Understanding Deep Neural Function Approximation in Reinforcement Learning with ϵ-Greedy

Anonymous Author(s)
Affiliation
Address
email

Abstract

This paper provides a theoretical study of deep neural function approximation in reinforcement learning (RL) employed with the ϵ-greedy exploration under the online setting. This problem setting is motivated by the successful deep Q-networks (DQN) framework that works in this regime. We provide an initial attempt on theoretical understanding deep RL from the perspective of function class and neural networks architectures (e.g., width and depth). In this work, we focus on value based algorithm with the ϵ-greedy exploration under Besov and Barron function spaces, which aims at approximating an α-smooth Q-function in a d-dimensional feature space with T episodes. We prove that scaling the width $m = \widetilde{\mathcal{O}}(T^{\frac{d}{2\alpha+d}})$ and the depth $L = \widetilde{\mathcal{O}}(1)$ of the neural network for deep RL is sufficient for learning with sublinear regret in Besov spaces. Moreover, for a two layer neural network in the Barron space, scaling the width even in sublinearly in T is sufficient. In particular, for low smooth Q-function, the theory suggests that the depth should be chosen as the logarithm of the width. This matches the practical DQN setting. Our analysis relies on transforming the estimation of temporal difference error to a generalization problem under the conditionally independent but non-identically distributed data setting over the averaged empirical measure. This might have its own interest in reinforcement learning theory for better understanding ϵ-greedy exploration in deep RL.

1 Introduction

Efficient reinforcement learning (RL) under the large (or even infinite) state space and action space setting is increasingly important and relevant challenge [1, 2, 3]. One of the first successful approaches towards this problem is the deep Q-network (DQN) [4, 5] framework, which deploys powerful nonlinear function approximation techniques via Deep Neural Networks (DNNs) [6] to concisely approximate state and action spaces.

Despite its impressive practical success, there is still a gap between practical uses and theoretical understanding on DQN that is equipped with the ϵ-greedy policy under the online setting. In contrast, many works directly focus on linear function approximation [7, 8], linear mixtures [9, 10] with confidence-bound based scheme for exploration. In fact, the ϵ-greedy policy is a popular choice in deep RL even though it has exponential sample complexity in the worst case [11].

Existing non-linear function approximation results on RL are largely based on neural tangent kernel (NTK) [12, 13], Bellman rank [14, 15], and Eluder dimension [16, 17, 18]. Nevertheless, these approaches fail in truly capturing the highly non-linear properties of DQN. Indeed, NTK (or lazy

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

training [19]) essentially works in a linear regime [20, 21, 22], and can not efficiently learn even a single ReLU neuron [23, 24, 25] as it requires $\Omega(\varepsilon^{-d})$ samples to achieve ε approximation error, where d is the (original) or transformed feature dimension input.

Interestingly, even for two-layer neural networks, the corresponding Eluder dimension is at least in an exponential order [26, 27], which appears difficult to fully demonstrate the success of practical deep RL both theoretically and empirically. Regarding to ensuring a sublinear regret, the width of neural networks under the NTK regime [13] is required to be $m = \Omega(T^{13})$, where T is the number of episodes, which does not match the used DQN in practice with small width/depth under large episodes [4, 28].

To bridge the large theory-practice gap, we study value iteration based on deep neural function approximation with the ϵ-greedy policy under the online setting, which broadly capture the key features of deep RL. Our analysis framework is equipped with two-layer as well as deep fully-connected neural networks in the respective Barron and Besov spaces, which can fully capture the properties of Q-functions, e.g., smoothness by neural networks.

Besides, our results allow for reasonably finite width and depth in practice, give precise characterization of these parameters to obtain sublinear regret, and demonstrate the relationship between the problem-dependent smoothness of Q-function and regret bounds. These results could also motivate practitioners to consider different architectures of implementations of deep RL.

1.1 Technical challenges and contributions

Most previous RL theory results on function approximation in the online setting work with "optimism in the face of uncertainty" principle for exploration, leading to a series of upper confidence bound (UCB)-type algorithms. However, conceptually, optimism is sometimes too aggressive and UCB-style algorithms can suffer exponential sample complexity even for nonlinear bandits [27]. Empirical model-based RL has also not found optimism to be very useful [29].

In the RL setting, linear/kernel function approximation [7, 13, 30] depends on a known feature mapping or the NTK matrix to design a UCB-type bonus function for optimism, which appears invalid for DNNs as the used confidence ellipsoid and elliptical potential lemma are not applicable for data-dependent feature mapping. To avoid explicitly designing a bonus function, Thompson sampling [31, 32] appears promising in a Bayesian perspective by using randomized (i.e., perturbed) versions of the estimated model or value function [33]. Nevertheless, the bonus function is still implicitly included in confidence estimate of perturbations in their UCB-style algorithm.

This work analyzes a value iteration algorithm with ϵ-greedy by neural networks. Designing a provably efficient exploration mechanism is beyond the scope of our work but it remains an interesting future direction.

The technical contribution in our analysis is how to control the temporal difference (TD) error without bonus function design. In this case, the ϵ-greedy exploration is not enough to ensure the TD error negative as conducted in UCB-style algorithms.

By a measure transform, we analyze the TD error in an L^2-integrable space wrt historical state-action pairs, introduce the δ-covering of the value function set to break the dependence between the episodes[1], and transform to generalization bounds under the independent but non-identically distributed data setting. To this end, we need to employ proper function spaces beyond linear/kernel Markov decision processes (MDP) in our analysis: one is the Barron space [34], as the largest[2] function space for two-layer ReLU neural networks without curse of dimensionality [35]; the other is the Besov space [36, 37] for deep ReLU neural networks, which allows for continuous/non-continuous functions and thus is more general than Hölder spaces used in offline RL [38]. Under these proper hypothesis

[1] In fact, the indpendence requirement can be achieved by the target networks in DQN.

[2] The terminology "largest" means two-layer neural networks with bounded norm and functions in the Barron space can be efficiently represented without curse of dimensionality, refer to Appendix A for details.

spaces, such generalization problem can be addressed by uniform convergence via Rademacher complexity of these spaces under the averaged empirical measure.

Under attainable assumptions, our results show that (i) the problem-dependent smoothness of Q-function affects the deep Q-learning efficiency, which can be improved by the model complexity. We use α as a parameter indicating the smoothness degree of Q-function. A larger α indicates smoother functions. (ii) for two-layer neural networks, the width $m = \Omega(\sqrt{T})$ suffices to ensure sublinear regret. (iii) for deep neural networks, the width $m = \widetilde{\mathcal{O}}(T^{\frac{d}{2\alpha+d}})$ and the depth $L = \widetilde{\mathcal{O}}(1)$ are enough for sublinear regret, where $\widetilde{\mathcal{O}}(\cdot)$ omits the $\log$ terms. Furthermore, our regret bounds can be (nearly) independent of the feature dimension d, supporting the premise of practical, high-dimensional deep RL.

1.2 Related work

Recent work on neural network function approximation beyond NTK (or the Eluder dimension) mainly restrict on the generative setting [39, 40] by assuming a simulator in which the agent can require any state and action, and the offline setting [38, 41]. In sequel, we review RL with function approximation under the online setting that DQN falls into this regime. We also mention that, theoretical understanding of DQN can be conducted by Q learning [42], neural fitted Q-iteration algorithm [43, 38, 44], and analyses of the target network [45].

RL with linear/kernel function approximation: RL with linear function approximation results in regret bounds $\widetilde{\mathcal{O}}(\sqrt{d^3H^3T})$ [7] and $\widetilde{\mathcal{O}}(dH^2\sqrt{T})$ [30], where H is the length of each episode; and linear mixtures with regret $\widetilde{\mathcal{O}}(d\sqrt{H^3T})$ [9] and improved to $\widetilde{\mathcal{O}}(dH\sqrt{T})$ [10]; kernel regime with regret $\widetilde{\mathcal{O}}(\delta_{\mathcal{F}}H^2\sqrt{T})$ [30, 13], where $\delta_{\mathcal{F}}$ is the intrinsic complexity (e.g., effective dimension) of the function class RKHS $\mathcal{F}$. The above bounds are based on confidence ellipsoid to quantify the uncertainty in an explicit bonus function by feature mapping/kernel matrix; while Thompson sampling [32, 31] utilizes an implicit bonus function in probability estimation on uncertainty quantification, which leads to $\widetilde{\mathcal{O}}(d^2H^2\sqrt{T})$ [33] regret in linear function approximation.

RL with general function approximation: One prototypical scheme uses the Eluder dimension [16], which measures the degree of dependence among action rewards, resulting in $\widetilde{\mathcal{O}}(\texttt{poly}(\delta_{\mathcal{F}}H)\sqrt{T})$ regret [17, 18], where the complexity $\delta_{\mathcal{F}}$ depends on the Eluder dimension. The sublinear regret under ϵ-greedy exploration can be achieved by [46] using Eluder dimension. Bellman rank [14] is also another metric for general function approximation [15] by measuring the error of the function class under the Bellman operator. Jin et al. [47] introduce a new metric, Bellman Eluder dimension, by combining these two metrics, and achieve $\widetilde{\mathcal{O}}(H\sqrt{\delta_{\mathcal{F}}T})$, where $\delta_{\mathcal{F}}$ depends on this metric.

Overall, none of the above metrics can handle the nonlinear spaces that concern us.

2 Background and preliminaries

In this section, we provide some preliminaries with respect to online reinforcement learning based on episodic MDPs, function spaces of two (and deep)-layer ReLU neural networks.

Notation: We denote by $a(n) \lesssim b(n)$: there exists a positive constant c independent of n such that $a(n) \leqslant cb(n)$. We use the shorthand $[n] := \{1, 2, \ldots, n\}$ for some positive n. Let $\mathcal{X} = [0, 1]^d$ be a domain of the functions, we denote the L^p-integrable space by $L^p(\mathcal{X})$ endowed by the norm $\|f\|_{L^p(\mathcal{X})} = \big(\int_{\mathcal{X}} |f(\boldsymbol{x})|^p \mathrm{d}\boldsymbol{x}\big)^{1/p}$, and the μ-integrable L^p space by $L^p(\mathrm{d}\mu)$ for a probability measure μ on $\mathcal{X}$ and the norm is given by $\|f\|_{L^p(\mathrm{d}\mu)} = \big(\int_{\mathcal{X}} |f(\boldsymbol{x})|^p \mathrm{d}\mu(\boldsymbol{x})\big)^{1/p}$.

2.1 Episodic Markov decision processes

A (finite-horizon) episodic MDPs is denoted as $\mathrm{MDP}(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$, where $\mathcal{S}$ is the state space with possibly infinite states; $\mathcal{A}$ is the action space, often with finite actions; H is the number of steps in each episode; $\mathbb{P} := \{\mathbb{P}_h\}_{h=1}^H$ is the Markov transition kernel with the transition probability $\mathbb{P}_h(\cdot|s, a)$ on action a taken at state $s \in \mathcal{S}$ in the h-th step; the reward functions $r := \{r_h\}_{h=1}^H$ are assumed to be

deterministic. For notational simplicity, denote $\mathcal{X} = \mathcal{S} \times \mathcal{A}$ and $\boldsymbol{x} = (s, a)$, we assume $\mathcal{X} = [0, 1]^d$ as a compact space of $\mathbb{R}^d$ and $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$ at h-th step.

A non-stationary policy π is a collection of H functions $\pi := \{\pi_h : \mathcal{S} \to \mathcal{A}\}_{h=1}^H$. Given a policy π, the (state) value function $V_h^\pi \colon \mathcal{S} \to [0, H]$ is defined as the expected cumulative reward of the MDP starting from step $h \in [H]$, i.e., $V_h^\pi(s) = \mathbb{E}_\pi\big[\sum_{h'=h}^H r_{h'}(s_{h'}, a_{h'})\big| s_h = s\big], \forall s \in \mathcal{S}, h \in [H]$ where $\mathbb{E}_\pi[\cdot]$ denotes the expectation with respect to the randomness of the trajectory $\{(s_h, a_h)\}_{h=1}^H$ obtained by the policy π. Likewise, the action-value function $Q_h^\pi : \mathcal{S} \times \mathcal{A} \to [0, H]$ is defined as $Q_h^\pi(s, a) = \mathbb{E}_\pi\big[\sum_{h'=h}^H r_{h'}(s_{h'}, a_{h'}) \,\big|\, s_h = s,\, a_h = a\big]$.

Moreover, since the action space and episode length are both finite, there always exists an optimal policy $\pi^\star$ such that $V_h^\star(s) = \sup_\pi V_h^\pi(s)$ for all $s \in \mathcal{S}$ and $h \in [H]$ [48]. To simplify the notation, denote $(\mathbb{P}_h V)(s, a) := \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,a)}[V(s')]$ and the Bellman operator $(\mathbb{T}_h V)(s, a) = r_h(s, a) + (\mathbb{P}_h V)(s, a)$ for any measurable function $V \colon \mathcal{S} \to [0, H]$. Using this notation, the Bellman equation associated with a policy π can be formulated as

$$Q_h^\pi(s, a) = (\mathbb{T}_h V_{h+1}^\pi)(s, a), \qquad V_h^\pi(s) = \langle Q_h^\pi(s, \cdot), \pi_h(\cdot|s)\rangle_{\mathcal{A}}, \qquad V_{H+1}^\pi(s) = 0\,. \tag{1}$$

Similarly, the Bellman optimality equation is given by

$$Q_h^\star(s, a) = (\mathbb{T}_h V_{h+1}^\star)(s, a), \qquad V_h^\star(s) = \max_{a \in \mathcal{A}} Q_h^\star(s, a), \qquad V_{H+1}^\star(s) = 0\,. \tag{2}$$

Accordingly, the optimal policy $\pi^\star$ is the greedy policy with respect to $\{Q_h^\star\}_{h=1}^H$. Due to $r_h \in [0, 1]$ yielding $Q_h \in [0, H - h + 1]$, we introduce the *projection operator* $\mathcal{K}(f)(s_h, a_h) := \min\{f(s_h, a_h), H-h+1\}_+$ in our analysis to obtain the tighter bound. Hence the Bellman optimality operator $\mathbb{T}_h^\star$ is defined as

$$(\mathbb{T}_h^\star Q)(s_h, a_h) = r_h(s_h, a_h) + \mathbb{E}_{s_{h+1} \sim \mathbb{P}_h(\cdot|s_h,a_h)} \mathcal{K}[\max_{a \in \mathcal{A}} Q(s_{h+1}, a)], \qquad \forall\, Q : \mathcal{S} \times \mathcal{A} \to [0, H]\,.$$

By definition, the Bellman equation in Eq. (2) is equivalent to $Q_h^\star = \mathbb{T}_h^\star Q_{h+1}^\star, \forall h \in [H]$.

In the **online setting**, the goal is to learn the optimal policy $\pi^\star$ by minimizing the cumulative regret under the interaction with the environment over a number of episodes. For any policy π, the difference between V_1^π and $V_1^\star$ quantifies its sub-optimality. Thus, after T (fixed but large) episodes, the total (expected) regret is defined as $\mathrm{Regret}(T) = \sum_{t=1}^T \big[V_1^\star(s_1^t) - V_1^{\pi^t}(s_1^t)\big]$, where π^t is the policy executed in the t-th episode and s_1^t is the initial state.

2.2 Barron spaces and Besov spaces

We begin with introduction on the Barron space associated with two-layer ReLU neural networks and then give an overview of Besov spaces for deep ReLU neural networks. More details refer to Appendix A. For description simplicity, we focus on the ReLU activation function in this work.

Barron spaces: A two-layer neural network with m neurons can be represented as[3] $f(\boldsymbol{x}) = \frac{1}{m}\sum_{k=1}^m b_k \sigma(\boldsymbol{w}_k^\top \boldsymbol{x})$, where $\sigma(x) = \max\{x, 0\}$ is the ReLU activation function used in this work. If the parameters $\{(b_k, \boldsymbol{w}_k)\}_{k=1}^m$ are associated with a signed Radon measure with finite total variation [23, 49], this function space is in fact a Barron space $\widetilde{\mathcal{P}}$ [34], of which the Barron norm can be roughly substituted by with the ℓ_1-path norm, i.e., $\|f\|_{\widetilde{\mathcal{P}}} \leqslant \|f\|_{\mathcal{P}} := \frac{1}{m}\sum_{k=1}^m |b_k| \|\boldsymbol{w}_k\|_1 \leqslant 2\|f\|_{\widetilde{\mathcal{P}}}$. Accordingly, it is natural to use $\|f\|_{\mathcal{P}}$ to denote the Barron norm, as the discrete version.

The Barron space can be regarded as the *largest* function space for two layer neural networks in two folds [34, 35]: 1) *direct approximation:* Any function in Barron spaces can be efficiently approximated by two-layer neural networks with bounded ℓ_1-path norm; 2) *inverse approximation:* Any continuous function that can be efficiently approximated by two-layer neural networks with bounded ℓ_1-path norm belongs to a Barron space.

Besov spaces: Different from two-layer neural networks in Barron spaces, the complete function space of deep ReLU neural networks is still open. Previous work in approximation theory focuses

[3] It allows for neural networks with bias, by taking $(\boldsymbol{x}, 1) \in \mathbb{R}^{d+1}$ as $\boldsymbol{x}$ and $(\boldsymbol{w}, b) \in \mathbb{R}^{d+1}$ as $\boldsymbol{w}$.

on the "smoothness" of the function, e.g., Hölder spaces [50, 38] and Sobolev spaces [51]. Here we consider the concept of α-smooth from modulus of smoothness [37], *cf.*, Appendix A.

Based on this, we consider a more general function space beyond Hölder spaces and Sobolev spaces, i.e., Besov spaces [36, 37], which allows for spatially inhomogeneous smoothness with spikes and jumps. The Besov space is defined by $\mathcal{B}^{\alpha}_{p,q}(\mathcal{X}) = \{f \in L^p(\mathcal{X}) \mid \|f\|_{\mathcal{B}^{\alpha}_{p,q}} < \infty\}$, where the Besov norm is $\|f\|_{\mathcal{B}^{\alpha}_{p,q}} := \|f\|_{L^p(\mathcal{X})} + |f|_{\mathcal{B}^{\alpha}_{p,q}}$. The smoothness parameter α indicates which function at a certain smoothness degree can be represented. For example, if $\alpha > d/p$, then the related Besov space is continuously embedded in the set of the continuous functions; if $\alpha < d/p$, then the functions in the Besov space are no longer continuous. The formal definition and relations to Hölder spaces and Sobolev spaces are deferred to Appendix A.

Mixed smooth Besov spaces: Learning in Besov spaces [52] still faces the *curse of dimensionality* since the Besov space is too general. To tackle this issue, we use the mixed Besov space $\widetilde{\mathcal{B}}^{\alpha}_{p,q}(\mathcal{X})$ [53, 37], as defined by $\|f\|_{\widetilde{\mathcal{B}}^{\alpha}_{p,q}} := \|f\|_{L^p(\mathcal{X})} + |f|_{\widetilde{\mathcal{B}}^{\alpha}_{p,q}}$, and consider the differentiation over all coordinates, which does not heavily depend on the feature dimension d. Intuitively, the Besov space ensures a function with boundedness of derivatives, of which the "sum" of the orders are smaller than α; while the mixed smooth Besov space allows for the "max" of the orders of derivatives over coordinates smaller than α. Hence, we obtain the approximation performance and the separation on the *dimension-dependent* convergence rate for sublinear regret.

3 Value iteration via neural networks under ϵ-greedy exploration

In this section, we lay out our algorithm 1 via value iteration by DNNs under the ϵ-greedy policy. After initialization, at t-th episode, at h-th time step, we have observed $t-1$ transition tuples, $\{(s_h^\tau, a_h^\tau, s_{h+1}^\tau)\}_{\tau=1}^{t-1}$ and attempt to estimate $\{Q_h^\star\}_{h=1}^H$ by the following least squares regression problem $\widehat{Q}_h^t = \operatorname{argmin}_{f\in\mathcal{F}} \widehat{\mathcal{E}}_h^t(f)$ with

$$\widehat{\mathcal{E}}_h^t(f) = \frac{1}{t-1}\sum_{\tau=1}^{t-1}\left[\mathcal{K}f(s_h^\tau, a_h^\tau) - r_h(s_h^\tau, a_h^\tau) - V_{h+1}^t(s_{h+1}^\tau)\right]^2 , \tag{3}$$

where $\mathcal{K}$ is the projection operator defined before for tight bound. The function class $\mathcal{F}$ is given by $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_H$, including $\mathcal{F}_{\texttt{SNN}}$ for two-layer (Shallow) ReLU neural networks and $\mathcal{F}_{\texttt{DNN}}$ for deep ReLU neural networks as below.

$$\mathcal{F}_{\texttt{SNN}} = \left\{ f : [0,1]^d \to \mathbb{R} \middle| f(\boldsymbol{x}) = \frac{1}{m}\sum_{k=1}^m b_k \sigma(\boldsymbol{w}_k^\top \boldsymbol{x} + c_k), \|f\|_{\mathcal{P}} \leqslant B \right\} , \tag{4}$$

where $B > 0$ is the ℓ_1-path norm constraint parameter, and deep ReLU neural networks [37] as

$$\begin{aligned}\mathcal{F}_{\texttt{DNN}}(L, m, S, B) := \Bigg\{ f : [0,1]^d \to \mathbb{R} \Bigg| f(\boldsymbol{x}) = (\boldsymbol{W}^{(L)}\sigma(\cdot) + b^{(L)}) \circ \cdots \circ (\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)}) , \\ \sum_{i=1}^L (\|\boldsymbol{W}^{(i)}\|_0 + \|\boldsymbol{b}^{(i)}\|_0) \leqslant S, \ \max_i \|\boldsymbol{W}^{(i)}\|_\infty \vee \|\boldsymbol{b}^{(i)}\|_\infty \leqslant B \Bigg\} ,\end{aligned} \tag{5}$$

where the weight parameters are $\boldsymbol{W}^{(1)} \in \mathbb{R}^{m\times d}$, $\boldsymbol{W}^l \in \mathbb{R}^{m\times m}$, $\forall l \in \{2, 3, \ldots, L-1\}$, and $\boldsymbol{W}^L \in \mathbb{R}^m$; the bias parameter are $\boldsymbol{b}^{(l)} \in \mathbb{R}^m$, $\forall l \in [L-1]$ and $b^{(L)} \in \mathbb{R}$. The depth L, the width m, the sparsity parameter S and the norm parameter B can be further determined to achieve good approximation and estimation performance.

In our setting, we use the historical $t-1$ episodic data $\{(s_h^\tau, a_h^\tau, s_{h+1}^\tau)\}_{\tau=1}^{t-1}$ for value function approximation by neural networks. It can be found that, for each time step h, these $t-1$ examples are merely conditionally independent snf non-identically distributed. Formally, we assume each state-action pair is sampled from a respective (unknown) probability measure, i.e., $(s_h^\tau, a_h^\tau) \sim \mu_h^\tau, \forall \tau \in [t-1]$, where $\mu_h^\tau \in \mathscr{P}(\mathcal{S}\times\mathcal{A})$ is from the collection of all probability distribution on $\mathcal{S}\times\mathcal{A}$. Taking the averaged measure $\bar{\mu}_h^t := \frac{1}{t-1}\sum_{\tau=1}^{t-1}\mu_h^\tau$, the expected risk of Eq. (3) is defined as

$$\mathcal{E}_h(f) = \mathbb{E}_{(s_h,a_h)\sim\bar{\mu}_h^t, s_{h+1}\sim\mathbb{P}_h(\cdot|s_h,a_h)}\left[\mathcal{K}f(s_h,a_h) - r_h(s_h,a_h) - V_{h+1}^t(s_{h+1})\right]^2 . \tag{6}$$

Algorithm 1 Value Iteration via neural networks under ϵ-greedy exploration

Input: Function class $\mathcal{F}$, the number of episodes T, and the parameter $\epsilon \in (0,1)$.
for episode $t = 1, \ldots, T$ **do**
  Receive the initial state s_1^t.
  Set V_{H+1}^t as the zero function.
  for step $h = H, \ldots, 1$ **do**
    Obtain $\widehat{Q}_h^t := \operatorname{argmin}_{f \in \mathcal{F}} \sum_{\tau=1}^{t-1} \left[\mathcal{K}f(s_h^\tau, a_h^\tau) - r_h(s_h^\tau, a_h^\tau) - V_{h+1}^t(s_{h+1}^\tau)\right]^2$.
    Obtain $Q_h^t = \min\{\widehat{Q}_h^t, H-h+1\}^+$ and $V_h^t(\cdot) = \max_{a \in \mathcal{A}} Q_h^t(\cdot, a)$.
  end for
  //ϵ-`greedy for exploration`
  Take the policy $\{\pi_h^t\}_{h=1}^H$ to be greedy policy with probability $1-\epsilon$ or any policy with probability ϵ.
  for step $h = 1, \ldots, H$ **do**
    Take $\tilde{a}_h^t \sim \pi_h^t(\cdot|s_h^t)$.
    Observe the reward $r_h(s_h^t, \tilde{a}_h^t)$ and obtain the next state s_{h+1}^t.
  end for
end for

Note that, Eq. (3) is a biased estimator of Eq. (6) due to an extra variance term caused by the stochastic transition [54]. However, this issue can be avoided by introducing extra ε-net of the value function set [7] in theory. In practice, DQN [4] uses target networks to reduce the biased estimate or even eliminate it [38], and some variants [55] can also reduce the biased estimate and performs well without target networks. Accordingly, both practical uses and theoretical guarantees provide a good justification to not strictly distinguish this biased estimation in our analysis for better understanding deep RL.

4 Theoretical guarantees of efficient value iteration via DNNs

In this section, we present our results for value iteration achieving sublinear regret under two-layer and deep ReLU neural networks.

4.1 Efficient value iteration under two-layer ReLU neural networks

In this setting, we consider $\widehat{Q}_h^t = \operatorname{argmin}_{f \in \mathcal{F}_{\texttt{SNN}}} \widehat{\mathcal{E}}_h^t(f)$ in Eq. (3), where $\mathcal{F}_{\texttt{SNN}}$ is the function space of two-layer ReLU neural networks defined in Eq. (4). We give an assumption for two-layer ReLU neural networks, where the Bellman optimality operator maps any bounded value function to a bounded Barron space ball. This assumption is commonly used in RL theory under various function classes [7, 13].

Assumption 1. *Let $\widetilde{R} > 0$ be a fixed constant. Define $\mathcal{P}_{\widetilde{R}} = \{f \in \mathcal{P} : \|f\|_{\mathcal{P}} \leq \widetilde{R}\}$ in the Barron space, and assume that for any $h \in [H]$ and $Q\colon \mathcal{S} \times \mathcal{A} \to [0, H]$, we have $\mathbb{T}_h^\star Q \in \mathcal{P}_{\widetilde{R}}$.*

Remark: Due to $Q_h \in [0, H-h+1]$, the radius R in fact depends on H, and thus we have $\widetilde{R} \asymp H$. Based on this assumption, we have the following regret bounds for two-layer ReLU neural networks, with the proof deferred to Appendix E.

Theorem 1. *Under Assumption 1, given $K \in [1, H]$, considering that the Q-function is learned by two-layer ReLU neural networks with width m and bounded ℓ_1 norm B in Eq.* (4)*, then under the ϵ-greedy exploration, for any $\delta \in (0,1)$, the total expected regret can be upper bounded with probability at least $1-\delta$*

$$\begin{aligned}\textit{Regret}(T) &\lesssim \left(\frac{\epsilon}{A}\right)^{-\frac{K}{2}} H^2 T^{\frac{3}{4}} \left[B(\log d)^{\frac{1}{4}} + \log^{\frac{1}{4}}\left(\frac{2}{\delta}\right)\right] + \left(\frac{\epsilon}{A}\right)^{-\frac{K}{2}} \frac{H^2 T}{\sqrt{m}} + H\sqrt{T} + \epsilon H T \\ &\lesssim \widetilde{\mathcal{O}}(H^{\frac{K+4}{K+2}} K^{\frac{K}{K+2}} T^{\frac{2K+3}{2K+4}}), \quad \textit{by taking } m = \Omega(\sqrt{T}) \textit{ and } \epsilon = \mathcal{O}\left((HK)^{\frac{2}{K+2}} T^{-\frac{1}{2(K+2)}}\right).\end{aligned}$$

Remark: The exponential dependence on H (in the worst case for any MDP) due to the employed ϵ-greedy can be avoided at an additional cost of worsening T dependence, i.e., from $\widetilde{O}(T^{\frac{3}{4}})$ to

$\widetilde{\mathcal{O}}(T^{\frac{2K+3}{2K+4}})$. Note that $\epsilon = \mathcal{O}\left((HK)^{\frac{2}{K+2}} T^{-\frac{1}{2(K+2)}}\right)$ is still satisfied with $\epsilon \in (0,1)$ for any $K \in [1, H]$ under a large episode T. Besides, our regret bound almost gets rid of the feature dimension d, which allows for DQN for high-dimensional image data in practice. We will discuss the achieved regret later with linear function approximation under the ϵ-greedy exploration, refer to Remark in Theorem 2.

4.2 Efficient value iteration under deep ReLU neural networks

In this setting, we consider $\widehat{Q}_h^t = \operatorname{argmin}_{f \in \mathcal{F}_{\texttt{DNN}}} \widehat{\mathcal{E}}_h^t(f)$ in Eq. (3), where $\mathcal{F}_{\texttt{DNN}}$ is the function space of deep ReLU neural networks defined in Eq. (5). Similar to Assumption 1, we have the following assumption on the Besov space $\mathcal{B}$, where the Bellman optimality operator maps any bounded value function to a bounded Besov space ball.

Assumption 2. *Let $\widetilde{R} \asymp H$ be a fixed constant. Define $\mathcal{B}_{\widetilde{R}} = \{f \in \mathcal{B}_{p,q}^{\alpha}(\mathcal{X}) : \|f\|_{\mathcal{B}} \leqslant \widetilde{R}\}$ in the Besov space and assume that for any $h \in [H]$ and $Q\colon \mathcal{S} \times \mathcal{A} \to [0, H]$, we have $\mathbb{T}_h^{\star} Q \in \mathcal{B}_{\widetilde{R}}$.*

Based on this assumption, we have the following theorem on regret bound in the Besov space for efficient deep Q-learning, with the proof deferred to Appendix F.1.

Theorem 2. *Under Assumptions 2 with the smoothness parameter $\alpha > d(1/p - 1/4)_+$ in the Besov space $\mathcal{B}_{p,q}^{\alpha}(\mathcal{X})$, given a constant $K \in [1, H]$, considering that the Q-function is learned by deep ReLU neural networks in Eq. (5) under the ϵ-greedy exploration, taking*

$$\text{the depth } L \asymp \tfrac{d}{2\alpha+d} \log T, \qquad \text{the width } m \asymp \tfrac{d}{2\alpha+d} T^{\frac{d}{2\alpha+d}} \log T\,,$$

then for any $\delta \in (0,1)$, the regret bound holds with probability at least $1-\delta$

$$\begin{aligned}\textit{Regret}(T) &\lesssim \left(\frac{\epsilon}{A}\right)^{-\frac{K}{2}} \left(H^{\frac{3}{2}} T^{\frac{\alpha+d}{2\alpha+d}} \log^3 T + H^2 \sqrt{T} \sqrt{\log\left(\frac{1}{\delta}\right)} \log T\right) + \epsilon H T \\ &\lesssim \widetilde{\mathcal{O}}(H^{\frac{H+4}{H+2}} K^{\frac{2}{K+2}} A^{\frac{K}{K+2}} T^{\frac{\alpha K + (\alpha+d)(K+2)}{(2\alpha+d)(K+2)}})\,, \quad \textit{taking } \epsilon = \mathcal{O}((HK)^{\frac{2}{K+2}} A^{\frac{K}{K+2}} T^{-\frac{2\alpha}{(2\alpha+d)(K+2)}})\,.\end{aligned}$$

Remark: We make the following remarks.

i) If the smoothness $\alpha \to \infty$ (i.e., the target Q function is sufficiently smooth), which holds for linear function approximation, we have

$$\text{Regret} \lesssim \widetilde{\mathcal{O}}(H^{\frac{H+4}{H+2}} K^{\frac{2}{K+2}} A^{\frac{K}{K+2}} T^{\frac{K+1}{K+2}})\,,$$

which recovers the regret bound $\widetilde{\mathcal{O}}(T^{\frac{K+1}{K+2}})$ in [46, Theorem 3]. In the best case, taking $K = 1$, our regret bound implies $\widetilde{\mathcal{O}}(H^{\frac{4}{3}} A^{\frac{1}{3}} T^{\frac{2}{3}})$ with $H \geqslant 4$, which matches the optimal regret bound for the contextual bandits problem in terms of dependence on T or A under the ϵ-greedy exploration [56]. In the worst case (for any MDP case), we can still obtain sublinear regret $\widetilde{\mathcal{O}}(T^{\frac{H+1}{H+2}})$.

ii) It can be found that, one hand, $\widetilde{O}(1)$ depth and $\widetilde{O}(T^{\frac{d}{2\alpha+d}})$ are enough for sublinear regret in deep Q-learning. On the other hand, the change of problem-dependent smoothness leads to less changes on the depth but more changes on the width. To be specific, if the Q-function becomes less smooth from α_1 to α_2 with $\alpha_1 > \alpha_2$, we have different depths $L_1,\ L_2$ and widths $m_1,\ m_2$ under these two settings, then we have $m_2 - m_1 \geqslant T^{\frac{d}{2\alpha_1+d}}(L_2 - L_1)$.

iii) According to the width-depth ratio $m/L = T^{\frac{d}{2\alpha+d}}$, our theory indicates that less problem-dependent smoothness of Q-function requires DNNs to be wider. In practice, if we work in the limited budget of parameters N in neural networks, e.g., $N \asymp m^2 L$, our theory implies that there is a tradeoff between the depth and width on smoothness, i.e., the depth $L := N^{1/3} T^{-\frac{2d}{3(2\alpha+d)}}$ increasing with α (or T) and the width $m = N^{1/3} T^{\frac{d}{3(2\alpha+d)}}$ decreasing with α (or T).

iv) Two-layer neural networks endowed by Barron spaces cannot be regarded as a special case of DNNs endowed by Besov spaces. This is because, our results in Barron spaces accommodate dense two-layer neural networks; while our result in Besov spaces here depends on the sparsity of deep ReLU neural networks [57] for statistical complexity estimation of $\mathcal{F}_{\texttt{DNN}}$.

$$\begin{cases} \text{Rademacher complexity on non-i.i.d data: Lem. 6} \Leftarrow \text{Lem. 5} \\ \text{two layers: Thm. 1} \Leftarrow \mathcal{O}(H^2 B\sqrt{\log d/n}) \Leftarrow \text{Lem. 10: Rademacher complexity of Barron spaces} \\ \text{DNNs:} \begin{cases} \text{Besov spaces: Thm. 2} \Leftarrow \mathcal{O}(n^{-\frac{2\alpha}{2\alpha+d}}) \Leftarrow \text{Prop. 3} \Leftarrow \text{Lem. 11 on LRC for Besov spaces} \\ \text{mixed smooth Besov spaces: Prop. 1} \Leftarrow \mathcal{O}(n^{-\frac{2\alpha}{2\alpha+1}}) \text{ (or faster)} \Leftarrow \text{Prop. 4} \end{cases} \end{cases}$$

Figure 1: Proof framework of temporal difference error under generalization on non-iid data. We denote LRC by local Rademacher complexity for short. The results omitted in the main text are deferred to the appendix.

It can be found that, though the sub-linear regret can be achieved by Theorem 2 by choosing $\widetilde{\mathcal{O}}(1)$ depth and $\widetilde{\mathcal{O}}(T^{\frac{d}{2\alpha+d}})$ width, the regret bound heavily depends on the feature dimension d, failing in the *curse of dimensionality*, which appears ineffective on high dimensional data in deep RL. To tackle this issue, we consider *mixed smooth* Besov space $\widetilde{\mathcal{B}}$ to alleviate the curse of dimensionality.

Assumption 3. *Let $\widetilde{R} \asymp H$ be a fixed constant. Define $\widetilde{\mathcal{B}}_{\widetilde{R}} = \{f \in \widetilde{\mathcal{B}}^{\alpha}_{p,q} : \|f\|_{\widetilde{\mathcal{B}}} \leqslant \widetilde{R}\}$ in the mixed smooth Besov space and assume that $\forall h \in [H]$ and $Q\colon \mathcal{S} \times \mathcal{A} \to [0, H]$, then we have $\mathbb{T}^{\star}_h Q \in \widetilde{\mathcal{B}}_{\widetilde{R}}$.*

Based on this assumption, we have the following proposition on regret bound in the mixed-smooth Besov space, with the proof deferred to Appendix F.2.

Proposition 1. *Under Assumptions 3 with the smoothness parameter $\alpha > 0$ and $u := (1 - 1/q)_+$ defined in the mixed smooth Besov space $\widetilde{\mathcal{B}}^{\alpha}_{p,q}$ with $p = 4$, given a constant $K \in [1, H]$, considering that the Q-function is learned by deep ReLU neural networks in Eq.* (5) *under the ϵ-greedy exploration, taking*

$$\text{the depth } L \asymp \widetilde{\mathcal{O}}(1), \qquad \text{the width } m \asymp \widetilde{\mathcal{O}}(T^{\frac{1}{2\alpha+1}}) \ ,$$

then for any $\delta \in (0, 1)$ with probability at least $1 - \delta$, we have

$$\textit{Regret}(T) \lesssim \begin{cases} \widetilde{\mathcal{O}}\left(H^{\frac{H+4}{H+2}} K^{\frac{2}{K+2}} T^{\frac{(\alpha+1)(K+1)+1}{(2\alpha+1)(K+2)}}\right), \text{ if } \alpha \leqslant u \log_2 e \text{ and taking } \epsilon = \mathcal{O}((HK)^{\frac{2}{K+2}} T^{-\frac{2\alpha}{(2\alpha+1)(K+2)}}) \\ \widetilde{\mathcal{O}}\left(H^{\frac{H+4}{H+2}} K^{\frac{2}{K+2}} T^{\frac{\tilde{c}(K+1)+1}{\tilde{c}(K+2)}}\right), \text{ if } \alpha > u \log_2 e \text{ and taking } \epsilon = \mathcal{O}\left((HK)^{\frac{2}{K+2}} T^{\frac{2(\log_2 e - \alpha)}{\tilde{c}(K+2)}}\right). \end{cases}$$

where $\tilde{c} := 1 + 2\alpha + (1 - 2u) \log_2 e$.

Remark: Our regret bound gets rid of the feature dimension d and is close to the $\widetilde{\mathcal{O}}(T^{\frac{K+1}{K+2}})$ bound if the Q-function is sufficiently smooth, which holds for DQN on high dimensional image data in practice.

5 Proof outline

In this section, we outline the proof of value iteration via DNNs with sublinear regret bounds presented in Section 4. Our proof framework includes three parts: regret decomposition, transformation between TD error and generalization, and generalization bounds on non-iid data for TD error analysis. The last two parts are the main technical contribution in this work and provide width and depth guideline for efficient deep RL. The complete proof is reported in the appendix.

Regret decomposition: This part is standard and commonly studied in RL theory, e.g., [58, 7, 13]. We briefly include here for self-completeness. Define the temporal-difference (TD) error as

$$\Gamma^t_h(s, a) = r_h(s, a) + (\mathbb{P}_h V^t_{h+1})(s, a) - Q^t_h(s, a) = (\mathbb{T}_h V^t_{h+1})(s, a) - Q^t_h(s, a), \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}, \tag{7}$$

where Γ^t_h is a function on $\mathcal{S} \times \mathcal{A}$ for all $h \in [H]$ and $t \in [T]$. Accordingly, the regret can be decomposed into two main terms related to the TD error and ϵHT due to the ϵ-greedy exploration, respectively (*c.f.* Lemma 1).

Transforming TD error to generalization: To bound the TD error, previous theory work focuses on designing bonus function via confidence based algorithms to ensure TD error negative, which mismatches with practical DQN that works with ϵ-greedy exploration. Accordingly, we attempt to

built the connection between the regret error and $\|\Gamma_h^t\|_{L^2(\mathrm{d}\bar{\mu}_h^t)}$, see Lemma 3; and then transform estimation on $\|\Gamma_h^t\|_{L^2(\mathrm{d}\bar{\mu}_h^t)}$ to the following two terms: generalization error and approximation error, respectively (*c.f.* Lemma 4)

$$\|\Gamma_h^t\|_{L^2(\mathrm{d}\bar{\mu}_h^t)}^2 \leqslant \left[\mathcal{E}_h(\widehat{Q}_h^t) - \min_{f\in\mathcal{F}} \mathcal{E}_h(f)\right] + \inf_{f\in\mathcal{F}} \|f - \mathbb{T}_h^\star Q_{h+1}^t\|_{L^2(\mathrm{d}\bar{\mu}_h^t)}^2 .$$

The convergence rate of approximation error can be directly obtained by previous work, e.g., $\mathcal{O}(1/m)$ for two-layer ReLU networks in Barron spaces [34] and $\mathcal{O}(n^{-2\alpha/(2\alpha+d)})$ in [37] for deep ReLU networks; and in the next we elucidate the generalization error bounds.

Analysis of TD error based on generalization bounds for non-iid data: The key part left is to bound the generalization error under our non-i.i.d data for TD error, see the proof framework in Figure 1. Following a covering argument for the function approximation class used for value functions [7], we introduce two additional terms to handle conditional independence and biased estimation, which can be upper bounded by ε-covering of $\mathcal{F}$. By doing so, we firstly verify that the maximum error in estimating the mean of any function $f \in \mathcal{F}$ can be still bounded by the Rademacher complexity of $\mathcal{F}$ in Lemma 5, and then generalization bounds by Rademacher complexity still holds by Lemma 6 via the averaged measure, which only requires the data to be independent.

To be specific, for two-layer neural networks, by computing the Rademacher complexity of $\mathcal{F}_{\texttt{SNN}}$ in Lemma 10, we obtain the generalization error at a certain $\mathcal{O}(H^2 B\sqrt{\log d/n})$ convergence rate; combining the result of approximation error in Barron spaces, we conclude the proof of Theorem 1 with a sublinear regret for efficient deep RL.

For deep neural networks, by computing the local Rademacher complexity of $\mathcal{F}_{\texttt{DNN}}$ in Lemma 11, by choosing proper parameters in Eq. (5), we present the generalization bounds in Besov spaces (*c.f.* Proposition 3) with $\mathcal{O}(n^{-\frac{2\alpha}{2\alpha+d}})$ and mixed smooth Besov spaces (*c.f.* Proposition 4) with $\mathcal{O}(n^{-\frac{2\alpha}{2\alpha+1}})$ (or faster) respectively. Finally, combining the result of approximation error in these two spaces, the sublinear regret can be achieved in Theorem 2 that depends on d and Proposition 1 that gets rid of d, respectively. The above results demonstrate that, $\widetilde{\mathcal{O}}(1)$ depth and the width in sublinear T are suffice to efficient deep value iteration.

6 Conclusion and discussion

This paper provides an in-depth understanding on neural network function approximation with the ϵ-greedy exploration under the online setting beyond linear/kernel approximation, and eluder dimension. Our result sheds light on the theoretical guarantees of sub-linear regret bounds, with width and depth guidelines in deep RL, i.e., 1) the width in sublinear of T and depth in a constant order are sufficient to sublinear regret; 2) problem-dependent smoothness depending on the width/depth ratio: for realistic low values of α, we observe that the depth L should be chosen approximately as $\log m$. Interestingly, we notice that this result is closely matching practical implementation of DQN. For example, the choices of [4] $m = 512$ and $L = 5$ can be explained with our theory, indeed $\log(512) \approx 6$.

The analysis of this work is built on the ϵ greedy for exploration, which are satisfied in practical cases when employing DQN. Nevertheless, designing a provably efficient exploration mechanism for deep RL could be an interesting future direction in both practice and theory.

Furthermore, our work is built on value iteration based algorithm, which is different from practical DQN that adapts Q-learning via one-step gradient descent. We remark that, this work focuses on neural network function approximation beyond "linear" regime, which shares the same spirit and main intention of DQN but has a large step towards theoretically understanding DQN in a comprehensive view. For example, the experience replay and target network in DQN are not studied in this work, as a significant role on its success in practice, and would be an interesting direction for better understanding DQN, e.g., [45].

References

[1] Csaba Szepesvári. Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1–103, 2010. 1

[2] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 1

[3] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018. 1

[4] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015. 1, 2, 6, 9

[5] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016. 1

[6] LeCun Yann, Bengio Yoshua, and Hinton Geoffrey. Deep learning. *Nature*, 521(7553):436–444, 2015. 1

[7] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020. 1, 2, 3, 6, 8, 9, 26, 28, 30, 32, 33, 34

[8] Yining Wang, Ruosong Wang, Simon Shaolei Du, and Akshay Krishnamurthy. Optimism in reinforcement learning with generalized linear function approximation. In *International Conference on Learning Representations*, 2020. 1

[9] Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pages 463–474. PMLR, 2020. 1, 3

[10] Dongruo Zhou, Quanquan Gu, and Csaba Szepesvari. Nearly minimax optimal reinforcement learning for linear mixture markov decision processes. In *Conference on Learning Theory*, pages 4532–4576. PMLR, 2021. 1, 3

[11] Ian Osband, Benjamin Van Roy, Daniel J Russo, Zheng Wen, et al. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019. 1

[12] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, pages 8571–8580, 2018. 1

[13] Zhuoran Yang, Chi Jin, Zhaoran Wang, Mengdi Wang, and Michael I Jordan. On function approximation in reinforcement learning: Optimism in the face of large state spaces. In *Advances in Neural Information Processing Systems*, 2020. 1, 2, 3, 6, 8, 17

[14] Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017. 1, 3

[15] Kefan Dong, Jian Peng, Yining Wang, and Yuan Zhou. Root-n-regret for learning in markov decision processes with function approximation and low bellman rank. In *Conference on Learning Theory*, pages 1554–1557. PMLR, 2020. 1, 3

[16] Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. *Advances in Neural Information Processing Systems*, 26, 2013. 1, 3

[17] Ruosong Wang, Russ R Salakhutdinov, and Lin Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. In *Advances in Neural Information Processing Systems*, volume 33, pages 6123–6135, 2020. 1, 3

[18] Haque Ishfaq, Qiwen Cui, Viet Nguyen, Alex Ayoub, Zhuoran Yang, Zhaoran Wang, Doina Precup, and Lin Yang. Randomized exploration in reinforcement learning with general value function approximation. In *International Conference on Machine Learning*, pages 4607–4616. PMLR, 2021. 1, 3

[19] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, pages 2933–2943, 2019. 2

[20] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*, pages 8570–8581, 2019. 2

[21] Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pages 3635–3673. PMLR, 2020. 2

[22] Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11):113301, 2020. 2

[23] Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(1):629–681, 2017. 2, 4, 15

[24] Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding neural networks. In *Advances in Neural Information Processing Systems*, pages 6594–6604, 2019. 2

[25] Michael Celentano, Theodor Misiakiewicz, and Andrea Montanari. Minimum complexity interpolation in random features models. *arXiv preprint arXiv:2103.15996*, 2021. 2

[26] Gene Li, Pritish Kamath, Dylan J Foster, and Nathan Srebro. Eluder dimension and generalized rank. *arXiv preprint arXiv:2104.06970*, 2021. 2

[27] Kefan Dong, Jiaqi Yang, and Tengyu Ma. Provable model-based nonlinear bandit and reinforcement learning: Shelve optimism, embrace virtual curvature. In *Advances in Neural Information Processing Systems*, volume 34, 2021. 2

[28] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2

[29] Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. In *International Conference on Learning Representations*, 2018. 2

[30] Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pages 10746–10756. PMLR, 2020. 2, 3

[31] Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018. 2, 3

[32] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 1–39. JMLR Workshop and Conference Proceedings, 2012. 2, 3

[33] Andrea Zanette, David Brandfonbrener, Emma Brunskill, Matteo Pirotta, and Alessandro Lazaric. Frequentist regret bounds for randomized least-squares value iteration. In *International Conference on Artificial Intelligence and Statistics*, pages 1954–1964. PMLR, 2020. 2, 3

[34] Weinan E, Chao Ma, and Lei Wu. The barron space and the flow-induced function spaces for neural network models. *Constructive Approximation*, pages 1–38, 2021. 2, 4, 9, 15, 25, 26

[35] Weinan E and Stephan Wojtowytsch. Representation formulas and pointwise properties for barron functions. *arXiv preprint arXiv:2006.05982*, 2020. 2, 4, 15

[36] Yoshihiro Sawano. *Theory of Besov spaces*, volume 56. Springer, 2018. 2, 5, 16

[37] Taiji Suzuki. Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality. In *International Conference on Learning Representations*, 2019. 2, 5, 9, 15, 16, 17, 27, 28, 32

[38] Jianqing Fan, Zhaoran Wang, Yuchen Xie, and Zhuoran Yang. A theoretical analysis of deep q-learning. In *Learning for Dynamics and Control*, pages 486–489, 2020. 2, 3, 5, 6

[39] Baihe Huang, Kaixuan Huang, Sham Kakade, Jason D Lee, Qi Lei, Runzhe Wang, and Jiaqi Yang. Going beyond linear rl: Sample efficient neural function approximation. In *Advances in Neural Information Processing Systems*, 2021. 3

[40] Jihao Long, Jiequn Han, et al. An l^2 analysis of reinforcement learning in high dimensions with kernel and neural network approximation. *arXiv preprint arXiv:2104.07794*, 2021. 3

[41] Thanh Nguyen-Tang, Sunil Gupta, Hung Tran-The, and Svetha Venkatesh. Sample complexity of offline reinforcement learning with deep relu networks. *arXiv preprint arXiv:2103.06671*, 2021. 3

[42] Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? In *Advances in neural information processing systems*, 2018. 3

[43] Martin Riedmiller. Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*, pages 317–328. Springer, 2005. 3

[44] Pan Xu and Quanquan Gu. A finite-time analysis of q-learning with neural network function approximation. In *International Conference on Machine Learning*, pages 10555–10565. PMLR, 2020. 3

[45] Andrea Zanette and Martin J Wainwright. Stabilizing Q-learning with linear architectures for provably efficient learning. *arXiv preprint arXiv:2206.00796*, 2022. 3, 9

[46] Chris Dann, Yishay Mansour, Mehryar Mohri, Ayush Sekhari, and Karthik Sridharan. Guarantees for epsilon-greedy reinforcement learning with function approximation. In *International Conference on Machine Learning*, pages 4666–4689, 2022. 3, 7

[47] Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[48] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014. 4

[49] Yoshua Bengio, Nicolas Le Roux, Pascal Vincent, Olivier Delalleau, and Patrice Marcotte. Convex neural networks. In *Advances in Neural Information Processing Systems*, pages 123–130, 2005. 4, 15

[50] Minshuo Chen, Haoming Jiang, Wenjing Liao, and Tuo Zhao. Efficient approximation of deep ReLU networks for functions on low dimensional manifolds. *Advances in neural information processing systems*, 32:8174–8184, 2019. 5

[51] Ahmed Abdeljawad and Philipp Grohs. Approximations with deep neural networks in sobolev time-space. *arXiv preprint arXiv:2101.06115*, 2020. 5

[52] Gérard Kerkyacharian and Dominique Picard. Density estimation in besov spaces. *Statistics & probability letters*, 13(1):15–24, 1992. 5, 16

[53] Winfried Sickel and Tino Ullrich. Tensor products of sobolev–besov spaces and applications to approximation from the hyperbolic cross. *Journal of Approximation Theory*, 161(2):748–786, 2009. 5, 16, 17

[54] Steven J Bradtke and Andrew G Barto. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1):33–57, 1996. 6

[55] Seungchan Kim, Kavosh Asadi, Michael Littman, and George Konidaris. Deepmellow: removing the need for a target network in deep q-learning. In *International Joint Conference on Artificial Intelligence*, 2019. 6

[56] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020. 7

[57] Boris Hanin and David Rolnick. Deep relu networks have surprisingly few activation patterns. *Advances in Neural Information Processing Systems*, 32, 2019. 7

[58] Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pages 1283–1294, 2020. 8, 17

[59] Weinan E, Chao Ma, and Lei Wu. A priori estimates of the population risk for two-layer neural networks. *Communications in Mathematical Sciences*, 17(5):1407–1425, 2019. 15

[60] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401. PMLR, 2015. 15

[61] Evarist Giné and Richard Nickl. *Mathematical foundations of infinite-dimensional statistical models*. Cambridge University Press, 2021. 16

[62] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018. 21, 22

[63] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019. 22, 25

[64] Shahar Mendelson. Improving the sample complexity using global data. *IEEE transactions on Information Theory*, 48(7):1977–1991, 2002. 22, 24

[65] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014. 22, 23, 24

[66] Peter L Bartlett, Olivier Bousquet, and Shahar Mendelson. Local rademacher complexities. *Annals of Statistics*, 33(4):1497–1537, 2005. 24

[67] Yunwen Lei, Lixin Ding, and Yingzhou Bi. Local rademacher complexity bounds based on covering numbers. *Neurocomputing*, 218:320–330, 2016. 24, 30

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We clearly discuss the limitation of this work in the Conclusion section.
   (c) Did you discuss any potential negative societal impacts of your work? [No] Our work is theoretical and generally will have no negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] The assumptions are clearly stated and well discussed.
   (b) Did you include complete proofs of all theoretical results? [Yes] All of the proofs can be found in the Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]