r_1}$, and $h_3 \colon \mathcal{S} \to \mathbb{R}^{r_2}$ can be viewed as feature mappings, and $h_1 \colon \mathbb{R}^{r_1+r_2} \to \mathbb{R}$ is a bivariate function. We define function $h_4 \colon \mathbb{R}^{r_1} \to \mathbb{R}$ by

$$h_4(u) = \int_{\mathcal{S}} \bigl[\max_{a' \in \mathcal{A}} f(s', a')\bigr] h_1[u, h_3(s')] \mathrm{d}s'.$$

Then by (4.3) we have $(Tf)(s,a) = g_1(s) + h_4 \circ h_2(s,a)$. Then Assumption 4.2 holds if h_4 is Hölder smooth and both g_1 and h_2 can be represented as compositions of Hölder functions. Thus, Assumption 4.2 holds if both the reward function and the transition density of the MDP are sufficiently smooth.

Moreover, even when the transition density is not smooth, we could also expect Assumption 4.2 to hold. Consider the extreme case where the MDP has deterministic transitions, that is, the next state s' is a function of s and a, which is denoted by $s' = h(s,a)$. In this case, for any ReLU network f, we have $(Tf)(s,a) = r(s,a) + \gamma \cdot \max_{a' \in \mathcal{A}} f[h(s,a), a']$. Since

$$\Bigl|\max_{a' \in \mathcal{A}} f(s_1, a') - \max_{a' \in \mathcal{A}} f(s_2, a')\Bigr| \le \max_{a' \in \mathcal{A}} \bigl| f(s_1, a') - f(s_2, a')\bigr|$$

for any $s_1, s_2 \in \mathcal{S}$, and network $f(\cdot, a)$ is Lipschitz continuous for any fixed $a \in \mathcal{A}$, function $m_1(s) = \max_{a'} f(s, a')$ is Lipschitz on $\mathcal{S}$. Thus, for any fixed $a \in \mathcal{A}$, if both $g_1(s) = r(s,a)$ and $m_2(s) = h(s,a)$ are compositions of Hölder functions, so is $(Tf)(s,a) = g_1(s) + m_1 \circ m_2(s)$. Therefore, even if the MDP has deterministic dynamics, if both the reward function $r(s,a)$ and the transition function $h(s,a)$ are sufficiently nice, Assumption 4.2 still holds true.

In the following, we define the concentration coefficients, which measures the similarity between two probability distributions under the MDP.

Assumption 4.3 (Concentration Coefficients)**.** Let $\nu_1, \nu_2 \in \mathcal{P}(\mathcal{S} \times \mathcal{A})$ be two probability measures that are absolutely continuous with respect to the Lebesgue measure on $\mathcal{S} \times \mathcal{A}$. Let $\{\pi_t\}_{t \ge 1}$ be a sequence of policies. Suppose the initial state-action pair (S_0, A_0) of the MDP has distribution ν_1, and we take action A_t according to policy π_t. For any integer m, we denote by $P^{\pi_m} P^{\pi_{m-1}} \cdots P^{\pi_1} \nu_1$ the distribution of (S_m, A_m). Then we define the m-th concentration coefficient as

$$\kappa(m; \nu_1, \nu_2) = \sup_{\pi_1, \ldots, \pi_m} \biggl[\mathbb{E}_{\nu_2} \biggl| \frac{\mathrm{d}(P^{\pi_m} P^{\pi_{m-1}} \cdots P^{\pi_1} \nu_1)}{\mathrm{d}\nu_2} \biggr|^2 \biggr]^{1/2}, \tag{4.4}$$

where the supremum is taken over all possible policies.

Furthermore, let σ be the sampling distribution in Algorithm 1 and let μ be a fixed distribution on $\mathcal{S} \times \mathcal{A}$. We assume that there exists a constant $\phi_{\mu,\sigma} < \infty$ such that

$$(1-\gamma)^2 \cdot \sum_{m \ge 1} \gamma^{m-1} \cdot m \cdot \kappa(m; \mu, \sigma) \le \phi_{\mu,\sigma}, \tag{4.5}$$

where $(1-\gamma)^2$ in (4.5) is a normalization term, since $\sum_{m \ge 1} \gamma^{m-1} \cdot m = (1-\gamma)^{-2}$.

By definition, concentration coefficients in (4.4) quantifies the similarity between ν_2 and the distribution of the future states of the MDP when starting from ν_1. Moreover, (4.5) is a standard assumption in the literature. See, e.g., Munos and Szepesvári (2008); Lazaric et al. (2016);

Scherrer et al. (2015); Farahmand et al. (2010, 2016). This assumption holds for large class of systems MDPs and specifically for MDPs whose top-Lyapunov exponent is finite. See Munos and Szepesvári (2008); Antos et al. (2007) for more detailed discussions on this assumption.

Now we are ready to present the main theorem.

Theorem 4.4. Under Assumptions 4.2 and 4.3, let $\mathcal{F}_0$ be defined in (4.1) based on the family of sparse ReLU networks $\mathcal{F}(L^*, \{d_j^*\}_{j=0}^{L^*+1}, s^*)$ and let $\mathcal{G}_0$ be given in (4.2). Moreover, for any $j \in [q-1]$, we define $\beta_j^* = \beta_j \cdot \prod_{\ell=j+1} \min(\beta_\ell, 1)$; let $\beta_q^* = 1$. In addition, let $\alpha^* = \max_{j\in[q]} t_j/(2\beta_j^* + t_j)$. For the parameters of $\mathcal{G}_0$, we assume that there exists a constant $\xi > 0$ such that

$$\max_{j\in[q]} t_j \leq \xi, \quad \sum_{j\in[q]} \log t_j \lesssim (\log n)^{\xi} \quad \text{and} \quad \max_{j\in[q]} p_j \lesssim (\log n)^{\xi}. \tag{4.6}$$

For the hyperparameters L^*, $\{d_j^*\}_{j=0}^{L^*+1}$, and s^* of the ReLU network, we set $d_0^* = 0$ and $d_{L^*+1}^* = 1$. Moreover, we set

$$L^* \lesssim (\log n)^{\xi'}, \quad \max_{j\in[q]}\{p_{j+1} \cdot t_j\} \cdot n^{\alpha^*} \lesssim \min_{i\in[L^*]} d_j^* \leq \max_{j\in[L^*]} d_j^* \lesssim n^{\xi'}, \quad \text{and} \quad s^* \asymp n^{\alpha^*} \cdot (\log n)^{\xi'} \tag{4.7}$$

for some constant $\xi' > 0$. For any $K \in \mathbb{N}$, let Q^{π_K} be the action-value function corresponding to policy π_K, which is returned by Algorithm 1 based on function class $\mathcal{F}_0$. Then there exists constants ξ^* and C such that

$$\|Q^* - Q^{\pi_K}\|_{1,\mu} \leq C \cdot \frac{\phi_{\mu,\sigma} \cdot \gamma}{(1-\gamma)^2} \cdot |\mathcal{A}| \cdot (\log n)^{\xi^*} \cdot n^{(\alpha^*-1)/2} + \frac{4\gamma^{K+1}}{(1-\gamma)^2} \cdot R_{\max}. \tag{4.8}$$

This theorem implies that the statistical rate of convergence is the sum of a statistical error and an algorithmic error. The algorithmic error converges to zero in linear rate as the algorithm proceeds, whereas the statistical error reflects the fundamental difficulty of the problem. Thus, when the number of iterations satisfy

$$K \geq C' \cdot \big[\log |A| + (1-\alpha^*) \cdot \log n\big] / \log(1/\gamma)$$

iterations, where C' is a sufficiently large constant, the algorithmic error is dominated by the statistical error. In this case, if we view both γ and $\phi_{\mu,\sigma}$ as constants and ignore the polylogarithmic term, Algorithm 1 achieves error rate

$$|\mathcal{A}| \cdot n^{(\alpha^*-1)/2} = |\mathcal{A}| \cdot \max_{j\in[q]} n^{-\beta_j^*/(2\beta_j^*+t_j)}, \tag{4.9}$$

which scales linearly with the capacity of the action space, and decays to zero when the n goes to infinity. Furthermore, the rates $\{n^{-\beta_j^*/(2\beta_j^*+t_j)}\}_{j\in[q]}$ in (4.9) recovers the statistical rate of nonparametric regression in ℓ_2-norm, whereas our statistical rate $n^{(\alpha^*-1)/2}$ in (4.9) is the fastest among these nonparametric rates, which illustrates the benefit of compositional structure of $\mathcal{G}_0$.

Furthermore, as a concrete example, we assume that both the reward function and the Markov transition kernel are Hölder smooth with smoothness parameter β. As stated below Assumption 4.2, for any $f \in \mathcal{F}_0$, we have $(Tf)(\cdot, a) \in \mathcal{C}_r(\mathcal{S}, \beta, H')$. Then Theorem 4.4 implies that Algorithm 1 achieves error rate $|\mathcal{A}| \cdot n^{-\beta/(2\beta+r)}$ when K is sufficiently large. Since $|\mathcal{A}|$ is finite, this rate achieves the minimax-optimal statistical rate of convergence within the class of Hölder smooth functions defined on $[0,1]^d$ (Stone, 1982) and thus cannot be further improved.

In the sequel, we conclude this section by sketching the proof of Theorem 4.4; the detailed proof is deferred to §6.

Proof Sketch of Theorem 4.4. Since $\{\widetilde{Q}_k\}_{k\in[K]}$ is constructed by a iterative algorithm, it is crucial to relate $\|Q^* - Q^{\pi_K}\|_{1,\mu}$, the quantity of interest, to the errors incurred in the previous steps, namely $\{\widetilde{Q}_k - T\widetilde{Q}_{k-1}\}_{k\in[K]}$. Thus, in the first step of the proof, we establish the following theorem on error propagation. For the state-value function Similar result is established in Munos and Szepesvári (2008) for fitted value iteration, which is further extended by Lazaric et al. (2016); Scherrer et al. (2015); Farahmand et al. (2010, 2016) for other batch reinforcement learning methods.

Theorem 4.5 (Error Propagation)**.** Recall that $\{\widetilde{Q}_k\}_{0\le k\le K}$ are the iterates of Algorithm 1. Let π_K be the one-step greedy policy with respect to $\widetilde{Q}_K$, and let Q^{π_K} be the action-value function corresponding to π_K. Under Assumption 4.3, we have

$$\|Q^* - Q^{\pi_K}\|_{1,\mu} \le \frac{2\phi_{\mu,\sigma}\cdot\gamma}{(1-\gamma)^2}\cdot\max_{k\in[K]}\|\widetilde{Q}_k - T\widetilde{Q}_{k-1}\|_\sigma + \frac{4\gamma^{K+1}}{(1-\gamma)^2}\cdot R_{\max}, \tag{4.10}$$

where $\phi_{\mu,\sigma}$, given in (4.5), is a constant that only depends on distributions μ and σ.

The upper bound in (4.10) shows that

$$\max_{k\in[K]}\|\widetilde{Q}_k - T\widetilde{Q}_{k-1}\|_\sigma$$

is essentially the statistical error of DQN. This implies that the fundamental difficulty of DQN is captured by the error incurred in a single step. Moreover, the proof of this theorem depends on bounding $\widetilde{Q}_\ell - T\widetilde{Q}_{\ell-1}$ using $\widetilde{Q}_k - T\widetilde{Q}_{k-1}$ for any $\ell > k$, which characterizes how the one-step error $\widetilde{Q}_k - T\widetilde{Q}_{k-1}$ propagates as the algorithm proceeds. See §C.1 for a detailed proof.

In addition, in the second step of the proof, for any $k \in [K]$, we establish the following upper bound for $\|\widetilde{Q}_k - T\widetilde{Q}_{k-1}\|_\sigma$ using tools from nonparametric regression.

Theorem 4.6 (One-step Approximation Error)**.** Recall that we define the Bellman optimality operator T in (2.6) and define function class $\mathcal{F}_0$ in (4.1). For any $\delta > 0$, let $\mathcal{N}(\delta, \mathcal{F}_0, \|\cdot\|_\infty)$ be the minimal δ-covering of $\mathcal{F}$ with respect to ℓ_∞-norm, and we denote by N_δ its cardinality. Under Assumption 4.2, for any $k \in [K]$, we have

$$\|\widehat{Q} - TQ\|_\sigma^2 \le 4\cdot[\mathrm{dist}_\infty(\mathcal{F}_0, \mathcal{G}_0)]^2 + C\cdot V_{\max}^2/n\cdot\log N_\delta + C\cdot V_{\max}\cdot\delta, \tag{4.11}$$

where $C > 0$ is an absolute constant, and

$$\mathrm{dist}_\infty(\mathcal{F}_0, \mathcal{G}_0) = \sup_{f'\in\mathcal{G}_0}\inf_{f\in\mathcal{F}_0}\|f - f'\|_\infty \tag{4.12}$$

is the ℓ_∞-error of approximating functions in $\mathcal{G}_0$ using functions in $\mathcal{F}_0$.

This theorem characterizes the bias and variance that arise in estimating the action-value functions using deep ReLU networks. Specifically, $[\mathrm{dist}_\infty(\mathcal{F}_0, \mathcal{G}_0)]^2$ corresponds to the bias incurred by approximating functions in $\mathcal{G}_0$ using ReLU networks. We note that such a bias is measured in the ℓ_∞-norm. In addition, $V_{\max}^2/n \cdot \log N_\delta + V_{\max} \cdot \delta$ controls the variance of the estimator, where the covering number N_δ is used to obtain a uniform bound over $\mathcal{F}_0$.

In the sequel, we fix $\delta = 1/n$ in (4.11), which implies that

$$\|\widetilde{Q}_{k+1} - T\widetilde{Q}_k\|_\sigma^2 \le 4 \cdot \mathrm{dist}_\infty^2(\mathcal{F}_0, \mathcal{G}_0) + C' \cdot V_{\max}^2/n \cdot \log N_\delta, \tag{4.13}$$

where $C' > 0$ is an absolute constant.

In the subsequent proof, we establish upper bounds for $\mathrm{dist}(\mathcal{F}_0, \mathcal{G}_0)$ defined in (4.12) and $\log N_\delta$, respectively. Recall that the family of composite Hölder smooth functions $\mathcal{G}_0$ is defined in (4.2).

By the definition of $\mathcal{G}_0$ in (4.2), for any $f \in \mathcal{G}_0$ and any $a \in \mathcal{A}$, $f(\cdot, a) \in \mathcal{G}(\{(p_j, t_j, \beta_j, H_j)\}_{j\in[q]})$ is a composition of Hölder smooth functions, that is, $f(\cdot, a) = g_q \circ \cdots \circ g_1$. Recall that, as defined in Definition 2.3, g_{jk} is the k-th entry of the vector-valued function g_j. Here $g_{jk} \in \mathcal{C}_{t_j}([a_j, b_j]^{t_j}, \beta_j, H_j)$ for each $k \in [p_{j+1}]$ and $j \in [q]$. To construct a ReLU network that is $f(\cdot, a)$, we first show that $f(\cdot, a)$ can be reformulated as a composition of Hölder functions defined on a hypercube. Specifically, let $h_1 = g_1/(2H_1) + 1/2$, $h_q(u) = g_q(2H_{q-1}u - H_{q-1})$, and

$$h_j(u) = g_j(2H_{j-1}u - H_{j-1})/(2H_j) + 1/2$$

for all $j \in \{2, \ldots, q-1\}$. Then we immediately have

$$f(\cdot, a) = g_q \circ \cdots \circ g_1 = h_q \circ \cdots \circ h_1. \tag{4.14}$$

Furthermore, by the definition of Hölder smooth functions in Definition 2.2, for any $j \in [q]$ and $k \in [p_{j+1}]$, it is not hard to verify that $h_{jk} \in \mathcal{C}_{t_j}\big([0, 1]^{t_j}, W\big)$, where we define $W > 0$ by

$$W = \max\Big\{\max_{1\le j\le q-1} (2H_{j-1})^{\beta_j}, H_q(2H_{q-1})^{\beta_q}\Big\}, \tag{4.15}$$

Now we employ Lemma 6.3, obtained from Schmidt-Hieber (2017), to construct a ReLU network that approximates each h_{jk}, which, combined with (4.14), yields a ReLU network that is close to $f(\cdot, a)$ in the ℓ_∞-norm.

To apply Lemma 6.3 we set $m = \eta \cdot \lceil \log_2 n \rceil$ for a sufficiently large constant $\eta > 1$, and set N to be a sufficiently large integer that depends on n, which will be specified later. In addition, we set

$$L_j = 11 + (m + 5) \cdot (1 + \lceil \log_2 t_j \rceil).$$

Then by Lemma 6.3, there exists a ReLU network $\widetilde{h}_{jk}$ such that

$$\|\widetilde{h}_{jk} - h_{jk}\|_\infty \lesssim N \cdot n^{-\eta} + N^{-\beta_j/t_j}. \tag{4.16}$$

Furthermore, we have

$$\widetilde{h}_{jk} \in \mathcal{F}(L_j, \{t_j, 12t_jN, \ldots, 12t_jN, 1\}, s_j),$$

where the sparsity s_j satisfies

$$s_j \lesssim N \cdot (m+6) \cdot (1 + \lceil \log_2 t_j \rceil). \tag{4.17}$$

Now we define $\widetilde{f} \colon \mathcal{S} \to \mathbb{R}$ by $\widetilde{f} = \widetilde{h}_q \circ \cdots \circ \widetilde{h}_1$ and set

$$N = \Big\lceil \max_{1 \le j \le q} C \cdot n^{t_j/(2\beta_j^* + t_j)} \Big\rceil. \tag{4.18}$$

For this choice of N, we show that $\widetilde{f}$ belongs to function class $\mathcal{F}(L^*, \{d_j^*\}_{j=1}^{L^*+1}, s^*)$. Moreover, we define $\lambda_j = \prod_{\ell=j+1}^{q} (\beta_\ell \wedge 1)$ for any $j \in [q-1]$, and set $\lambda_q = 1$. Then we have $\beta_j \cdot \lambda_j = \beta_j^*$ for all $j \in [q]$. Furthermore, we show that $\widetilde{f}$ is a good approximation of $f(\cdot, a)$. Specifically, we have

$$\| f(\cdot, a) - \widetilde{f} \|_\infty \lesssim \sum_{j=1}^{q} \| \widetilde{h}_j - h_j \|_\infty^{\lambda_j}.$$

Combining this with (4.16) and (4.18), we obtain that

$$\big[\mathrm{dist}(\mathcal{F}_0, \mathcal{G}_0) \big]^2 \lesssim n^{\alpha^* - 1}. \tag{4.19}$$

Moreover, using classical results on the covering number of neural networks (Anthony and Bartlett, 2009), we further show that

$$\log N_\delta \lesssim |\mathcal{A}| \cdot s^* \cdot L^* \max_{j \in [L^*]} \log(d_j^*) \lesssim |\mathcal{A}| \cdot n^{\alpha^*} \cdot (\log n)^{\xi^*} \tag{4.20}$$

for some constant $\xi^* > 0$, where $\delta = 1/n$. Therefore, combining (4.11), (4.13), (4.19), and (4.20), we conclude the proof. □

5 Extension to Two-Player Zero-Sum Markov Games

In this section, we propose the Minimax-DQN algorithm, which combines DQN and the Minimax-Q learning for two-player zero-sum Markov games. We first present the background of zero-sum Markov games and introduce the the algorithm in §5.1. Borrowing the analysis for DQN in the previous section, we provide theoretical guarantees for the proposed algorithm in §5.2.

5.1 Minimax-DQN Algorithm

As one of the simplistic extension of MDP to the multi-agent setting, two-player zero-sum Markov game is denoted by $(\mathcal{S}, \mathcal{A}, \mathcal{B}, P, R, \gamma)$, where $\mathcal{S}$ is state space, $\mathcal{A}$ and $\mathcal{B}$ are the action spaces of the first and second player, respectively. In addition, $P \colon \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathcal{P}(\mathcal{S})$ is the Markov transition kernel, and $R \colon \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathcal{P}(\mathbb{R})$ is the distribution of immediate reward received by the first player. At any time t, the two players simultaneously take actions $A_t \in \mathcal{A}$ and $B_t \in \mathcal{B}$ at state $S_t \in \mathcal{S}$, then the first player receives reward $R_t \sim R(S_t, A_t, B_t)$ and the second player obtains $-R_t$. The goal of each agent is to maximize its own cumulative discounted return.

Furthermore, let $\pi\colon \mathcal{S} \to \mathcal{P}(\mathcal{A})$ and $\nu\colon \mathcal{S} \to \mathcal{P}(\mathcal{B})$ be policies of the first and second players, respectively. Then, we similarly define the action-value function $Q^{\pi,\nu} : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ as

$$Q^{\pi,\nu}(s,a,b) = \mathbb{E}\biggl[\sum_{t=0}^{\infty} \gamma^t \cdot R_t \biggm| (S_0, A_0, B_0) = (s,a,b), A_t \sim \pi(\cdot\,|S_t), B_t \sim \nu(\cdot\,|S_t)\biggr], \tag{5.1}$$

and define the state-value function $V^{\pi,\nu} : \mathcal{S} \to \mathbb{R}$ as

$$V^{\pi,\nu}(s) = \mathbb{E}\bigl[Q^{\pi,\nu}(s,A,B) \bigm| A \sim \pi(\cdot\,|\,s), B \sim \nu(\cdot\,|\,s)\bigr]. \tag{5.2}$$

Note that these two value functions are defined by the rewards of the first player. Thus, at any state-action tuple (s,a,b), the two players aim to solve $\max_\pi \min_\nu Q^{\pi,\nu}(s,a,b)$ and $\min_\nu \max_\pi Q^{\pi,\nu}(s,a,b)$, respectively. By the von Neumann's minimax theorem (Von Neumann and Morgenstern, 1947; Patek, 1997), there exists a minimax function of the game, $Q^*\colon \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathbb{R}$, such that

$$Q^*(s,a,b) = \max_{\pi} \min_{\nu} Q^{\pi,\nu}(s,a,b) = \min_{\nu} \max_{\pi} Q^{\pi,\nu}(s,a,b). \tag{5.3}$$

Moreover, for joint policy (π,ν) of two players, we define the Bellman operators $T^{\pi,\nu}$ and T by

$$(T^{\pi,\nu}Q)(s,a,b) = r(s,a,b) + \gamma \cdot (P^{\pi,\nu}Q)(s,a,b), \tag{5.4}$$

$$(TQ)(s,a,b) = r(s,a,b) + \gamma \cdot (P^*Q)(s,a,b), \tag{5.5}$$

where $r(s,a,b) = \int rR(\mathrm{d}r\,|\,s,a,b)$, and we define operators $P_{\pi,\nu}$ and P^* by

$$(P^{\pi,\nu}Q)(s,a,b) = \mathbb{E}_{s'\sim P(\cdot\,|\,s,a,b), a'\sim\pi(\cdot\,|\,s'), b'\sim\nu(\cdot\,|\,s')}\bigl[Q(s',a',b')\bigr],$$

$$(P^*Q)(s,a,b) = \mathbb{E}_{s'\sim P(\cdot\,|\,s,a,b)}\Bigl\{\max_{\pi'\in\mathcal{P}(\mathcal{A})} \min_{\nu'\in\mathcal{P}(\mathcal{B})} \mathbb{E}_{a'\sim\pi', b'\sim\nu'}\bigl[Q(s',a',b')\bigr]\Bigr\}.$$

Note that P^* is defined by solving a zero-sum matrix game based on $Q(s',\cdot,\cdot) \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{B}|}$, which could be achieved via linear programming. It can be shown that both $T^{\pi,\nu}$ and T are γ-contractive, with $Q^{\pi,\nu}$ defined in (5.1) and Q^* defined in (5.3) being the unique fixed points, respectively. Furthermore, similar to (2.3), in zero-sum Markov games, for any action-value function Q, the equilibrium joint policy with respect to Q is defined as

$$\bigl[\pi_Q(\cdot\,|\,s), \nu_Q(\cdot\,|\,s)\bigr] = \mathop{\mathrm{argmax}}_{\pi'\in\mathcal{P}(\mathcal{A})} \mathop{\mathrm{argmin}}_{\nu'\in\mathcal{P}(\mathcal{B})} \mathbb{E}_{a\sim\pi', b\sim\nu'}\bigl[Q(s,a,b)\bigr], \qquad \forall s \in \mathcal{S}. \tag{5.6}$$

That is, $\pi_Q(\cdot\,|\,s)$ and $\nu_Q(\cdot\,|\,s)$ solves the zero-sum matrix game based on $Q(s,\cdot,\cdot)$ for all $s \in \mathcal{S}$. By this definition, we obtain that the equilibrium joint policy with respect to the minimax function Q^* defined in (5.3) achieves the Nash equilibrium of the Markov game.

Therefore, to learn the Nash equilibrium, it suffices to estimate Q^*, which is the unique fixed point of the Bellman operator T. Similar to the standard Q-learning for MDP, Littman (1994) proposes the Minimax-Q learning algorithm, which constructs a sequence of action-value functions that converges to Q^*. Specifically, in each iteration, based on a transition (s,a,b,s'), Minimax-Q learning updates the current estimator of Q^*, denoted by Q, via

$$Q(s,a,b) \leftarrow (1-\alpha)\cdot Q(s,a,b) + \alpha \cdot \Bigl\{r(s,a,b) + \gamma \cdot \max_{\pi'\in\mathcal{P}(\mathcal{A})} \min_{\nu'\in\mathcal{P}(\mathcal{B})} \mathbb{E}_{a'\sim\pi', b'\sim\nu'}\bigl[Q(s',a',b')\bigr]\Bigr\},$$

where $\alpha \in (0, 1)$ is the stepsize.

Motivated by this algorithm, we propose the Minimax-DQN algorithm which extend DQN to two-player zero-sum Markov games. Specifically, we parametrize the action-value function using a deep neural network $Q_\theta \colon \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ and store the transition $(S_t, A_t, B_t, R_t, S_{t+1})$ into the replay memory $\mathcal{M}$ at each time-step. Parameter θ of the Q-network is updated as follows. Let Q_{θ^*} be the target network. With n independent samples $\{(s_i, a_i, b_i, r_i, s_i')\}_{i\in[n]}$ from $\mathcal{M}$, for all $i \in [n]$, we compute the target

$$Y_i = r_i + \gamma \cdot \max_{\pi' \in \mathcal{P}(\mathcal{A})} \min_{\nu' \in \mathcal{P}(\mathcal{B})} \mathbb{E}_{a\sim\pi', b\sim\nu'}\big[Q_{\theta^*}(s_i', a, b)\big], \tag{5.7}$$

which can be attained via linear programming. Then we update θ in the direction of $\nabla_\theta L(\theta)$, where $L(\theta) = n^{-1}\sum_{i\in[n]}[Y_i - Q_\theta(s_i, a_i, b_i)]^2$. Finally, the target network Q_{θ^*} is updated every T_{target} steps by letting $\theta^* = \theta$. For brevity, we defer the details of Minimax-DQN to Algorithm 4 in §A.

To understand the theoretical aspects of this algorithm, we similarly utilize the framework of batch reinforcement learning for statistical analysis. With the insights gained in §3, we consider a modification of Minimax-DQN based on neural fitted Q-iteration, whose details are stated in Algorithm 2. As in the MDP setting, we replace sampling from the replay memory by sampling i.i.d. state-action tuples from a fixed distribution $\sigma \in \mathcal{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{B})$, and estimate Q^* in (5.3) by solving a sequence of least-squares regression problems.

Algorithm 2 Fitted Q-Iteration Algorithm for Zero-Sum Markov Games (Minimax-FQI)

```
Input: Two-player zero-sum Markov game $(\mathcal{S}, \mathcal{A}, \mathcal{B}, P, R, \gamma)$, function class $\mathcal{F}$, distribution $\sigma \in \mathcal{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{B})$, number of iterations $K$, number of samples $n$, the initial estimator $\widetilde{Q}_0 \in \mathcal{F}$.
for $k = 0, 1, 2, \ldots, K-1$ do
    Sample $n$ i.i.d. observations $\{(S_i, A_i, B_i)\}_{i\in[n]}$ from $\sigma$, obtain $R_i \sim R(\cdot \,|S_i, A_i, B_i)$ and $S_i' \sim P(\cdot \,|S_i, A_i, B_i)$.
    Compute $Y_i = R_i + \gamma \cdot \max_{\pi' \in \mathcal{P}(\mathcal{A})} \min_{\nu' \in \mathcal{P}(\mathcal{B})} \mathbb{E}_{a\sim\pi', b\sim\nu'}\big[\widetilde{Q}_k(s_i', a, b)\big]..$
    Update the action-value function:
        $\widetilde{Q}_{k+1} \leftarrow \operatorname*{argmin}_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\big[Y_i - f(S_i, A_i, B_i)\big]^2.$
end for
Let $(\pi_K, \nu_K)$ be the equilibrium joint policy with respect to $\widetilde{Q}_K$, which is defined in (5.6).
Output: An estimator $\widetilde{Q}_K$ of $Q^*$ and joint policy $(\pi_K, \nu_K)$.
```

5.2 Theoretical Results for Minimax-FQI

Following the theoretical results established in §4, in this subsection, we provide statistical guarantees for the Minimax-FQI algorithm with $\mathcal{F}$ being a family of deep neural networks with ReLU activation. Hereafter, without loss of generality, we assume $\mathcal{S} = [0, 1]^r$ with r being a fixed integer, and the action spaces $\mathcal{A}$ and $\mathcal{B}$ are both finite. To evaluate the performance of the algorithm, we first introduce the best-response policy as follows.

Definition 5.1. For any policy $\pi\colon \mathcal{S} \to \mathcal{P}(\mathcal{A})$ of player one, the best-response policy against π, denoted by ν_π^*, is defined as the optimal policy of second player when the first player follows π. In other words, for all $s \in \mathcal{S}$, we have $\nu_\pi^*(\cdot \,|\, s) = \mathrm{argmin}_\nu V^{\pi,\nu}(s)$, where $V^{\pi,\nu}$ is defined in (5.2).

Note that when the first player adopt a fixed policy π, from the perspective of the second player, the Markov game becomes a MDP. Thus, ν_π^* is the optimal policy of the MDP induced by π. Moreover, it can be shown that, for any policy π, $Q^*(s,a,b) \ge Q^{\pi,\nu_\pi^*}(s,a,b)$ holds for every state-action tuple (s,a,b). Thus, by considering the adversarial case where the opponent always plays the best-response policy, the difference between $Q^{\pi.\nu_\pi^*}$ and Q^* servers as a characterization of the suboptimality of π. Hence, to quantify the performance of Algorithm 2, we consider the closeness between Q^* and $Q^{\pi_K,\nu^*_{\pi_K}}$, which will be denoted by Q_K^* hereafter for simplicity. Specifically, in the following we establish an upper bound for $\|Q^* - Q_K^*\|_{1,\mu}$ for some distribution $\mu \in \mathcal{P}(\mathcal{S}\times\mathcal{A}\times\mathcal{B})$.

We first specify the function class $\mathcal{F}$ in Algorithm 2 as follows.

Assumption 5.2 (Function Classes)**.** Following Definition 4.1, let $\mathcal{F}(L, \{d_j\}_{j=0}^{L+1}, s)$ and $\mathcal{G}(\{p_j, t_j, \beta_j, H_j\}_{j\in[q]})$ be the family of sparse ReLU networks and the set of composition of Hölder smooth functions defined on $\mathcal{S}$, respectively. Similar to (4.1), we define $\mathcal{F}_1$ by

$$\mathcal{F}_1 = \bigl\{ f\colon \mathcal{S}\times\mathcal{A} \to \mathbb{R}\colon f(\cdot,a,b) \in \mathcal{F}(L, \{d_j\}_{i=0}^{L+1}, s) \text{ for any } (a,b)\in\mathcal{A}\times\mathcal{B}\bigr\}. \tag{5.8}$$

For the Bellman operator T defined in (5.5), we assume that for any $f \in \mathcal{F}_1$ and any state-action tuple (s,a,b), we have $(Tf)(\cdot,a,b) \in \mathcal{G}(\{p_j,t_j,\beta_j,H_j\}_{j\in[q]})$.

We remark that this Assumption is in the same flavor as Assumption 4.2. As discussed in §4, this assumption holds if both the reward function and the transition density of the Markov game are sufficiently smooth.

In the following, we define the concentration coefficients for Markov games.

Assumption 5.3 (Concentration Coefficient for Zero-Sum Markov Games)**.** Let $\{\tau_t\colon \mathcal{S} \to \mathcal{P}(\mathcal{A}\times\mathcal{B})\}$ be a sequence of joint policies for the two players in the zero-sum Markov game. Let $\nu_1, \nu_2 \in \mathcal{P}(\mathcal{S}\times\mathcal{A}\times\mathcal{B})$ be two absolutely continuous probability measures. Suppose the initial state-action pair (S_0, A_0, B_0) has distribution ν_1, the future states are sampled according to the Markov transition kernel, and the action (A_t, B_t) is sampled from policy τ_t. For any integer m, we denote by $P^{\tau_m}P^{\tau_{m-1}}\cdots P^{\tau_1}\nu_1$ the distribution of (S_m, A_m, B_m). Then, the m-th concentration coefficient is defined as

$$\kappa(m;\nu_1,\nu_2) = \sup_{\tau_1,\ldots,\tau_m}\biggl[\mathbb{E}_{\nu_2}\biggl|\frac{\mathrm{d}(P^{\tau_m}P^{\tau_{m-1}}\cdots P^{\tau_1}\nu_1)}{\mathrm{d}\nu_2}\biggr|^2\biggr]^{1/2}, \tag{5.9}$$

where the supremum is taken over all possible joint policy sequences $\{\tau_t\}_{t\in[m]}$.

Furthermore, for some $\mu \in \mathcal{P}(\mathcal{S}\times\mathcal{A}\times\mathcal{B})$, we assume that there exists a finite constant $\phi_{\mu,\sigma}$ such that $(1-\gamma)^2 \cdot \sum_{m\ge1}\gamma^{m-1}\cdot m\cdot\kappa(m;\mu,\sigma) \le \phi_{\mu,\sigma}$, where σ is the sampling distribution in Algorithm 2 and $\kappa(m;\mu,\sigma)$ is the m-th concentration coefficient defined in (5.9).

We remark that the definition of the m-th concentration coefficient is the same as in (4.4) if we replace the action space $\mathcal{A}$ of the MDP by $\mathcal{A} \times \mathcal{B}$ of the Markov game. Thus, Assumptions 4.3 and 5.3 are of the same nature, which are standard in the literature.

Now we are ready to present the main theorem.

Theorem 5.4. Under Assumptions 5.2 and 5.3, consider the Minimax-FQI algorithm with the function class $\mathcal{F}$ being $\mathcal{F}_1$ defined in (5.8) based on the family of sparse ReLU networks $\mathcal{F}(L^*, \{d_j^*\}_{j=0}^{L^*+1}, s^*)$. We make the same assumptions on $\mathcal{F}(L^*, \{d_j^*\}_{j=0}^{L^*+1}, s^*)$ and $\mathcal{G}(\{p_j, t_j, \beta_j, H_j\}_{j\in[q]})$ as in (4.6) and (4.7). Then for any $K \in \mathbb{N}$, let (π_K, ν_K) be the policy returned by the algorithm and let Q_K^* be the action-value function corresponding to $(\pi_K, \nu_{\pi_K}^*)$. Then there exists constants ξ^* and C such that

$$\|Q^* - Q_K^*\|_{1,\mu} \le C \cdot \frac{\phi_{\mu,\sigma} \cdot \gamma}{(1-\gamma)^2} \cdot |\mathcal{A}| \cdot |\mathcal{B}| \cdot (\log n)^{\xi^*} \cdot n^{(\alpha^*-1)/2} + \frac{4\gamma^{K+1}}{(1-\gamma)^2} \cdot R_{\max}, \tag{5.10}$$

where $\alpha^* = \max_{j\in[q]} t_j/(2\beta_j^* + t_j)$ and $\phi_{\mu,\sigma}$ is specified in Assumption 5.3.

Similar to Theorem 4.4, the bound in (5.10) shows that closeness between (π_K, ν_K) returned by Algorithm 2 and the Nash equilibrium policy (π_{Q^*}, ν_{Q^*}), measured by $\|Q^* - Q_K^*\|_{1,\mu}$, is bounded by the sum of statistical error and an algorithmic error. Specifically, the statistical error balances the bias and variance of estimating the value functions using the family of deep ReLU neural networks, which exhibits the fundamental difficulty of the problem. Whereas the algorithmic error decay to zero geometrically as K increases. Thus, when K is sufficiently large, both γ and $\phi_{\mu,\sigma}$ are constants, and the polylogarithmic term is ignored, Algorithm 2 achieves error rate

$$|\mathcal{A}| \cdot |\mathcal{B}| \cdot n^{\alpha^*-1} = |\mathcal{A}| \cdot |\mathcal{B}| \cdot \max_{j\in[q]} n^{-\beta_j^*/(2\beta_j^*+t_j)}, \tag{5.11}$$

which scales linearly with the capacity of joint action space. Besides, if $|\mathcal{B}| = 1$, The Minimax-FQI algorithm reduces to Algorithm 1. In this case, (5.11) also recovers the error rate of Algorithm 1. Furthermore, the statistical rate $n^{(\alpha^*-1)/2}$ achieves the optimal ℓ_2-norm error of regression for nonparametric regression with a compositional structure, which indicates that the statistical error in (5.10) can not be further improved.

Proof. See §D for a detailed proof. □

6 Proof of the Main Theorem

In this section, we present a detailed proof of Theorem 4.4.

Proof. The proof requires two key ingredients. First in Theorem 6.1 we quantify how the error of action-value function approximation propagates through each iteration of Algorithm 1. Then in Theorem 6.2 we analyze such one-step approximation error for ReLU networks.

Theorem 6.1 (Error Propagation)**.** Recall that $\{\widetilde{Q}_k\}_{0\le k\le K}$ are the iterates of Algorithm 1. Let π_K be the one-step greedy policy with respect to $\widetilde{Q}_K$, and let Q^{π_K} be the action-value function corresponding to π_K. Under Assumption 4.3, we have

$$\|Q^* - Q^{\pi_K}\|_{1,\mu} \le \frac{2\phi_{\mu,\sigma}\gamma}{(1-\gamma)^2}\cdot \varepsilon_{\max} + \frac{4\gamma^{K+1}}{(1-\gamma)^2}\cdot R_{\max}, \tag{6.1}$$

where we define the maximum one-step approximation error $\varepsilon_{\max} = \max_{k\in[K]} \|T\widetilde{Q}_{k-1} - \widetilde{Q}_k\|_\sigma$. Here $\phi_{\mu,\sigma}$ is a constant that only depends on the probability distributions μ and σ.

Proof. See §C.1 for a detailed proof. □

In the sequel, we establish an upper bound for the one-step approximation error $\|T\widetilde{Q}_{k-1} - \widetilde{Q}_k\|_\sigma$ for each $k \in [K]$.

Theorem 6.2 (One-step Approximation Error)**.** Let $\mathcal{F} \subseteq \mathcal{B}(\mathcal{S}\times\mathcal{A}, V_{\max})$ be a class of measurable functions on $\mathcal{S}\times\mathcal{A}$ that are bounded by $V_{\max} = R_{\max}/(1-\gamma)$, and let σ be a probability distribution on $\mathcal{S}\times\mathcal{A}$. Also, let $\{(S_i, A_i)\}_{i\in[n]}$ be n i.i.d. random variables in $\mathcal{S}\times\mathcal{A}$ following σ. For each $i\in[n]$, let R_i and S_i' be the reward and the next state corresponding to (S_i, A_i). In addition, for any fixed $Q\in\mathcal{F}$, we define $Y_i = R_i + \gamma\cdot\max_{a\in\mathcal{A}} Q(S_i', a)$. Based on $\{(X_i, A_i, Y_i)\}_{i\in[n]}$, we define $\widehat{Q}$ as the solution to the least-squares problem

$$\underset{f\in\mathcal{F}}{\text{minimize}}\ \frac{1}{n}\sum_{i=1}^{n}\big[f(S_i, A_i) - Y_i\big]^2. \tag{6.2}$$

Meanwhile, for any $\delta > 0$, let $\mathcal{N}(\delta, \mathcal{F}, \|\cdot\|_\infty)$ be the minimal δ-covering of $\mathcal{F}$ with respect to ℓ_∞-norm, and we denote by N_δ its cardinality. Then for any $\epsilon\in(0,1]$ and any $\delta > 0$, we have

$$\|\widehat{Q} - TQ\|_\sigma^2 \le (1+\epsilon)^2\cdot\omega(\mathcal{F}) + C\cdot V_{\max}^2/(n\cdot\epsilon)\cdot\log N_\delta + C'\cdot V_{\max}\cdot\delta, \tag{6.3}$$

where C and C' are two positive absolute constants and $\omega(\mathcal{F})$ is defined as

$$\omega(\mathcal{F}) = \sup_{g\in\mathcal{F}}\inf_{f\in\mathcal{F}} \|f - Tg\|_\sigma^2. \tag{6.4}$$

Proof. See §C.2 for a detailed proof. □

To obtain an upper bound for $\|T\widetilde{Q}_{k-1} - \widetilde{Q}_k\|_\sigma$ as required in Theorem 6.1, we set $Q = \widetilde{Q}_{k-1}$ in Theorem 6.2. Then according to Algorithm 1, $\widehat{Q}$ defined in (6.2) becomes $\widetilde{Q}_k$. We set the function class $\mathcal{F}$ in Theorem 6.2 to be the family of ReLU Q-networks $\mathcal{F}_0$ defined in (4.1). By setting $\epsilon = 1$ and $\delta = 1/n$ in Theorem 6.2, we obtain

$$\|\widetilde{Q}_{k+1} - T\widetilde{Q}_k\|_\sigma^2 \le 4\cdot\omega(\mathcal{F}_0) + C\cdot V_{\max}^2/n\cdot\log N_0, \tag{6.5}$$

where C is a positive absolute constant and

$$N_0 = \big|\mathcal{N}(1/n, \mathcal{F}_0, \|\cdot\|_\infty)\big| \tag{6.6}$$

is the $1/n$-covering number of $\mathcal{F}_0$. In the subsequent proof, we establish upper bounds for $\omega(\mathcal{F}_0)$ defined in (6.4) and $\log N_0$, respectively. Recall that the family of composite Hölder smooth functions $\mathcal{G}_0$ is defined in (4.2). By Assumption 4.2, we have $Tg \in \mathcal{G}_0$ for any $g \in \mathcal{F}_0$. Hence, we have

$$\omega(\mathcal{F}_0) = \sup_{f' \in \mathcal{G}_0} \inf_{f \in \mathcal{F}_0} \|f - f'\|_\sigma^2 \leq \sup_{f' \in \mathcal{G}_0} \inf_{f \in \mathcal{F}_0} \|f - f'\|_\infty^2, \tag{6.7}$$

where the right-hand side is the ℓ_∞-error of approximating the functions in $\mathcal{G}_0$ using the family of ReLU networks $\mathcal{F}_0$.

By the definition of $\mathcal{G}_0$ in (4.2), for any $f \in \mathcal{G}_0$ and any $a \in \mathcal{A}$, $f(\cdot, a) \in \mathcal{G}(\{(p_j, t_j, \beta_j, H_j)\}_{j \in [q]})$ is a composition of Hölder smooth functions, that is, $f(\cdot, a) = g_q \circ \cdots \circ g_1$. Recall that, as defined in Definition 2.3, g_{jk} is the k-th entry of the vector-valued function g_j. Here $g_{jk} \in \mathcal{C}_{t_j}([a_j, b_j]^{t_j}, \beta_j, H_j)$ for each $k \in [p_{j+1}]$ and $j \in [q]$. In the sequel, we construct a ReLU network to approximate $f(\cdot, a)$ and establish an upper bound of the approximation error on the right-hand side of (6.7). We first show that $f(\cdot, a)$ can be reformulated as a composition of Hölder functions defined on a hypercube. We define $h_1 = g_1/(2H_1) + 1/2$,

$$h_j(u) = g_j(2H_{j-1}u - H_{j-1})/(2H_j) + 1/2, \quad \text{for all} \quad j \in \{2, \ldots, q-1\},$$

and $h_q(u) = g_q(2H_{q-1}u - H_{q-1})$. Then we immediately have

$$f(\cdot, a) = g_q \circ \cdots \circ g_1 = h_q \circ \cdots \circ h_1. \tag{6.8}$$

Furthermore, by the definition of Hölder smooth functions in Definition 2.2, for any $k \in [p_2]$, we have that h_{1k} takes value in $[0, 1]$ and $h_{1k} \in \mathcal{C}_{t_1}([0, 1]^{t_1}, \beta_1, 1)$. Similarly, for any $j \in \{2, \ldots, q-1\}$ and $k \in [p_{j+1}]$, h_{jk} also takes value in $[0, 1]$ and

$$h_{jk} \in \mathcal{C}_{t_j}\big([0, 1]^{t_j}, \beta_j, (2H_{j-1})^{\beta_j}\big).$$

Finally, recall that we use the convention that $p_{q+1} = 1$, that is, h_q is a scalar-valued function that satisfies

$$h_q \in \mathcal{C}_{t_q}\big([0, 1]^{t_q}, \beta_q, H_q(2H_{q-1})^{\beta_q}\big).$$

Now we employ the following lemma of Schmidt-Hieber (2017) to construct a ReLU network that approximates each h_{jk}, which combined with (6.8) yields a ReLU network that is close to $f(\cdot, a)$. Recall that, as defined in Definition 2.2, we denote by $\mathcal{C}_r(\mathcal{D}, \beta, H)$ the family of Hölder smooth functions with parameters β and H on $\mathcal{D} \subseteq \mathbb{R}^r$.

Lemma 6.3 (Approximation of Hölder Smooth Function)**.** For any integers $m \geq 1$ and $N \geq \max\{(\beta+1)^r, (H+1)\}$, let $L = 8 + (m+5) \cdot (1 + \lceil \log_2 r \rceil)$, $d_0 = r$, $d_j = 12rN$ for each $j \in [L]$, and $d_{L+1} = 1$. For any $g \in \mathcal{C}_r([0, 1]^r, \beta, H)$, there exists a ReLU network $f \in \mathcal{F}(L, \{d_j\}_{j=0}^{L+1}, s, V_{\max})$ as defined in Definition 2.1 such that

$$\|f - g\|_\infty \leq (2H + 1) \cdot 3^{r+1} \cdot N \cdot 2^{-m} + H \cdot 2^\beta \cdot N^{-\beta/r},$$

where the parameter s satisfies $s \leq 94 \cdot r^2 \cdot (\beta + 1)^{2r} \cdot N \cdot (m + 6) \cdot (1 + \lceil \log_2 r \rceil)$.

Proof. See Theorem 3 of Schmidt-Hieber (2017) for a detailed proof. □

We apply Lemma 6.3 to $h_{jk}\colon [0,1]^{t_j} \to [0,1]$ for any $j \in [q]$ and $k \in [p_{j+1}]$. We set $m = \eta \cdot \lceil \log_2 n \rceil$ for a sufficiently large constant $\eta > 1$, and set N to be a sufficiently large integer depending on n, which will be specified later. In addition, we set

$$L_j = 9 + (m+5) \cdot (1 + \lceil \log_2 t_j \rceil) \tag{6.9}$$

and define

$$W = \max\bigl\{ \max_{1 \le j \le q-1} (2H_{j-1})^{\beta_j}, H_q(2H_{q-1})^{\beta_q} \bigr\}, \tag{6.10}$$

which is a constant. Without loss of generality, we assume $W \ge 1$ hereafter. Then by Lemma 6.3, there exists a ReLU network $\widehat{h}_{jk}$ such that

$$\|\widehat{h}_{jk} - h_{jk}\|_\infty \le (2W+1) \cdot 3^{t_j} \cdot N \cdot n^{-\eta} + W \cdot 2^{\beta_j} \cdot N^{-\beta_j/t_j}. \tag{6.11}$$

Furthermore, we have $\widehat{h}_{jk} \in \mathcal{F}(L_j, \{t_j, 12t_jN, \ldots, 12t_jN, 1\}, s_j)$ with

$$s_j \le 94 \cdot t_j^2 \cdot (\beta_j + 1)^{2t_j} \cdot N \cdot (m+6) \cdot (1 + \lceil \log_2 t_j \rceil). \tag{6.12}$$

Meanwhile, since $h_{j+1} = (h_{(j+1)k})_{k \in [p_{j+2}]}$ takes input from $[0,1]^{t_{j+1}}$, we need to further transform $\widehat{h}_{jk}$ so that it takes value in $[0,1]$. In particular, we define $\sigma(u) = 1 - (1-u)_+ = \min\{\max\{u,0\},1\}$ for any $u \in \mathbb{R}$. Note that σ can be represented by a two-layer ReLU network with four nonzero weights. Then we define $\widetilde{h}_{jk} = \sigma \circ \widehat{h}_{jk}$ and $\widetilde{h}_j = (\widetilde{h}_{jk})_{k \in [p_{j+1}]}$. Note that by the definition of $\widetilde{h}_{jk}$, we have $\widetilde{h}_{jk} \in \mathcal{F}(L_j + 2, \{t_j, 12t_jN, \ldots, 12t_jN, 1\}, s_j + 4)$, which yields

$$\widetilde{h}_j \in \mathcal{F}\bigl(L_j + 2, \{t_j, 12t_jN \cdot p_{j+1}, \ldots, 12t_jN \cdot p_{j+1}, p_{j+1}\}, (s_j+4) \cdot p_{j+1}\bigr).$$

Moreover, since both $\widetilde{h}_{jk}$ and h_{jk} take value in $[0,1]$, by (6.11) we have

$$\begin{aligned}\|\widetilde{h}_{jk} - h_{jk}\|_\infty &= \|\sigma \circ \widehat{h}_{jk} - \sigma \circ h_{jk}\|_\infty \le \|\widehat{h}_{jk} - h_{jk}\|_\infty \\ &\le (2W+1) \cdot 3^{t_j} \cdot N \cdot n^{-\eta} + W \cdot 2^{\beta_j} \cdot N^{-\beta_j/t_j},\end{aligned} \tag{6.13}$$

where the constant W is defined in (6.10). Now we define $\widetilde{f}\colon \mathcal{S} \to \mathbb{R}$ by $\widetilde{f} = \widetilde{h}_q \circ \cdots \circ \widetilde{h}_1$, which falls in the function class

$$\mathcal{F}(\widetilde{L}, \{r, \widetilde{d}, \ldots, \widetilde{d}, 1\}, \widetilde{s}), \tag{6.14}$$

where we define $\widetilde{L} = \sum_{j=1}^q (L_j + 2)$, $\widetilde{d} = \max_{j \in [q]} 12t_j \cdot p_{j+1} \cdot N$, and $\widetilde{s} = \sum_{j=1}^q (s_j + 4) \cdot p_{j+1}$. Recall that, as defined in (6.9), we have $L_j = 9 + (m+5) \cdot (1 + \lceil \log_2 t_j \rceil)$. Then when n is sufficiently large, we have

$$\widetilde{L} \le \sum_{i=1}^q \bigl[11 + (\log_2 n + 5) \cdot (1 + \lceil \log_2 t_j \rceil)\bigr] \le \sum_{i=1}^q 4 \log_2 t_i \cdot \log_2 n \lesssim (\log n)^{1+\xi}, \tag{6.15}$$

where $\xi > 0$ is an absolute constant. Here the last inequality follows from (4.6). Moreover, for $\widetilde{d}$ in (6.14), by (4.6) we have

$$N \cdot \max_{j\in[q]}\{p_{j+1}\cdot t_j\} \lesssim \widetilde{d} \le 12 \cdot N \cdot \big(\max_{j\in[q]} p_j\big)\cdot\big(\max_{j\in[q]} t_j\big) \lesssim N\cdot(\log n)^{\xi}. \tag{6.16}$$

In addition, combining (6.12), (4.6), and the fact that $t_j \le p_j$, we obtain

$$\widetilde{s} \lesssim N\cdot \log n \cdot \big(\max_{j\in[q]} p_j\big)\cdot\biggl(\sum_{j=1}^{q}\log t_j\biggr) \lesssim N\cdot(\log n)^{1+2\xi}. \tag{6.17}$$

Now we show that the function class in (6.14) can be embedded in $\mathcal{F}(L^*, \{d_j^*\}_{j=1}^{L^*+1}, s^*)$, where L^*, $\{d_j^*\}_{j=1}^{L^*+1}$, and s^* are specified in (4.7). To this end, we set

$$N = \bigl\lceil \max_{1\le j\le q} C\cdot n^{t_j/(2\beta_j^*+t_j)}\bigr\rceil, \tag{6.18}$$

where the absolute constant $C > 0$ is sufficiently large. Note that we define $\alpha^* = \max_{j\in[q]} t_j/(2\beta_j^* + t_j)$. Then (6.18) implies that $N \asymp n^{\alpha^*}$. When ξ' in (4.7) satisfies $\xi' \ge 1+2\xi$, by (6.15) we have

$$\widetilde{L} \le L^* \lesssim (\log n)^{\xi'}.$$

In addition, (6.16) and (4.7) implies that we can set $d_j^* \ge \widetilde{d}$ for all $j \in [L^*]$. Finally, by (6.17) and (6.18), we have $\widetilde{s} \lesssim n^{\alpha^*}\cdot(\log n)^{\xi'}$, which implies $\widetilde{s} + (L^* - \widetilde{L})\cdot r \le s^*$. For an $\widetilde{L}$-layer ReLU network in (6.14), we can make it an L^*-layer ReLU network by inserting $L^* - \widetilde{L}$ identity layers, since the inputs of each layer are nonnegative. Thus, ReLU networks in (6.14) can be embedded in

$$\mathcal{F}\bigl[L^*, \{r, r, \ldots, r, \widetilde{d}, \ldots, \widetilde{d}, 1\}, \widetilde{s} + (L^* - \widetilde{L})r\bigr],$$

which is a subset of $\mathcal{F}(L^*, \{d_j^*\}_{j=1}^{L+1}, s^*)$ by (4.7).

To obtain the approximation error $\|\widetilde{f} - f(\cdot, a)\|_\infty$, we define $G_j = h_j \circ \cdots \circ h_1$ and $\widetilde{G}_j = \widetilde{h}_j \circ \cdots \circ \widetilde{h}_1$ for any $j \in [q]$. By triangle inequality, for any $j > 1$ we have

$$\begin{aligned}\|G_j - \widetilde{G}_j\|_\infty &\le \|h_j\circ\widetilde{G}_{j-1} - h_j \circ G_{j-1}\|_\infty + \|\widetilde{h}_j\circ\widetilde{G}_{j-1} - h_j\circ\widetilde{G}_{j-1}\|_\infty\\ &\le W\cdot\|G_{j-1} - \widetilde{G}_{j-1}\|_\infty^{\beta_j\wedge 1} + \|h_j - \widetilde{h}_j\|_\infty,\end{aligned} \tag{6.19}$$

where the second inequality holds since h_j is Hölder smooth. To simplify the notation, we define $\lambda_j = \prod_{\ell=j+1}^{q}(\beta_\ell \wedge 1)$ for any $j \in [q-1]$, and set $\lambda_q = 1$. By applying recursion to (6.19), we obtain

$$\|f(\cdot, a) - \widetilde{f}\|_\infty = \|G_q - \widetilde{G}_q\|_\infty \le W\sum_{j=1}^{q}\|\widetilde{h}_j - h_j\|_\infty^{\lambda_j}, \tag{6.20}$$

where the constant W is defined in (6.10). Here in (6.20) we use the fact that $(a+b)^\alpha \le a^\alpha + b^\alpha$ for all $\alpha \in [0,1]$ and $a, b > 0$.

In the sequel, we combine (6.7), (6.13), (6.20), and (6.18) to obtain the final bound on $\omega(\mathcal{F}_0)$. Since we can set the constant η in (6.13) to be sufficiently large, the second term on the right-hand side of (6.13) is the leading term asymptotically, that is,

$$\|\widetilde{h}_{jk} - h_{jk}\|_\infty \lesssim N^{-\beta_j/t_j}. \tag{6.21}$$

Also note that $\beta_j^* = \beta_j \cdot \prod_{\ell=j+1}^q (\beta_\ell \wedge 1) = \beta_j \cdot \lambda_j$ for all $j \in [q-1]$. Thus we have $\beta_j^* = \beta_j \cdot \lambda_j$ for all $j \in [q]$. Combining (6.20) and (6.21), we have

$$\|f(\cdot, a) - \widetilde{f}\|_\infty \lesssim \sum_{j=1}^q \big(N^{-\beta_j/t_j}\big)^{\lambda_j} = \sum_{j=1}^q N^{-\beta_j^*/t_j} \lesssim \max_{j\in[q]} N^{-\beta_j^*/t_j}. \tag{6.22}$$

Thus, we combine (6.7), (6.18), and (6.22) to obtain

$$\omega(\mathcal{F}_0) \le \big(\max_{j\in[q]} N^{-\beta_j^*/t_j}\big)^2 \asymp \max_{j\in[q]} n^{-2\beta_j^*/(2\beta_j^*+t_j)} = n^{\alpha^*-1}. \tag{6.23}$$

As the final step of the proof, it remains to control the covering number of $\mathcal{F}_0$ defined in (4.1). By definition, for any $f \in \mathcal{F}_0$, we have $f(\cdot, a) \in \mathcal{F}(L^*, \{d_j^*\}_{j=1}^{L^*+1}, s^*)$ for any $a \in \mathcal{A}$. For notational simplicity, we denote by $\mathcal{N}_\delta$ the δ-covering of $\mathcal{F}(L^*, \{d_j^*\}_{j=1}^{L^*+1}, s^*)$, that is, we define

$$\mathcal{N}_\delta = \mathcal{N}\big[\delta, \mathcal{F}(L^*, \{d_j^*\}_{j=1}^{L^*+1}, s^*), \|\cdot\|_\infty\big].$$

By the definition of covering, for any $f \in \mathcal{F}_0$ and any $a \in \mathcal{A}$, there exists $g_a \in \mathcal{N}_\delta$ such that $\|f(\cdot,a) - g_a\|_\infty \le \delta$. Then we define a function $g\colon \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ by $g(s,a) = g_a(s)$ for any $(s,a) \in \mathcal{S}\times\mathcal{A}$. By the definition of g, it holds that $\|f - g\|_\infty \le \delta$. Therefore, the cardinality of $\mathcal{N}(\delta, \mathcal{F}_0, \|\cdot\|_\infty)$ satisfies

$$\big|\mathcal{N}(\delta, \mathcal{F}_0, \|\cdot\|_\infty)\big| \le |\mathcal{N}_\delta|^{|\mathcal{A}|}. \tag{6.24}$$

Now we utilize the following lemma in Anthony and Bartlett (2009) to obtain an upper bound of the cardinality of $\mathcal{N}_\delta$.

Lemma 6.4 (Covering Number of ReLU Network)**.** Recall that the family of ReLU networks $\mathcal{F}(L, \{d_j\}_{j=0}^{L+1}, s, V_{\max})$ is given in Definition 2.1. Let $D = \prod_{\ell=1}^{L+1}(d_\ell + 1)$. For any $\delta > 0$, we have

$$\log\Big|\mathcal{N}\big[\delta, \mathcal{F}(L, \{d_j\}_{j=0}^{L+1}, s, V_{\max}), \|\cdot\|_\infty\big]\Big| \le (s+1)\cdot \log\big[2\delta^{-1}\cdot(L+1)\cdot D^2\big].$$

Proof. See Theorem 14.5 of Anthony and Bartlett (2009) for a detailed proof. □

Recall that we denote $\mathcal{N}(1/n, \mathcal{F}_0, \|\cdot\|_\infty)$ by N_0 in (6.6). By combining (6.24) with Lemma 6.4 and setting $\delta = 1/n$, we obtain that

$$\log N_0 \le |\mathcal{A}| \cdot \log|\mathcal{N}_\delta| \le |\mathcal{A}|\cdot(s^*+1)\cdot\log\big[2n\cdot(L^*+1)\cdot D^2\big],$$

where $D = \prod_{\ell=1}^{L^*+1}(d_\ell^* + 1)$. By the choice of L^*, s^*, and $\{d_j^*\}_{j=0}^{L^*+1}$ in (4.7), we conclude that

$$\log N_0 \lesssim |\mathcal{A}|\cdot s^* \cdot L^* \max_{j\in[L^*]} \log(d_j^*) \lesssim n^{\alpha^*}\cdot(\log n)^{1+2\xi'}. \tag{6.25}$$

Finally, combining (6.1), (6.5), (6.23), and (6.25), we conclude the proof of Theorem 4.4. □

7 Conclusion

We study deep Q-network from the statistical perspective. Specifically, by neglecting the computational issues, we consider the fitted Q-iteration algorithm with ReLU networks, which can be viewed as a modification of DQN that fully captures its key features. Under mild assumptions, we show that DQN creates a sequence of policies whose corresponding value functions converge to the optimal value function, when both the sample size and the number of iteration go to infinity. Moreover, we establish a precise characterization of both the statistical and the algorithmic rates of convergence. As a byproduct, our results provide theoretical justification for the trick of using a target network in DQN. Furthermore, for we extend DQN to two-player zero-sum Markov games by proposing the Minimax-DQN algorithm. Utilizing the analysis of DQN, we establish theoretical guarantees for Minimax-DQN. To further extend this work, one future direction is to analyze reinforcement learning methods targeting at MDP with continuous action spaces, e.g., example, soft Q-learning (Haarnoja et al., 2017) and deep deterministic policy gradient (DDPG) (Lillicrap et al., 2015). Another promising direction is to combine results on optimization for deep learning with our statistical analysis to gain a unified understanding of the statistical and computational aspects of DQN.

A Deep Q-Network

We first present the DQN algorithm for MDP in details, which is proposed by Mnih et al. (2015) and adapted here to discounted MDP. As shown in Algorithm 3 below, DQN features two key tricks that lead to its empirical success, namely, experience replay and target network.

Algorithm 3 Deep Q-Network (DQN)

```
Input: MDP $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, replay memory $\mathcal{M}$, number of iterations $T$, minibatch size $n$, exploration probability $\epsilon \in (0, 1)$, a family of deep Q-networks $Q_\theta \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, an integer $T_{\text{target}}$ for updating the target network, and a sequence of stepsizes $\{\alpha_t\}_{t \geq 0}$.
Initialize the replay memory $\mathcal{M}$ to be empty.
Initialize the Q-network with random weights $\theta$.
Initialize the weights of the target network with $\theta^\star = \theta$.
Initialize the initial state $S_0$.
for $t = 0, 1, \ldots, T$ do
    With probability $\epsilon$, choose $A_t$ uniformly at random from $\mathcal{A}$, and with probability $1 - \epsilon$, choose $A_t$ such that $Q_\theta(S_t, A_t) = \max_{a \in \mathcal{A}} Q_\theta(S_t, a)$.
    Execute $A_t$ and observe reward $R_t$ and the next state $S_{t+1}$.
    Store transition $(S_t, A_t, R_t, S_{t+1})$ in $\mathcal{M}$.
    Experience replay: Sample random minibatch of transitions $\{(s_i, a_i, r_i, s_i')\}_{i \in [n]}$ from $\mathcal{M}$.
    For each $i \in [n]$, compute the target $Y_i = r_i + \gamma \cdot \max_{a \in \mathcal{A}} Q_{\theta^\star}(s_i', a)$.
    Update the Q-network: Perform a gradient descent step
        $\theta \leftarrow \theta - \alpha_t \cdot \frac{1}{n} \sum_{i \in [n]} \big[ Y_i - Q_\theta(s_i, a_i) \big] \cdot \nabla_\theta Q_\theta(s_i, a_i).$
    Update the target network: Update $\theta^\star \leftarrow \theta$ every $T_{\text{target}}$ steps.
end for
Define policy $\overline{\pi}$ as the greedy policy with respect to $Q_\theta$.
Output: Action-value function $Q_\theta$ and policy $\overline{\pi}$.
```

Furthermore, in the following, we present the details of the Minimax-DQN algorithm that extends DQN to two-player zero-sum Markov games introduced in §5. Similar to DQN, this algorithm also utilizes the experience replay and target networks. The main difference is that here the target Y_i in (5.7) is obtained by solving a zero-sum matrix game. In Algorithm 4 we present the algorithm for the second player, which can be easily modified for the first player. We note that for the second player, similar to (5.6),the equilibrium joint policy is defined as

$$\big[\widetilde{\pi}_Q(\cdot \,|\, s), \widetilde{\nu}_Q(\cdot \,|\, s) \big] = \mathop{\mathrm{argmax}}_{\nu' \in \mathcal{P}(\mathcal{B})} \mathop{\mathrm{argmin}}_{\pi' \in \mathcal{P}(\mathcal{A})} \mathbb{E}_{a \sim \pi', b \sim \nu'} \big[Q(s, a, b) \big], \qquad \forall s \in \mathcal{S}. \tag{A.1}$$

Algorithm 4 Minimax Deep Q-Network (Minimax-DQN) for the second player

```
Input: Zero-Sum Markov game $(\mathcal{S}, \mathcal{A}, \mathcal{B}, P, R, \gamma)$, replay memory $\mathcal{M}$, number of iterations $T$, minibatch size $n$, exploration probability $\epsilon \in (0,1)$, a family of deep Q-networks $Q_\theta \colon \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathbb{R}$, an integer $T_{\text{target}}$ for updating the target network, and a sequence of stepsizes $\{\alpha_t\}_{t \geq 0}$.
Initialize the replay memory $\mathcal{M}$ to be empty.
Initialize the Q-network with random weights $\theta$.
Initialize the weights of the target network by letting $\theta^\star = \theta$.
Initialize the initial state $S_0$.
for $t = 0, 1, \ldots, T$ do
    With probability $\epsilon$, choose $B_t$ uniformly at random from $\mathcal{B}$, and with probability $1 - \epsilon$, sample $B_t$ according to the equilibrium policy $\widetilde{\nu}_{Q_\theta}(\cdot \,|\, S_t)$ defined in (A.1).
    Execute $B_t$ and observe the first player's action $A_t$, reward $R_t$ satisfying $-R_t \sim R(S_t, A_t, B_t)$, and the next state $S_{t+1} \sim P(\cdot \,|\, S_t, A_t, B_t)$.
    Store transition $(S_t, A_t, B_t, R_t, S_{t+1})$ in $\mathcal{M}$.
    Experience replay: Sample random minibatch of transitions $\{(s_i, a_i, b_i, r_i, s'_i)\}_{i \in [n]}$ from $\mathcal{M}$.
    For each $i \in [n]$, compute the target
        $$Y_i = r_i + \gamma \cdot \max_{\nu' \in \mathcal{P}(\mathcal{B})} \min_{\pi' \in \mathcal{P}(\mathcal{A})} \mathbb{E}_{a \sim \pi', b \sim \nu'} \bigl[ Q_{\theta^*}(s'_i, a, b) \bigr].$$
    Update the Q-network: Perform a gradient descent step
        $$\theta \leftarrow \theta - \alpha_t \cdot \frac{1}{n} \sum_{i \in [n]} \bigl[ Y_i - Q_\theta(s_i, a_i, b_i) \bigr] \cdot \nabla_\theta Q_\theta(s_i, a_i, b_i).$$
    Update the target network: Update $\theta^\star \leftarrow \theta$ every $T_{\text{target}}$ steps.
end for
Output: Q-network $Q_\theta$ and equilibrium joint policy with respect to $Q_\theta$.
```

B Computational Aspect of DQN

Recall that in Algorithm 1 we assume the global optima of the nonlinear least-squares problem in (3.1) is obtained in each iteration. We make such an assumption as our focus is on the statistical analysis. In terms of optimization, it has been shown recently that, when the neural network is overparametrized, (stochastic) gradient descent converges to the global minima of the empirical function. Moreover, the generalization error of the obtained neural network can also be established. The intuition behind these results is that, when the neural network is overparametrized, it behaves similar to the random feature model (Rahimi and Recht, 2008). See, e.g., Du et al. (2018b,a); Zou et al. (2018); Chizat and Bach (2018); Allen-Zhu et al. (2018a,b); Jacot et al. (2018); Cao and Gu (2019); Arora et al. (2019); Ma et al. (2019); Mei et al. (2019); Yehudai and Shamir (2019) and the references therein. Also see Fan et al. (2019) for a detailed survey. In this section, we make an initial attempt in providing a unified statistical and computational analysis of DQN.

In the sequel, we consider the reinforcement learning problem with the state space $\mathcal{S} = [0,1]^r$ and a finite action space $\mathcal{A}$. To simplify the notation, we represent action a using one-hot embedding and thus identify it as an element in $\{0,1\}^{|\mathcal{A}|} \subseteq \mathbb{R}^{|\mathcal{A}|}$. In practice, categorical actions are often embedded into the Euclidean space (Dulac-Arnold et al., 2015). Thus, we can pack the state s and the action a together and obtain a vector (s,a) in $\mathbb{R}^d$, where we denote $r + |\mathcal{A}|$ by d.

We represent the Q-network by the family of two-layer neural networks

$$Q(s,a;b,W) = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} b_j \cdot \sigma[W_j^\top (s,a)], \qquad \forall (s,a) \in \mathcal{S} \times \mathcal{A}. \tag{B.1}$$

Here m is the number of neurons, $b_j \in \mathbb{R}$ and $W_j \in \mathbb{R}^d$ for all $j \in [m]$, and $\sigma(u) = \max\{u, 0\}$ is the ReLU activation function. Here $b = (b_1, \ldots, b_m)^\top \in \mathbb{R}^m$ and $W = (W_1, \ldots, W_m) \in \mathbb{R}^{d\times m}$ are the weights of the neural network. Then, in the k-th iteration of Algorithm 1, the optimization problem in (3.1) becomes

$$\underset{b,W}{\text{minimize}} \, \frac{1}{2n} \sum_{i=1}^{n} \bigl[Y_i - Q(S_i, A_i; b, W)\bigr]^2, \tag{B.2}$$

where $Y_i = R_i + \gamma \cdot \max_{a\in\mathcal{A}} \widetilde{Q}_k(S_i', a)$ is the target and $\widetilde{Q}_k$ is the Q-network computed in the previous iteration. Notice that this problem is the least-squares regression with two-layer neural networks in the overparametrized setting. We solve the optimization in (B.2) via gradient descent. Specifically, we initialize the parameters via $b_j \overset{\text{i.i.d.}}{\sim} \text{Unif}(\{-1, 1\})$ and $W_j \overset{\text{i.i.d.}}{\sim} N(0, I_d/d)$, where I_d is the identity matrix in $\mathbb{R}^d$. Moreover, for ease of presentation, during training we keeping $\{b_1, \ldots, b_m\}$ fixed as the random initialization and only optimize over W. Moreover, let $W(0) \in \mathbb{R}^{d\times m}$ be the initialization of W. We restrict the weight W to a Frobenius ball centered at $W(0)$ with radius $B > 0$, i.e., we define

$$\mathcal{B}_B = \bigl\{ W \in \mathbb{R}^{d\times m} \colon \|W - W(0)\|_{\text{fro}} \le B \bigr\}, \tag{B.3}$$

where B is a sufficiently large constant. Thus, the problem in (B.2) is transformed into

$$\underset{W\in\mathcal{B}_B}{\text{minimize}} \, L_n(W) = \frac{1}{2n} \sum_{i=1}^{n} \bigl[Y_i - Q(S_i, A_i; b, W)\bigr]^2, \tag{B.4}$$

where b is fixed to the initial value. We solve this optimization problem via projected gradient descent, which generates a sequence of weights $\{W(t)\}_{t\ge 0} \subseteq \mathcal{B}_B$ satisfying

$$W(t+1) = \Pi_{\mathcal{B}_B}\Bigl[W(t) - \frac{\eta}{n} \sum_{i=1}^{n} \bigl[Y_i - Q\bigl(S_i, A_i; b, W(t)\bigr)\bigr] \cdot \nabla_W Q\bigl(S_i, A_i; b, W(t)\bigr) \Bigr], \tag{B.5}$$

where $\Pi_{\mathcal{B}_B}$ is the projection operator onto $\mathcal{B}_B$ and $\eta > 0$ is the step size.

To understand the convergence of the updates in (B.4), we utilize the fact that overparametrized two-layer neural networks behave similar to the random feature model. Specifically, notice that

$$Q(s,a;b,W) = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} b_j \cdot \mathbb{1}\{W_j^\top (s,a) > 0\} \cdot W_j^\top (s,a), \tag{B.6}$$

$$\nabla_{W_j} Q(s,a;b,W) = \frac{1}{\sqrt{m}} b_j \cdot \mathbb{1}\{W_j^\top (s,a) > 0\}, \qquad \forall j \in [m]. \tag{B.7}$$

We define function class

$$\mathcal{F}_{B,m}^{(t)} = \biggl\{ Q(s,a) = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} b_j \cdot \mathbb{1}\{W_j(t)^\top (s,a) > 0\} \cdot W_j^\top (s,a) \colon W \in \mathcal{B}_B \biggr\}. \tag{B.8}$$

By (B.6) and (B.7), every $Q \in \mathcal{F}_{B,m}^{(t)}$ can be written as

$$\begin{aligned} Q(s,a) &= Q\bigl(s,a;b,W(t)\bigr) + \frac{1}{\sqrt{m}} \sum_{j=1}^{m} b_j \cdot \mathbb{1}\{W_j(t)^\top (s,a) > 0\} \cdot [W_j - W_j(t)]^\top (s,a) \\ &= Q\bigl(s,a;b,W(t)\bigr) + \bigl\langle \nabla_W Q\bigl(s,a;b,W(t)\bigr), W - W(t) \bigr\rangle. \end{aligned}$$

Thus, $\mathcal{F}_{B,m}^{(t)}$ contains first-order approximations of $Q\bigl(s,a;b,W(t)\bigr)$. Furthermore, as shown in various work including Du et al. (2018b,a); Arora et al. (2019), when m is sufficiently large, the overall effect of the scaled indicators $\{1/\sqrt{m} \cdot \mathbb{1}\{W_j(t)^\top (s,a) > 0\}_{j\in[m]}$ are well approximated by the indicators produced by the initial weights. That is, when m approaches infinity, for each fixed t, $\mathcal{F}_{B,m}^{(t)}$ defined in (B.8) is close to

$$\mathcal{F}_{B,m} = \biggl\{ Q(s,a) = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} b_j \cdot \mathbb{1}\{W_j(0)^\top (s,a) > 0\} \cdot W_j^\top (s,a) \colon W \in \mathcal{B}_B \biggr\}. \tag{B.9}$$

Notice that $\{\phi_j(s,a) = 1/\sqrt{m} \cdot b_j \cdot \mathbb{1}\{W_j(0)^\top (s,a) > 0\}$ are i.i.d. random variables. Thus, function class $\mathcal{F}_{B,m}$ is family of functions that can be written as combinations of random features, i.e.,

$$\mathcal{F}_{B,m} = \biggl\{ Q(s,a) = Q\bigl(s,a;b,W(0)\bigr) + \sum_{j=1}^{m} \phi_j(s,a)^\top W_j \colon \|W\|_{\text{fro}} \le B \biggr\}, \tag{B.10}$$

where we utilize the definition of $\mathcal{B}_B$. More importantly, when m goes to infinity, the empirical distribution of the random features $\{\phi_j(s,a)\}_{j\in[m]}$ converges to its population distribution. We denote $\phi(\cdot,\cdot;\beta,w)$ as the random feature, where $\beta \in \text{Unif}(\{-1,1\})$ and $w \sim N(I_d/d)$. We denote μ as the joint distribution of β and w. Then, $\mathcal{F}_{B,m}$ in (B.10) converges to a set $\mathcal{F}_B^*$ given by

$$\mathcal{F}_B^* = \biggl\{ Q(s,a) = Q_0(s,a) + \int \phi(s,a;\beta,w)^\top \alpha(\beta,w) \,\mathrm{d}\mu(\beta,w) \colon \int \|\alpha(\beta,w)\|_2^2 \,\mathrm{d}\mu(\beta,w) \le B^2 \biggr\}, \tag{B.11}$$

where α is a function over $\{-1,1\} \times \mathbb{R}^d$ and $Q_0(s,a) = \lim_{m\to\infty} Q\bigl(s,a;b,W(0)\bigr)$.

Furthermore, it can be shown that $\mathcal{F}_B^*$ is a subset of a reproducing kernel Hilbert space $\mathcal{H}$ (Rahimi and Recht, 2008) generated by kernel

$$K\bigl[(s,a),(s',a')\bigr] = \mathbb{E}_{N(I_d/d)}\bigl[\mathbb{1}\{w^\top (s,a) > 0\} \cdot \mathbb{1}\{w^\top (s',a') > 0\} \langle (s,a),(s',a') \rangle\bigr]. \tag{B.12}$$

Besides, the inner product induced by the RKHS norm between two functions

$$f_1 = \int \phi(\cdot,\cdot;\beta,w)^\top \alpha_1(\beta,w) \,\mathrm{d}\mu(\beta,w) \qquad \text{and} \qquad f_2 = \int \phi(\cdot,\cdot;\beta,w)^\top \alpha_2(\beta,w) \,\mathrm{d}\mu(\beta,w)$$

is given by $\langle f_1, f_2\rangle_{\mathcal{H}} = \int \langle \alpha_1(\beta, w), \alpha_2(\beta, w)\rangle \, \mathrm{d}\mu(\beta, w)$. Thus, $\mathcal{F}_B^*$ given in (B.11) can be written a RKHS-norm ball

$$\mathcal{F}_B^* = \{Q = Q_0 + f : \|f\|_{\mathcal{H}} \leq B\}$$

with radius B. As a result, when both m and n go to infinity, the population problem correspond to (B.2) becomes minimize$_{Q\in\mathcal{F}_B^*} \|Q - T\widetilde{Q}_k\|_\sigma^2$.

Furthermore, we utilize a recent result on optimization for overparametrized two-layer neural networks (Arora et al., 2019), which states that the generalization error of projected gradient descent has $\mathcal{O}(1/\sqrt{n})$ generalization error when both the number of hidden neurons and the number of iterations are sufficiently large. That is, when the Q-netowrk is overparametrized, the Q-networks obtained by the projected gradient descent algorithm in (B.5) satisfy

$$\|Q(\cdot, \cdot, W(t)) - T\widetilde{Q}_k\|_\sigma^2 - \inf_{Q\in\mathcal{F}_B^*} \|Q - T\widetilde{Q}_k\|_\sigma^2 \lesssim 1/\sqrt{n}$$

where t is sufficiently large. Here we use the notation $a \lesssim b$ to indicate that there is a constant C that does not depend on n, m and t such that $a \leq Cb$. Therefore, we have

$$\|Q(\cdot, \cdot, W(t)) - T\widetilde{Q}_k\|_\sigma^2 \lesssim \inf_{Q\in\mathcal{F}_B^*} \|Q - T\widetilde{Q}_k\|_\sigma^2 + 1/\sqrt{n} \lesssim 1/\sqrt{n} + \mathrm{dist}(\mathcal{F}_B^*, \sigma, \mathcal{T}), \tag{B.13}$$

where $\mathrm{dist}(\mathcal{F}_B^*, \sigma, \mathcal{T})$ is defined as

$$\mathrm{dist}(\mathcal{F}_B^*, \sigma, \mathcal{T}) = \inf_{f\in\mathcal{F}_B^*} \sup_{g\in\mathcal{F}_B^*} \|f - Tg\|_\sigma^2,$$

which measures the approximation error of functions in the RKHS ball $\mathcal{F}_B^*$ with respect the Bellman operator and the sampling distribution σ. Finally, combining (B.13) with the error propagation result in Theorem 6.1, we obtain the error of FQI algorithm with overparametrized two-layer neural networks. Specifically, for any $k \geq 0$ let $\widetilde{Q}_{k+1}$ be the output of the projected gradient descent in (B.5) when the target is computed by $\widetilde{Q}_k$. Then the Algorithm 3 outputs $\widetilde{Q}_K$ and its greedy policy π_K for some sufficiently large K. Combining Theorem 6.1 and (B.13) we have

$$\|Q^* - Q^{\pi_K}\|_{1,\mu} \lesssim \frac{2\phi_{\mu,\sigma}\gamma}{(1-\gamma)^2} \cdot \left[\frac{1}{\sqrt{n}} + \mathrm{dist}(\mathcal{F}_B^*, \sigma, \mathcal{T})\right] + \frac{4\gamma^{K+1}}{(1-\gamma)^2} \cdot R_{\max}. \tag{B.14}$$

Thus, when both n and K is sufficiently large, the dominating term on right-hand side of (B.14) is $\phi_{\mu,\sigma}\gamma \cdot (1-\gamma)^{-2} \cdot \mathrm{dist}(\mathcal{F}_B^*, \sigma, \mathcal{T})$, which indicates that the error of using functions in $\mathcal{F}_B^*$ to approximate $\{TQ : Q \in \mathcal{F}_B^*\}$ determines the performance of DQN with overparametrized neural networks.

C Proofs of Auxiliary Results

In this section, we present the proofs for Theorems 6.1 and 6.2, which are used in the §6 to establish our main theorem.

C.1 Proof of Theorem 6.1

Proof. Before we present the proof, we introduce some notation. For any $k \in \{0, \ldots, K-1\}$, we denote $T\widetilde{Q}_k$ by Q_{k+1} and define

$$\varrho_k = Q_k - \widetilde{Q}_k. \tag{C.1}$$

Also, we denote by π_k the greedy policy with respect to $\widetilde{Q}_k$. In addition, throughout the proof, for two functions $Q_1, Q_2 \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, we use the notation $Q_1 \geq Q_2$ if $Q_1(s,a) \geq Q_2(s,a)$ for any $s \in \mathcal{S}$ and any $a \in \mathcal{A}$, and define $Q_1 \leq Q_2$ similarly. Furthermore, for any policy π, recall that in (2.4) we define the operator P^π by

$$(P^\pi Q)(s,a) = \mathbb{E}\bigl[Q(S', A') \,\big|\, S' \sim P(\cdot \,|\, s,a), A' \sim \pi(\cdot \,|\, S')\bigr]. \tag{C.2}$$

In addition, we define the operator T^π by

$$(T^\pi Q)(s,a) = r(s,a) + \gamma \cdot (P^\pi Q)(s,a).$$

Finally, we denote $R_{\max}/(1-\gamma)$ by $V_{\max}$. Now we are ready to present the proof, which consists of three key steps.

Step (i): In the first step, we establish a recursion that relates $Q^* - \widetilde{Q}_{k+1}$ with $Q^* - \widetilde{Q}_k$ to measure the sub-optimality of the value function $\widetilde{Q}_k$. In the following, we first establish an upper bound for $Q^* - \widetilde{Q}_{k+1}$ as follows. For each $k \in \{0, \ldots, K-1\}$, by the definition of ϱ_{k+1} in (C.1), we have

$$\begin{aligned} Q^* - \widetilde{Q}_{k+1} &= Q^* - (Q_{k+1} - \varrho_{k+1}) = Q^* - Q_{k+1} + \varrho_{k+1} = Q^* - T\widetilde{Q}_k + \varrho_{k+1} \\ &= Q^* - T^{\pi^*}\widetilde{Q}_k + (T^{\pi^*}\widetilde{Q}_k - T\widetilde{Q}_k) + \varrho_{k+1}, \end{aligned} \tag{C.3}$$

where π^* is the greedy policy with respect to Q^*. Now we leverage the following lemma to show $T^{\pi^*}\widetilde{Q}_k \leq T\widetilde{Q}_k$.

Lemma C.1. For any action-value function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and any policy π, it holds that

$$T^{\pi_Q} Q = TQ \geq T^\pi Q.$$

Proof. Note that we have $\max_{a'} Q(s', a') \geq Q(s', a')$ for any $s' \in \mathcal{S}$ and $a' \in \mathcal{A}$. Thus, it holds that

$$\begin{aligned} (TQ)(s,a) &= r(s,a) + \gamma \cdot \mathbb{E}\bigl[\max_{a'} Q(S', a') \,\big|\, S' \sim P(\cdot \,|\, s,a)\bigr] \\ &\geq r(s,a) + \gamma \cdot \mathbb{E}\bigl[Q(S', A') \,\big|\, S' \sim P(\cdot \,|\, s,a), A' \sim \pi(\cdot \,|\, S')\bigr] = (T^\pi Q)(s,a). \end{aligned}$$

Recall that π_Q is the greedy policy with respect to Q such that

$$\mathbb{P}\bigl[A \in \operatorname*{argmax}_a Q(s,a) \,\big|\, A \sim \pi_Q(\cdot \,|\, s)\bigr] = 1,$$

which implies

$$\mathbb{E}\bigl[Q(s', A') \,\big|\, A' \sim \pi_Q(\cdot \,|\, s')\bigr] = \max_{a'} Q(s', a').$$

Consequently, we have

$$
\begin{aligned}
(T^{\pi_Q} Q)(s,a) &= r(s,a) + \gamma \cdot \mathbb{E}\big[Q(S',A') \,\big|\, S' \sim P(\cdot \,|\, s,a), A' \sim \pi_Q(\cdot \,|\, S')\big] \\
&= r(s,a) + \gamma \cdot \mathbb{E}\big[\max_{a'} Q(S',a') \,\big|\, S' \sim P(\cdot \,|\, s,a)\big] = (TQ)(s,a),
\end{aligned}
$$

which concludes the proof of Lemma C.1. □

By Lemma C.1, we have $T\widetilde{Q}_k \geq T^{\pi^*}\widetilde{Q}_k$. Also note that Q^* is the unique fixed point of T^{π^*}. Thus, by (C.3) we have

$$
Q^* - \widetilde{Q}_{k+1} = (T^{\pi^*}Q^* - T^{\pi^*}\widetilde{Q}_k) + (T^{\pi^*}\widetilde{Q}_k - T\widetilde{Q}_k) + \varrho_{k+1} \leq (T^{\pi^*}Q^* - T^{\pi^*}\widetilde{Q}_k) + \varrho_{k+1}, \tag{C.4}
$$

In the following, we establish a lower bound for $Q^* - \widetilde{Q}_{k+1}$ based on $\widetilde{Q}^* - \widetilde{Q}_k$. Note that, by Lemma C.1, we have $T^{\pi_k}\widetilde{Q}_k = T\widetilde{Q}_k$ and $TQ^* \geq T^{\pi_k}Q^*$. Similar to (C.3), since Q^* is the unique fixed point of T, it holds that

$$
\begin{aligned}
Q^* - \widetilde{Q}_{k+1} &= Q^* - T\widetilde{Q}_k + \varrho_{k+1} = Q^* - T^{\pi_k}\widetilde{Q}_k + \varrho_{k+1} = Q^* - T^{\pi_k}Q^* + (T^{\pi_k}Q^* - T^{\pi_k}\widetilde{Q}_k) + \varrho_{k+1} \\
&= (TQ^* - T^{\pi_k}Q^*) + (T^{\pi_k}Q^* - T^{\pi_k}\widetilde{Q}_k) + \varrho_{k+1} \geq (T^{\pi_k}Q^* - T^{\pi_k}\widetilde{Q}_k) + \varrho_{k+1}. \qquad \text{(C.5)}
\end{aligned}
$$

Thus, combining (C.4) and (C.5) we obtain that, for any $k \in \{0, \ldots, K-1\}$,

$$
T^{\pi_k}Q^* - T^{\pi_k}\widetilde{Q}_k + \varrho_{k+1} \leq Q^* - \widetilde{Q}_{k+1} \leq T^{\pi^*}Q^* - T^{\pi^*}\widetilde{Q}_k + \varrho_{k+1}. \tag{C.6}
$$

The inequalities in (C.6) show that the error $Q^* - \widetilde{Q}_{k+1}$ can be sandwiched by the summation of a term involving $Q^* - \widetilde{Q}_k$ and the error ϱ_{k+1}, which is defined in (C.1) and induced by approximating the action-value function. Using P^π defined in (C.2), we can write (C.6) in a more compact form,

$$
\gamma \cdot P^{\pi^*}(Q^* - \widetilde{Q}_k) + \varrho_{k+1} \geq Q^* - \widetilde{Q}_{k+1} \geq \gamma \cdot P^{\pi_k}(Q^* - \widetilde{Q}_k) + \varrho_{k+1}. \tag{C.7}
$$

Meanwhile, note that P^π defined in (C.2) is a linear operator. In fact, P^π is the Markov transition operator for the Markov chain on $\mathcal{S} \times \mathcal{A}$ with transition dynamics

$$
S_{t+1} \sim P(\cdot \,|\, S_t, A_t), \qquad A_{t+1} \sim \pi(\cdot \,|\, S_{t+1}).
$$

By the linearity of the operator P^π and the one-step error bound in (C.6), we have the following characterization of the multi-step error.

Lemma C.2 (Error Propagation)**.** For any $k, \ell \in \{0, 1, \ldots, K-1\}$ with $k < \ell$, we have

$$
Q^* - \widetilde{Q}_\ell \leq \sum_{i=k}^{\ell-1} \gamma^{\ell-1-i} \cdot (P^{\pi^*})^{\ell-1-i}\varrho_{i+1} + \gamma^{\ell-k} \cdot (P^{\pi^*})^{\ell-k}(Q^* - \widetilde{Q}_k), \tag{C.8}
$$

$$
Q^* - \widetilde{Q}_\ell \geq \sum_{i=k}^{\ell-1} \gamma^{\ell-1-i} \cdot (P^{\pi_{\ell-1}}P^{\pi_{\ell-2}} \cdots P^{\pi_{i+1}})\varrho_{i+1} + \gamma^{\ell-k} \cdot (P^{\pi_{\ell-1}}P^{\pi_{\ell-2}} \cdots P^{\pi_k})(Q^* - \widetilde{Q}_k). \tag{C.9}
$$

Here ϱ_{i+1} is defined in (C.1) and we use $P^\pi P^{\pi'}$ and $(P^\pi)^k$ to denote the composition of operators.

Proof. Note that P^π is a linear operator for any policy π. We obtain (C.8) and (C.9) by iteratively applying the inequalities in (C.7). □

Lemma C.2 gives the upper and lower bounds for the propagation of error through multiple iterations of Algorithm 1, which concludes the first step of our proof.

Step (ii): The results in the first step only concern the propagation of error $Q^* - \widetilde{Q}_k$. In contrast, the output of Algorithm 1 is the greedy policy π_k with respect to $\widetilde{Q}_k$. In the second step, our goal is to quantify the suboptimality of Q^{π_k}, which is the action-value function corresponding to π_k. In the following, we establish an upper bound for $Q^* - Q^{\pi_k}$.

To begin with, we have $Q^* \geq Q^{\pi_k}$ by the definition of Q^* in (2.5). Note that we have $Q^* = T^{\pi^*} Q^*$ and $Q^{\pi_k} = T^{\pi_k} Q^{\pi_k}$. Hence, it holds that

$$\begin{aligned} Q^* - Q^{\pi_k} &= T^{\pi^*} Q^* - T^{\pi_k} Q^{\pi_k} = T^{\pi^*} Q^* + (-T^{\pi^*}\widetilde{Q}_k + T^{\pi^*}\widetilde{Q}_k) + (-T^{\pi_k}\widetilde{Q}_k + T^{\pi_k}\widetilde{Q}_k) - T^{\pi_k} Q^{\pi_k} \\ &= (T^{\pi^*}\widetilde{Q}_k - T^{\pi_k}\widetilde{Q}_k) + (T^{\pi^*} Q^* - T^{\pi^*}\widetilde{Q}_k) + (T^{\pi_k}\widetilde{Q}_k - T^{\pi_k} Q^{\pi_k}). \end{aligned} \tag{C.10}$$

Now we quantify the three terms on the right-hand side of (C.10) respectively. First, by Lemma C.1, we have

$$T^{\pi^*}\widetilde{Q}_k - T^{\pi_k}\widetilde{Q}_k = T^{\pi^*}\widetilde{Q}_k - T\widetilde{Q}_k \leq 0. \tag{C.11}$$

Meanwhile, by the definition of the operator P^π in (C.2), we have

$$T^{\pi^*} Q^* - T^{\pi^*}\widetilde{Q}_k = \gamma \cdot P^{\pi^*}(Q^* - \widetilde{Q}_k), \qquad T^{\pi_k}\widetilde{Q}_k - T^{\pi_k} Q^{\pi_k} = \gamma \cdot P^{\pi_k}(\widetilde{Q}_k - Q^{\pi_k}). \tag{C.12}$$

Plugging (C.11) and (C.12) into (C.10), we obtain

$$\begin{aligned} Q^* - Q^{\pi_k} &\leq \gamma \cdot P^{\pi^*}(Q^* - \widetilde{Q}_k) + \gamma \cdot P^{\pi_k}(\widetilde{Q}_k - Q^{\pi_k}) \\ &= \gamma \cdot (P^{\pi^*} - P^{\pi_k})(Q^* - \widetilde{Q}_k) + \gamma \cdot P^{\pi_k}(Q^* - Q^{\pi_k}), \end{aligned}$$

which further implies that

$$(I - \gamma \cdot P^{\pi_k})(Q^* - Q^{\pi_k}) \leq \gamma \cdot (P^{\pi^*} - P^{\pi_k})(Q^* - \widetilde{Q}_k).$$

Here I is the identity operator. Since T^π is a γ-contractive operator for any policy π, $I - \gamma \cdot P^\pi$ is invertible. Thus, we obtain

$$0 \leq Q^* - Q^{\pi_k} \leq \gamma \cdot (I - \gamma \cdot P^{\pi_k})^{-1} \big[P^{\pi^*}(Q^* - \widetilde{Q}_k) - P^{\pi_k}(Q^* - \widetilde{Q}_k)\big], \tag{C.13}$$

which relates $Q^* - Q^{\pi_k}$ with $Q^* - \widetilde{Q}_k$. In the following, we plug Lemma C.2 into (C.13) to obtain the multiple-step error bounds for Q^{π_k}. First note that, by the definition of P^π in (C.2), for any functions $f_1, f_2 : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ satisfying $f_1 \geq f_2$, we have $P^\pi f_1 \geq P^\pi f_2$. Combining this inequality with the upper bound in (C.8) and the lower bound in (C.9), we have that, for any $k < \ell$,

$$P^{\pi^*}(Q^* - \widetilde{Q}_\ell) \leq \sum_{i=k}^{\ell-1} \gamma^{\ell-1-i} \cdot (P^{\pi^*})^{\ell-i} \varrho_{i+1} + \gamma^{\ell-k} \cdot (P^{\pi^*})^{\ell-k+1}(Q^* - \widetilde{Q}_k), \tag{C.14}$$

$$\begin{aligned} P^{\pi_\ell}(Q^* - \widetilde{Q}_\ell) \geq \sum_{i=k}^{\ell-1} &\gamma^{\ell-1-i} \cdot (P^{\pi_\ell} P^{\pi_{\ell-1}} \cdots P^{\pi_{i+1}}) \varrho_{i+1} \\ &+ \gamma^{\ell-k} \cdot (P^{\pi_\ell} P^{\pi_{\ell-1}} \cdots P^{\pi_k})(Q^* - \widetilde{Q}_k). \end{aligned} \tag{C.15}$$

Then we plug (C.14) and (C.15) into (C.13) and obtain

$$0 \le Q^* - Q^{\pi_\ell} \le (I - \gamma \cdot P^{\pi_\ell})^{-1} \biggl\{ \sum_{i=k}^{\ell-1} \gamma^{\ell-i} \cdot \bigl[(P^{\pi^*})^{\ell-i} - (P^{\pi_\ell} P^{\pi_{\ell-1}} \cdots P^{\pi_{i+1}}) \bigr] \varrho_{i+1} + \gamma^{\ell+1-k} \cdot \bigl[(P^{\pi^*})^{\ell-k+1} - (P^{\pi_\ell} P^{\pi_{\ell-1}} \cdots P^{\pi_k}) \bigr] (Q^* - \widetilde{Q}_k) \biggr\} \tag{C.16}$$

for any $k < \ell$. To quantify the error of Q^{π_K}, we set $\ell = K$ and $k = 0$ in (C.16) to obtain

$$0 \le Q^* - Q^{\pi_K} \le (I - \gamma P^{\pi_K})^{-1} \biggl\{ \sum_{i=0}^{K-1} \gamma^{K-i} \cdot \bigl[(P^{\pi^*})^{K-i} - (P^{\pi_K} P^{\pi_{K-1}} \cdots P^{\pi_{i+1}}) \bigr] \varrho_{i+1} + \gamma^{K+1} \cdot \bigl[(P^{\pi^*})^{K+1} - (P^{\pi_K} P^{\pi_{K-1}} \cdots P^{\pi_0}) \bigr] (Q^* - \widetilde{Q}_0) \biggr\}. \tag{C.17}$$

For notational simplicity, we define

$$\alpha_i = \frac{(1-\gamma)\gamma^{K-i-1}}{1-\gamma^{K+1}}, \quad \text{for} \quad 0 \le i \le K-1, \quad \text{and} \quad \alpha_K = \frac{(1-\gamma)\gamma^K}{1-\gamma^{K+1}}. \tag{C.18}$$

One can show that $\sum_{i=0}^{K} \alpha_i = 1$. Meanwhile, we define $K+1$ linear operators $\{O_k\}_{k=0}^{K}$ by

$$O_i = (1-\gamma)/2 \cdot (I - \gamma P^{\pi_K})^{-1} \bigl[(P^{\pi^*})^{K-i} + (P^{\pi_K} P^{\pi_{K-1}} \cdots P^{\pi_{i+1}}) \bigr], \quad \text{for} \quad 0 \le i \le K-1,$$
$$O_K = (1-\gamma)/2 \cdot (I - \gamma P^{\pi_K})^{-1} \bigl[(P^{\pi^*})^{K+1} + (P^{\pi_K} P^{\pi_{K-1}} \cdots P^{\pi_0}) \bigr].$$

Using this notation, for any $(s,a) \in \mathcal{S} \times \mathcal{A}$, by (C.17) we have

$$\bigl| Q^*(s,a) - Q^{\pi_K}(s,a) \bigr| \le \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2} \cdot \biggl[\sum_{i=0}^{K-1} \alpha_i \cdot \bigl(O_i |\varrho_{i+1}| \bigr)(s,a) + \alpha_K \cdot \bigl(O_K |Q^* - \widetilde{Q}_0| \bigr)(s,a) \biggr], \tag{C.19}$$

where both $O_i|\varrho_{i+1}|$ and $O_K|Q^* - \widetilde{Q}_0|$ are functions defined on $\mathcal{S} \times \mathcal{A}$. Here (C.19) gives a uniform upper bound for $Q^* - Q^{\pi_K}$, which concludes the second step.

Step (iii): In this step, we conclude the proof by establishing an upper bound for $\|Q^* - Q^{\pi_K}\|_{1,\mu}$ based on (C.19). Here $\mu \in \mathcal{P}(\mathcal{S} \times \mathcal{A})$ is a fixed probability distribution. To simplify the notation, for any measurable function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, we denote $\mu(f)$ to be the expectation of f under μ, that is, $\mu(f) = \int_{\mathcal{S} \times \mathcal{A}} f(s,a) \mathrm{d}\mu(s,a)$. Using this notation, by (C.19) we bound $\|Q^* - Q^{\pi_\ell}\|_{1,\mu}$ by

$$\|Q^* - Q^{\pi_K}\|_{1,\mu} = \mu\bigl(|Q^* - Q^{\pi_K}| \bigr) \le \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2} \cdot \mu \biggl[\sum_{i=0}^{K-1} \alpha_i \cdot \bigl(O_i |\varrho_{i+1}| \bigr) + \alpha_K \cdot \bigl(O_K |Q^* - \widetilde{Q}_0| \bigr) \biggr]. \tag{C.20}$$

By the linearity of expectation, (C.20) implies

$$\|Q^* - Q^{\pi_K}\|_{1,\mu} \le \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2} \cdot \biggl[\sum_{i=0}^{K-1} \alpha_i \cdot \mu\bigl(O_i |\varrho_{i+1}| \bigr) + \alpha_K \cdot \mu\bigl(O_K |Q^* - \widetilde{Q}_0| \bigr) \biggr]. \tag{C.21}$$

Furthermore, since both Q^* and $\widetilde{Q}_0$ are bounded by $V_{\max} = R_{\max}/(1-\gamma)$ in ℓ_∞-norm, we have

$$\mu\big(O_K|Q^* - \widetilde{Q}_0|\big) \le 2 \cdot R_{\max}/(1-\gamma). \tag{C.22}$$

Moreover, for any $i \in \{0, \ldots, K-1\}$, by expanding $(1-\gamma P^{\pi_K})^{-1}$ into a infinite series, we have

$$\begin{aligned}\mu\big(O_i|\varrho_{i+1}|\big) &= \mu\Big\{\frac{1-\gamma}{2} \cdot (1-\gamma P^{\pi_K})^{-1}\big[(P^{\pi^*})^{K-i} + (P^{\pi_K} P^{\pi_{K-1}} \cdots P^{\pi_{i+1}})\big]|\varrho_{i+1}|\Big\} \\ &= \frac{1-\gamma}{2} \cdot \mu\Big\{\sum_{j=0}^{\infty} \gamma^j \cdot \big[(P^{\pi_K})^j (P^{\pi^*})^{K-i} + (P^{\pi_K})^{j+1}(P^{\pi_{K-1}} \cdots P^{\pi_{i+1}})\big]|\varrho_{i+1}|\Big\}. \end{aligned} \tag{C.23}$$

To upper bound the right-hand side of (C.23), we consider the following quantity

$$\mu\big[(P^{\pi_K})^j (P^{\tau_m} P^{\tau_{m-1}} \cdots P^{\tau_1}) f\big] = \int_{\mathcal{S}\times\mathcal{A}} \big[(P^{\pi_K})^j (P^{\tau_m} P^{\tau_{m-1}} \cdots P^{\tau_1}) f\big](s,a) \mathrm{d}\mu(s,a). \tag{C.24}$$

Here $\tau_1, \ldots, \tau_m$ are m policies. Recall that P^π is the transition operator of a Markov process defined on $\mathcal{S}\times\mathcal{A}$ for any policy π. Then the integral on the right-hand side of (C.24) corresponds to the expectation of the function $f(X_t)$, where $\{X_t\}_{t\ge 0}$ is a Markov process defined on $\mathcal{S}\times\mathcal{A}$. Such a Markov process has initial distribution $X_0 \sim \mu$. The first m transition operators are $\{P^{\tau_j}\}_{j\in[m]}$, followed by j identical transition operators P^{π_K}. Hence, $(P^{\pi_K})^j (P^{\tau_m} P^{\tau_{m-1}} \cdots P^{\tau_1})\mu$ is the marginal distribution of X_{j+m}, which we denote by $\widetilde{\mu}_j$ for notational simplicity. Hence, (C.24) takes the form

$$\mu\big[(P^{\pi_K})^j (P^{\tau_m} P^{\tau_{m-1}} \cdots P^{\tau_1}) f\big] = \mathbb{E}\big[f(X_{j+m})\big] = \widetilde{\mu}_j(f) = \int_{\mathcal{S}\times\mathcal{A}} f(s,a) \mathrm{d}\widetilde{\mu}_j(s,a) \tag{C.25}$$

for any measurable function f on $\mathcal{S}\times\mathcal{A}$. By Cauchy-Schwarz inequality, we have

$$\widetilde{\mu}_j(f) \le \bigg[\int_{\mathcal{S}\times\mathcal{A}} \Big|\frac{\mathrm{d}\widetilde{\mu}_j}{\mathrm{d}\sigma}(s,a)\Big|^2 \mathrm{d}\sigma(s,a)\bigg]^{1/2} \bigg[\int_{\mathcal{S}\times\mathcal{A}} |f(s,a)|^2 \mathrm{d}\sigma(s,a)\bigg]^{1/2}, \tag{C.26}$$

in which $\mathrm{d}\widetilde{\mu}_j/\mathrm{d}\sigma \colon \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ is the Radon-Nikodym derivative. Recall that the $(m+j)$-th order concentration coefficient $\kappa(m+j;\mu,\sigma)$ is defined in (4.4). Combining (C.25) and (C.26), we obtain

$$\widetilde{\mu}_j(f) \le \kappa(m+j;\mu,\sigma) \cdot \|f\|_\sigma.$$

Thus, by (C.23) we have

$$\begin{aligned}\mu\big(O_i|\varrho_{i+1}|\big) &= \frac{1-\gamma}{2} \cdot \sum_{j=0}^{\infty} \gamma^j \cdot \Big\{\mu\big[(P^{\pi_K})^j (P^{\pi^*})^{K-i}|\varrho_{i+1}|\big] + \mu\big[(P^{\pi_K})^{j+1}(P^{\pi_{K-1}} \cdots P^{\pi_{i+1}})|\varrho_{i+1}|\big]\Big\} \\ &\le (1-\gamma) \cdot \sum_{j=0}^{\infty} \gamma^j \cdot \kappa(K-i+j;\mu,\sigma) \cdot \|\varrho_{i+1}\|_\sigma. \end{aligned} \tag{C.27}$$

Now we combine (C.21), (C.22), and (C.27) to obtain

$$\begin{aligned}\|Q^* - Q^{\pi_K}\|_{1,\mu} &\le \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2} \cdot \bigg[\sum_{i=0}^{K-1} \alpha_i \cdot \mu\big(O_i|\varrho_{i+1}|\big) + \alpha_K \cdot \mu\big(O_K|Q^* - \widetilde{Q}_0|\big)\bigg] \\ &\le \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)} \cdot \bigg[\sum_{i=0}^{K-1}\sum_{j=0}^{\infty} \alpha_i \cdot \gamma^j \cdot \kappa(K-i+j;\mu,\sigma) \cdot \|\varrho_{i+1}\|_\sigma\bigg] + \frac{4\gamma(1-\gamma^{K+1})}{(1-\gamma)^3} \cdot \alpha_K \cdot R_{\max}. \end{aligned}$$

Recall that in Theorem 6.1 and (C.1) we define $\varepsilon_{\max} = \max_{i\in[K]} \|\varrho_i\|_\sigma$. We have that $\|Q^* - Q^{\pi_K}\|_{1,\mu}$ is further upper bounded by

$$
\begin{aligned}
&\|Q^* - Q^{\pi_K}\|_{1,\mu} \qquad\qquad (\text{C.28})\\
&\quad\le \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)} \cdot \biggl[\sum_{i=0}^{K-1}\sum_{j=0}^{\infty} \alpha_i \cdot \gamma^j \cdot \kappa(K-i+j;\mu,\sigma)\biggr] \cdot \varepsilon_{\max} + \frac{4\gamma(1-\gamma^{K+1})}{(1-\gamma)^3} \cdot \alpha_K \cdot R_{\max}\\
&\quad= \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)} \cdot \biggl[\sum_{i=0}^{K-1}\sum_{j=0}^{\infty} \frac{(1-\gamma)\gamma^{K-i-1}}{1-\gamma^{K+1}} \cdot \gamma^j \cdot \kappa(K-i+j;\mu,\sigma)\biggr] \cdot \varepsilon_{\max} + \frac{4\gamma^{K+1}}{(1-\gamma)^2} \cdot R_{\max},
\end{aligned}
$$

where the last equality follows from the definition of $\{\alpha_i\}_{0\le i\le K}$ in (C.18). We simplify the summation on the right-hand side of (C.28) and use Assumption 4.3 to obtain

$$
\begin{aligned}
&\sum_{i=0}^{K-1}\sum_{j=0}^{\infty} \frac{(1-\gamma)\gamma^{K-i-1}}{1-\gamma^{K+1}} \cdot \gamma^j \cdot \kappa(K-i+j;\mu,\sigma)\\
&\qquad= \frac{1-\gamma}{1-\gamma^{K+1}} \sum_{j=0}^{\infty}\sum_{i=0}^{K-1} \gamma^{K-i+j-1} \cdot \kappa(K-i+j;\mu,\sigma)\\
&\qquad\le \frac{1-\gamma}{1-\gamma^{K+1}} \sum_{m=0}^{\infty} \gamma^{m-1} \cdot m \cdot \kappa(m;\mu,\sigma) \le \frac{\phi_{\mu,\sigma}}{(1-\gamma^{K+1})(1-\gamma)}, \qquad (\text{C.29})
\end{aligned}
$$

where the last inequality follows from (4.5) in Assumption 4.3. Finally, combining (C.28) and (C.29), we obtain

$$
\|Q^* - Q^{\pi_K}\|_{1,\mu} \le \frac{2\gamma \cdot \phi_{\mu,\sigma}}{(1-\gamma)^2} \cdot \varepsilon_{\max} + \frac{4\gamma^{K+1}}{(1-\gamma)^2} \cdot R_{\max},
$$

which concludes the third step and hence the proof of Theorem 6.1. □

C.2 Proof of Theorem 6.2

Proof. Recall that in Algorithm 1 we define $Y_i = R_i + \gamma \cdot \max_{a\in\mathcal{A}} Q(S_{i+1}, a)$, where Q is any function in $\mathcal{F}$. By definition, we have $\mathbb{E}(Y_i \,|\, S_i = s, A_i = a) = (TQ)(s,a)$ for any $(s,a) \in \mathcal{S}\times\mathcal{A}$. Thus, TQ can be viewed as the underlying truth of the regression problem defined in (6.2), where the covariates and responses are $\{(S_i, A_i)\}_{i\in[n]}$ and $\{Y_i\}_{i\in[n]}$, respectively. Moreover, note that TQ is not necessarily in function class $\mathcal{F}$. We denote by Q^* the best approximation of TQ in $\mathcal{F}$, which is the solution to

$$
\operatorname*{minimize}_{f\in\mathcal{F}} \|f - TQ\|_\sigma^2 = \mathbb{E}\Bigl\{\bigl[f(S_i, A_i) - Q(S_i, A_i)\bigr]^2\Bigr\}. \qquad (\text{C.30})
$$

For notational simplicity, in the sequel we denote (S_i, A_i) by X_i for all $i \in [n]$. For any $f \in \mathcal{F}$, we define $\|f\|_n^2 = 1/n \cdot \sum_{i=1}^n [f(X_i)]^2$. Since both $\widehat{Q}$ and TQ are bounded by $V_{\max} = R_{\max}/(1-\gamma)$, we only need to consider the case where $\log N_\delta \le n$. Here N_δ is the cardinality of $\mathcal{N}(\delta, \mathcal{F}, \|\cdot\|_\infty)$. Moreover, let $f_1, \ldots, f_{N_\delta}$ be the centers of the minimal δ-covering of $\mathcal{F}$. Then by the definition of

δ-covering, there exists $k^* \in [N_\delta]$ such that $\|\widehat{Q} - f_{k^*}\|_\infty \le \delta$. It is worth mentioning that k^* is a random variable since $\widehat{Q}$ is obtained from data.

In the following, we prove (6.3) in two steps, which are bridged by $\mathbb{E}[\|\widehat{Q} - TQ\|_n^2]$.

Step (i): We relate $\mathbb{E}[\|\widehat{Q} - TQ\|_n^2]$ with its empirical counterpart $\|\widehat{Q} - TQ\|_n^2$. Recall that we define $Y_i = R_i + \gamma \cdot \max_{a \in \mathcal{A}} Q(S_{i+1}, a)$ for each $i \in [n]$. By the definition of $\widehat{Q}$, for any $f \in \mathcal{F}$ we have

$$\sum_{i=1}^n \bigl[Y_i - \widehat{Q}(X_i)\bigr]^2 \le \sum_{i=1}^n \bigl[Y_i - f(X_i)\bigr]^2. \tag{C.31}$$

For each $i \in [n]$, we define $\xi_i = Y_i - (TQ)(X_i)$. Then (C.31) can be written as

$$\|\widehat{Q} - TQ\|_n^2 \le \|f - TQ\|_n^2 + \frac{2}{n}\sum_{i=1}^n \xi_i \cdot \bigl[\widehat{Q}(X_i) - f(X_i)\bigr]. \tag{C.32}$$

Since both f and Q are deterministic, we have $\mathbb{E}(\|f - TQ\|_n^2) = \|f - TQ\|_\sigma^2$. Moreover, since $\mathbb{E}(\xi_i \,|\, X_i) = 0$ by definition, we have $\mathbb{E}[\xi_i \cdot g(X_i)] = 0$ for any bounded and measurable function g. Thus, it holds that

$$\mathbb{E}\biggl\{\sum_{i=1}^n \xi_i \cdot \bigl[\widehat{Q}(X_i) - f(X_i)\bigr]\biggr\} = \mathbb{E}\biggl\{\sum_{i=1}^n \xi_i \cdot \bigl[\widehat{Q}(X_i) - (TQ)(X_i)\bigr]\biggr\}. \tag{C.33}$$

In addition, by triangle inequality and (C.33), we have

$$\begin{aligned} &\biggl|\mathbb{E}\biggl\{\sum_{i=1}^n \xi_i \cdot \bigl[\widehat{Q}(X_i) - (TQ)(X_i)\bigr]\biggr\}\biggr| \\ &\qquad \le \biggl|\mathbb{E}\biggl\{\sum_{i=1}^n \xi_i \cdot \bigl[\widehat{Q}(X_i) - f_{k^*}(X_i)\bigr]\biggr\}\biggr| + \biggl|\mathbb{E}\biggl\{\sum_{i=1}^n \xi_i \cdot \bigl[f_{k^*}(X_i) - (TQ)(X_i)\bigr]\biggr\}\biggr|, \end{aligned} \tag{C.34}$$

where f_{k^*} satisfies $\|f_{k^*} - \widehat{Q}\|_\infty \le \delta$. In the following, we upper bound the two terms on the right-hand side of (C.34) respectively. For the first term, by applying Cauchy-Schwarz inequality twice, we have

$$\begin{aligned} \biggl|\mathbb{E}\biggl\{\sum_{i=1}^n \xi_i \cdot \bigl[\widehat{Q}(X_i) - f_{k^*}(X_i)\bigr]\biggr\}\biggr| &\le \sqrt{n} \cdot \biggl|\mathbb{E}\biggl[\biggl(\sum_{i=1}^n \xi_i^2\biggr)^{1/2} \cdot \|\widehat{Q} - f_{k^*}\|_n\biggr]\biggr| \\ &\le \sqrt{n} \cdot \biggl[\mathbb{E}\biggl(\sum_{i=1}^n \xi_i^2\biggr)\biggr]^{1/2} \cdot \bigl[\mathbb{E}\bigl(\|\widehat{Q} - f_{k^*}\|_n^2\bigr)\bigr]^{1/2} \le n\delta \cdot \bigl[\mathbb{E}(\xi_i^2)\bigr]^{1/2}, \end{aligned} \tag{C.35}$$

where we use the fact that $\{\xi_i\}_{i \in [n]}$ have the same marginal distributions and $\|\widehat{Q} - f_{k^*}\|_n \le \delta$. Since both Y_i and TQ are bounded by $V_{\max}$, ξ_i is a bounded random variable by its definition. Thus, there exists a constant $C_\xi > 0$ depending on ξ such that $\mathbb{E}(\xi_i^2) \le C_\xi^2 \cdot V_{\max}^2$. Then (C.35) implies

$$\biggl|\mathbb{E}\biggl\{\sum_{i=1}^n \xi_i \cdot \bigl[\widehat{Q}(X_i) - f_{k^*}(X_i)\bigr]\biggr\}\biggr| \le C_\xi \cdot V_{\max} \cdot n\delta. \tag{C.36}$$

It remains to upper bound the second term on the right-hand side of (C.34). We first define N_δ self-normalized random variables

$$Z_j = \frac{1}{\sqrt{n}} \sum_{i=1}^n \xi_i \cdot \bigl[f_j(X_i) - (TQ)(X_i)\bigr] \cdot \|f_j - (TQ)\|_n^{-1} \tag{C.37}$$

for all $j \in [N_\delta]$. Here recall that $\{f_j\}_{j\in[N_\delta]}$ are the centers of the minimal δ-covering of $\mathcal{F}$. Then we have

$$\begin{aligned} &\biggl|\mathbb{E}\biggl\{\sum_{i=1}^n \xi_i \cdot \bigl[f_{k^*}(X_i) - (TQ)(X_i)\bigr]\biggr\}\biggr| = \sqrt{n}\cdot \mathbb{E}\bigl[\|f_{k^*} - TQ\|_n \cdot |Z_{k^*}|\bigr] \\ &\quad \le \sqrt{n}\cdot\mathbb{E}\Bigl\{\bigl[\|\widehat{Q} - TQ\|_n + \|\widehat{Q} - f_{k^*}\|_n\bigr]\cdot |Z_{k^*}|\Bigr\} \le \sqrt{n}\cdot\mathbb{E}\Bigl\{\bigl[\|\widehat{Q} - TQ\|_n + \delta\bigr]\cdot |Z_{k^*}|\Bigr\}, \end{aligned} \tag{C.38}$$

where the first inequality follows from triangle inequality and the second inequality follows from the fact that $\|\widehat{Q} - f_{k^*}\|_\infty \le \delta$. Then applying Cauchy-Schwarz inequality to the last term on the right-hand side of (C.38), we obtain

$$\begin{aligned} \mathbb{E}\Bigl\{\bigl[\|\widehat{Q} - TQ\|_n + \delta\bigr]\cdot|Z_{k^*}|\Bigr\} &\le \biggl(\mathbb{E}\Bigl\{\bigl[\|\widehat{Q} - TQ\|_n + \delta\bigr]^2\Bigr\}\biggr)^{1/2} \cdot \bigl[\mathbb{E}(Z_{k^*}^2)\bigr]^{1/2} \\ &\le \biggl(\Bigl\{\mathbb{E}\bigl[\|\widehat{Q} - TQ\|_n^2\bigr]\Bigr\}^{1/2} + \delta\biggr)\cdot\Bigl[\mathbb{E}\bigl(\max_{j\in[N]} Z_j^2\bigr)\Bigr]^{1/2}, \end{aligned} \tag{C.39}$$

where the last inequality follows from

$$\mathbb{E}\bigl[\|\widehat{Q} - TQ\|_n\bigr] \le \Bigl\{\mathbb{E}\bigl[\|\widehat{Q} - TQ\|_n^2\bigr]\Bigr\}^{1/2}.$$

Moreover, since ξ_i is centered conditioning on $\{X_i\}_{i\in[n]}$ and is bounded by $2V_{\max}$, ξ_i is a sub-Gaussian random variable. In specific, there exists an absolute constant $H_\xi > 0$ such that $\|\xi_i\|_{\psi_2} \le H_\xi \cdot V_{\max}$ for each $i \in [n]$. Here the ψ_2-norm of a random variable $W \in \mathbb{R}$ is defined as

$$\|W\|_{\psi_2} = \sup_{p\ge 1} p^{-1/2}\bigl[\mathbb{E}(|W|^p)\bigr]^{1/p}.$$

By the definition of Z_j in (C.37), conditioning on $\{X_i\}_{i\in[n]}$, $\xi_i \cdot [f_j(X_i) - (TQ)(X_i)]$ is a centered and sub-Gaussian random variable with

$$\bigl\|\xi_i \cdot \bigl[f_j(X_i) - (TQ)(X_i)\bigr]\bigr\|_{\psi_2} \le H_\xi \cdot V_{\max} \cdot \bigl|f_j(X_i) - (TQ)(X_i)\bigr|.$$

Moreover, since Z_j is a summation of independent sub-Gaussian random variables, by Lemma 5.9 of Vershynin (2010), the ψ_2-norm of Z_j satisfies

$$\|Z_j\|_{\psi_2} \le C\cdot H_\xi \cdot V_{\max}\cdot \|f_j - TQ\|_n^{-1}\cdot\biggl[\frac{1}{n}\sum_{i=1}^n \bigl|[f_j(X_i) - (TQ)(X_i)]\bigr|^2\biggr]^{1/2} \le C\cdot H_\xi\cdot V_{\max},$$

where $C > 0$ is an absolute constant. Furthermore, by Lemmas 5.14 and 5.15 of Vershynin (2010), Z_j^2 is a sub-exponential random variable, and its the moment-generating function is bounded by

$$\mathbb{E}\bigl[\exp(t\cdot Z_j^2)\bigr] \le \exp(C\cdot t^2 \cdot H_\xi^4 \cdot V_{\max}^4) \tag{C.40}$$

for any t satisfying $C' \cdot |t| \cdot H_\xi^2 \cdot V_{\max}^2 \le 1$, where C and C' are two positive absolute constants. Moreover, by Jensen's inequality, we bound the moment-generating function of $\max_{j\in[N_\delta]} Z_j^2$ by

$$\mathbb{E}\Big[\exp\big(t \cdot \max_{j\in[N_\delta]} Z_j^2\big)\Big] \le \sum_{j\in[N_\delta]} \mathbb{E}\big[\exp(t\cdot Z_j^2)\big]. \tag{C.41}$$

Combining (C.40) and (C.41), we have

$$\mathbb{E}\big(\max_{j\in[N]} Z_j^2\big) \le C^2 \cdot H_\xi^2 \cdot V_{\max}^2 \cdot \log N_\delta, \tag{C.42}$$

where $C > 0$ is an absolute constant. Hence, plugging (C.42) into (C.38) and (C.39), we upper bound the second term of the right-hand side of (C.33) by

$$\begin{aligned}&\bigg|\mathbb{E}\bigg\{\sum_{i=1}^n \xi_i \cdot \big[f_{k^*}(X_i) - (TQ)(X_i)\big]\bigg\}\bigg| \\ &\qquad \le \bigg(\Big\{\mathbb{E}\big[\|\widehat{Q} - TQ\|_n^2\big]\Big\}^{1/2} + \delta\bigg) \cdot C \cdot H_\xi \cdot V_{\max} \cdot \sqrt{n \cdot \log N_\delta}.\end{aligned} \tag{C.43}$$

Finally, combining (C.32), (C.36) and (C.43), we obtain the following inequality

$$\begin{aligned}\mathbb{E}\big[\|\widehat{Q} - TQ\|_n^2\big] &\le \inf_{f\in\mathcal{F}} \mathbb{E}\big[\|f - TQ\|_n^2\big] + C_\xi \cdot V_{\max} \cdot \delta \\ &\qquad + \Big(\big\{\mathbb{E}\big[\|\widehat{Q} - (TQ)\|_n^2\big]\big\}^{1/2} + \delta\Big) \cdot C \cdot H_\xi \cdot V_{\max} \cdot \sqrt{\log N_\delta / n} \\ &\le C \cdot V_{\max}\sqrt{\log N_\delta/n} \cdot \Big\{\mathbb{E}\big[\|\widehat{Q} - (TQ)\|_n^2\big]\Big\}^{1/2} + \inf_{f\in\mathcal{F}} \mathbb{E}\big[\|f - TQ\|_n^2\big] + C' \cdot V_{\max}\delta,\end{aligned} \tag{C.44}$$

where C and C' are two positive absolute constants. Here in the first inequality we take the infimum over $\mathcal{F}$ because (C.31) holds for any $f \in \mathcal{F}$, and the second inequality holds because $\log N_\delta \le n$.

Now we invoke a simple fact to obtain the final bound for $\mathbb{E}[\|\widehat{Q} - TQ\|_n^2]$ from (C.44). Let a, b, and c be positive numbers satisfying $a^2 \le 2ab + c$. For any $\epsilon \in (0, 1]$, since $2ab \le \epsilon \cdot a^2/(1+\epsilon) + (1+\epsilon) \cdot b^2/\epsilon$, we have

$$a^2 \le (1+\epsilon)^2 \cdot b^2/\epsilon + (1+\epsilon)\cdot c. \tag{C.45}$$

Therefore, applying (C.45) to (C.44) with $a^2 = \mathbb{E}[\|\widehat{Q} - TQ\|_n^2]$, $b = C \cdot V_{\max} \cdot \sqrt{\log N_\delta/n}$, and $c = \inf_{f\in\mathcal{F}} \mathbb{E}[\|f - TQ\|_n^2] + C' \cdot V_{\max} \cdot \delta$, we obtain

$$\mathbb{E}\big[\|\widehat{Q} - TQ\|_n^2\big] \le (1+\epsilon) \cdot \inf_{f\in\mathcal{F}} \mathbb{E}\big[\|f - TQ\|_n^2\big] + C \cdot V_{\max}^2 \cdot \log N_\delta/(n\epsilon) + C' \cdot V_{\max} \cdot \delta, \tag{C.46}$$

where C and C' are two positive absolute constants. Now we conclude the first step.

Step (ii). In this step, we relate the population risk $\|\widehat{Q} - TQ\|_\sigma^2$ with $\mathbb{E}[\|\widehat{Q} - TQ\|_n^2]$, which is characterized in the first step. To begin with, we generate n i.i.d. random variables $\{\widetilde{X}_i = (\widetilde{S}_i, \widetilde{A}_i)\}_{i\in[n]}$ following σ, which are independent of $\{(S_i, A_i, R_i, S_i')\}_{i\in[n]}$. Since $\|\widehat{Q} - f_{k^*}\|_\infty \le \delta$, for any $x \in \mathcal{S} \times \mathcal{A}$, we have

$$\begin{aligned}&\Big|\big[\widehat{Q}(x) - (TQ)(x)\big]^2 - \big[f_{k^*}(x) - (TQ)(x)\big]^2\Big| \\ &\qquad = \big|\widehat{Q}(x) - f_{k^*}(x)\big| \cdot \big|\widehat{Q}(x) + f_{k^*}(x) - 2(TQ)(x)\big| \le 4V_{\max} \cdot \delta,\end{aligned} \tag{C.47}$$

where the last inquality follows from the fact that $\|TQ\|_\infty \le V_{\max}$ and $\|f\|_\infty \le V_{\max}$ for any $f \in \mathcal{F}$. Then by the definition of $\|\widehat{Q} - TQ\|_\sigma^2$ and (C.47), we have

$$\begin{aligned}
\|\widehat{Q} - TQ\|_\sigma^2 &= \mathbb{E}\biggl\{\frac{1}{n}\sum_{i=1}^n \bigl[\widehat{Q}(\widetilde{X}_i) - (TQ)(\widetilde{X}_i)\bigr]^2\biggr\} \\
&\le \mathbb{E}\biggl\{\|\widehat{Q} - TQ\|_n^2 + \frac{1}{n}\sum_{i=1}^n \bigl[f_{k^*}(\widetilde{X}_i) - (TQ)(\widetilde{X}_i)\bigr]^2 - \frac{1}{n}\sum_{i=1}^n \bigl[f_{k^*}(X_i) - (TQ)(\widetilde{X}_i)\bigr]^2\biggr\} + 8V_{\max}\cdot\delta \\
&= \mathbb{E}\bigl(\|\widehat{Q} - TQ\|_n^2\bigr) + \mathbb{E}\biggl[\frac{1}{n}\sum_{i=1}^n h_{k^*}(X_i, \widetilde{X}_i)\biggr] + 8V_{\max}\cdot\delta, \qquad \text{(C.48)}
\end{aligned}$$

where we apply (C.47) to obtain the first inequality, and in the last equality we define

$$h_j(x,y) = \bigl[f_j(y) - (TQ)(y)\bigr]^2 - \bigl[f_j(x) - (TQ)(x)\bigr]^2, \qquad \text{(C.49)}$$

for any $(x,y) \in \mathcal{S}\times\mathcal{A}$ and any $j \in [N_\delta]$. Note that h_{k^*} is a random function since k^* is random. By the definition of h_j in (C.49), we have $|h_j(x,y)| \le 4V_{\max}^2$ for any $(x,y) \in \mathcal{S}\times\mathcal{A}$ and $\mathbb{E}[h_j(X_i, \widetilde{X}_i)] = 0$ for any $i \in [n]$. Moreover, the variance of $h_j(X_i, \widetilde{X}_i)$ is upper bounded by

$$\begin{aligned}
\operatorname{Var}\bigl[h_j(X_i, \widetilde{X}_i)\bigr] &= 2\operatorname{Var}\Bigl\{\bigl[f_j(X_i) - (TQ)(X_i)\bigr]^2\Bigr\} \\
&\le 2\mathbb{E}\Bigl\{\bigl[f_j(X_i) - (TQ)(X_i)\bigr]^4\Bigr\} \le 8\Upsilon^2\cdot V_{\max}^2,
\end{aligned}$$

where we define Υ by letting

$$\Upsilon^2 = \max\Bigl(4V_{\max}^2\cdot\log N_\delta/n, \max_{j\in[N_\delta]} \mathbb{E}\Bigl\{\bigl[f_j(X_i) - (TQ)(X_i)\bigr]^2\Bigr\}\Bigr). \qquad \text{(C.50)}$$

Furthermore, we define

$$T = \sup_{j\in[N_\delta]} \biggl|\sum_{i=1}^n h_j(X_i, \widetilde{X}_i)/\Upsilon\biggr|. \qquad \text{(C.51)}$$

Combining (C.48) and (C.51), we obtain

$$\|\widehat{Q} - TQ\|_\sigma^2 \le \mathbb{E}\bigl[\|\widehat{Q} - TQ\|_n^2\bigr] + \Upsilon/n\cdot\mathbb{E}(T) + 8V_{\max}\cdot\delta. \qquad \text{(C.52)}$$

In the sequel, we utilize Bernstein's inequality to establish an upper bound for $\mathbb{E}(T)$, which is stated as follows for completeness.

Lemma C.3 (Bernstein's Inequality)**.** Let $U_1, \ldots U_n$ be n independent random variables satisfying $\mathbb{E}(U_i) = 0$ and $|U_i| \le M$ for all $i \in [n]$. Then for any $t > 0$, we have

$$\mathbb{P}\biggl(\biggl|\sum_{i=1}^n U_i\biggr| \ge t\biggr) \le 2\exp\biggl(\frac{-t^2}{2M\cdot t/3 + 2\sigma^2}\biggr),$$

where $\sigma^2 = \sum_{i=1}^n \operatorname{Var}(U_i)$ is the variance of $\sum_{i=1}^n U_i$.

We first apply Bernstein's inequality by setting $U_i = h_j(X_i, \widetilde{X}_i)/\Upsilon$ for each $i \in [n]$. Then we take a union bound for all $j \in [N_\delta]$ to obtain

$$\mathbb{P}(T \geq t) = \mathbb{P}\biggl[\sup_{j\in[N_\delta]} \frac{1}{n}\biggl|\sum_{i=1}^{n} h_j(X_i, \widetilde{X}_i)/\Upsilon\biggr| \geq t\biggr] \leq 2N_\delta \cdot \exp\biggl\{\frac{-t^2}{8V_{\max}^2 \cdot [t/(3\Upsilon) + n]}\biggr\}. \tag{C.53}$$

Since T is nonnegative, we have $\mathbb{E}(T) = \int_0^\infty \mathbb{P}(T \geq t)\mathrm{d}t$. Thus, for any $u \in (0, 3\Upsilon \cdot n)$, by (C.53) it holds that

$$\begin{aligned}\mathbb{E}(T) &\leq u + \int_u^\infty \mathbb{P}(T \geq t)\mathrm{d}t \leq u + 2N_\delta \int_u^{3\Upsilon\cdot n} \exp\biggl(\frac{-t^2}{16V_{\max}^2 \cdot n}\biggr)\,\mathrm{d}t + 2N_\delta \int_{3\Upsilon\cdot n}^\infty \exp\biggl(\frac{-3\Upsilon \cdot t}{16V_{\max}^2}\biggr)\,\mathrm{d}t \\ &\leq u + 32N_\delta \cdot V_{\max}^2 \cdot n/u \cdot \exp\biggl(\frac{-u^2}{16V_{\max}^2 \cdot n}\biggr) + 32N_\delta \cdot V_{\max}^2/(3\Upsilon) \cdot \exp\biggl(\frac{-9\Upsilon^2 \cdot n}{16V_{\max}^2}\biggr), \end{aligned} \tag{C.54}$$

where in the second inequality we use the fact that $\int_s^\infty \exp(-t^2/2)\mathrm{d}t \leq 1/s \cdot \exp(-s^2/2)$ for all $s > 0$. Now we set $u = 4V_{\max}\sqrt{n \cdot \log N_\delta}$ in (C.54) and plug in the definition of Υ in (C.50) to obtain

$$\mathbb{E}(T) \leq 4V_{\max}\sqrt{n \cdot \log N_\delta} + 8V_{\max}\sqrt{n/\log N_\delta} + 6V_{\max}\sqrt{n/\log N_\delta} \leq 8V_{\max}\sqrt{n \cdot \log N_\delta}, \tag{C.55}$$

where the last inequality holds when $\log N_\delta \geq 4$. Moreover, the definition of Υ in (C.50) implies that $\Upsilon \leq \max[2V_{\max}\sqrt{\log N_\delta/n}, \|\widehat{Q} - TQ\|_\sigma + \delta]$. In the following, we only need to consider the case where $\Upsilon \leq \|\widehat{Q} - TQ\|_\sigma + \delta$, since we already have (6.3) if $\|\widehat{Q} - TQ\|_\sigma + \delta \leq 2V_{\max}\sqrt{\log N_\delta/n}$, which concludes the proof.

Then, when $\Upsilon \leq \|\widehat{Q} - TQ\|_\sigma + \delta$ holds, combining (C.52) and (C.55) we obtain

$$\begin{aligned}\|\widehat{Q} - TQ\|_\sigma^2 &\leq \mathbb{E}\bigl[\|\widehat{Q} - TQ\|_n^2\bigr] + 8V_{\max}\sqrt{\log(N)/n} \cdot \|\widehat{Q} - TQ\|_\sigma + 8V_{\max}\sqrt{\log N_\delta/n} \cdot \delta + 8V_{\max} \cdot \delta \\ &\leq \mathbb{E}\bigl[\|\widehat{Q} - TQ\|_n^2\bigr] + 8V_{\max}\sqrt{\log N_\delta/n} \cdot \|\widehat{Q} - TQ\|_\sigma + 16V_{\max} \cdot \delta. \end{aligned} \tag{C.56}$$

We apply the inequality in (C.45) to (C.56) with $a = \|\widehat{Q} - TQ\|_\sigma$, $b = 8V_{\max}\sqrt{\log N_\delta/n}$, and $c = \mathbb{E}[\|\widehat{Q} - TQ\|_n^2] + 16V_{\max} \cdot \delta$. Hence we finally obtain that

$$\begin{aligned}\|\widehat{Q} - TQ\|_\sigma^2 &\leq (1 + \epsilon) \cdot \mathbb{E}\bigl[\|\widehat{Q} - TQ\|_n^2\bigr] \\ &\quad + (1 + \epsilon)^2 \cdot 64V_{\max} \cdot \log(N)/(n \cdot \epsilon) + (1 + \epsilon) \cdot 18V_{\max} \cdot \delta, \end{aligned} \tag{C.57}$$

which concludes the second step of the proof.

Finally, combining these two steps together, namely, (C.46) and (C.57), we conclude that

$$\|\widehat{Q} - TQ\|_\sigma^2 \leq (1 + \epsilon)^2 \cdot \inf_{f\in\mathcal{F}} \mathbb{E}\bigl[\|f - TQ\|_n^2\bigr] + C_1 \cdot V_{\max}^2 \cdot \log N_\delta/(n \cdot \epsilon) + C_2 \cdot V_{\max} \cdot \delta,$$

where C_1 and C_2 are two absolute constants. Moreover, since $Q \in \mathcal{F}$, we have

$$\inf_{f\in\mathcal{F}} \mathbb{E}\bigl[\|f - TQ\|_n^2\bigr] \leq \sup_{Q\in\mathcal{F}}\Bigl\{\inf_{f\in\mathcal{F}} \mathbb{E}\bigl[\|f - TQ\|_n^2\bigr]\Bigr\},$$

which concludes the proof of Theorem 6.2. □

D Proof of Theorem 5.4

In this section, we present the proof of Theorem 5.4. The proof is similar to that of Theorem 4.4, which is presented in §6 in details. In the following, we follow the proof in §6 and only highlight the differences for brevity.

Proof. The proof requires two key ingredients, namely the error propagation and the statistical error incurred by a single step of Minimax-FQI. We note that Perolat et al. (2015) establish error propagation for the state-value functions in the approximate modified policy iteration algorithm, which is more general than the FQI algorithm.

Theorem D.1 (Error Propagation)**.** Recall that $\{\widetilde{Q}_k\}_{0\leq k\leq K}$ are the iterates of Algorithm 2 and (π_K, ν_K) is the equilibrium policy with respect to $\widetilde{Q}_K$. Let Q^*_K be the action-value function corresponding to $(\pi_K, \nu^*_{\pi_K})$, where $\nu^*_{\pi_K}$ is the best-response policy of the second player against π_K. Then under Assumption 5.3, we have

$$\|Q^* - Q^*_K\|_{1,\mu} \leq \frac{2\phi_{\mu,\rho}\cdot\gamma}{(1-\gamma)^2}\cdot\varepsilon_{\max} + \frac{4\gamma^{K+1}}{(1-\gamma)^2}\cdot R_{\max}, \tag{D.1}$$

where we define the maximum one-step approximation error $\varepsilon_{\max} = \max_{k\in[K]} \|T\widetilde{Q}_{k-1} - \widetilde{Q}_k\|_\sigma$, and constant $\phi_{\mu,\nu}$ is specified in Assumption 5.3.

Proof. We note that the proof of Theorem 6.1 cannot be directly applied to prove this theorem. The main reason is that here we also need to consider the role played by the opponent, namely player two. Different from the MDP setting, here Q^*_K is a fixed point of a nonlinear operator due to the fact that player two adopts the optimal policy against π_K. Thus, we need to conduct a more refined analysis. See §D.1 for a detailed proof. □

By this theorem, we need to derive an upper bound of $\varepsilon_{\max}$. We achieve such a goal by studying the one-step approximation error $\|T\widetilde{Q}_{k-1} - \widetilde{Q}_k\|_\sigma$ for each $k\in[K]$.

Theorem D.2 (One-step Approximation Error)**.** Let $\mathcal{F}\subseteq\mathcal{B}(\mathcal{S}\times\mathcal{A}\times\mathcal{B}, V_{\max})$ be a family of measurable functions on $\mathcal{S}\times\mathcal{A}\times\mathcal{B}$ that are bounded by $V_{\max} = R_{\max}/(1-\gamma)$. Also, let $\{(S_i, A_i, B_i)\}_{i\in[n]}$ be n i.i.d. random variables following distribution $\sigma\in\mathcal{P}(\mathcal{S}\times\mathcal{A}\times\mathcal{B})$. . For each $i\in[n]$, let R_i and S'_i be the reward obtained by the first player and the next state following (S_i, A_i, B_i). In addition, for any fixed $Q\in\mathcal{F}$, we define the response variable as

$$Y_i = R_i + \gamma\cdot\max_{\pi'\in\mathcal{P}(\mathcal{A})}\min_{\nu'\in\mathcal{P}(\mathcal{B})}\mathbb{E}_{a\sim\pi', b\sim\nu'}\big[Q(S'_i, a, b)\big]. \tag{D.2}$$

Based on $\{(X_i, A_i, Y_i)\}_{i\in[n]}$, we define $\widehat{Q}$ as the solution to the least-squares problem

$$\min_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n\big[f(S_i, A_i) - Y_i\big]^2. \tag{D.3}$$

Then for any $\epsilon \in (0,1]$ and any $\delta > 0$, we have

$$\|\widehat{Q} - TQ\|_\sigma^2 \le (1+\epsilon)^2 \cdot \sup_{g \in \mathcal{F}} \inf_{f \in \mathcal{F}} \|f - Tg\|_\sigma^2 + C \cdot V_{\max}^2/(n \cdot \epsilon) \cdot \log N_\delta + C' \cdot V_{\max} \cdot \delta, \tag{D.4}$$

where C and C' are two positive absolute constants, T is the Bellman operator defined in (5.5), N_δ is the cardinality of the minimal δ-covering of $\mathcal{F}$ with respect to ℓ_∞-norm.

Proof. By the definition of Y_i in (D.2), for any $(s,a,b) \in \mathcal{S} \times \mathcal{A} \times \min_{\nu' \in \mathcal{P}(\mathcal{B})}$, we have

$$\begin{aligned}
&\mathbb{E}(Y_i \,|\, S_i = s, A_i = a, B_i = b) \\
&\quad = r(s,a) + \gamma \cdot \mathbb{E}_{s' \sim P(\cdot \,|\, s,a,b)} \Big\{ \max_{\pi' \in \mathcal{P}(\mathcal{A})} \min_{\nu' \in \mathcal{P}(\mathcal{B})} \mathbb{E}_{a' \sim \pi', b' \sim \nu'} \big[Q(s', a, b) \big] \Big\} = (TQ)(s,a,b).
\end{aligned}$$

Thus, TQ can be viewed as the ground truth of the nonlinear least-squares regression problem in (D.3). Therefore, following the same proof of Theorem 6.2, we obtain the desired result. □

Now we let $\mathcal{F}$ be the family of ReLU Q-networks $\mathcal{F}_1$ defined in (5.8) and set $Q = \widetilde{Q}_{k-1}$ in Theorem D.2. In addition, setting $\epsilon = 1$ and $\delta = 1/n$ in (D.4), we obtain

$$\begin{aligned}
\|\widetilde{Q}_{k+1} - T\widetilde{Q}_k\|_\sigma^2 &\le 4 \cdot \sup_{g \in \mathcal{F}_1} \inf_{f \in \mathcal{F}_1} \|f - Tg\|_\sigma^2 + C \cdot V_{\max}^2/n \cdot \log N_1 \\
&\le 4 \cdot \sup_{f' \in \mathcal{G}_1} \inf_{f \in \mathcal{F}_1} \|f - f'\|_\infty^2 + C \cdot V_{\max}^2/n \cdot \log N_1,
\end{aligned} \tag{D.5}$$

where C is a positive absolute constant, N_1 is the $1/n$-covering number of $\mathcal{F}_1$, and function class $\mathcal{G}_1$ is defined as

$$\mathcal{G}_1 = \big\{ f \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R} \colon f(\cdot, a, b) \in \mathcal{G}(\{p_j, t_j, \beta_j, H_j\}_{j \in [q]}) \text{ for any } (a,b) \in \mathcal{A} \times \mathcal{B} \big\}. \tag{D.6}$$

Here the second inequality follows from Assumption 5.2.

Thus, it remains to bound the ℓ_∞-error of approximating functions in $\mathcal{G}_1$ using ReLU Q-networks in $\mathcal{F}_1$ and the $1/n$-covering number of $\mathcal{F}_1$. In the sequel, obtain upper bounds for these two terms.

By the definition of $\mathcal{G}_1$ in (D.6), for any $f \in \mathcal{G}_1$ and any $(a,b) \in \mathcal{A} \times \mathcal{B}$, we have $f(\cdot, a, b) \in \mathcal{G}(\{(p_j, t_j, \beta_j, H_j)\}_{j \in [q]})$. Following the same construction as in §C.2, we can find a function $\widetilde{f}$ in $\mathcal{F}(L^*, \{d_j^*\}_{j=1}^{L^*+1}, s^*)$ such that

$$\|f(\cdot, a, b) - \widetilde{f}\|_\infty \lesssim \max_{j \in [q]} n^{-2\beta_j^*/(2\beta_j^* + t_j)} = n^{\alpha^* - 1},$$

which implies that

$$\sup_{f' \in \mathcal{G}_1} \inf_{f \in \mathcal{F}_1} \|f - f'\|_\infty^2 \lesssim n^{\alpha^* - 1}. \tag{D.7}$$

Moreover, for any $f \in \mathcal{F}_1$ and any $(a,b) \in \mathcal{A} \times \mathcal{B}$, we have $f(\cdot, a, b) \in \mathcal{F}(L^*, \{d_j^*\}_{j=1}^{L^*+1}, s^*)$. Let $\mathcal{N}_\delta$ be the δ-covering of $\mathcal{F}(L^*, \{d_j^*\}_{j=1}^{L^*+1}, s^*)$ in the ℓ_∞-norm. Then for any $f \in \mathcal{F}_1$ and any

$(a,b) \in \mathcal{A} \times \mathcal{B}$, there exists $g_{ab} \in \mathcal{N}_\delta$ such that $\|f(\cdot,a,b) - g_{a,b}\|_\infty \le \delta$. Thus, the cardinality of the $\mathcal{N}(\delta, \mathcal{F}_1, \|\cdot\|_\infty)$ satisfies

$$\big|\mathcal{N}(\delta, \mathcal{F}_1, \|\cdot\|)\big| \le |\mathcal{N}_\delta|^{|\mathcal{A}|\cdot|\mathcal{B}|}. \tag{D.8}$$

Combining (D.8) with Lemma 6.4 and setting $\delta = 1/n$, we obtain that

$$\begin{aligned} \log N_1 &\le |\mathcal{A}| \cdot |\mathcal{B}| \cdot \log |\mathcal{N}_\delta| \le |\mathcal{A}| \cdot |\mathcal{B}| \cdot (s^* + 1) \cdot \log\big[2n \cdot (L^* + 1) \cdot D^2\big] \\ &\le |\mathcal{A}| \cdot |\mathcal{B}| \cdot s^* \cdot L^* \max_{j \in [L^*]} \log(d_j^*) \lesssim |\mathcal{A}| \cdot |\mathcal{B}| \cdot n^{\alpha^*} \cdot (\log n)^{1+2\xi'}, \end{aligned} \tag{D.9}$$

where $D = \prod_{\ell=1}^{L^*+1}(d_\ell^* + 1)$ and the second inequality follows from (4.7).

Finally, combining (D.1), (D.5), (D.7), and (D.9), we conclude the proof of Theorem 5.4. □

D.1 Proof of Theorem D.1

Proof. The proof is similar to the that of Theorem D.1. Before presenting the proof, we first introduce the following notation for simplicity. For any $k \in \{0, \ldots, K-1\}$, we denote $T\widetilde{Q}_k$ by Q_{k+1} and define $\varrho_k = Q_k - \widetilde{Q}_k$. In addition, throughout the proof, for two action-value functions Q_1 and Q_2, we write $Q_1 \le Q_2$ if $Q_1(s,a,b) \ge Q_2(s,a,b)$ for any $(s,a,b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}$, and define $Q_1 \ge Q_2$ similarly. Furthermore, we denote by (π_k, ν_k) and (π^*, ν^*) the equilibrium policies with respect to $\widetilde{Q}_k$ by Q^*, respectively. Besides, in addition to the Bellman operators $T^{\pi,\nu}$ and T defined in (5.4) and (5.5), for any policy π of the first player, we define

$$T^\pi Q(s,a,b) = r(s,a,b) + \gamma \cdot \mathbb{E}_{s' \sim P(\cdot\,|\,s,a,b)}\Big\{\min_{\nu' \in \mathcal{P}(\mathcal{B})} \mathbb{E}_{a' \sim \pi, b' \sim \nu'}\big[Q(s',a',b')\big]\Big\}, \tag{D.10}$$

corresponds to the case where the first player follows policy π and player 2 adopts the best policy in response to π. By this definition, it holds that $Q^* = T^{\pi^*}Q^*$. Unlike the MDP setting, here T^π is a nonlinear operator due to the minimization in (D.10). Furthermore, for any fixed action-value function Q, we define the best-response policy against π with respect to Q, denote by $\nu(\pi, Q)$, as

$$\nu(\pi, Q)(\cdot\,|\,s) = \operatorname*{argmin}_{\nu' \in \mathcal{P}(\mathcal{B})} \mathbb{E}_{a \sim \pi, b \sim \nu'}\big[Q(s,a,b)\big]. \tag{D.11}$$

Using this notation, we can write (D.10) equivalently as

$$T^\pi Q(s,a,b) = r(s,a,b) + \gamma \cdot \big(P^{\pi,\nu(\pi,Q)}\big)(s,a,b).$$

Notice that $P^{\pi,\nu(\pi,Q)}$ is a linear operator and that $\nu_Q = \nu(\pi_Q, Q)$ by definition.

Now we are ready to present the proof, which can be decomposed into three key steps.

Step (i): In the first step, we establish recursive upper and lower bounds for $\{Q^* - \widetilde{Q}_k\}_{0 \le k \le K}$. For each $k \in \{0, \ldots, K-1\}$, similar to the decomposition in (C.3), we have

$$Q^* - \widetilde{Q}_{k+1} = Q^* - T^{\pi^*}\widetilde{Q}_k + (T^{\pi^*}\widetilde{Q}_k - T\widetilde{Q}_k) + \varrho_{k+1}, \tag{D.12}$$

where π^* is part of the equilibrium policy with respect to Q^* and T^{π^*} is defined in (D.10).

Similar to Lemma C.1, we utilize the following lemma to show $T^{\pi^*}\widetilde{Q}_k \ge T\widetilde{Q}_k$.

Lemma D.3. For any action-value function $Q : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathbb{R}$, let (π_Q, ν_Q) be the equilibrium policy with respect to Q. Then for and any policy π of the first player, it holds that

$$T^{\pi_Q} Q = TQ \geq T^{\pi} Q.$$

Furthermore, for any policy $\pi \colon \mathcal{S} \to \mathcal{P}(\mathcal{A})$ of player one and any action-value function Q, we have

$$T^{\pi, \nu(\pi, Q)} Q = T^{\pi} Q \leq T^{\pi, \nu} Q \tag{D.13}$$

for any policy $\nu \colon \mathcal{S} \to \mathcal{P}(\mathcal{B})$, where $\nu(\pi, Q)$ is the best-response policy defined in (D.11).

Proof. Note that for any $s' \in \mathcal{S}$, by the definition of equilibrium policy, we have

$$\max_{\pi' \in \mathcal{P}(\mathcal{A})} \min_{\nu' \in \mathcal{P}(\mathcal{B})} \mathbb{E}_{a' \sim \pi', b' \sim \nu'} \big[Q(s', a', b') \big] = \min_{\nu' \in \mathcal{P}(\mathcal{B})} \mathbb{E}_{a' \sim \pi_Q, b' \sim \nu'} \big[Q(s', a', b') \big].$$

Thus, for any state-action tuple (s, a, b), taking conditional expectations of s with respect to $P(\cdot \,|\, s, a, b)$ on both ends of this equation, we have

$$\begin{aligned} (T^{\pi_Q} Q)(s, a, b) &= r(s, a, b) + \gamma \cdot \mathbb{E}_{s' \sim P(\cdot \,|\, s, a, b)} \Big\{ \min_{\nu' \in \mathcal{P}(\mathcal{B})} \mathbb{E}_{a' \sim \pi_Q, b' \sim \nu'} \big[Q(s', a', b') \big] \Big\} \\ &= r(s, a, b) + \gamma \cdot \mathbb{E}_{s' \sim P(\cdot \,|\, s, a, b)} \Big\{ \max_{\pi' \in \mathcal{P}(\mathcal{A})} \min_{\nu' \in \mathcal{P}(\mathcal{B})} \mathbb{E}_{a' \sim \pi', b' \sim \nu'} \big[Q(s', a', b') \big] \Big\} = (TQ)(s, a, b), \end{aligned}$$

which proves $T^{\pi_Q} Q = TQ$. Moreover, for any policy π of the first player, it holds that

$$\max_{\pi' \in \mathcal{P}(\mathcal{A})} \min_{\nu' \in \mathcal{P}(\mathcal{B})} \mathbb{E}_{a' \sim \pi', b' \sim \nu'} \big[Q(s', a', b') \big] \geq \min_{\nu' \in \mathcal{P}(\mathcal{B})} \mathbb{E}_{a' \sim \pi, b' \sim \nu'} \big[Q(s', a', b') \big].$$

Taking expectations with respect to $s' \sim P(\cdot \,|\, s, a, b)$ on both ends, we establish $TQ \geq T^{\pi} Q$.

It remains to show the second part of Lemma D.3. By the definition of $\nu(\pi, Q)$, we have

$$\mathbb{E}_{a' \sim \pi, b' \sim \nu(\pi, Q)} \big[Q(s', a', b') \big] = \min_{\nu' \in \mathcal{P}(\mathcal{B})} \mathbb{E}_{a' \sim \pi, b' \sim \nu'} \big[Q(s', a', b') \big],$$

which, combined with the definition of T^{π} in (D.10), implies that $T^{\pi, \nu(\pi, Q)} Q = T^{\pi} Q$. Finally, for any policy ν of player two, we have

$$\min_{\nu' \in \mathcal{P}(\mathcal{B})} \mathbb{E}_{a' \sim \pi, b' \sim \nu'} \big[Q(s', a', b') \big] \geq \mathbb{E}_{a' \sim \pi, b' \sim \nu} \big[Q(s', a', b') \big],$$

which yields $T^{\pi} Q \leq T^{\pi, \nu} Q$. Thus, we conclude the proof of this lemma. □

Hereafter, for notational simplicity, for each k, let (π_k, ν_k) be the equilibrium joint policy with respect to $\widetilde{Q}_k$, and we denote $\nu(\pi^*, \widetilde{Q}_k)$ and $\nu(\pi_k, Q^*)$ by $\widetilde{\nu}_k$ and $\bar{\nu}_k$, respectively. Applying Lemma D.3 to (D.12) and utilizing the fact that $Q^* = T^{\pi^*} Q^*$, we have

$$\begin{aligned} Q^* - \widetilde{Q}_{k+1} &\leq (Q^* - T^{\pi^*} \widetilde{Q}_k) + \varrho_{k+1} = (T^{\pi^*} Q^* - T^{\pi^*} \widetilde{Q}_k) + \varrho_{k+1} \\ &\leq \big(T^{\pi^*, \widetilde{\nu}_k} Q^* - T^{\pi^*, \widetilde{\nu}_k} \widetilde{Q}_k \big) + \varrho_{k+1} = \gamma \cdot P^{\pi^*, \widetilde{\nu}_k} (Q^* - \widetilde{Q}_k) + \varrho_{k+1}, \end{aligned} \tag{D.14}$$

where the last inequality follows from (D.13). Furthermore, for a lower bound of $Q^* - \widetilde{Q}_{k+1}$, similar to (C.5), we have

$$\begin{aligned} Q^* - \widetilde{Q}_{k+1} &= (TQ^* - T^{\pi_k}Q^*) + (T^{\pi_k}Q^* - T^{\pi_k}\widetilde{Q}_k) + \varrho_{k+1} \\ &\geq (T^{\pi_k}Q^* - T^{\pi_k}\widetilde{Q}_k) + \varrho_{k+1} \geq \gamma \cdot P^{\pi_k,\bar{\nu}_k}(Q^* - \widetilde{Q}_k) + \varrho_{k+1}, \end{aligned} \tag{D.15}$$

where the both inequalities follow from Lemma D.3. Thus, combining (D.14) and (D.15) we have

$$\gamma \cdot P^{\pi^*,\widetilde{\nu}_k}(Q^* - \widetilde{Q}_k) + \varrho_{k+1} \geq Q^* - \widetilde{Q}_{k+1} \geq \gamma \cdot P^{\pi_k,\bar{\nu}_k}(Q^* - \widetilde{Q}_k) + \varrho_{k+1}. \tag{D.16}$$

for any $k \in \{0, \ldots, K-1\}$. Similar to the proof of Lemma C.2 , by applying recursion to (D.16), we obtain the following upper and lower bounds for the error propagation of Algorithm 2.

Lemma D.4 (Error Propagation)**.** For any $k, \ell \in \{0, 1, \ldots, K-1\}$ with $k < \ell$, we have

$$\begin{aligned} Q^* - \widetilde{Q}_\ell \leq \sum_{i=k}^{\ell-1} \gamma^{\ell-1-j} \cdot \big(P^{\pi^*,\widetilde{\nu}_{\ell-1}} P^{\pi^*,\widetilde{\nu}_{\ell-2}} \cdots P^{\pi^*,\widetilde{\nu}_{i+1}}\big) \varrho_{i+1} \\ + \gamma^{\ell-k} \cdot \big(P^{\pi^*,\widetilde{\nu}_{\ell-1}} P^{\pi^*,\widetilde{\nu}_{\ell-2}} \cdots P^{\pi^*,\widetilde{\nu}_k}\big)(Q^* - \widetilde{Q}_k), \end{aligned} \tag{D.17}$$

$$\begin{aligned} Q^* - \widetilde{Q}_\ell \geq \sum_{i=k}^{\ell-1} \gamma^{\ell-1-i} \cdot \big(P^{\pi_{\ell-1},\bar{\nu}_{\ell-1}} P^{\pi_{\ell-2},\bar{\nu}_{\ell-2}} \cdots P^{\pi_{i+1},\bar{\nu}_{i+1}}\big) \varrho_{i+1} \\ + \gamma^{\ell-k} \cdot \big(P^{\pi_{\ell-1},\bar{\nu}_{\ell-1}} P^{\pi_{\ell-2},\bar{\nu}_{\ell-2}} \cdots P^{\pi_{i+1},\bar{\nu}_k}\big)(Q^* - \widetilde{Q}_k). \end{aligned} \tag{D.18}$$

Proof. The desired results follows from applying the inequalities in (D.16) multiple times and the linearity of the operator $P^{\pi,\nu}$ for any joint policy (π, ν). □

The above lemma establishes recursive upper and lower bounds for the error terms $\{Q^* - \widetilde{Q}_k\}_{0 \leq k \leq K-1}$, which completes the first step of the proof.

Step (ii): In the second step, we characterize the suboptimality of the equilibrium policies constructed by Algorithm 2. Specifically, for each π_k, we denote by Q_k^* the action-value function obtained when agent one follows π_k while agent two adopt the best-response policy against π_k. In other words, Q_k^* is the fixed point of Bellman operator T^{π_k} defined in (D.10). In the following, we obtain an upper bound of $Q^* - Q_k^*$, which establishes the a notion of suboptimality of policy (π_k, ν_k) from the perspective of the first player.

To begin with, for any k, we first decompose $Q^* - Q_k^*$ by

$$Q^* - Q_k^* = \big(T^{\pi^*}Q^* - T^{\pi^*}\widetilde{Q}_k\big) + \big(T^{\pi^*}\widetilde{Q}_k - T^{\pi_k}\widetilde{Q}_k\big) + \big(T^{\pi_k}\widetilde{Q}_k - T^{\pi_k}Q_k^*\big). \tag{D.19}$$

Since π_k is the equilibrium policy with respect to $\widetilde{Q}_k$, by Lemma D.3, we have $T^{\pi^*}\widetilde{Q}_k \leq T^{\pi_k}\widetilde{Q}_k$. Recall that (π^*, ν^*) is the joint equilibrium policy with respect to Q^*. The second argument of Lemma D.3 implies that

$$T^{\pi^*}Q^* \leq T^{\pi^*,\widetilde{\nu}_k}Q^*, \qquad T^{\pi_k}\widetilde{Q}_k \leq T^{\pi_k,\widehat{\nu}_k}\widetilde{Q}_k, \tag{D.20}$$

where $\widetilde{\nu}_k = \nu(\pi^*, \widetilde{Q}_k)$ and we define $\widehat{\nu}_k = \nu(\pi_k, Q_k^*)$. Thus, combining (D.19) and (D.20) yields that

$$\begin{aligned} 0 \le Q^* - Q_k^* &\le \gamma \cdot P^{\pi^*, \widetilde{\nu}_k} \bigl(Q^* - \widetilde{Q}_k\bigr) + \gamma \cdot P^{\pi_k, \widehat{\nu}_k} \bigl(\widetilde{Q}_k - Q_k^*\bigr) \\ &= \gamma \cdot \bigl(P^{\pi^*, \widetilde{\nu}_k} - P^{\pi_k, \widehat{\nu}_k}\bigr)\bigl(Q^* - \widetilde{Q}_k\bigr) + \gamma \cdot P^{\pi_k, \widehat{\nu}_k} \bigl(Q^* - Q_k^*\bigr). \end{aligned} \tag{D.21}$$

Furthermore, since $I - \gamma \cdot P^{\pi_k, \widehat{\nu}_k}$ is invertible, by (D.21) we have

$$0 \le Q^* - Q_k^* \le \gamma \cdot \bigl(I - \gamma \cdot P^{\pi_k, \widehat{\nu}_k}\bigr)^{-1} \cdot \bigl[P^{\pi^*, \widetilde{\nu}_k} \bigl(Q^* - \widetilde{Q}_k\bigr) - P^{\pi_k, \widehat{\nu}_k} \bigl(Q^* - \widetilde{Q}_k\bigr)\bigr]. \tag{D.22}$$

Now we apply Lemma D.4 to the right-hand side of (D.22). Then for any $k \le \ell$, we have

$$\begin{aligned} P^{\pi^*, \widetilde{\nu}_\ell} \bigl(Q^* - \widetilde{Q}_\ell\bigr) \le \sum_{i=k}^{\ell-1} \gamma^{\ell-1-j} \cdot \bigl(P^{\pi^*, \widetilde{\nu}_\ell} P^{\pi^*, \widetilde{\nu}_{\ell-1}} \cdots P^{\pi^*, \widetilde{\nu}_{i+1}}\bigr) \varrho_{i+1} \\ + \gamma^{\ell-k} \cdot \bigl(P^{\pi^*, \widetilde{\nu}_\ell} P^{\pi^*, \widetilde{\nu}_{\ell-1}} \cdots P^{\pi^*, \widetilde{\nu}_k}\bigr)(Q^* - \widetilde{Q}_k), \end{aligned} \tag{D.23}$$

$$\begin{aligned} P^{\pi_\ell, \widehat{\nu}_\ell} \bigl(Q^* - \widetilde{Q}_\ell\bigr) \ge \sum_{i=k}^{\ell-1} \gamma^{\ell-1-i} \cdot \bigl(P^{\pi_\ell, \bar{\nu}_\ell} P^{\pi_\ell, \bar{\nu}_{\ell-2}} \cdots P^{\pi_{i+1}, \bar{\nu}_{i+1}}\bigr) \varrho_{i+1} \\ + \gamma^{\ell-k} \cdot \bigl(P^{\pi_\ell, \bar{\nu}_\ell} P^{\pi_{\ell-1}, \bar{\nu}_{\ell-1}} \cdots P^{\pi_k, \bar{\nu}_k}\bigr)(Q^* - \widetilde{Q}_k). \end{aligned} \tag{D.24}$$

Thus, setting $\ell = K$ and $k = 0$ in (D.23) and (D.24), we have

$$\begin{aligned} &Q^* - Q_K^* \le \bigl(I - \gamma \cdot P^{\pi_K, \widehat{\nu}_K}\bigr)^{-1} \cdot \\ &\qquad \biggl\{\sum_{i=0}^{K-1} \gamma^{K-1} \cdot \Bigl[\bigl(P^{\pi^*, \widetilde{\nu}_K} P^{\pi^*, \widetilde{\nu}_{K-1}} \cdots P^{\pi^*, \widetilde{\nu}_{i+1}}\bigr) - \bigl(P^{\pi_K, \bar{\nu}_K} P^{\pi_{K-1}, \bar{\nu}_{K-1}} \cdots P^{\pi_{i+1}, \bar{\nu}_{i+1}}\bigr)\Bigr] \varrho_{i+1} \\ &\quad + \gamma^{K+1} \cdot \Bigl[\bigl(P^{\pi^*, \widetilde{\nu}_K} P^{\pi^*, \widetilde{\nu}_{K-1}} \cdots P^{\pi^*, \widetilde{\nu}_0}\bigr) - \bigl(P^{\pi_K, \bar{\nu}_K} P^{\pi_{K-1}, \bar{\nu}_{K-1}} \cdots P^{\pi_0, \bar{\nu}_0}\bigr)\Bigr](Q^* - \widetilde{Q}_0)\biggr\}. \end{aligned} \tag{D.25}$$

To simplify the notation, we define $\{\alpha_i\}_{i=0}^K$ as in (C.18). Note that we have $\sum_{i=0}^K \alpha_i = 1$ by definition. Moreover, we define $K+1$ linear operators $\{O_k\}_{k=0}^K$ as follows. For any $i \le K-1$, let

$$O_i = \frac{1-\gamma}{2} \cdot (I - \gamma \cdot P^{\pi_K, \widehat{\nu}_K})^{-1} \Bigl[\bigl(P^{\pi^*, \widetilde{\nu}_K} P^{\pi^*, \widetilde{\nu}_{K-1}} \cdots P^{\pi^*, \widetilde{\nu}_{i+1}}\bigr) - \bigl(P^{\pi_K, \bar{\nu}_K} P^{\pi_{K-1}, \bar{\nu}_{K-1}} \cdots P^{\pi_{i+1}, \bar{\nu}_{i+1}}\bigr)\Bigr].$$

Moreover, we define O_K by

$$O_K = \frac{1-\gamma}{2} \cdot (I - \gamma \cdot P^{\pi_K, \widehat{\nu}_K})^{-1} \Bigl[\bigl(P^{\pi^*, \widetilde{\nu}_K} P^{\pi^*, \widetilde{\nu}_{K-1}} \cdots P^{\pi^*, \widetilde{\nu}_0}\bigr) - \bigl(P^{\pi_K, \bar{\nu}_K} P^{\pi_{K-1}, \bar{\nu}_{K-1}} \cdots P^{\pi_0, \bar{\nu}_0}\bigr)\Bigr].$$

Therefore, taking absolute values on both sides of (D.25), we obtain that

$$\begin{aligned} &\bigl|Q^*(s,a,b) - Q_K^*(s,a,b)\bigr| \\ &\qquad \le \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2} \cdot \biggl[\sum_{i=0}^{K-1} \alpha_i \cdot \bigl(O_i |\varrho_{i+1}|\bigr)(s,a,b) + \alpha_K \cdot \bigl(O_K |Q^* - \widetilde{Q}_0|\bigr)(s,a,b)\biggr], \end{aligned} \tag{D.26}$$

for any $(s, a, b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}$, which concludes the second step of the proof.

Step (iii): We note that (D.26) is nearly the same as (C.19) for the MDP setting. Thus, in the last step, we follow the same proof strategy as in **Step (iii)** in §C.1. For notational simplicity, for any function $f : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ and any probability distribution $\mu \in \mathcal{P}(\mathcal{S} \times \mathcal{A} \times \mathcal{B})$, we denote the expectation of f under μ by $\mu(f)$. By taking expectation with respect to μ in (D.26), we have

$$\|Q^* - Q_K^*\|_{1,\mu} \le \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)^2} \cdot \biggl[\sum_{i=0}^{K-1} \alpha_i \cdot \mu\bigl(O_i|\varrho_{i+1}|\bigr) + \alpha_K \cdot \mu\bigl(O_K|Q^* - \widetilde{Q}_0|\bigr)\biggr]. \tag{D.27}$$

By the definition of O_i, we can write $\mu(O_i|\varrho_{i+1}|)$ as

$$\begin{aligned}\mu\bigl(O_i|\varrho_{i+1}|\bigr) = \frac{1-\gamma}{2} \cdot \mu\biggl\{\sum_{j=0}^{\infty} \gamma^j \cdot \Bigl[\bigl(P^{\pi_K, \widehat{\nu}_K}\bigr)^j \bigl(P^{\pi^*, \widetilde{\nu}_K} P^{\pi^*, \widetilde{\nu}_{K-1}} \cdots P^{\pi^*, \widetilde{\nu}_{i+1}}\bigr) \\ + \bigl(P^{\pi_K, \widehat{\nu}_K}\bigr)^j \bigl(P^{\pi_K, \bar{\nu}_K} P^{\pi_{K-1}, \bar{\nu}_{K-1}} \cdots P^{\pi_{i+1}, \bar{\nu}_{i+1}}\bigr)\Bigr] |\varrho_{i+1}|\biggr\}.\end{aligned} \tag{D.28}$$

To upper bound the right-hand side of (D.28), we consider the following quantity

$$\mu\bigl[P^{\tau_m} \cdots P^{\tau_1})f\bigr] = \int_{\mathcal{S}\times\mathcal{A}\times\mathcal{B}} \bigl[(P^{\tau_m} P^{\tau_{m-1}} \cdots P^{\tau_1})f\bigr](s,a,b) \mathrm{d}\mu(s,a,b),$$

where $\{\tau_t : \mathcal{S} \to \mathcal{P}(\mathcal{A} \times \mathcal{B})\}_{t \in [m]}$ are m joint policies of the two-players. By Cauchy-Schwarz inequality, it holds that

$$\begin{aligned}\mu\bigl[P^{\tau_m} \cdots P^{\tau_1})f\bigr] \le \biggl[\int_{\mathcal{S}\times\mathcal{A}\times\mathcal{B}} \biggl|\frac{\mathrm{d}(P^{\tau_m} P^{\tau_{m-1}} \cdots P^{\tau_1} \mu)}{\mathrm{d}\sigma}(s,a,b)\biggr|^2 \mathrm{d}\sigma(s,a,b)\biggr]^{1/2} \\ \cdot \biggl[\int_{\mathcal{S}\times\mathcal{A}\times\mathcal{B}} |f(s,a,b)|^2 \mathrm{d}\sigma(s,a,b)\biggr]^{1/2} \le \kappa(m;\mu,\sigma) \cdot \|f\|_\sigma,\end{aligned}$$

where $\kappa(m;\mu,\sigma)$ is the m-th concentration parameter defined in (5.9). Thus, by (D.28)we have

$$\mu\bigl(O_i|\varrho_{i+1}|\bigr) \le (1-\gamma) \cdot \sum_{j=0}^{\infty} \gamma^j \cdot \kappa(K-i+j;\mu,\nu) \cdot \|\varrho_{i+1}\|_\sigma. \tag{D.29}$$

Besides, since both Q^* and $\widetilde{Q}_0$ are bounded by $R_{\max}/(1-\gamma)$ in ℓ_∞-norm, we have

$$\mu\bigl(O_K|Q^* - \widetilde{Q}_0|\bigr) \le 2 \cdot R_{\max}/(1-\gamma). \tag{D.30}$$

Finally, combining (D.27), (D.29), and (D.30), we obtain that

$$\begin{aligned}&\|Q^* - Q^{\pi_K}\|_{1,\mu} \\ &\quad\le \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)} \cdot \biggl[\sum_{i=0}^{K-1}\sum_{j=0}^{\infty} \alpha_i \cdot \gamma^j \cdot \kappa(K-i+j;\mu,\nu) \cdot \|\varrho_{i+1}\|_\sigma\biggr] + \frac{4\gamma(1-\gamma^{K+1})}{(1-\gamma)^3} \cdot \alpha_K \cdot R_{\max} \\ &\quad\le \frac{2\gamma(1-\gamma^{K+1})}{(1-\gamma)} \cdot \biggl[\sum_{i=0}^{K-1}\sum_{j=0}^{\infty} \frac{(1-\gamma)\gamma^{K-i-1}}{1-\gamma^{K+1}} \cdot \gamma^j \cdot \kappa(K-i+j;\mu,\nu)\biggr] \cdot \varepsilon_{\max} + \frac{4\gamma^{K+1}}{(1-\gamma)^2} \cdot R_{\max},\end{aligned}$$

where the last inequality follows from the fact that $\varepsilon_{\max} = \max_{i\in[K]} \|\varrho_i\|_\sigma$. Note that in (C.29) we show that it holds under Assumption 5.3 that

$$\sum_{i=0}^{K-1}\sum_{j=0}^{\infty} \frac{(1-\gamma)\gamma^{K-i-1}}{1-\gamma^{K+1}} \cdot \gamma^j \cdot \kappa(K-i+j;\mu,\nu) \le \frac{\phi_{\mu,\nu}}{(1-\gamma^{K+1})(1-\gamma)}.$$

Hence, we obtain (D.1) and thus conclude the proof of Theorem D.1. □

References

Allen-Zhu, Z., Li, Y. and Liang, Y. (2018a). Learning and generalization in overparameterized neural networks, going beyond two layers. *arXiv preprint arXiv:1811.04918*.

Allen-Zhu, Z., Li, Y. and Song, Z. (2018b). A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:1811.03962*.

Anthony, M. and Bartlett, P. L. (2009). *Neural network learning: Theoretical foundations*. Cambridge University Press.

Antos, A., Szepesvári, C. and Munos, R. (2007). Value-iteration based fitted policy iteration: learning with a single trajectory. In *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*. IEEE.

Antos, A., Szepesvári, C. and Munos, R. (2008a). Fitted Q-iteration in continuous action-space mdps. In *Advances in neural information processing systems*.

Antos, A., Szepesvári, C. and Munos, R. (2008b). Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, **71** 89–129.

Arora, S., Du, S. S., Hu, W., Li, Z. and Wang, R. (2019). Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *arXiv preprint arXiv:1901.08584*.

Arulkumaran, K., Deisenroth, M. P., Brundage, M. and Bharath, A. A. (2017). A brief survey of deep reinforcement learning. *arXiv preprint arXiv:1708.05866*.

Baird, L. (1995). Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*. Elsevier, 30–37.

Bartlett, P. L. (1998). The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE transactions on Information Theory*, **44** 525–536.

Bartlett, P. L., Foster, D. J. and Telgarsky, M. J. (2017a). Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*.

Bartlett, P. L., Harvey, N., Liaw, C. and Mehrabian, A. (2017b). Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *arXiv preprint arXiv:1703.02930*.

Bartlett, P. L., Maiorov, V. and Meir, R. (1999). Almost linear VC dimension bounds for piecewise polynomial networks. In *Advances in Neural Information Processing Systems*.

Bauer, B., Kohler, M. et al. (2019). On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *The Annals of Statistics*, **47** 2261–2285.

Bellemare, M. G., Naddaf, Y., Veness, J. and Bowling, M. (2013). The Arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, **47** 253–279.

Bowling, M. (2001). Rational and convergent learning in stochastic games. In *International Conference on Artificial Intelligence*.

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

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

Cao, Y. and Gu, Q. (2019). A generalization theory of gradient descent for learning over-parameterized deep relu networks. *arXiv preprint arXiv:1902.01384*.

Chen, H., Liu, X., Yin, D. and Tang, J. (2017). A survey on dialogue systems: Recent advances and new frontiers. *arXiv preprint arXiv:1711.01731*.

Chizat, L. and Bach, F. (2018). A note on lazy training in supervised differentiable programming. *arXiv preprint arXiv:1812.07956*.

Conitzer, V. and Sandholm, T. (2007). AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, **67** 23–43.

Du, S. S., Lee, J. D., Li, H., Wang, L. and Zhai, X. (2018a). Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*.

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

Dulac-Arnold, G., Evans, R., Sunehag, P. and Coppin, B. (2015). Reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679*.

Dziugaite, G. K. and Roy, D. M. (2017). Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*.

Ernst, D., Geurts, P. and Wehenkel, L. (2005). Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, **6** 503–556.

Fan, J., Ma, C. and Zhong, Y. (2019). A selective overview of deep learning. *arXiv preprint arXiv:1904.05526*.

Farahmand, A.-m., Ghavamzadeh, M., Szepesvári, C. and Mannor, S. (2009). Regularized fitted Q-iteration for planning in continuous-space Markovian decision problems. In *American Control Conference*. IEEE.

Farahmand, A.-m., Ghavamzadeh, M., Szepesvári, C. and Mannor, S. (2016). Regularized policy iteration with nonparametric function spaces. *The Journal of Machine Learning Research*, **17** 4809–4874.

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

Geist, M., Scherrer, B. and Pietquin, O. (2019). A theory of regularized markov decision processes. *arXiv preprint arXiv:1901.11275*.

Golowich, N., Rakhlin, A. and Shamir, O. (2017). Size-independent sample complexity of neural networks. *arXiv preprint arXiv:1712.06541*.

Haarnoja, T., Tang, H., Abbeel, P. and Levine, S. (2017). Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*.

Han, S., Mao, H. and Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*.

Hausknecht, M. and Stone, P. (2015). Deep recurrent Q-learning for partially observable mdps. In *2015 AAAI Fall Symposium Series*.

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

Kawaguchi, K., Kaelbling, L. P. and Bengio, Y. (2017). Generalization in deep learning. *arXiv preprint arXiv:1710.05468*.

Klusowski, J. M. and Barron, A. R. (2016). Risk bounds for high-dimensional ridge function combinations including neural networks. *arXiv preprint arXiv:1607.01434*.

Kober, J. and Peters, J. (2012). Reinforcement learning in robotics: A survey. In *Reinforcement Learning*. Springer, 579–610.

Lagoudakis, M. G. and Parr, R. (2002). Value function approximation in zero-sum markov games. In *Uncertainty in Artificial Intelligence*.

Lagoudakis, M. G. and Parr, R. (2003). Least-squares policy iteration. *Journal of machine learning research*, **4** 1107–1149.

Lange, S., Gabel, T. and Riedmiller, M. (2012). Batch reinforcement learning. In *Reinforcement learning*. Springer, 45–73.

Lazaric, A., Ghavamzadeh, M. and Munos, R. (2012). Finite-sample analysis of least-squares policy iteration. *Journal of Machine Learning Research*, **13** 3041–3074.

Lazaric, A., Ghavamzadeh, M. and Munos, R. (2016). Analysis of classification-based policy iteration algorithms. *The Journal of Machine Learning Research*, **17** 583–612.

Liang, T., Poggio, T., Rakhlin, A. and Stokes, J. (2017). Fisher-Rao metric, geometry, and complexity of neural networks. *arXiv preprint arXiv:1711.01530*.

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

Lin, L.-J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, **8** 293–321.

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

Liu, B., Wang, M., Foroosh, H., Tappen, M. and Pensky, M. (2015). Sparse convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition*.

Liu, R. and Zou, J. (2017). The effects of memory replay in reinforcement learning. *arXiv preprint arXiv:1710.06574*.

Ma, C., Wu, L. et al. (2019). A comparative analysis of the optimization and generalization property of two-layer neural network and random feature models under gradient descent dynamics. *arXiv preprint arXiv:1904.04326*.

Maass, W. (1994). Neural nets with superlinear VC-dimension. *Neural Computation*, **6** 877–884.

Mei, S., Misiakiewicz, T. and Montanari, A. (2019). Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. *arXiv preprint arXiv:1902.06015*.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D. and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. et al. (2015). Human-level control through deep reinforcement learning. *Nature*, **518** 529–533.

Mohri, M., Rostamizadeh, A. and Talwalkar, A. (2012). *Foundations of machine learning*. MIT Press.

Munos, R. and Szepesvári, C. (2008). Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, **9** 815–857.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *International conference on machine learning*.

Neyshabur, B., Bhojanapalli, S., McAllester, D. and Srebro, N. (2017a). Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*.

Neyshabur, B., Bhojanapalli, S., McAllester, D. and Srebro, N. (2017b). A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*.

Neyshabur, B., Salakhutdinov, R. R. and Srebro, N. (2015a). Path-SGD: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*.

Neyshabur, B., Tomioka, R. and Srebro, N. (2015b). Norm-based capacity control in neural networks. In *Conference on Learning Theory*.

Patek, S. D. (1997). *Stochastic and shortest path games: theory and algorithms*. Ph.D. thesis, Massachusetts Institute of Technology.

Pérolat, J., Piot, B., Geist, M., Scherrer, B. and Pietquin, O. (2016a). Softened approximate policy iteration for markov games. In *International Conference on Machine Learning*.

Pérolat, J., Piot, B. and Pietquin, O. (2018). Actor-critic fictitious play in simultaneous move multistage games. In *International Conference on Artificial Intelligence and Statistics*.

Pérolat, J., Piot, B., Scherrer, B. and Pietquin, O. (2016b). On the use of non-stationary strategies for solving two-player zero-sum markov games. In *International Conference on Artificial Intelligence and Statistics*.

Perolat, J., Scherrer, B., Piot, B. and Pietquin, O. (2015). Approximate dynamic programming for two-player zero-sum markov games. In *International Conference on Machine Learning (ICML 2015)*.

Prasad, H., LA, P. and Bhatnagar, S. (2015). Two-timescale algorithms for learning Nash equilibria in general-sum stochastic games. In *International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems.

Rahimi, A. and Recht, B. (2008). Random features for large-scale kernel machines. In *Advances in neural information processing systems*.

Riedmiller, M. (2005). Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*. Springer.

Schaul, T., Quan, J., Antonoglou, I. and Silver, D. (2016). Prioritized experience replay. *International Conference on Learning Representations*.

Scherrer, B., Ghavamzadeh, M., Gabillon, V., Lesner, B. and Geist, M. (2015). Approximate modified policy iteration and its application to the game of tetris. *Journal of Machine Learning Research*, **16** 1629–1676.

Schmidt-Hieber, J. (2017). Nonparametric regression using deep neural networks with ReLU activation function. *arXiv preprint arXiv:1708.06633*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M. and Moritz, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning*.

Shapley, L. S. (1953). Stochastic games. *Proceedings of the national academy of sciences*, **39** 1095–1100.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, **529** 484–489.

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

Srinivasan, S., Lanctot, M., Zambaldi, V., Pérolat, J., Tuyls, K., Munos, R. and Bowling, M. (2018). Actor-critic policy optimization in partially observable multiagent environments. In *Advances in Neural Information Processing Systems*.

Stone, C. J. (1982). Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics* 1040–1053.

Sutton, R. S. and Barto, A. G. (2011). *Reinforcement learning: An introduction*. MIT Press.

Sutton, R. S., McAllester, D. A., Singh, S. P. and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*.

Suzuki, T. (2018). Adaptivity of deep relu network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality. *arXiv preprint arXiv:1810.08033*.

Tagorti, M. and Scherrer, B. (2015). On the rate of convergence and error bounds for LSTD (λ). In *International Conference on Machine Learning*.

Tosatto, S., Pirotta, M., D'Eramo, C. and Restelli, M. (2017). Boosted fitted Q-iteration. In *International Conference on Machine Learning*.

van Hasselt, H., Guez, A. and Silver, D. (2016). Deep reinforcement learning with double Q-learning. In *AAAI Conference on Artificial Intelligence*.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*.

Von Neumann, J. and Morgenstern, O. (1947). *Theory of games and economic behavior*. Princeton University Press.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M. and Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. In *International Conference on Machine Learning*.

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

Wei, C.-Y., Hong, Y.-T. and Lu, C.-J. (2017). Online reinforcement learning in stochastic games. In *Advances in Neural Information Processing Systems*.

Yehudai, G. and Shamir, O. (2019). On the power and limitations of random features for understanding neural networks. *arXiv preprint arXiv:1904.00687*.

Zhang, K., Yang, Z., Liu, H., Zhang, T. and Başar, T. (2018). Finite-sample analyses for fully decentralized multi-agent reinforcement learning. *arXiv preprint arXiv:1812.02783*.

Zhang, S. and Sutton, R. S. (2017). A deeper look at experience replay. *arXiv preprint arXiv:1712.01275*.

Zou, D., Cao, Y., Zhou, D. and Gu, Q. (2018). Stochastic gradient descent optimizes over-parameterized deep relu networks. *arXiv preprint arXiv:1811.08888*.