Journal of Machine Learning Research 25 (2024) 1-67 Submitted 1/24; Revised 7/24; Published 8/24

Sample Complexity of Neural Policy Mirror Descent for Policy Optimization on Low-Dimensional Manifolds

Zhenghao Xu ZHENGHAOXU@GATECH.EDU
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

Xiang Ji XIANGJ@PRINCETON.EDU
Department of Electrical and Computer Engineering
Princeton University
Princeton, NJ 08544, USA

Minshuo Chen MINSHUOCHEN@PRINCETON.EDU
Department of Electrical and Computer Engineering
Princeton University
Princeton, NJ 08544, USA

Mengdi Wang MENGDIW@PRINCETON.EDU
Department of Electrical and Computer Engineering
Princeton University
Princeton, NJ 08544, USA

Tuo Zhao TOURZHAO@GATECH.EDU
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

Editor: Qiang Liu

Abstract

Policy gradient methods equipped with deep neural networks have achieved great success in solving high-dimensional reinforcement learning (RL) problems. However, current analyses cannot explain why they are resistant to the curse of dimensionality. In this work, we study the sample complexity of the neural policy mirror descent (NPMD) algorithm with deep convolutional neural networks (CNN). Motivated by the empirical observation that many high-dimensional environments have state spaces possessing low-dimensional structures, such as those taking images as states, we consider the state space to be a d-dimensional manifold embedded in the D-dimensional Euclidean space with intrinsic dimension $d \ll D$. We show that in each iteration of NPMD, both the value function and the policy can be well approximated by CNNs. The approximation errors are controlled by the size of the networks, and the smoothness of the previous networks can be inherited. As a result, by properly choosing the network size and hyperparameters, NPMD can find an ϵ-optimal policy with $\widetilde{O}(\epsilon^{-\frac{d}{\alpha}-2})$ samples in expectation, where $\alpha \in (0, 1]$ indicates the smoothness of environment. Compared to previous work, our result exhibits that NPMD can leverage the low-dimensional structure of state space to escape from the curse of dimensionality, explaining the efficacy of deep policy gradient algorithms.

Keywords: deep reinforcement learning, policy optimization, function approximation, convolutional neural network, Riemannian manifold

1. Introduction

Deep Reinforcement Learning (DRL) is a popular approach for solving complex decision-making problems in various domains. DRL methods, especially policy-based ones including DDPG (Lillicrap et al., 2015), TRPO (Schulman et al., 2015), and PPO (Schulman et al., 2017), are able to handle high-dimensional state space efficiently by leveraging function approximation with neural networks. For instance, in Atari games (Brockman et al., 2016), the states are images of size 210×160 with RGB color channels, resulting in a continuous state space of dimension $100,800$, to which tabular algorithms such as policy iteration (Puterman, 1994) and policy mirror descent (PMD, Lan 2023) are not applicable. Surprisingly, such a high dimension does not seem to significantly impact the efficacy of the aforementioned DRL algorithms.

Despite the empirical success of these DRL methods in high dimensions, there currently exist no satisfactory results in theory that can explain the reason behind them. Most of the existing works about function approximation in RL focus on linear function class (Agarwal et al., 2021; Alfano and Rebeschini, 2022; Yuan et al., 2023). They assume the value function and the policy can be well approximated by linear functions of features representing states and actions (Jin et al., 2020), which is restrictive and requires feature engineering. One way to relax such a linearity assumption is to consider the reproducing kernel Hilbert space (RKHS) which allows nonlinear function approximation (Agarwal et al., 2020; Yang et al., 2020) through random features (Rahimi and Recht, 2007). However, the commonly used reproducing kernels such as the Gaussian radial basis function and randomized ReLU kernel suffer from the curse of dimensionality in sample complexity without additional smoothness assumptions (Bach, 2017; Yehudai and Shamir, 2019; Hsu et al., 2021).

Moreover, some researchers study neural network approximation in the neural tangent kernel (NTK) regime, which is equivalent to an RKHS (Jacot et al., 2018; Liu et al., 2019; Wang et al., 2019; Cayci et al., 2022; Alfano et al., 2023). Consequently, these results inherit the curse of dimensionality from general RKHS. Some other works investigate neural network approximation from a non-parametric perspective (Fan et al., 2020; Nguyen-Tang et al., 2022), but they consider value-based methods where only value function is approximated and also suffer from the curse of dimensionality. There are alternative lines of work on policy optimization with general function approximation, but they either assume the functions can be well approximated without any further verification (Lan, 2022; Mondal and Aggarwal, 2023), or require some strong assumptions such as third-time differentiability (Yin et al., 2022).

One possible explanation for the empirical effectiveness of DRL algorithms is the adaptivity of neural networks to the intrinsic low-dimensional structure of the state space. In the Atari games example, the images share common textures and are rendered using just a small number of internal states, such as the type, position, and angle of each object, thus the intrinsic dimension of the state space is in fact very small compared to the data dimension of 100,800. However, the extent of this adaptivity is yet to be explored in DRL literature.

Algorithm	Approximation	Regularity	Complexity	Remark
NPG (P) (Yuan et al., 2023)	Linear	Linear (D)	$\widetilde{O}(\epsilon^{-2})$	strong realizability
NPPO (P) (Liu et al., 2019)	NTK (2-layer ReLU)	RKHS (D)	$O(\epsilon^{-14})$	realizability
NNAC (P) (Cayci et al., 2022)	NTK (2-layer ReLU)	RKHS (D)	$\widetilde{O}(\epsilon^{-6})$	realizability
AMPO (P) (Alfano et al., 2023)	NTK (2-layer ReLU)	RKHS (D)	$\widetilde{O}(\epsilon^{-4})$ [1]	realizability
FQI (V) (Fan et al., 2020)	FNN (Deep ReLU)	Hölder (I)	$\widetilde{O}(\epsilon^{-\frac{D}{\alpha}-2})$	curse of dimension
FQI (V) (Nguyen-Tang et al., 2022)	FNN (Deep ReLU)	Besov (I)	$\widetilde{O}(\epsilon^{-\frac{2D}{\alpha}-2})$	curse of dimension
NPMD (P) (This paper)	CNN (Deep ReLU)	Lipschitz (I) [2]	$\widetilde{O}(\epsilon^{-\frac{d}{\alpha}-2})$	$d \ll D$

Table 1: Comparison with existing results. The algorithms are classified as value-based (V) or policy-based (P, including actor-critic methods involving policy gradient). The regularity assumptions are algorithm-dependent (D) or algorithm-independent (I). We consider sample complexity for arbitrary $\epsilon > 0$, for which some previous works require additional realizability assumptions to eliminate the error floors.

To bridge this gap between theory and practice, we propose to investigate neural policy optimization within environments possessing low-dimensional state space structures. Specifically, we consider the infinite-horizon discounted Markov decision process (MDP) with continuous state space $\mathcal{S}$, finite action space $\mathcal{A}$, and discount factor γ. We focus on the sample complexity of the neural policy mirror descent (NPMD) method. NPMD is based on the actor-critic framework (Konda and Tsitsiklis, 1999) where both the policy (actor) and value function (critic) are approximated by neural networks. It is an implementation of the general PMD scheme (Lan, 2022) with neural network approximation. The NPMD-type methods including TRPO (Schulman et al., 2015)) and PPO (Schulman et al., 2017) are widely used in applications like inventory systems (Guo et al., 2024), game AI (Berner et al., 2019) and fine-tuning large language models (Ziegler et al., 2019; Ouyang et al., 2022). Moreover, instead of working on general Euclidean space, we assume the state space to be a d-dimensional manifold embedded in the D-dimensional Euclidean space where $d \ll D$.

We summarize our main contributions as follows:

(1) We first investigate the universal approximation of the convolutional neural network (CNN), a popular architecture for image data, on the d-dimensional manifold. We

1. The results in Liu et al. (2019); Cayci et al. (2022); Alfano et al. (2023) implicitly suffer from the curse of dimensionality due to hidden constants related to NTK, including the width of the network and the RKHS norm. These constants can have exponential dependence on D for realizability (Yehudai and Shamir, 2019).
2. We define Lipschitz continuity for all $\alpha \in (0, 1]$, which reduces to the usually defined Lipschitz condition when $\alpha = 1$ and reduces to the usually defined Hölder continuity when $\alpha \in (0, 1)$.

show that under the Lipschitz MDP condition (Assumption 4), CNN with sufficient parameters can well approximate both the value function and the policy (Theorem 16, Theorem 21). Compared to previous work on policy-based methods, our analysis decouples the regularity conditions from algorithmic specifications. For example, in Liu et al. (2019), the value functions are assumed to be in a network width-dependent set that approximates the NTK-induced RKHS, while our regularity assumptions are not based on the network architecture in advance. In Yuan et al. (2023), the approximation error highly depends on the design of the feature map.

(2) Based on CNN function approximation, we then derive $\widetilde{O}\big(|\mathcal{A}|^{\frac{d}{2\alpha}+2}(1-\gamma)^{-\frac{2d}{\alpha}-5}\epsilon^{-\frac{d}{\alpha}-2}\big)$ sample complexity bound for NPMD with CNN approximation to find a policy whose value function is at most ϵ to the global optimal in expectation (Theorem 24). Here, $\alpha \in (0, 1]$ is the exponent of the Lipschitz condition and $\widetilde{O}(\cdot)$ hides the logarithmic terms and some coefficients related to distribution mismatch and concentrability (see Assumptions 2 and 3). Compared to the results in Fan et al. (2020) and Nguyen-Tang et al. (2022), the curse of dimensionality (exponential dependence on D) is avoided by exploiting the intrinsic d-dimensional structure. To the best of our knowledge, this is the first sample complexity result for policy gradient methods with deep neural network function approximation.

Some preliminary results for this work have been first presented in our conference paper Ji et al. (2022), which focuses on policy evaluation only. We extend the scope to policy optimization with a full characterization of both iteration complexity and sample complexity.

1.1 Related Work

Our work is based on previous studies on policy gradient methods with function approximation as well as deep supervised learning on manifolds.

Policy gradient methods. The policy gradient method (Williams, 1992; Sutton et al., 1999) is first developed under the compatible function approximation framework. The natural policy gradient (NPG) method (Kakade, 2001) extends the policy gradient method by incorporating the geometry of the parameter space to improve convergence properties. Trust region policy optimization (TRPO, Schulman et al. 2015) and proximal policy optimization (PPO, Schulman et al. 2017) are modern variants of policy gradient methods with neural network function approximation that use constraints or penalties to prevent aggressive updates, resulting in more stable and efficient learning. These modern methods are often used to handle high-dimensional state spaces and have been shown to achieve state-of-the-art results in a variety of RL domains. For example, the PPO algorithm and its variants are used in training some of the most advanced artificial intelligence, such as OpenAI Five (Berner et al., 2019) and GPT-4 (OpenAI, 2023). From a theoretical perspective, policy gradient methods such as NPG and PPO can be unified under the PMD framework (Geist et al., 2019; Shani et al., 2020; Lan, 2023), whose fast linear rate of convergence has been established for the tabular case (Cen et al., 2022; Xiao, 2022; Zhan et al., 2023).

Linear function approximation. The majority of existing research on function approximation considers the linear function class (Agarwal et al., 2021; Alfano and Rebeschini, 2022; Yuan et al., 2023), which is the only known option for the compatible function approx-

imation framework by far (Sutton et al., 1999). However, these linear function approximation methods are restrictive. Only in simple environments, such as linear MDP (Jin et al., 2020), can high approximation quality be guaranteed, which necessitates carefully designed features. Regrettably, the task of crafting such features is either infeasible or demands substantial effort from domain experts, and any misspecification of features could lead to an exponential gap (Du et al., 2020).

Reproducing kernel approach. The reproducing kernel Hilbert space (RKHS) has been adopted to relax the limitation of the linear function class and to enable more expressive nonlinear function approximation (Agarwal et al., 2020; Yang et al., 2020). To achieve efficient computation, random features are employed (Rahimi and Recht, 2007). Nevertheless, the RKHS suffers from the curse of dimensionality, which hinders its performance on high-dimensional problems.

Neural tangent kernel. One approach to investigating the function approximation capabilities of neural networks is through the use of the neural tangent kernel (NTK, Jacot et al. 2018; Liu et al. 2019; Wang et al. 2019; Cayci et al. 2022; Alfano et al. 2023). The NTK approach can be viewed as training a neural network with gradient descent under a specific regime, and as the width of the neural network approaches infinity, it converges to an RKHS. As a consequence, like other RKHS approaches, the NTK approach suffers from the curse of dimensionality, limiting its performance on high-dimensional problems. Additionally, some literature has pointed out that the NTK is susceptible to the kernel degeneracy problem (Chen and Xu, 2020; Huang et al., 2020), which may impact its overall learnability.

Non-parametric neural network approximation. The non-parametric approach has been adopted to study the sample complexity of neural function approximation in RL under mild smoothness assumptions, such as Fan et al. (2020) and Nguyen-Tang et al. (2022). These analyses are mainly focused on value-based methods and do not apply to policy gradient methods due to the lack of smoothness in neural policies.

Deep supervised learning on manifolds. Parallel to DRL, existing work on deep supervised learning extensively studies the adaptivity of neural networks to the intrinsic low-dimensional data manifold embedded in high-dimensional ambient space, and how this adaptivity helps neural networks escape from the curse of dimensionality. In deep supervised learning, it has been shown that the sample complexity's exponential dependence on the ambient dimension D can be replaced by the dependence on the manifold dimension d (Chen et al., 2019; Schmidt-Hieber, 2019; Liu et al., 2021). These analyses focus on fitting a single target function whose smoothness is predetermined by the nature of the learning task, while in our setting, the target functions include policies whose smoothness can get worse in each iteration.

1.2 Notation

For $n \in \mathbb{N}$, $[n] \coloneqq \{i \mid 1 \leq i \leq n\}$. For $a \in \mathbb{R}$, $\lceil a \rceil$ denotes the smallest integer no less than a. For $a, b \in \mathbb{R}$, $a \vee b \coloneqq \max(a, b)$ and $a \wedge b \coloneqq \min(a, b)$. For a vector, $\|\cdot\|_p$ denotes the p-norm for $1 \leq p \leq +\infty$. For a matrix, $\|\cdot\|_\infty$ denotes the maximum magnitude of entries. For a finite set, $|\cdot|$ denotes its cardinality. For a function $f \colon \mathcal{X} \to \mathbb{R}$, $\|f\|_\infty$ denotes the maximal value of $|f|$ over $\mathcal{X}$. Given distribution ρ on $\mathcal{X}$, we use $f(\rho) \coloneqq \mathbb{E}_{x \sim \rho}\big[f(x)\big]$ to denote the expected value of $f(x)$ where $x \sim \rho$. Given distributions μ and ν on $\mathcal{X}$, the

total variation distance is defined as $d_{\mathrm{TV}}(\mu,\nu) \coloneqq \sup_{A\in\Sigma} |\mu(A)-\nu(A)|$, where Σ contains all measurable sets on $\mathcal{X}$. When μ is absolutely continuous with respect to ν, denoted as $\mu \ll \nu$, the Pearson χ^2-divergence is defined as $\chi^2(\mu,\nu) \coloneqq \mathbb{E}_\nu[(\frac{\mathrm{d}\mu}{\mathrm{d}\nu}-1)^2]$, where $\frac{\mathrm{d}\mu}{\mathrm{d}\nu}$ denotes the Radon–Nikodym derivative.

Let $\mathcal{A}$ be a finite set, we denote $P^{|\mathcal{A}|} \coloneqq \{(p_a)_{a\in\mathcal{A}} \mid p_a \in P\}$ as the Cartesian product of P's indexed by $\mathcal{A}$, $\mathbf{1} \coloneqq (1)_{a\in\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}$ as the vector with all entries being 1, $\Delta_{\mathcal{A}} \coloneqq \left\{p \in \mathbb{R}^{|\mathcal{A}|} \mid \sum_{a\in\mathcal{A}} p_a = 1, p_a \geq 0\right\}$ as the probability simplex over $\mathcal{A}$, and define the inner product $\langle\cdot,\cdot\rangle : \mathbb{R}^{|\mathcal{A}|} \times \mathbb{R}^{|\mathcal{A}|} \to \mathbb{R}$ as $\langle p,q\rangle \coloneqq \sum_{a\in\mathcal{A}} p_a q_a$. Let $\pi\colon \mathcal{S} \to \Delta_{\mathcal{A}}$ be a map, we use $h^\pi(s) \coloneqq \langle \log\pi(s), \pi(s)\rangle$ to denote the negative entropy of π at $s \in \mathcal{S}$ where $\log(\cdot)$ is performed entrywise, and denote the Kullback-Leibler (KL) divergence between two distributions $\pi'(s)$ and $\pi(s)$ by $D_\pi^{\pi'}(s) \coloneqq \langle \log\pi'(s) - \log\pi(s), \pi'(s)\rangle \geq 0$.

1.3 Roadmap

The rest of this paper is organized as follows: Section 2 briefly introduces some preliminaries; Section 3 presents the neural policy mirror descent algorithm; Section 4 presents the theoretical analysis; Section 5 presents the experimental results to back up our theory; Section 6 discusses our results with the related work and draws a brief conclusion.

2. Background

We introduce the problem setting and briefly review the Markov decision process, Riemannian manifold, and convolutional neural networks.

2.1 Markov Decision Process

We consider an infinite-horizon discounted Markov decision process (MDP) denoted as $\mathcal{M} = (\mathcal{S},\mathcal{A},\mathcal{P},c,\gamma)$, where $\mathcal{S} \subseteq \mathbb{R}^D$ is a continuous state space in $\mathbb{R}^D$, $\mathcal{A}$ is a finite action space, $\mathcal{P}$ is the transition kernel that describes the next state distribution $s' \sim \mathcal{P}(\cdot|s,a)$ at state $s \in \mathcal{S}$ when action $a \in \mathcal{A}$ is taken, $c\colon \mathcal{S}\times\mathcal{A} \to [0,C]$ is a cost function bounded by some constant $C > 0$, and $\gamma \in (0,1)$ is a discount factor.

A *stochastic policy* $\pi\colon \mathcal{S} \to \Delta_{\mathcal{A}}$ describes the behavior of an agent. For any state $s \in \mathcal{S}$, $\pi(\cdot|s) \in \Delta_{\mathcal{A}}$ gives a conditional probability distribution over the action space $\mathcal{A}$, where $\pi(a|s)$ is the probability of taking action a at state s.

Given a policy π, the expected cost starting from state s is given by the *state value function*

$$V^\pi(s) = \mathbb{E}_{\substack{a_t\sim\pi(\cdot|s_t),\\ s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t c(s_t,a_t) \;\middle|\; s_0 = s\right].$$

The goal of policy optimization is to learn an optimal policy $\pi^\star$ by solving a stochastic optimization problem, where the objective function is the expected value function for a given initial state distribution ρ:[3]

$$V^\star(\rho) \coloneqq V^{\pi^\star}(\rho) = \operatorname*{minimize}_\pi \; \mathbb{E}_{s\sim\rho}\big[V^\pi(s)\big]. \tag{1}$$

3. The optimal policy $\pi^\star$ does not depend on the choice of ρ.

A policy π is called ϵ-*optimal*, if

$$V^{\pi}(\rho) - V^{\star}(\rho) \leq \epsilon.$$

In the reinforcement learning setting, the algorithm cannot directly access the transition kernel $\mathcal{P}$ and the cost function c. Instead, the algorithm can only start from an initial state from ρ and interact with the environment for the immediate cost $c_{s,a} = c(s,a)$ and the next state $s' \sim \mathcal{P}(\cdot|s,a)$. Each interaction is through a sample oracle. The (expected) number of sample oracle calls required to obtain an ϵ-optimal policy is referred to as the *sample complexity* of the algorithm.

The state value function is closely related to the *state-action value function*, which is the expected cost starting from state s and taking action a:

$$Q^{\pi}(s,a) = \mathbb{E}_{\substack{s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_t),\\ a_{t+1}\sim\pi(\cdot|s_{t+1})}}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t) \;\middle|\; s_0 = s, a_0 = a\right].$$

By definition, the value functions are bounded:

$$0 \leq V^{\pi}(s) \leq \frac{C}{1-\gamma}, \quad 0 \leq Q^{\pi}(s,a) \leq \frac{C}{1-\gamma}. \tag{2}$$

The value functions satisfy the following relations:

$$V^{\pi}(s) = \langle Q^{\pi}(s,\cdot), \pi(\cdot|s)\rangle = \mathbb{E}_{a\sim\pi(\cdot|s)}[Q^{\pi}(s,a)], \tag{3}$$

$$Q^{\pi}(s,a) = c(s,a) + \gamma\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V^{\pi}(s')]. \tag{4}$$

For the convenience of analysis, we define recursively

$$\mathcal{P}_0^{\pi} = \rho, \quad \mathcal{P}_{t+1}^{\pi} = \mathbb{E}_{s\sim\mathcal{P}_t^{\pi}, a\sim\pi(\cdot|s)}[\mathcal{P}(\cdot|s,a)], \tag{5}$$

and define the *state visitation distribution* and the *state-action visitation distribution* respectively:

$$\nu_{\rho}^{\pi} = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\mathcal{P}_t^{\pi}, \tag{6}$$

$$\bar{\nu}_{\rho}^{\pi}(s,a) = \nu_{\rho}^{\pi}(s) \times \pi(a|s), \ \forall s \in \mathcal{S}, a \in \mathcal{A}. \tag{7}$$

The prefactor $1-\gamma$ in (6) makes ν_{ρ}^{π} be a distribution. The visitation distributions reflect the frequency of visiting state s or state-action pair (s,a) along the trajectories starting from $s_0 \sim \rho$ and taking actions according to policy π. It follows immediately from the definition that the state visitation distribution is lower bounded by the initial distribution (in terms of Radon–Nikodym derivative) with factor $1-\gamma$:

$$\frac{\mathrm{d}\nu_{\rho}^{\pi}}{\mathrm{d}\rho} = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\frac{\mathrm{d}\mathcal{P}_t^{\pi}}{\mathrm{d}\rho} \geq (1-\gamma)\frac{\mathrm{d}\mathcal{P}_0^{\pi}}{\mathrm{d}\rho} = 1-\gamma. \tag{8}$$

We can rewrite the value functions using the visitation distributions:

$$V^{\pi}(\rho) = \sum_{t=0}^{\infty}\gamma^t\int_{\mathcal{S}}\sum_{a\in\mathcal{A}}c(s,a)\pi(a|s)\,\mathrm{d}\mathcal{P}_t^{\pi}(s) = \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim\bar{\nu}_{\rho}^{\pi}}[c(s,a)], \tag{9}$$

$$Q^{\pi}(s,a) = c(s,a) + \gamma V^{\pi}(\mathcal{P}(\cdot|s,a)) = c(s,a) + \frac{\gamma}{1-\gamma}\mathbb{E}_{(s',a')\sim\bar{\nu}_{\mathcal{P}(\cdot|s,a)}^{\pi}}[c(s',a')], \tag{10}$$

where (10) is from (9) and (4).

2.2 Riemannian Manifold

We consider the state space $\mathcal{S}$ to be a d-dimensional Riemannian manifold isometrically embedded in $\mathbb{R}^D$. A *chart* for $\mathcal{S}$ is a pair (U, ϕ) such that $U \subset \mathcal{S}$ is open and $\phi : U \to \mathbb{R}^d$ is a homeomorphism, i.e., ϕ is a bijection; its inverse and itself are continuous. Two charts (U, ϕ) and (V, ψ) are called C^k *compatible* if and only if

$$\phi \circ \psi^{-1} \colon \psi(U \cap V) \to \phi(U \cap V) \quad \text{and} \quad \psi \circ \phi^{-1} \colon \phi(U \cap V) \to \psi(U \cap V)$$

are both C^k functions (k times continuously differentiable). A C^k *atlas* of $\mathcal{S}$ is a collection of C^k compatible charts $\{(U_i, \phi_i)\}_{i \in I}$ such that $\bigcup_{i \in I} U_i = \mathcal{S}$. An atlas of $\mathcal{S}$ contains an open cover of $\mathcal{S}$ and mappings from each open cover to $\mathbb{R}^d$.

Definition 1 (Smooth manifold) *A manifold $\mathcal{S}$ is smooth if it has a C^∞ atlas.*

We introduce the *reach* (Federer, 1959; Niyogi et al., 2008) of a manifold to characterize the curvature of $\mathcal{S}$.

Definition 2 (Reach) *The medial axis of $\mathcal{S}$ is defined as $\overline{\mathcal{T}}(\mathcal{S})$, which is the closure of*

$$\mathcal{T}(\mathcal{S}) = \{x \in \mathbb{R}^D \mid \exists x_1 \neq x_2 \in \mathcal{S}, \|x - x_1\|_2 = \|x - x_2\|_2 = \inf_{y \in \mathcal{S}} \|x - y\|_2\}.$$

The reach ω of $\mathcal{S}$ is the minimum distance between $\mathcal{S}$ and $\overline{\mathcal{T}}(\mathcal{S})$, that is,

$$\omega = \inf_{x \in \overline{\mathcal{T}}(\mathcal{S}), y \in \mathcal{S}} \|x - y\|_2.$$

Roughly speaking, reach measures how fast a manifold "bends". A manifold with a large reach "bends" relatively slowly. On the contrary, a small ω signifies more complicated local geometric structures, which are possibly hard to fully capture.

2.3 Convolutional Neural Networks

We consider one-sided stride-one convolutional neural networks (CNNs) with the rectified linear unit (ReLU) activation function $\mathrm{ReLU}(z) = \max(z, 0)$. Specifically, a CNN we consider consists of a padding layer, several convolutional blocks, and finally a fully connected output layer.

Given an input vector $x \in \mathbb{R}^D$, the network first applies a padding operator $P : \mathbb{R}^D \to \mathbb{R}^{D \times C}$ for some integer $C \geq 1$ such that

$$Z = P(x) = \begin{bmatrix} x & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{D \times C}.$$

Then the matrix Z is passed through M convolutional blocks. We will denote the input matrix to the m-th block as Z_m and its output as Z_{m+1} (so that $Z_1 = Z$).

We now define convolution as illustrated in Figure 1. Let $\mathcal{W} = (\mathcal{W}_{j,i,l})_{j,i,l} \in \mathbb{R}^{C' \times I \times C}$ be a filter where C' is the output channel size, I is the filter size and C is the input channel size. For $Z \in \mathbb{R}^{D \times C}$, the convolution of Z with $\mathcal{W}$, denoted with $\mathcal{W} * Z$, results in $Y \in \mathbb{R}^{D \times C'}$ with

$$Y_{k,j} = \sum_{i=1}^{I} \sum_{l=1}^{C} \mathcal{W}_{j,i,l} Z_{k+i-1,l},$$

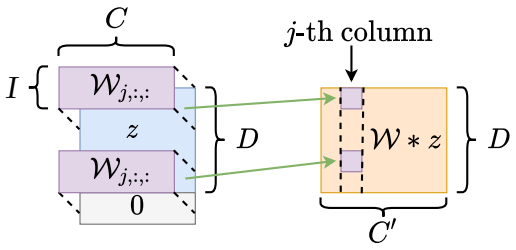


Figure 1: Convolution of $\mathcal{W} * Z$. $\mathcal{W}_{j,:,:}$ is a $I \times C$ matrix for the j-th output channel.

where we set $Z_{k+i-1,l} = 0$ for $k + i - 1 > D$.

In the m-th convolutional block, let $\mathcal{W}_m = \{\mathcal{W}_m^{(1)}, \ldots, \mathcal{W}_m^{(L_m)}\}$ be a collection of filters and $\mathcal{B}_m = \{\mathcal{B}_m^{(1)}, \ldots, \mathcal{B}_m^{(L_m)}\}$ be a collection of biases of proper sizes. The m-th block maps its input matrix $Z_m \in \mathbb{R}^{D\times C}$ to $Z_{m+1} \in \mathbb{R}^{D\times C}$ by

$$Z_{m+1} = \mathrm{ReLU}\left(\mathcal{W}_m^{(L_m)} * \cdots * \mathrm{ReLU}\left(\mathcal{W}_m^{(1)} * Z_m + \mathcal{B}_m^{(1)}\right) \cdots + \mathcal{B}_m^{(L_m)}\right) \tag{11}$$

with ReLU applied entrywise. For notational simplicity, we denote this series of operations in the m-th block with a single operator from $\mathbb{R}^{D\times C}$ to $\mathbb{R}^{D\times C}$ with $\mathrm{Conv}_{\mathcal{W}_m,\mathcal{B}_m}$, so (11) can be abbreviated as

$$Z_{m+1} = \mathrm{Conv}_{\mathcal{W}_m,\mathcal{B}_m}(Z_m).$$

Overall, we denote the mapping from input x to the output of the M-th convolutional block as

$$G(x) = (\mathrm{Conv}_{\mathcal{W}_M,\mathcal{B}_M}) \circ \cdots \circ (\mathrm{Conv}_{\mathcal{W}_1,\mathcal{B}_1}) \circ P(x). \tag{12}$$

Given (12), a CNN applies an additional fully connected layer to G and outputs

$$f(x) = W \otimes G(x) + b,$$

where $W \in \mathbb{R}^{D\times C}$ and $b \in \mathbb{R}$ are a weight matrix and a bias, respectively, and $\otimes$ denotes the sum of entrywise product, that is, $W \otimes G(x) = \sum_{i,j} W_{i,j}[G(x)]_{i,j}$. Thus, we define a class of CNNs of the same architecture as

$$\begin{aligned}
&\mathcal{F}(M, L, J, I, R_1, R_2) = \\
&\big\{f \mid f(x) = W \otimes G(x) + b, \|W\|_\infty \vee |b| \le R_2, \text{ where } G(x) \text{ is in (12) with } M \text{ blocks.} \\
&\quad \mathcal{W}_m^{(l)} \in \mathbb{R}^{C_m^{(l+1)} \times I_m^{(l)} \times C_m^{(l)}}, \mathcal{B}_m^{(l)} \in \mathbb{R}^{D\times C_m^{(l+1)}}, \text{ where } C_m^{(l)} \le J, I_m^{(l)} \le I, \ \forall l \in [L], m \in [M]; \\
&\quad \max_{m,l} \|\mathcal{W}_m^{(l)}\|_\infty \vee \|\mathcal{B}_m^{(l)}\|_\infty \le R_1\big\}.
\end{aligned} \tag{13}$$

3. Neural Policy Mirror Descent

In this section, we present the neural policy mirror descent (NPMD) algorithm. It is an extension of the policy mirror descent (PMD) method with a *critic network* Q_w parameterized

by w to approximate the state-action value function, and an *actor network* f_θ parameterized by θ to determine the policy. Both networks belong to a neural network function class $\mathcal{F}$, which we will specify later in Section 4.

The NPMD algorithm starts from a uniform policy π_0. At the k-th iteration (indexed from 0), the policy π_k is determined by the actor network f_{θ_k} along with a hyperparameter λ_k. The NPMD algorithm first performs a critic update, training the critic network Q_{w_k} to fit the state-action value function of the current policy. Then, the NPMD algorithm performs an actor update, indirectly obtaining an improved policy π_{k+1} by updating the actor network to $f_{\theta_{k+1}}$.

3.1 Critic Update

For the critic update at the k-th iteration, the goal is to approximate the exact state-action value function Q^{π_k} with the critic network Q_{w_k}. The component of Q_{w_k} corresponding to each action $a \in \mathcal{A}$ is a neural network $Q_{w_k}(\cdot, a) \in \mathcal{F}$ parameterized by $w_{k,a}$, which takes $s \in \mathcal{S}$ as input and outputs a scalar. For simplicity, we let all $|\mathcal{A}|$ networks share the same architecture and denote $w_k \coloneqq (w_{k,a})_{a\in\mathcal{A}} \in \mathcal{W}_k$. We define the critic loss as

$$\mathcal{L}_{\text{critic}}(w_k; \pi_k) = \mathbb{E}_{s\sim\nu_\rho^{\pi_k}} \|Q_{w_k}(s,\cdot) - Q^{\pi_k}(s,\cdot)\|_2^2, \tag{14}$$

where $Q_{w_k}(s,\cdot)$ and $Q^{\pi_k}(s,\cdot)$ are $|\mathcal{A}|$-dimensional vectors and $\nu_\rho^{\pi_k}$ is the state visitation distribution defined as (6).

Directly minimizing the critic loss (14) is difficult since Q^{π_k} is unknown in advance. Instead, we sample N states $\{s_{a,i}\}_{i=1}^N$ independently from the distribution $\nu_\rho^{\pi_k}$ for every action $a \in \mathcal{A}$ and use the empirical risk on these samples to approximate (14). For notation simplicity, we omit the iteration index k of samples. The empirical risk $\widehat{\mathcal{L}}_{\text{critic}}$ is defined as

$$\widehat{\mathcal{L}}_{\text{critic}}(w_k; \Xi_k) = \frac{1}{N}\sum_{a\in\mathcal{A}}\sum_{i=1}^{N}\left|Q_{w_k}(s_{a,i},a) - c(s_{a,i},a) - \frac{\gamma}{1-\gamma}c(s'_{a,i},a'_{a,i})\right|^2, \tag{15}$$

where N is the sample size, each pair $(s'_{a,i}, a'_{a,i})$ is sampled from distribution $\bar{\nu}^{\pi_k}_{\mathcal{P}(\cdot|s_{a,i},a)}$,[4] and Ξ_k denotes the collection of samples. We let w_k be the solution to the empirical risk minimization (ERM) problem, namely

$$w_k = \operatorname*{argmin}_{w\in\mathcal{W}_k} \widehat{\mathcal{L}}_{\text{critic}}(w; \Xi_k). \tag{16}$$

3.2 Actor Update

For the actor update, the goal is to learn an improved policy. If no actor function approximation is considered, an ideal PMD update is given by (see Lan 2023):

$$\pi^\star_{k+1}(s) = \operatorname*{argmin}_{\pi(\cdot|s)\in\Delta_\mathcal{A}} \langle Q_{w_k}(s,\cdot), \pi(\cdot|s)\rangle + \frac{1}{\eta_k} D^\pi_{\pi_k}(s),\ \forall s \in \mathcal{S}, \tag{17}$$

4. We can acquire one sample from this distribution once the sampling algorithm terminates at $s_{a,i}$. Indeed, we can take action a and restart sampling without resetting the environment, so the distribution would be $\bar{\nu}^{\pi_k}_{\mathcal{P}(\cdot|s_{a,i},a)}$ as desired.

where $D_{\pi_k}^{\pi}(s)$ is the Kullback-Leibler (KL) divergence between π and π_k and η_k is the step size. The PMD update (17) coincides with the KL-penalty version of the PPO algorithm (Schulman et al., 2017; Liu et al., 2019).

With neural function approximation, we train a neural policy π_{k+1} to approximate the ideal policy $\pi_{k+1}^{\star}$. For any $k \geq 0$, the neural policy π_k takes the form

$$\pi_k(a|s) = \frac{\exp\big(\lambda_k^{-1} f_{\theta_k}(s,a)\big)}{\sum_{a' \in \mathcal{A}} \exp\big(\lambda_k^{-1} f_{\theta_k}(s,a')\big)}, \tag{18}$$

where $\theta_k \coloneqq (\theta_{k,a})_{a \in \mathcal{A}}$ is the collection of neural network parameters and $\lambda_k > 0$ is a temperature parameter (will be discussed later). For any $a \in \mathcal{A}$, $f_{\theta_k}(\cdot, a) \in \mathcal{F}$ is a neural network parameterized by $\theta_{k,a}$, which takes $s \in \mathcal{S}$ as input and outputs a scalar. Again, we let all $|\mathcal{A}|$ neural networks share the same parameter space and denote $\theta_k \coloneqq (\theta_{k,a})_{a \in \mathcal{A}} \in \Theta_k$. With definition (18), the ideal PMD update (17) admits a closed-form solution.

Lemma 3 *The exact solution of* (17) *with neural policy* π_k *defined as* (18) *is given by*

$$\pi_{k+1}^{\star}(a|s) = \frac{\exp\big(g_{k+1}^{\star}(s,a)\big)}{\sum_{a' \in \mathcal{A}} \exp\big(g_{k+1}^{\star}(s,a')\big)}, \tag{19}$$

where $g_{k+1}^{\star} = \lambda_k^{-1} f_{\theta_k} - \eta_k Q_{w_k}$.

The proof of Theorem 3 is given in Appendix B.1. In view of Theorem 3, approximating $\pi_{k+1}^{\star}$ with π_{k+1} is equivalent to approximating $g_{k+1}^{\star}$ with the scaled actor network $\lambda_{k+1}^{-1} f_{\theta_{k+1}}$. We define the actor loss to be minimized as

$$\mathcal{L}_{\text{actor}}(\theta_{k+1}; \theta_k, w_k) = \mathbb{E}_{s \sim \nu_\rho^{\pi_k}} \big\| \lambda_{k+1}^{-1} f_{\theta_{k+1}}(s,\cdot) - \lambda_k^{-1} f_{\theta_k}(s,\cdot) + \eta_k Q_{w_k}(s,\cdot) \big\|_2^2, \tag{20}$$

where λ_k is the current temperature, λ_{k+1} is the next temperature, and η_k is the step size. For notation simplicity, we omit the hyperparameters η_k, λ_{k+1} and λ_k in $\mathcal{L}_{\text{actor}}$. Similar to the critic update, instead of minimizing (20) directly, we minimize the empirical risk:

$$\begin{aligned} &\widehat{\mathcal{L}}_{\text{actor}}(\theta_{k+1}; \theta_k, w_k, \Xi_k) \\ =&\frac{1}{N} \sum_{a \in \mathcal{A}} \sum_{i=1}^{N} \big| \lambda_{k+1}^{-1} f_{\theta_{k+1}}(s_{a,i}, a) - \lambda_k^{-1} f_{\theta_k}(s_{a,i}, a) + \eta_k Q_{w_k}(s_{a,i}, a) \big|^2, \end{aligned} \tag{21}$$

where Ξ_k contains the same sampled states $\{s_{a,i}\}_{i=1}^N$ from $\nu_\rho^{\pi_k}$ as used in the critic update. The improved actor parameter θ_{k+1} is given by the solution to the ERM problem:

$$\theta_{k+1} = \operatorname*{argmin}_{\theta \in \Theta_{k+1}} \widehat{\mathcal{L}}_{\text{actor}}(\theta; \theta_k, w_k, \Xi_k). \tag{22}$$

When the sample size N is sufficiently large, we have $\lambda_{k+1}^{-1} f_{\theta_{k+1}} \approx g_{k+1}^{\star}$ and hence $\pi_{k+1} \approx \pi_{k+1}^{\star}$.

Algorithm 1: Neural Policy Mirror Descent

Input: Iteration number K, initial distribution ρ, sample size per iteration N, step size $\eta_k > 0$, temperature parameter $\lambda_k > 0$, discount factor $\gamma \in (0,1)$, neural network parameter space $\mathcal{W}, \Theta$

Initialize $\theta_0 = 0$, $\Xi_k = \emptyset,\ \forall k \geq 0$;

for $k = 0$ **to** $K-1$ **do**

  for $a \in \mathcal{A}$ **do**

    Sample $\{s_{a,i}\}_{i=1}^N$ with $s_{a,i} \sim \nu_\rho^{\pi_k}$;

    Sample $\{(s'_{a,i}, a'_{a,i})\}_{i=1}^N$ with $(s'_{a,i}, a'_{a,i}) \sim \bar{\nu}^{\pi_k}_{\mathcal{P}(\cdot|s_{a,i},a)}$;

    Update $\Xi_k \leftarrow \Xi_k \cup \{(s_{a,i}, s'_{a,i}, a'_{a,i})\}_{i=1}^N$;

  end

  Update $w_k \leftarrow \operatorname{argmin}_{w \in \mathcal{W}_k} \widehat{\mathcal{L}}_{\text{critic}}(w; \pi_k, \Xi_k)$ as (16) ; `// Critic update`

  Update $\theta_{k+1} \leftarrow \operatorname{argmin}_{\theta \in \Theta_{k+1}} \widehat{\mathcal{L}}_{\text{actor}}(\theta; \theta_k, w_k, \Xi_k)$ as (22) ; `// Actor update`

end

Output: θ_K as the policy parameter

Remark 4 *The temperature parameter λ_k is introduced mainly for technical reasons. For any infinite-horizon discounted MDP, there always exists a deterministic optimal policy $\pi^\star$ (Puterman, 1994), while the neural policy π_k adopted to approximate $\pi^\star$ is fully stochastic in the sense that $\pi_k(a|s) > 0$ for any $(s,a) \in \mathcal{S} \times \mathcal{A}$. Using the temperature parameter λ_k allows us to control the spikiness of π_k. As λ_k approaches zero, π_k is prone to the action with the maximal value of f_{θ_k}. This makes stochastic policy π_k closer to the deterministic policy $\pi^\star$.*

Remark 5 *Our algorithm uses neural policy π_{k+1} to approximate the ideal policy $\pi^\star_{k+1}$. This allows us to keep the most up-to-date policy with only one actor network. If no actor network is used to approximate $\pi^\star_{k+1}$, ideally, we can obtain an implicitly defined policy by iteratively calling* (17)*. However, this requires us to keep all the $k+1$ critic networks to compute $\pi^\star_{k+1}$, which is not scalable. On the other hand, the critic network, serving solely for training the improved policy π_{k+1}, is dispensable. Consequently, we can remove the whole critic part and replace Q_{w_k} in* (21) *by target values in* (15)*. This streamlined approach makes policy evaluation implicit and reduces computation overhead. Nevertheless, it cannot simplify function approximation for an improved sample complexity, as the (lack of) smoothness in f_{θ_k} remains a bottleneck (see Section 4.2), and the critic error does not vanish but resides in the form of sample noise. We keep the actor-critic framework for a more elucidated analysis.*

We summarize NPMD in Algorithm 1. Note that Algorithm 1 requires samples from the visitation distributions. We provide a sampling algorithm in Appendix A.

4. Main Results

In this section, we present our main results on the sample complexity of Algorithm 1. As mentioned in Section 1, we focus on RL environments with low-dimensional structures, for which we make the following smooth manifold assumption on the state space.

Assumption 1 (State space manifold) *The state space $\mathcal{S}$ is a d-dimensional compact Riemannian manifold isometrically embedded in $\mathbb{R}^D$ where $d \ll D$. There exists $B > 0$ such that $\|x\|_\infty \leq B$ for any $x \in \mathcal{S}$. The surface area of $\mathcal{S}$ is $\mathrm{Area}(\mathcal{S}) < \infty$, and the reach of $\mathcal{S}$ is $\omega > 0$.*

We first derive the iteration complexity with well-approximated value functions and neural policies, then derive the number of samples to meet the requirement for approximation. Combining the results together, we establish the overall sample complexity for Algorithm 1.

4.1 Iteration Complexity

We make the following assumptions on the visitation distributions for iteration complexity.

Assumption 2 (Mismatch) *There exists $\kappa_\nu < \infty$ such that for all $k \geq 0$ iterations of Algorithm 1,*

$$\left\| \frac{\mathrm{d}\nu_\rho^{\pi^\star}}{\mathrm{d}\nu_\rho^{\pi_k}} \right\|_\infty \leq \kappa_\nu.$$

Accordingly, we can define the shifted discount factor as $\gamma_\rho \coloneqq 1 - 1/\kappa_\nu$.

Assumption 2 requires that the distribution mismatch between visitation distributions corresponding to π_k and $\pi^\star$ are uniformly controlled by a mismatch coefficient κ_ν. The Radon–Nikodym derivative on the left of the inequality exists by the following lemma.

Lemma 6 *If $\pi(a|s) > 0$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, then $\nu_\rho^{\pi'} \ll \nu_\rho^\pi$ for any other policy π'.*

The proof is provided in Appendix C.1. Since π_k in Algorithm 1 has full support on $\mathcal{A}$ by construction, Theorem 6 ensures the absolute continuity of $\nu_\rho^{\pi^\star}$ with respect to $\nu_\rho^{\pi_k}$, hence the Radon–Nikodym derivative exists.

Assumption 2 holds, for example, when the initial distribution ρ has full support. Indeed, when ρ has full support, the visitation distribution ν_ρ^π also has full support by (8), regardless of the transition kernel and the given policy π. Suppose $\left\| \frac{\mathrm{d}\nu_\rho^{\pi^\star}}{\mathrm{d}\rho} \right\|_\infty = \kappa$, then Assumption 2 holds with $\kappa_\nu = \frac{\kappa}{1-\gamma}$ and $\gamma_\rho = 1 - \frac{1-\gamma}{\kappa}$. Assumption 2 can also hold without the full support of ρ. For instance, if different actions generate the same transition probability, then Assumption 2 trivially holds with $\kappa_\nu = 1$ regardless of ρ.

Assumption 3 (Concentrability) *There exists $C_\nu < \infty$ such that for all $k \geq 0$ iterations of Algorithm 1,*

$$\chi^2(\nu_\rho^\pi, \nu_\rho^{\pi_k}) + 1 \leq C_\nu,$$

where π takes π_{k+1} or $\pi^\star$, $\chi^2(\nu_\rho^\pi, \nu_\rho^{\pi_k}) = \mathbb{E}_{\nu_\rho^{\pi_k}}[(\frac{\mathrm{d}\nu_\rho^\pi}{\mathrm{d}\nu_\rho^{\pi_k}} - 1)^2]$ is the χ^2-divergence.

Assumption 3 requires the concentrability of visitation distributions. The distance between visitation distributions is measured by the χ^2-divergence, which is well-defined under

Assumption 2 as absolute continuity holds. This type of concentrability assumption is commonly adopted in the RL literature (Agarwal et al., 2021; Yuan et al., 2023) and is tighter than the absolute density ratio $\left\| \frac{\mathrm{d}\nu_\rho^\pi}{\mathrm{d}\nu_\rho^{\pi_k}} \right\|_\infty$. Assumptions 2 and 3 together require π_k at each iteration to be sufficiently exploratory so that the on-policy sampling distribution $\nu_\rho^{\pi_k}$ covers the optimal distribution $\nu_\rho^{\pi^\star}$ to a large extent. Intuitively, these are reasonable assumptions as we start from a uniform policy π_0 that is fully exploratory. Moreover, as the later iterate π_k approaches $\pi^\star$, their visitation distributions $\nu_\rho^{\pi_k}$ and $\nu_\rho^{\pi^\star}$ should have more overlap. However, rigorously verifying Assumptions 2 and 3 with quantified coefficients is challenging, as they depend on both the environment and the iterates.

With Assumptions 2 and 3, we have the following one-step improvement lemma for Algorithm 1. The proof is provided in Appendix C.3.

Lemma 7 *Suppose Assumptions 2 and 3 hold. Then Algorithm 1 yields*

$$\begin{aligned}\big(V^{\pi_{k+1}}(\rho) - V^\star(\rho)\big) + \frac{1}{(1-\gamma)\kappa_\nu \eta_k}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\big[D_{\pi_{k+1}}^{\pi^\star}(s)\big] \\ \le \gamma_\rho\big(V^{\pi_k}(\rho) - V^\star(\rho) + \frac{1}{(1-\gamma)\kappa_\nu\gamma_\rho\eta_k}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\big[D_{\pi_k}^{\pi^\star}(s)\big]\big) \\ + \frac{4\sqrt{C_\nu}}{(1-\gamma)}\Big(\sqrt{\mathcal{L}_{\mathrm{critic}}(w_k;\pi_k)} + \frac{1}{\eta_k}\sqrt{\mathcal{L}_{\mathrm{actor}}(\theta_{k+1};\theta_k,w_k)}\Big).\end{aligned}$$

Theorem 7 demonstrates that the optimality gap in value function decreases at rate γ_ρ up to approximation errors introduced by the critic and actor updates. When these errors are properly controlled, we can establish iteration complexity for Algorithm 1.

Theorem 8 *Suppose Assumptions 2 and 3 hold. If $\eta_k = \frac{1-\gamma_\rho}{C\gamma_\rho^{k+1}}$ for all $k \ge 0$, the critic loss and the actor loss satisfy respectively*

$$\mathbb{E}\big[\mathcal{L}_{\mathrm{critic}}(w_k;\pi_k)\big] \le C^2\gamma_\rho^{2(k+1)}, \quad \mathbb{E}\big[\mathcal{L}_{\mathrm{actor}}(\theta_{k+1};\theta_k,w_k)\big] \le (1-\gamma_\rho)^2,$$

then after $k \ge 1$ iterations, the expected optimality gap of π_k given by Algorithm 1 is

$$\mathbb{E}\big[V^{\pi_k}(\rho) - V^\star(\rho)\big] \le \gamma_\rho^k\big(C_1 + C_2(k+1)\big)\cdot\frac{C}{1-\gamma},$$

where $C_1 = 1 + \log|\mathcal{A}|$, $C_2 = 8\sqrt{C_\nu}$.

Moreover, for any $\epsilon > 0$, the number of iterations required for $\mathbb{E}\big[V^{\pi_k}(\rho) - V^\star(\rho)\big] \le \epsilon$ is

$$\widetilde{O}\left(\log_{\frac{1}{\gamma_\rho}}\left(\frac{C\left(\sqrt{C_\nu} + \log|\mathcal{A}|\right)}{(1-\gamma)(1-\gamma_\rho)\epsilon}\right)\right).$$

The proof of Theorem 8 provided in Appendix C.4 mostly follows the one for finite state space in Lan (2023), with a different set of error conditions that are applicable for continuous state space. We choose exponentially increasing step size η_k to establish (almost) linear convergence rate. To achieve this fast rate, we require the critic loss to be exponentially decreasing and the actor loss to be small for any consecutive temperatures λ_k and λ_{k+1}. In fact, if we convert the loss into function approximation error (see Appendices A.2 and A.3)

and choose λ_k to be exponentially decreasing at rate γ_ρ, then the requirements on critic and actor networks become the same. We show in the following subsections that these requirements can be met by properly designing the CNN architecture and using sufficiently many training samples. As long as the conditions are satisfied, Theorem 8 guarantees to find an ϵ-optimal policy (in expectation) after $\widetilde{O}(\log \frac{1}{\epsilon})$ iterations.

Remark 9 *Theorem 8 suggests that we can use smaller networks or fewer samples in the early stage and gradually increase the sizes along iterations. When the tolerance ϵ is given, the error bound for the last iteration is $\sqrt{\mathcal{L}_{\text{critic}}(w_K;\pi_K)}+\frac{1}{\eta_K}\sqrt{\mathcal{L}_{\text{actor}}(\theta_{K+1};\theta_K,w_K)} = \widetilde{O}(\epsilon)$. Notably, this coincides (up to logarithm factors) with the stipulated approximation error criterion in prior literature (Yuan et al., 2023; Alfano et al., 2023), with the distinction that their criteria are applied uniformly to all iterations. If we use a constant step size, then the iteration complexity becomes $O(\epsilon^{-1})$ as in Alfano et al. (2023), while the error criteria for the last iterate remains $O(\epsilon)$. Consequently, the overall sample complexity becomes much worse as it requires more iterations to converge.*

4.2 Function Approximation on Lipschitz MDP with CNN

The iteration complexity in Theorem 8 is valid if the state-action value function Q^{π_k} and the policy $\pi^\star_{k+1}$ are well approximated by the critic and actor networks at each iteration. However, we have not yet specified the CNN architecture that can meet these requirements. In this section, we study the function approximation on *Lipschitz MDP*, which possesses Lipschitz transition kernel $\mathcal{P}$ and Lipschitz cost function c. Here, the Lipschitzness is defined with respect to the *geodesic distance* on the state space $\mathcal{S}$, which is a d-dimensional Riemannian manifold (Assumption 1). Recall the definition of geodesic distance:

Definition 10 (Geodesic distance) *The geodesic distance between two points $x, y \in \mathcal{S}$ is defined as*

$$d_{\mathcal{S}}(x,y) \coloneqq \inf_{\Gamma\colon [0,1]\to\mathcal{S}} \int_0^1 \left\|\Gamma'(t)\right\|_2 \mathrm{d}t \qquad (23)$$
$$s.t.\ \ \Gamma(0)=x,\ \Gamma(1)=y,\ \Gamma \text{ is piecewise } C^1.$$

One can show the existence of a solution to the minimization problem (23) under mild conditions and that $d_{\mathcal{S}}(\cdot,\cdot)$ is indeed a distance. More references can be found in Do Carmo and Flaherty Francis (1992). With geodesic distance, we define Lipschitz functions on the Riemannian manifold $\mathcal{S}$.

Definition 11 (Lipschitz function) *Let $L \geq 0$ and $\alpha \in (0,1]$ be constants. A function $f\colon \mathcal{S}\to\mathbb{R}$ is called (L,α)-Lipschitz if for any $x, y \in \mathcal{S}$,*

$$|f(x)-f(y)| \leq L\cdot d^\alpha_{\mathcal{S}}(x,y).$$

For any fixed α, the Lipschitz constant L in Theorem 11 measures the smoothness of the function. A function is considered smooth if it possesses a small Lipschitz constant, whereas a non-smooth function will exhibit a large Lipschitz constant. Throughout the remainder of this paper, when we mention Lipschitzness, we specifically mean the property of being Lipschitz continuous with a moderate constant.

Remark 12 *The geodesic distance $d_{\mathcal{S}}$ in Theorem 11 is a global distance rather than a local one. This makes our definition of Lipschitz functions different from those based on local Euclidean distance and partition of unity as in Chen et al. (2019) and Liu et al. (2021). The two ways of defining Lipschitzness have some technical differences, but they agree with each other in our setting up to constant factors.*

When the atlas $\{(U_i, \phi_i)\}_{i\in I}$ are local projections onto tangent spaces as in Chen et al. (2019), the local Euclidean distance between two points in the same open set U_i is no greater than their Euclidean distance in $\mathbb{R}^D$, which is further less than their geodesic distance, hence the Lipschitzness defined with local distances implies Theorem 11.

On the other hand, when the curvature of manifold $\mathcal{S}$ is not too large compared to the radius of the open set U_i, then Theorem 11 also implies Lipschitzness in the Euclidean sense on each local coordinate $\phi_i(U_i) \subset [0,1]^d$ (Theorem 33). Here, we adopt the global definition for simplicity.

We now formally define the Lipschitz MDP condition, which ensures the Lipschitzness of the state-action value function Q^π for any policy π.

Assumption 4 (Lipschitz MDP) *There exist constants $L_{\mathcal{P}}, L_c \geq 0$ and $\alpha \in (0,1]$ such that for any tuple $(s, s', a) \in \mathcal{S} \times \mathcal{S} \times \mathcal{A}$, the cost function $c(\cdot, a)\colon \mathcal{S} \to \mathbb{R}$ is (L_c, α)-Lipschitz and the transition kernel $\mathcal{P}$ satisfies*

$$d_{\mathrm{TV}}(\mathcal{P}(\cdot|s,a), \mathcal{P}(\cdot|s',a)) \leq L_{\mathcal{P}} \cdot d_{\mathcal{S}}^{\alpha}(s,s'),$$

where $d_{\mathrm{TV}}(\cdot,\cdot)$ is the total variation distance.

Assumption 4 requires that when two states are close to each other, taking the same action would admit similar transition distributions and corresponding costs. This assumption holds for many spatially smooth environments, especially those driven by physical simulations such as MuJuCo (Todorov et al., 2012) and classic control environments (Brockman et al., 2016). Under Assumption 4, we show in Theorem 13 that the state-action value function Q^π is Lipschitz regardless of the evaluated policy π. The proof of Theorem 13 is provided in Appendix E.3.

Lemma 13 *If Assumption 4 holds, then for any policy π and any action $a \in \mathcal{A}$, the state-action value function $Q^\pi(\cdot, a)\colon \mathcal{S} \to \mathbb{R}$ is (L_Q, α)-Lipschitz with $L_Q = L_c + \frac{\gamma C}{1-\gamma} L_{\mathcal{P}}$ being the Lipschitz constant. That is, for any policy π and any tuple $(s, s', a) \in \mathcal{S} \times \mathcal{S} \times \mathcal{A}$, we have*

$$|Q^\pi(s,a) - Q^\pi(s',a)| \leq L_Q \cdot d_{\mathcal{S}}^{\alpha}(s,s').$$

The Lipschitz constant $L_Q = L_c + \frac{\gamma C}{1-\gamma} L_{\mathcal{P}}$ scales linearly with the magnitude of the cost function c as L_c does. In view of this property, we define a normalized Lipschitz constant which is invariant to the scaling of the cost:

$$\overline{L}_Q \coloneqq (1-\gamma) L_c / C + \gamma L_{\mathcal{P}}. \tag{24}$$

This normalized Lipschitz constant is a convex combination of L_c/C and $L_{\mathcal{P}}$, so it will not exceed the larger one for any γ.

We make a few remarks on the Lipschitz condition.

Remark 14 *Assumption 4 is a sufficient condition for the Lipschitzness of Q^{π_k}, regardless of the smoothness of π_k. The Lipschitzness of the target function is a minimal requirement for approximation theory and it is almost essential even for simple regression problems. However, even if Assumption 4 does not hold, Q^{π_k} being Lipschitz is still possible. An extreme example is when state space $\mathcal{S} = \mathbb{S}^1$ is a circle and the transition is a fixed rotation whichever action is taken. In this case, the transition kernel is not Lipschitz since the total variation distance between transitions is always 1. Meanwhile, Q^{π_k} is Lipschitz for any π_k provided that c is Lipschitz. Similar arguments can be found in Fan et al. (2020) and Nguyen-Tang et al. (2022) for non-smooth MDP having smooth Bellman operator, but they implicitly involve the smoothness of neural policy π_k.*

Remark 15 *In practice, many environments adhere to the Lipschitz condition, with only an extremely small portion of states being exceptions. For example, in the Box2D Car Racing environment (Brockman et al., 2016), the cost remains constant for each frame until a tile is reached, for which the agent will be given a huge reward. Even though this type of partially smooth environment does not fulfill the Lipschitz condition on a global scale, it is reasonable to expect the existence of an environment that is globally smooth and satisfies Assumption 4. Such a globally smooth environment could serve as a regularization of the original non-smooth environment, which inevitably introduces bias to the problem. When this bias is negligible compared to the extent of smoothness, we can study the smooth approximation under Assumption 4 without loss of generality.*

With Theorem 13 established, we show that a CNN of the form (13) can uniformly approximate $Q^{\pi_k}(\cdot, a)$ for any $a \in \mathcal{A}$. The approximation error depends on the specified CNN architecture.

Theorem 16 (Critic approximation) *Suppose Assumptions 1 and 4 hold. For any integers $I \in [2, D]$ and $\widetilde{M}, \widetilde{J} > 0$, we let*

$$M = O(\widetilde{M}),\ L = O(\log(\widetilde{M}\widetilde{J}) + D + \log D),\ J = O(D\widetilde{J}),$$
$$R_1 = (8ID)^{-1}\widetilde{M}^{-\frac{1}{L}} = O(1),\ \log R_2 = O(\log^2(\widetilde{M}\widetilde{J}) + D\log(\widetilde{M}\widetilde{J})),$$

where $O(\cdot)$ hides a constant depending on $\log L_Q$, $\log \frac{C}{1-\gamma}$, d, α, ω, B, and the surface area $\mathrm{Area}(\mathcal{S})$. Then for any policy π and any action $a \in \mathcal{A}$, there exists a CNN $Q_w(\cdot, a) \in \mathcal{F}(M, L, J, I, R_1, R_2)$ such that

$$\|Q_w(\cdot, a) - Q^{\pi}(\cdot, a)\|_{\infty} \leq \frac{C}{1-\gamma}(\bar{L}_Q + 1)(\widetilde{M}\widetilde{J})^{-\frac{\alpha}{d}}.$$

Here, L_Q and $\bar{L}_Q$ are defined as in Theorem 13 and (24).

We provide a proof overview for Theorem 16 in Appendix E.1 and the detailed proof in Appendix E.2. Compared to our preliminary work (Theorem 1 in Ji et al. 2022) that deals with a larger class of Besov functions by using cardinal B-spline approximation as a crucial step, we simplify the proof for Lipschitz functions where first-order spline approximation is sufficient.

To bound the approximation error by ϵ, we require the number of parameters in $Q_w(\cdot, a)$ to be $O(MLJ^2I) = \widetilde{O}(D^3 I \epsilon^{-\frac{d}{\alpha}})$. Note that the exponent over ϵ is the intrinsic dimension d rather than the data dimension D, and the hidden terms in $\widetilde{O}(\cdot)$ also have no exponential dependence on D.[5] This implies that CNN approximation does not suffer from the curse of dimensionality when the data support has a low-dimensional manifold structure.

Next, we consider function approximation for policy $\pi^\star_{k+1}$ in the actor update. As mentioned in Section 3.2, our goal is to learn a deterministic optimal policy, which is equivalent to a discrete mapping from $\mathcal{S}$ to $\mathcal{A}$. Such a mapping is not continuous given the discrete nature of $\mathcal{A}$, so it is difficult to be approximated directly. Instead, we iteratively update a temperature-controlled neural policy π_k in the form of (18) to approximate the deterministic optimal policy $\pi^\star$. Although π_k is a stochastic policy by construction, it can approximate the deterministic policy that chooses the action $a \in \mathcal{A}$ with the maximal value of $f_{\theta_k}(s, a)$ by using a sufficiently small temperature $\lambda_k > 0$. Therefore, as long as the actor network $f_{\theta_k}(s, \cdot)$ is learned to admit a maximizer $a \in \operatorname{supp}(\pi^\star(\cdot|s))$ for any state $s \in \mathcal{S}$, it can serve as a good approximation of the optimal policy $\pi^\star$.

To learn such an actor network, we iteratively train a new actor network $f_{\theta_{k+1}}$ based on the current critic and actor networks Q_{w_k} and f_{θ_k}. According to Theorem 3, the target function for the next actor network $f_{\theta_{k+1}}$ is given by

$$\lambda_{k+1} g^\star_{k+1} = \lambda_k^{-1} \lambda_{k+1} f_{\theta_k} - \eta_k \lambda_{k+1} Q_{w_k},$$

which is a weighted sum of the current critic and actor networks. This approximation target is not Lipschitz, since Q_{w_k} is just an approximation of the Lipschitz function Q^{π_k}, not a Lipschitz function in itself. Consequently, Theorem 16 cannot be directly transferred to actor approximation. To address the issue, we introduce the *approximately Lipschitz* condition to describe the smoothness inherited from approximating a Lipschitz function.

Definition 17 (Approximate Lipschitzness) *Let $L, \epsilon \geq 0$ and $\alpha \in (0, 1]$ be constants. A function $f\colon \mathcal{S} \to \mathbb{R}$ is called (L, α, ϵ)-approximately Lipschitz if for any $x, y \in \mathcal{S}$,*

$$|f(x) - f(y)| \leq L \cdot d^\alpha_{\mathcal{S}}(x, y) + 2\epsilon.$$

Here, L is called the Lipschitz constant, and ϵ is called the proximity constant. When $\epsilon = 0$, f is (L, α)-Lipschitz as defined in Theorem 11.

Theorem 17 relaxes the Lipschitz condition by allowing a proximity constant ϵ. When $\epsilon = 0$, the condition reduces to the Lipschitz continuity, and the Lipschitz constant is exactly the one in Theorem 11. When $\epsilon \geq \|f\|_\infty$, the condition is vacuously true for all $L \geq 0$. When ϵ is somewhere between 0 and $\|f\|_\infty$, a larger ϵ allows a potentially smaller Lipschitz constant L. As Theorem 18 shows, the approximate Lipschitzness is a shared property of all uniform approximators of a Lipschitz function.

Lemma 18 *If $\overline{f}_0\colon \mathcal{S} \to \mathbb{R}$ is (L, α)-Lipschitz, $f\colon \mathcal{S} \to \mathbb{R}$ satisfies $\left\|f - \overline{f}_0\right\|_\infty \leq \epsilon$ for some $\epsilon > 0$, then f is (L, α, ϵ)-approximately Lipschitz.*

5. The hidden constant can be as large as $\exp\big(C(\mathcal{S}, \alpha) \cdot d^2\big)$ where constant $C(\mathcal{S}, \alpha)$ depends on α and manifold-related parameters including ω, B, $\operatorname{Area}(\mathcal{S})$, as implied by the proof. Nevertheless, given the assumption that $d \ll D$, $O(\exp\big(C(\mathcal{S}, \alpha) \cdot d^2\big))$ is still much smaller than the curse of dimensionality $O(\exp(D))$ and hence can be treated as constant, as it would not affect our main conclusions.

Proof For any $x, y \in \mathcal{S}$,

$$\begin{aligned}|f(x) - f(y)| &= \left|\overline{f}_0(x) - \overline{f}_0(y) + f(x) - \overline{f}_0(x) - f(y) + \overline{f}_0(y)\right| \\ &\le \left|\overline{f}_0(x) - \overline{f}_0(y)\right| + \left|f(x) - \overline{f}_0(x)\right| + \left|-f(y) + \overline{f}_0(y)\right| \\ &\le L \cdot d_{\mathcal{S}}^{\alpha}(x, y) + 2\epsilon.\end{aligned}$$

The first inequality comes from the triangle inequality. The second inequality is from L-Lipschitz continuity of $\overline{f}_0$ and that $\left\|f - \overline{f}_0\right\|_\infty \le \epsilon$. ■

It follows immediately from Theorem 16 and Theorem 18 that there exists an $(L_Q, \epsilon, \epsilon_Q)$-approximately Lipschitz Q_{w_k} that can uniformly approximate Q^{π_k} up to ϵ_Q error. Therefore, we can impose (approximately) Lipschitz restrictions on the CNN function class without damaging its approximation power for the state-action value function. To be more precise, for any CNN class $\mathcal{F} = \mathcal{F}(M, L, J, I, R_1, R_2)$, we define its Lipschitz-restricted version $\mathcal{F}_{\text{Lip}}(A, L_f, \alpha, \epsilon_f)$ as

$$\mathcal{F}_{\text{Lip}}(A, L_f, \alpha, \epsilon_f) = \{f \in \mathcal{F} \mid \|f\|_\infty \le A,\ f \text{ is } (L_f, \alpha, \epsilon_f)\text{-approximately Lipschitz}\}. \quad (25)$$

Moreover, we denote the parameter space of the Lipschitz-restricted critic and actor network classes as $\mathcal{W}_{\text{Lip}}$ and Θ_{Lip} respectively:

$$\mathcal{W}_{\text{Lip}}(A, L_Q, \alpha, \epsilon_Q) = \left\{w \mid Q_w(\cdot, a) \in \mathcal{F}_{\text{Lip}}(A, L_Q, \alpha, \epsilon_Q),\ \forall a \in \mathcal{A}\right\}, \quad (26)$$

$$\Theta_{\text{Lip}}(A, L_f, \alpha, \epsilon_f) = \left\{\theta \mid f_\theta(\cdot, a) \in \mathcal{F}_{\text{Lip}}(A, L_f, \alpha, \epsilon_f),\ \forall a \in \mathcal{A}\right\}. \quad (27)$$

Then by setting $\mathcal{W}_k = \mathcal{W}_{\text{Lip}}$ and $\Theta_k = \Theta_{\text{Lip}}$ in Algorithm 1, we ensure the k-th critic network Q_{w_k} and actor network f_{θ_k} are both approximately Lipschitz, and such a restriction on $\mathcal{W}_k$ will not affect the approximation power of Q_{w_k} for Q^{π_k}. In addition, the target function $\lambda_{k+1} g_{k+1}^\star$ for the next actor is also approximately Lipschitz since it is a weighted sum of two approximately Lipschitz functions. By carefully selecting the temperature parameters to match the configuration of η_k in Theorem 8, the approximate Lipschitzness of target functions for actor updates in all iterations can be uniformly controlled.

Lemma 19 *For $k \ge 0$, we let $\eta_k = \frac{1-\gamma_\rho}{C\gamma_\rho^{k+1}}$, $\lambda_k = \frac{C\gamma_\rho^k}{1-\gamma_\rho}$, $\mathcal{W}_k = \mathcal{W}_{\text{Lip}}(\frac{C}{1-\gamma}, L_Q, \alpha, \epsilon_Q)$ and $\Theta_k = \Theta_{\text{Lip}}(\frac{C}{(1-\gamma_\rho)(1-\gamma)}, \frac{L_Q}{1-\gamma_\rho}, \alpha, \frac{\epsilon_Q}{1-\gamma_\rho})$ with some $\epsilon_Q \ge 0$. Then the target actor $\lambda_{k+1} g_{k+1}^\star$ defined in Theorem 3 is $(\frac{L_Q}{1-\gamma_\rho}, \alpha, \frac{\epsilon_Q}{1-\gamma_\rho})$-approximately Lipschitz and is uniformly bounded by $\frac{C}{(1-\gamma_\rho)(1-\gamma)}$.*

Proof By Theorem 3 and our choice of η_k and λ_k, the target function of the k-th critic update is

$$\begin{aligned}\lambda_{k+1} g_{k+1}^\star &= \lambda_{k+1}\lambda_k^{-1} f_{\theta_k} - \eta_k \lambda_{k+1} Q_{w_k} \\ &= \gamma_\rho f_{\theta_k} - Q_{w_k}.\end{aligned}$$

We initialize $\theta_0 = 0$ in Algorithm 1, thus $\theta_0 \in \Theta_{\text{Lip}}(0, 0, \alpha, 0) \subseteq \Theta_0$. It is easy to verify that if $f\colon \mathcal{S} \to \mathbb{R}$ is $(L_f, \alpha, \epsilon_f)$-approximately Lipschitz and $g\colon \mathcal{S} \to \mathbb{R}$ is $(L_g, \alpha, \epsilon_g)$-approximately

Lipschitz, $c \in \mathbb{R}$, then $f+g$ is $(L_f+L_g, \alpha, \epsilon_f+\epsilon_g)$-approximately Lipschitz, and $c \cdot f$ is $(|c|L_f, \alpha, |c|\epsilon_f)$-approximately Lipschitz. Combining this fact and our choice of $\mathcal{W}_k$ and Θ_k, we have that $\lambda_1 g_1^\star$ is $(\frac{L_Q}{1-\gamma_\rho}, \alpha, \frac{\epsilon_Q}{1-\gamma_\rho})$-approximately Lipschitz and is uniformly bounded by $\frac{C}{(1-\gamma_\rho)(1-\gamma)}$. ■

Theorem 19 shows the target actor is approximately Lipschitz in each iteration with $\mathcal{W}_k$ and Θ_k inserted. It remains to derive the approximation error for actor update with Lipschitz-restricted class $\Theta_{k+1} = \Theta_{\text{Lip}}$. We first show in Theorem 20 that any bounded and approximately Lipschitz function on $\mathcal{S}$ can be well approximated by a CNN with enough parameters, and this CNN is also bounded and approximately Lipschitz.

Theorem 20 *Suppose Assumption 1 holds, the target function $f_0 \colon \mathcal{S} \to \mathbb{R}$ is bounded and $(L_f, \alpha, \epsilon_f)$-approximately Lipschitz. For any integers $I \in [2, D]$ and $\widetilde{M}, \widetilde{J} > 0$, we let*

$$M = O(\widetilde{M}),\ L = O(\log(\widetilde{M}\widetilde{J}) + D + \log D),\ J = O(D\widetilde{J}),$$
$$R_1 = (8ID)^{-1}\widetilde{M}^{-\frac{1}{I}} = O(1),\ \log R_2 = O(\log^2(\widetilde{M}\widetilde{J}) + D\log(\widetilde{M}\widetilde{J})),$$

where $O(\cdot)$ hides a constant depending on $\log L_f$, $\log \|f_0\|_\infty$, α, ω, B, and the surface area $\text{Area}(\mathcal{S})$. Then there exists a CNN $f \in \mathcal{F}(M, L, J, I, R_1, R_2)$ such that

$$\|f - f_0\|_\infty \le (L_f + \|f_0\|_\infty)(\widetilde{M}\widetilde{J})^{-\frac{\alpha}{d}} + 2\epsilon_f.$$

Moreover, this f can be $(L_f, \alpha, \widehat{\epsilon}_f)$-approximately Lipschitz with $\widehat{\epsilon}_f = (L_f + \|f_0\|_\infty)(\widetilde{M}\widetilde{J})^{-\frac{\alpha}{d}}$ and uniformly bounded by $\|f_0\|_\infty$.

The proof of Theorem 20 is provided in Appendix E.4. Theorem 20 shows the existence of an approximately Lipschitz CNN that is close under L^∞ norm to any approximately Lipschitz target function on $\mathcal{S}$. As a corollary, we obtain the approximation error for the actor update.

Corollary 21 (Actor approximation) *Suppose Assumptions 1 and 4 hold. For any integers $I \in [2, D]$ and $\widetilde{M}, \widetilde{J} > 0$, we let*

$$M = O(\widetilde{M}),\ L = O(\log(\widetilde{M}\widetilde{J}) + D + \log D),\ J = O(D\widetilde{J}),$$
$$R_1 = (8ID)^{-1}\widetilde{M}^{-\frac{1}{I}} = O(1),\ \log R_2 = O(\log^2(\widetilde{M}\widetilde{J}) + D\log(\widetilde{M}\widetilde{J})),$$

where $O(\cdot)$ hides a constant depending on $\log L_Q$, $\log \frac{C}{1-\gamma}$, d, α, ω, B, and the surface area $\text{Area}(\mathcal{S})$. If η_k, λ_k, $\mathcal{W}_k$ and Θ_k are as specified in Theorem 19 for all $k \ge 0$, $\epsilon_Q = \frac{C}{1-\gamma}(\overline{L}_Q + 1)(\widetilde{M}\widetilde{J})^{-\frac{\alpha}{d}}$, then for any $w_k \in \mathcal{W}_k$ and $\theta_k \in \Theta_k$, there exists $\theta \in \Theta_{k+1}$ such that

$$\left\|f_\theta(\cdot, a) - \lambda_{k+1} g_{k+1}^\star(\cdot, a)\right\|_\infty \le \frac{3\epsilon_Q}{1-\gamma_\rho}.$$

Here, $g_{k+1}^\star$, L_Q and $\overline{L}_Q$ are defined as in Theorems 3 and 13 and (24).

We note that the requirement for proximity constant ϵ_Q in Theorem 21 is the same as the approximation error in Theorem 16. This alignment maintains the consistency of the Lipschitz constraints imposed on $\mathcal{W}_k$, Θ_k, and Θ_{k+1}. In this case, the actor approximation error is comparable to the critic approximation error, and they both depend on the CNN architecture. Therefore, a large CNN class $\mathcal{F} = \mathcal{F}(M, L, J, I, R_1, R_2)$ guarantees the existence of good approximations to both the state-action value function and the policy.

4.3 Sample Complexity

We have demonstrated that CNN approximation can be applied to both the state-action value function and the policy. To make sure that the solutions to the ERM subproblems (16) and (22) indeed provide good approximations for Q^{π_k} and $\pi^\star_{k+1}$, the number of samples must be sufficient. In this section, we derive the sample complexity for Algorithm 1. To be more precise, we consider the expected number of oracle accesses to the transition kernel $\mathcal{P}$ and the cost function c for Algorithm 1 to find a policy π_K that satisfies $\mathbb{E}\big[V^{\pi_K}(\rho) - V^\star(\rho)\big] \le \epsilon$.

We keep using the same notation for CNN class $\mathcal{F} = \mathcal{F}(M, L, J, I, R_1, R_2)$ and its Lipschitz-restricted version $\mathcal{F}_{\text{Lip}}$ as defined in (25), as well as the parameter spaces $\mathcal{W}_{\text{Lip}}$ and Θ_{Lip} as denoted in (26) and (27). We show the following lemma that characterizes the number of samples N sufficient for accurate critic update at the k-th iteration.

Theorem 22 (Critic sample size) *Suppose Assumptions 1 and 4 hold. For $k \ge 0$, we let $\eta_k = \frac{1-\gamma_\rho}{C\gamma_\rho^{k+1}}$ and $\mathcal{W}_k = \mathcal{W}_{\text{Lip}}(\frac{C}{1-\gamma}, L_Q, \alpha, \epsilon_Q)$ with*

$$M = O(N^{\frac{d}{d+2\alpha}}),\ L = O(\log N + D + \log D),\ J = O(D),\ I \in [2, D],\ R_1 = O(1),$$
$$\log R_2 = O(\log^2 N + D \log N),\ \epsilon_Q = (L_Q^2 + C^2/(1-\gamma)^2) D^{\frac{3\alpha}{2\alpha+d}} N^{-\frac{\alpha}{2\alpha+d}}.$$

If we take sample size $N = \widetilde{O}(\frac{\sqrt{|\mathcal{A}|}}{1-\gamma}\gamma_\rho^{-(K+1)})^{\frac{d}{\alpha}+2}$, then $\mathbb{E}\big[\mathcal{L}_{\text{critic}}(w_k; \pi_k)\big] \le C^2\gamma_\rho^{2(k+1)}$ holds for all $k \le K$ in Algorithm 1.

Moreover, if Assumptions 2 and 3 hold, then for any $\epsilon > 0$, it suffices to let

$$N = \widetilde{O}\left(\frac{\kappa_\nu C\left(\sqrt{C_\nu} + \log|\mathcal{A}|\right)\sqrt{|\mathcal{A}|}}{(1-\gamma)^2\epsilon}\right)^{\frac{d}{\alpha}+2}$$

so that $\mathbb{E}\big[\mathcal{L}_{\text{critic}}(w_k; \pi_k)\big] \le C^2\gamma_\rho^{2(k+1)}$ for all $k \le K$, where K is the iteration number given in Theorem 8 that guarantees $\mathbb{E}\big[V^{\pi_k}(\rho) - V^\star(\rho)\big] \le \epsilon$. Here, $O(\cdot)$ and $\widetilde{O}(\cdot)$ hide the constant depending on $D^{\frac{6\alpha}{2\alpha+d}}$, $\overline{L}_Q$, $\log L_Q$, $\log\frac{C}{1-\gamma}$, d, α, ω, B, and the surface area $\text{Area}(\mathcal{S})$. In particular, N has a cubic dependence on D.

The proof of Theorem 22 is provided in Appendix D.1. As shown in Theorem 22, when the iteration number k increases, the required number of samples N grows exponentially at rate $\widetilde{O}(\gamma^{-\frac{d}{\alpha}-2})$. By Theorem 8, the total number of iterations is $\widetilde{O}(\log\frac{1}{\epsilon})$, so the growing procedure will not continue for too long. As a result, the number of samples for the last iteration is $\widetilde{O}(\epsilon^{-\frac{d}{\alpha}-2})$, and the overall sample complexity for critic updates is in the same order up to logarithm terms. Moreover, the exponent over ϵ is again the intrinsic dimension d instead of the data dimension D, which implies avoidance from the curse of dimensionality.

We now turn to derive a similar bound for actor updates. As shown in Theorem 19, with adequately chosen temperature parameters, we can restrict the actor network class with constant parameters for all iterations, and the resulting target actor will have the same approximate Lipschitzness guarantee as the subsequent actor network. Hence we have the following Theorem 23 characterizing the sufficient sample size for accurate actor updates.

Theorem 23 (Actor sample size) *Suppose Assumptions 1 and 4 hold. For $k \geq 0$, we let $\eta_k = \frac{1-\gamma_\rho}{C\gamma_\rho^{k+1}}$, $\lambda_k = \frac{C\gamma_\rho^k}{1-\gamma_\rho}$, $\mathcal{W}_k = \mathcal{W}_{\mathrm{Lip}}(\frac{C}{1-\gamma}, L_Q, \alpha, \epsilon_Q)$ and $\Theta_k = \Theta_{\mathrm{Lip}}(\frac{C}{(1-\gamma_\rho)(1-\gamma)}, \frac{L_Q}{1-\gamma_\rho}, \alpha, \frac{\epsilon_Q}{1-\gamma_\rho})$ with*

$$M = O(N^{\frac{d}{d+2\alpha}}),\ L = O(\log N + D + \log D),\ J = O(D),\ I \in [2, D],\ R_1 = O(1),$$
$$\log R_2 = O(\log^2 N + D \log N),\ \epsilon_Q = (L_Q^2 + C^2/(1-\gamma)^2) D^{\frac{3\alpha}{2\alpha+d}} N^{-\frac{\alpha}{2\alpha+d}},$$

If we take sample size $N = \widetilde{O}\Big(\frac{\sqrt{|\mathcal{A}|}\gamma_\rho^{-(K+1)}}{(1-\gamma_\rho)(1-\gamma)}\Big)^{\frac{d}{\alpha}+2}$, then $\mathbb{E}\big[\mathcal{L}_{\mathrm{actor}}(\theta_{k+1}; \theta_k, w_k)\big] \leq (1-\gamma_\rho)^2$ holds for all $k \leq K$ in Algorithm 1.

Moreover, if Assumptions 2 and 3, then for any $\epsilon > 0$, it suffices to let

$$N = \widetilde{O}\left(\frac{{\kappa_\nu}^2 C\left(\sqrt{C_\nu} + \log|\mathcal{A}|\right)\sqrt{|\mathcal{A}|}}{(1-\gamma)^2\epsilon}\right)^{\frac{d}{\alpha}+2}$$

so that $\mathbb{E}\big[\mathcal{L}_{\mathrm{actor}}(\theta_{k+1}; \theta_k, w_k)\big] \leq (1-\gamma_\rho)^2$ for all $k \leq K$, where K is the iteration number given in Theorem 8 that guarantees $\mathbb{E}\big[V^{\pi_K}(\rho) - V^\star(\rho)\big] \leq \epsilon$. Here, $O(\cdot)$ and $\widetilde{O}(\cdot)$ hide the constant depending on $D^{\frac{6\alpha}{2\alpha+d}}$, $\overline{L}_Q$, $\log L_Q$, $\log \frac{C}{1-\gamma}$, d, α, ω, B, and the surface area $\mathrm{Area}(\mathcal{S})$. *In particular, N has a cubic dependence on D.*

The proof of Theorem 23 is provided in Appendix D.2, which is similar to the proof of Theorem 22. Compared to Theorem 22, Theorem 23 requires a $\widetilde{O}({\kappa_\nu}^{\frac{d}{\alpha}+2})$ times larger sample size because the target actor in each iteration has a worse approximate Lipschitzness than the target actor. Nevertheless, we can align the sample size to the larger one for actor updates so that both the actor and the critic will be accurate. Combining the results together, we establish the overall sample complexity for Algorithm 1.

Theorem 24 (Total sample complexity) *Suppose Assumptions 1 to 4 hold. If for $k \geq 0$, $\eta_k = \frac{1-\gamma_\rho}{C\gamma_\rho^{k+1}}$, $\lambda_k = \frac{C\gamma_\rho^k}{1-\gamma_\rho}$, $\mathcal{W}_k = \mathcal{W}_{\mathrm{Lip}}(\frac{C}{1-\gamma}, L_Q, \alpha, \epsilon_Q)$ and $\Theta_k = \Theta_{\mathrm{Lip}}(\frac{C}{(1-\gamma_\rho)(1-\gamma)}, \frac{L_Q}{1-\gamma_\rho}, \alpha, \frac{\epsilon_Q}{1-\gamma_\rho})$ with*

$$M = O(N^{\frac{d}{d+2\alpha}}),\ L = O(\log N + D + \log D),\ J = O(D),\ I \in [2, D],\ R_1 = O(1),$$
$$\log R_2 = O(\log^2 N + D \log N),\ \epsilon_Q^{(k)} = (L_Q^2 + C^2/(1-\gamma)^2) D^{\frac{3\alpha}{2\alpha+d}} N^{-\frac{\alpha}{2\alpha+d}}.$$

Then for $\epsilon > 0$, it suffices to set $N = \widetilde{O}\Big(\frac{{\kappa_\nu}^2 C\left(\sqrt{C_\nu}+\log|\mathcal{A}|\right)\sqrt{|\mathcal{A}|}}{(1-\gamma)^2\epsilon}\Big)^{\frac{d}{\alpha}+2}$, and the expected number of sample oracle calls for Algorithm 1 to find a π_K satisfying $\mathbb{E}\big[V^{\pi_K}(\rho) - V^\star(\rho)\big] \leq \epsilon$ is

$$\widetilde{O}\Big({\kappa_\nu}^{\frac{2d}{\alpha}+5} C_\nu^{\frac{d}{2\alpha}+1} |\mathcal{A}|^{\frac{d}{2\alpha}+2} (1-\gamma)^{-\frac{2d}{\alpha}-5} C^{\frac{d}{\alpha}+2} \epsilon^{-\frac{d}{\alpha}-2}\Big).$$

Here, $O(\cdot)$ and $\widetilde{O}(\cdot)$ hide the constant depending on $D^{\frac{6\alpha}{2\alpha+d}}$, $\overline{L}_Q$, $\log L_Q$, $\log \frac{C}{1-\gamma}$, d, α, ω, B, and the surface area $\mathrm{Area}(\mathcal{S})$. *In particular, the sample complexity has a cubic dependence on D.*

Proof Let K be the iteration number in Theorem 8. By Theorems 22 and 23, our specification of N ensures that $\mathbb{E}\big[\mathcal{L}_{\mathrm{critic}}(w_k;\pi_k)\big] \le C^2\gamma_\rho^{2(k+1)}$ and $\mathbb{E}\big[\mathcal{L}_{\mathrm{actor}}(\theta_{k+1};\theta_k,w_k)\big] \le (1-\gamma_\rho)^2$ for all $k \le K$. Note that we have $|\mathcal{A}|$ actions in total, and by Theorem 25, each sample requires $O(\frac{1}{1-\gamma})$ oracle calls. As a result, the overall sample complexity is

$$\begin{aligned} O\Big(\frac{KN|\mathcal{A}|}{1-\gamma}\Big) &= \widetilde{O}\Big(\frac{1}{\log\frac{1}{\gamma_\rho}}\kappa_\nu{}^{\frac{2d}{\alpha}+4}C_\nu^{\frac{d}{2\alpha}+1}|\mathcal{A}|^{\frac{d}{2\alpha}+2}C^{\frac{d}{\alpha}+2}(1-\gamma)^{-\frac{2d}{\alpha}-5}\epsilon^{-\frac{d}{\alpha}-2}\Big) \\ &\le \widetilde{O}\Big(\kappa_\nu{}^{\frac{2d}{\alpha}+5}C_\nu^{\frac{d}{2\alpha}+1}|\mathcal{A}|^{\frac{d}{2\alpha}+2}C^{\frac{d}{\alpha}+2}(1-\gamma)^{-\frac{2d}{\alpha}-5}\epsilon^{-\frac{d}{\alpha}-2}\Big), \end{aligned}$$

where the inequality uses $\frac{1}{\log\frac{1}{\gamma_\rho}} \le \frac{1}{1-\gamma_\rho} = \kappa_\nu$. ■

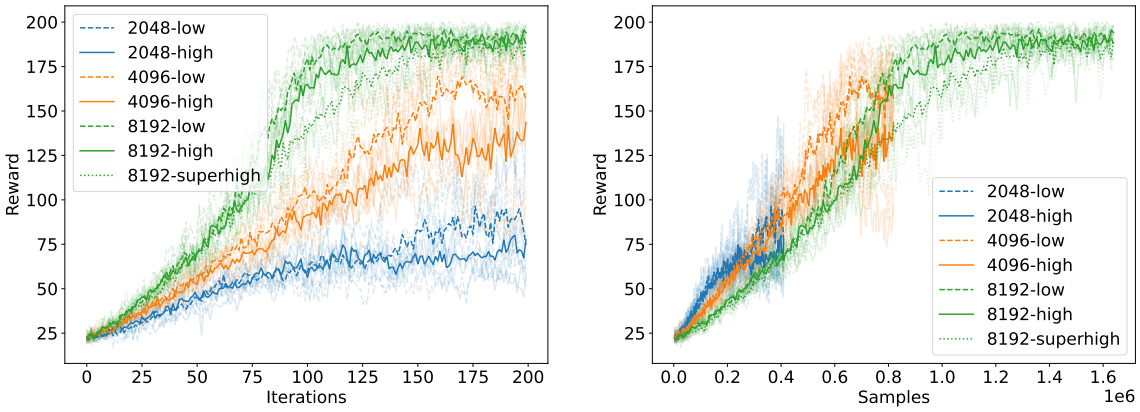
Theorem 24 characterizes the expected number of oracle access to the environment for finding an ϵ-optimal (in the sense of expected value function) policy π_K. The resulting sample complexity $\widetilde{O}\Big(\kappa_\nu{}^{\frac{2d}{\alpha}+5}C_\nu^{\frac{d}{2\alpha}+1}|\mathcal{A}|^{\frac{d}{2\alpha}+2}(1-\gamma)^{-\frac{2d}{\alpha}-5}C^{\frac{d}{\alpha}+2}\epsilon^{-\frac{d}{\alpha}-2}\Big)$ has no exponential dependence on the data dimension D. In our assumption, $d \ll D$, thus the sample complexity does not suffer from the curse of dimensionality.

5. Numerical Experiments

In this section, we present numerical experiments for NPMD to illustrate that its sample complexity does not necessarily grow exponentially with the ambient dimension D. We perform experiments on the CartPole environment (Barto et al., 1983) with visual display. The action space contains 2 discrete actions. The states are images of the cart and pole rendered from 4 internal factors indicating the status of the objects. Therefore, the intrinsic dimension $d = 4$, while the ambient dimension D scales with the image resolution. We consider three resolutions: low ($3\times20\times75$), high ($3\times40\times150$), and super high ($3\times60\times225$).

We set the discount factor $\gamma = 0.98$. We use 2048, 4096, and 8192 samples for low and high resolutions in each NPMD iteration. For further comparison, we use 8192 samples per iteration in the super high-resolution setting. For the CNN architecture, we set the pairs of kernel size and stride in the first layer as $(7,3)$, $(5,2)$, and $(3,1)$ for low, high, and super high resolutions respectively. The other CNN layers all share kernel size 3 and stride 1. We run 200 epochs of SGD to solve the ERM subproblems with batch size 256 and learning rate 0.001. To evaluate the obtained policy, we generate 32 independent trajectories using the policy and compute the average of their total rewards until termination or truncation after hitting the maximal reward limit of 200. We repeat the experiments 5 times with different random seeds, and the results are shown in Figure 2 and Table 2. More details about the environment and parameters are provided in Appendix H. The code is available at `https://github.com/zhenghaoxu-gatech/Neural-Policy-Mirror-Descent`.

The numerical results show that NPMD exhibits comparable performance across varying image resolutions, where the intrinsic structure of the CartPole environment remains the



(a) (b)

Figure 2: Evaluated rewards in different setups. The opaque lines are the average of 5 runs.

Sample size $N \cdot \lvert\mathcal{A}\rvert$	Metric	Resolution	Runs (low to high)					Range
8192	Max	Low	199.8	200.0	200.0	200.0	200.0	199.9 ± 0.1
		High	198.2	198.3	199.2	199.8	200.0	199.1 ± 0.9
		Super high	195.2	195.9	196.6	197.2	198.2	196.7 ± 1.5
	EMA	Low	186.1	190.0	190.4	193.1	193.4	189.8 ± 3.7
		High	185.2	186.2	190.6	193.7	194.2	189.7 ± 4.5
		Super high	181.1	181.7	186.4	186.8	190.4	185.8 ± 4.7
	Last	Low	178.0	181.3	187.7	193.0	198.1	188.1 ± 10.1
		High	188.7	191.1	194.4	197.6	198.6	193.7 ± 5.0
		Super high	175.7	187.7	190.5	194.0	197.5	186.6 ± 10.9

Table 2: Comparison of 5 runs in different resolutions with per iteration sample size 8192. The evaluation metrics include the maximal reward along the trajectory (Max), the exponential moving average of history rewards (EMA), and the last iterate reward (Last). The EMA is computed through $EMA_k = 0.9 * EMA_{k-1} + 0.1 * R_k$, where R_k is the reward at the k-th iteration.

same. While the ambient dimension D still affects the performance, its impact is not as much as an exponential function would do: Suppose the sample complexity depends exponentially on D, then moving from low to high or from high to super high resolution will increase D by factor 4, resulting in a quartic growth in sample complexity. However, the numerical results suggest that much fewer samples are required to get the same near-optimal performance (see Figure 2a and Table 2). Therefore, the sample complexity does not exhibit exponential dependence on D. We also observe that training becomes more stable as the per iteration sample size increases, and the average performance becomes better as the total sample size increases (see Figure 2b). These empirical results confirm our upper bound in Theorem 24, which does not depend exponentially on D.

6. Conclusion and Discussion

We have derived the overall sample complexity for NPMD (Algorithm 1) on Lipschitz MDP with intrinsically low-dimensional state space $\mathcal{S}$. Our result gives a concrete characterization of the expected number of samples required for an ϵ-optimal policy under mild regularity assumptions and shows no curse of dimensionality. We make a few remarks about this result.

Tightness in ϵ. The sample complexity $\widetilde{O}(\epsilon^{-\frac{d}{\alpha}-2})$ can be interpreted as two parts. The first part $\widetilde{O}(\epsilon^{-2})$ comes from iterations of the NPMD algorithm and is optimal up to logarithm terms. It matches the complexity of PMD in tabular case (Lan, 2023) and NPG on linear MDP (Yuan et al., 2023). The second part $\widetilde{O}(\epsilon^{-\frac{d}{\alpha}})$ comes from function approximation on d-dimensional state space manifold. Intuitively, it matches the number of states $|\mathcal{S}_{\text{dis}}|$ appeared in the complexity of tabular PMD if we discretize the continuous state space into a finite set of points $\mathcal{S}_{\text{dis}} \subset \mathcal{S}$.[6] This part scales with the intrinsic dimension d of $\mathcal{S}$ and can be interpreted as the result of neural networks adapting to the low-dimensional state space geometry. It is yet to be examined whether the overall complexity is tight in ϵ.

Dependence on the cost function scale. The sample complexity reasonably depends on $C^{-1}\epsilon$, which can be viewed as the relative error. Therefore, as long as ϵ scales with the cost function, the term $C^{\frac{d}{\alpha}+2}\epsilon^{-\frac{d}{\alpha}-2}$ in the complexity bound remains the same. Although the scaling of the cost function can still affect the hidden constant depending on $\log L_Q = \log\left(L_c + \frac{\gamma C}{1-\gamma} L_{\mathcal{P}}\right)$ and $\log \frac{C}{1-\gamma}$, it will not have a major impact since the dominating term is the normalized Lipschitz constant $\bar{L}_Q$, which is invariant to the scaling of the cost function.

Distribution mismatch and concentrability. The distribution mismatch coefficient κ_ν in Assumption 2 and concentrability coefficient C_ν in Assumption 3 have been widely used in the analysis of PMD-type methods in tabular setting (Xiao, 2022), linear function approximation (Agarwal et al., 2021; Alfano and Rebeschini, 2022; Yuan et al., 2023), and general function approximation (Alfano et al., 2023). The mismatch coefficient κ_ν occurs when the on-policy sampling distribution $\nu_\rho^{\pi_k}$ is different from the optimal visitation distribution $\nu_\rho^{\pi^\star}$. This mismatch coefficient seems unavoidable for analysis involving the performance difference lemma (Theorem 27) even in the tabular setting (Xiao, 2022) and can affect the iteration complexity through γ_ρ as shown in Theorem 8. Recently, Johnson et al. (2023) have established an alternative analysis that bypasses the performance difference lemma. Their result removes the mismatch coefficient for tabular PMD by using an adaptive step size. However, their sophisticated step size rule does not apply to our analysis, as it will make the smoothness of target actors uncontrollable.

The concentrability coefficient C_ν comes from the change of error measures. In the k-th iteration of NPMD, we cannot directly sample states from $\nu_\rho^{\pi_{k+1}}$ or $\nu_\rho^{\pi^\star}$ because the corresponding policies π_{k+1} and $\pi^\star$ are not available yet. Instead, we sample states from $\nu_\rho^{\pi_k}$ and solve the ERM subproblems with these samples. As a result, the actor and critic errors are naturally measured in $\nu_\rho^{\pi_k}$, and C_ν comes in when measuring the errors in $\nu_\rho^{\pi_{k+1}}$ and $\nu_\rho^{\pi^\star}$, which are related to the performance difference between consecutive policies. The concentrability coefficient is unavoidable when we have function approximation errors measured in the L^2 norm. To remove C_ν, one has to consider the exact PMD case where there is no

6. We note that directly performing discretization is difficult when $\mathcal{S}$ has a complicated geometric structure.

function approximation at all, or to derive L^∞ error bounds for critic and actor updates, both are intractable for continuous state space.

As suggested in Yuan et al. (2023) for finite state space, both coefficients can be potentially improved if we decouple the initial sampling distribution in Algorithm 1 and the evaluation distribution in the policy optimization problem (1).[7] Indeed, one can replace ρ in (1) by another distribution ρ', and replace the initial distribution in Algorithm 2 by ϱ. In this case, we can choose ρ' with full support so that Assumption 2 naturally holds and κ_ν becomes ${\kappa_\nu}' = \left\| \frac{\mathrm{d}\nu_{\rho'}^{\pi^\star}}{\mathrm{d}\nu_{\rho'}^{\pi_k}} \right\|_\infty \leq \frac{1}{1-\gamma} \left\| \frac{\mathrm{d}\nu_{\rho'}^{\pi^\star}}{\mathrm{d}\rho'} \right\|_\infty$. The optimality gap can be translated as

$$V^{\pi_k}(\rho) - V^\star(\rho) \leq \left\| \frac{\mathrm{d}\rho}{\mathrm{d}\rho'} \right\|_\infty (V^{\pi_k}(\rho') - V^\star(\rho')).$$

Similarly, Assumption 3 can be replaced by

$$\chi^2(\nu_{\rho'}^{\pi}, \nu_{\varrho}^{\pi_k}) + 1 \leq C_\nu'$$

for some C_ν'. However, we note that the pathological behavior of distribution mismatch and concentrability in the continuous state space is not always removable as it requires the Radon–Nikodym derivatives $\frac{\mathrm{d}\nu_{\rho'}^{\pi}}{\mathrm{d}\nu_{\varrho}^{\pi_k}}$ to exist, and it is difficult to get ${\kappa_\nu}' < \kappa_\nu$ and $C_\nu' < C_\nu$ for a better complexity. To the best of our knowledge, it remains an open problem to study function approximation policy gradient methods in continuous state space without Assumptions 2 and 3.

Dependence on the action space. The sample complexity depends on $|\mathcal{A}|^{\frac{d}{2\alpha}+2}$, which comes from two parts. One is from the aggregation of $|\mathcal{A}|$ networks, and the other part $|\mathcal{A}|^{\frac{d}{2\alpha}+1}$ is from function approximation, as $|\mathcal{A}|$ appears in the approximation error after critic/actor loss translation (see Appendices A.2 and A.3). We notice that some concurrent works do not have such dependence on $|\mathcal{A}|$, e.g. Yuan et al. (2023) and Alfano et al. (2023). Their results are based on the *state-action* concentrability condition (see e.g. Assumption 6 in Yuan et al. 2023), which requires bounded χ^2-divergence between state-action visitation distributions, rather than just state visitation distributions. Consequently, it is generally stronger than our Assumption 3.

Furthermore, these works directly make realizability assumptions on target functions and do not consider the approximation bias, which is one of the main focus of our work and is the cause of the polynomial dependence on the size of the action space. To avoid the $|\mathcal{A}|$ factor in error translation, it is necessary to consider fitting a single target function (either the value function or the target actor) defined over the product space $\mathcal{S} \times \mathcal{A}$, rather than fitting $|\mathcal{A}|$ functions on $\mathcal{S}$ separately. This requires the target function to be smooth in both states and actions. However, environments with discrete action space often exhibit non-smoothness in actions, since many of these actions have opposite functionalities (e.g. move left/right). Therefore, it is difficult to derive guarantees for function approximation even if we assume state-action concentrability. Nevertheless, we conjecture that the second part, $|\mathcal{A}|^{\frac{d}{2\alpha}+1}$, might be removable, and we leave it for future investigation.

7. The initial distribution in sampling is restricted by the environment in our setting (non-generative model), whereas the evaluation distribution can be arbitrarily chosen.

Most of our derivation can generalize to continuous state space with some modifications and additional assumptions. However, drawing actions from policy $\pi_k(\cdot|s) \propto e^{f_\theta(s,\cdot)}$ becomes a non-log-concave sampling problem on the action manifold $\mathcal{A}$, which significantly complicates the analysis considering that on-policy sampling (Algorithm 2) requires action $a_t \sim \pi_k(\cdot|s_t)$ at every step, and the inexactness of sampling at each step will accumulate, causing another layer of distribution shift.

Computational concerns. In each iteration of Algorithm 1, there are two ERM subproblems with approximately Lipschitz constraints, which are assumed to be solvable. However, solving such problems is non-trivial due to the non-convex objectives and the approximately Lipschitz constraints. In practice, one can use gradient descent (GD) and its variants to minimize the objectives, but little is known about their theoretical convergence behavior. Recently there has been some work studying the GD dynamics in the NTK or mean-field regime (Jacot et al., 2018; Song et al., 2018), but the gap remains as they cannot fully explain the convergence behavior of GD-like methods on deep models, and their results cannot adapt to the low-dimensional manifold structure.

Meanwhile, there is no existing result on optimization with approximate Lipschitzness constraints. One can apply Lipschitz regularization methods, such as spectral regularization (Yoshida and Miyato, 2017; Gogianu et al., 2021), gradient regularization (Gulrajani et al., 2017), projected gradient descent for Lipschitz constant constraint (Gouk et al., 2021), and adversarial training (Miyato et al., 2018). Most of these techniques are heuristic, so they cannot exactly control the Lipschitzness of networks. Nevertheless, they could result in approximately Lipschitz networks as it is a relaxed condition. In addition, recent studies have discovered that GD itself has some algorithmic regularization effect, which implicitly controls the Lipschitzness of the learned networks (Mulayoff et al., 2021). We leave the study of approximately Lipschitz-constrained optimization of neural networks for future work.

Overparameterization. In modern deep learning practice, there exists a propensity to employ overparameterized models that have more parameters than the number of data. Our current analysis is based on the classical bias-variance trade-off argument and cannot handle the overparameterization case. Recently, Zhang and Wang (2022) have established deep non-parametric regression results that apply to overparameterized models, but their work does not exploit the low-dimensional structure. It is an interesting future direction to examine if their work can be extended to the manifold setting and fit into our analysis. We expect it to close the theory-practice gap in DRL further.

Comparison with value-based methods. The sample complexity result for NPMD matches the bound for the value-based FQI method in Fan et al. (2020) when $d = D$, while our result is significantly better when the intrinsic dimension $d \ll D$. This shows that policy-based methods can achieve as good performance as value-based methods in theory. From a technical perspective, value-based methods only approximate smooth value functions (under the Bellman closedness assumption in Fan et al. 2020; Nguyen-Tang et al. 2022). On the other hand, policy-based methods require repetitively approximating new policies, whose Lipschitz constant will accumulate. We address the issue by introducing the notion of approximate Lipschitzness, imposing approximately Lipschcitz constraints on the neural networks, and establishing approximation theory for them. Our analysis framework can be applied to more general scenarios where there are iterative refittings of neural networks.

Beyond Lipschitz MDP. In this paper, we work on Lipschitz MDP (Assumption 4). In practice, the MDP can be either smoother or not as smooth as Lipschitz MDP. For the former case, one can consider Hölder smooth MDP with higher exponent, namely $\alpha > 1$, and expect there is a better sample complexity. If we only consider the policy evaluation and value-based algorithms, where the target value function is smooth, then this is possible as suggested by the results from deep supervised learning (Chen et al., 2022). However, for policy-based methods, it is unclear whether neural networks that uniformly approximate Hölder functions can have smoothness beyond approximate Lipschitzness, given that networks with ReLU activation are not differentiable. It is a future direction to examine the sample complexity of policy-based methods in smoother MDPs. For the latter case, one can consider extending the Lipschitz condition to the more general Sobolev or Besov conditions to deal with spatial inhomogeneity in smoothness. Also, as mentioned in Theorems 14 and 15, one can use a smooth approximation of the non-smooth MDP as a surrogate in this case.

Acknowledgments

The authors thank Yan Li for the discussion in the early stage of this work.

Appendix A. Algorithms

This section presents the missing sampling algorithm (Algorithm 2) for Algorithm 1, along with some auxiliary results related to Algorithms 1 and 2.

Algorithm 2: Sample $s \sim \nu_\rho^\pi$ or $(s,a) \sim \bar{\nu}_\rho^\pi$

Input: Distribution ρ, policy π, factor $\gamma \in (0,1)$
Initialize `flag` $=$ `True`, $t = 0$, $s_0 \sim \rho$, $a_0 \sim \pi(\cdot|s_0)$;
while `flag` *is* `True` **do**
  Sample $p \sim \mathrm{Unif}([0,1])$;
  if $p \leq \gamma$ **then**
    Sample $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$;
    Sample $a_{t+1} \sim \pi(\cdot|s_{t+1})$;
    $t \leftarrow t + 1$;
  else
    `flag` $=$ `False`;
  end
end
Output: s_t as s or (s_t, a_t) as (s,a)

A.1 Sample Complexity of Algorithm 2

We compute the expected number of sample oracle calls of Algorithm 2 to get $s \sim \nu_\rho^\pi$ or $(s,a) \sim \bar{\nu}_\rho^\pi$.

Lemma 25 *Algorithm 2 returns $(s,a) \sim \bar{\nu}_\rho^\pi$, and the expected number of sample oracle calls for each pair of (s,a) is $\frac{1}{1-\gamma}$.*

Proof Let T be the terminating time (trajectory length) of Algorithm 2, which has probability

$$\mathbb{P}(T=t) = \gamma^t(1-\gamma).$$

The probability distribution of the output s_t is then given by

$$\mathcal{P}_{\text{out}} = \sum_{t=0}^{\infty} \mathcal{P}_t^\pi \cdot \mathbb{P}(T=t) = \sum_{t=0}^{\infty} \mathcal{P}_t^\pi \cdot \gamma^t(1-\gamma),$$

which is exactly ν_ρ^π according to (6). Since $\mathbb{P}(a_t = a|s_t) = \pi(a|s_t)$, we have $(s,a) \sim \bar{\nu}_\rho^\pi$. The expected number of sample oracle calls is $\mathbb{E}[T+1]$, which is

$$\mathbb{E}[T+1] = \sum_{t=0}^{\infty}(t+1)\gamma^t(1-\gamma) = \frac{1}{1-\gamma}.$$

The expected trajectory length $\frac{1}{1-\gamma}$ is also called the *effective horizon.* ■

A.2 Critic Loss Translation

The critic loss can be translated to the mean squared error (MSE) of a regression problem. To see this, recall the state-action value function of the form (10):

$$Q^\pi(s,a) = c(s,a) + \frac{\gamma}{1-\gamma}\mathbb{E}_{(s',a')\sim\bar{\nu}^\pi_{\mathcal{P}(\cdot|s,a)}}\big[c(s',a')\big].$$

For any fixed $a \in \mathcal{A}$ and $k \geq 0$, let $X \sim \nu_\rho^{\pi_k}$ be a random variable on $\mathcal{S}$ and $Z = (S,A)$ be another random variable with conditional distribution $p_{Z|X} = \bar{\nu}^{\pi_k}_{\mathcal{P}(\cdot|X,a)}$. With this notation, reformulating (10) gives

$$c(X,a) + \frac{\gamma}{1-\gamma}c(S,A) = Q^{\pi_k}(X,a) + \zeta, \tag{28}$$

where

$$\zeta = \frac{\gamma}{1-\gamma}\Big(c(S,A) - \mathbb{E}_{(s',a')\sim\bar{\nu}^{\pi_k}_{\mathcal{P}(\cdot|X,a)}}\big[c(s',a')\big]\Big) \tag{29}$$

is a random variable. We can verify several properties of the noise term ζ.

Lemma 26 *The noise term ζ defined in* (29) *is a zero-mean sub-Gaussian random variable with variance proxy $\sigma^2 = \frac{\gamma^2 C^2}{4(1-\gamma)^2}$ and is uncorrelated with X.*

Proof By our construction, the randomness of ζ comes from the randomness of X and $Z = (S,A)$. When $X = x$ is fixed, $\bar{\nu}^{\pi_k}_{\mathcal{P}(\cdot|x,a)}$ is a fixed distribution, so we have

$$\begin{aligned}
\mathbb{E}[\zeta \mid X = x] &= \frac{\gamma}{1-\gamma}\mathbb{E}_{Z\sim p_{Z|X}}\Big[c(S,A) - \mathbb{E}_{(s',a')\sim\bar{\nu}^{\pi_k}_{\mathcal{P}(\cdot|x,a)}}[c(s',a')] \mid X = x\Big] \\
&= \frac{\gamma}{1-\gamma}\Big(\mathbb{E}_{(S,A)\sim\bar{\nu}^{\pi_k}_{\mathcal{P}(\cdot|x,a)}}[c(S,A)] - \mathbb{E}_{(s',a')\sim\bar{\nu}^{\pi_k}_{\mathcal{P}(\cdot|x,a)}}[c(s',a')]\Big) \\
&= 0.
\end{aligned}$$

Therefore, $\mathbb{E}[\zeta \mid X] = 0$, $\mathbb{E}[\zeta] = \mathbb{E}_X[\mathbb{E}[\zeta \mid X]] = \mathbb{E}_X[0] = 0$, ζ is uncorrelated with X.

By (2) we know that given any realization of X,

$$\frac{-\gamma\mu}{1-\gamma} \leq \zeta \leq \frac{\gamma(C-\mu)}{1-\gamma},$$

where $\mu = \mathbb{E}_{Z|X}\big[c(S,A)\big]$, thus ζ is sub-Gaussian with variance proxy $\sigma^2 = \frac{\gamma^2 C^2}{4(1-\gamma)^2}$. ■

Note that the target values $\{c(s_{a,i},a) + \frac{\gamma}{1-\gamma}c(s'_{a,i},a'_{a,i})\}_{i=1}^N$ in the empirical risk (15) are i.i.d. copies of the left-hand side of (28), and the right-hand side of (28) is a function plus a noise term. Therefore, the ERM subproblem (16) can be viewed as $|\mathcal{A}|$ independent regression problems, each corresponding to an action $a \in \mathcal{A}$. It follows immediately from the definition of the critic loss (14) that

$$\begin{aligned}
\mathbb{E}\left[\mathcal{L}_{\text{critic}}(w_k;\pi_k)\right] &= \mathbb{E}\left[\mathbb{E}_{s\sim\nu_\rho^{\pi_k}}\|Q_{w_k}(s,a) - Q^{\pi_k}(s,a)\|_2^2\right] \\
&= \mathbb{E}\left[\mathbb{E}_{s\sim\nu_\rho^{\pi_k}}\sum_{a\in\mathcal{A}}|Q_{w_k}(s,a) - Q^{\pi_k}(s,a)|^2\right] \\
&\leq |\mathcal{A}|\max_{a\in\mathcal{A}}\mathbb{E}\left[\mathbb{E}_X\left[|Q_{w_k}(X,a) - Q^{\pi_k}(X,a)|^2\right]\right], \qquad (30)
\end{aligned}$$

where the outer expectation is taken with respect to the estimated w_k, which depends on the samples used for ERM. As a result, the critic loss of Q_{w_k} can be upper bounded as long as we can bound the MSE of the regression problem for every $a \in \mathcal{A}$. We provide statistical results for the regression problem in Appendix G.

A.3 Actor Loss Translation

The actor loss can also be translated. We already know the exact solution to (17) is given by $g^\star_{k+1}$ defined in Theorem 3. The actor loss is thus translated into the error between the estimated function $f_{\theta_{k+1}}$ and the ground truth function $\lambda_{k+1}g^\star_{k+1}$:

$$\begin{aligned}
&\mathbb{E}\left[\mathcal{L}_{\text{actor}}(\theta_{k+1};\theta_k,w_k)\right] \\
=&\mathbb{E}\left[\mathbb{E}_{s\sim\nu_\rho^{\pi_k}}\left\|\lambda_{k+1}^{-1}f_{\theta_{k+1}}(s,\cdot) - \lambda_k^{-1}f_\theta(s,\cdot) + \eta_k Q_{w_k}(s,\cdot)\right\|_2^2\right] \\
=&\mathbb{E}\left[\mathbb{E}_{s\sim\nu_\rho^{\pi_k}}\sum_{a\in\mathcal{A}}\left|\lambda_{k+1}^{-1}f_{\theta_{k+1}}(s,a) - \lambda_k^{-1}f_\theta(s,a) + \eta_k Q_{w_k}(s,a)\right|^2\right] \\
\leq&\frac{|\mathcal{A}|}{\lambda_{k+1}^2}\max_{a\in\mathcal{A}}\mathbb{E}\left[\mathbb{E}_{s\sim\nu_\rho^{\pi_k}}\left|f_{\theta_{k+1}}(s,a) - \frac{\lambda_{k+1}}{\lambda_k}f_\theta(s,a) + \lambda_{k+1}\eta_k Q_{w_k}(s,a)\right|^2\right], \qquad (31)
\end{aligned}$$

where the outer expectation is taken with respect to θ_{k+1}, which depends on the samples used for empirical risk minimization. Once we derive an upper bound for the MSE of $f_{\theta_{k+1}}$, an upper bound for the actor update loss follows immediately.

Appendix B. Proofs in Section 3

B.1 Proof of Lemma 3

Proof For any $s \in \mathcal{S}$, (17) is a convex problem. Its Karush-Kuhn-Tucker condition yields

$$Q_{w_k}(s,\cdot) - \frac{1}{\eta_k}\log \pi_k(s,\cdot) + \frac{1}{\eta_k}\log \pi^\star_{k+1}(s,\cdot) + \mu_s^\star \mathbf{1} = 0$$

for some $\mu_s^\star \in \mathbb{R}$, where we denote $\pi^\star_{k+1}$ as the solution. This means for any $s \in \mathcal{S}$,

$$\pi^\star_{k+1}(a|s) \propto \exp(\log \pi_k(a|s) - \eta_k Q_{w_k}(s,a)).$$

By definition (18), we have

$$\log \pi_k(a|s) = \lambda_k^{-1} f_{\theta_k}(s,a) - \log \sum_{a' \in \mathcal{A}} \exp\big(\lambda_k^{-1} f_{\theta_k}(s,a')\big).$$

Since $\pi^\star_{k+1}(\cdot|s)$ is shift-invariant with $g^\star_{k+1}(s,\cdot)$, that is,

$$\pi^\star_{k+1}(a|s) = \frac{\exp\big(g^\star_{k+1}(s,a)\big)\exp(p(s))}{\sum_{a' \in \mathcal{A}} \exp\big(g^\star_{k+1}(s,a')\big)\exp(p(s))} = \frac{\exp\big(g^\star_{k+1}(s,a) + p(s)\big)}{\sum_{a' \in \mathcal{A}} \exp\big(g^\star_{k+1}(s,a') + p(s)\big)}$$

for any $p(s)$ independent of a, choosing $g^\star_{k+1} = \lambda_k^{-1} f_{\theta_k} - \eta_k Q_{w_k}$ suffices. ■

Appendix C. Proofs for Iteration Complexity

In this section, we present the missing proofs in Section 4.1 and auxiliary lemmas for them.

C.1 Proof of Lemma 6

We first present the missing proof for Theorem 6.

Proof By Assumption 1, $\mathcal{S}$ is compact, thus for any policy π' the following inequality holds:

$$\vartheta \coloneqq \sup_{s \in \mathcal{S}} \left\| \frac{\pi'(\cdot|s)}{\pi(\cdot|s)} \right\|_\infty < \infty,$$

where $\left\| \frac{\pi'(\cdot|s)}{\pi(\cdot|s)} \right\|_\infty$ exists given that $\pi(\cdot|s)$ has full support on $\mathcal{A}$. By (5), we have

$$\begin{aligned}
\mathcal{P}_1^\pi &= \mathbb{E}_{s\sim\rho} \sum_{a \in \mathcal{A}} \pi(a|s)\mathcal{P}(\cdot|s,a) \\
&\geq \mathbb{E}_{s\sim\rho} \sum_{a \in \mathcal{A}} \mathbb{1}[\pi'(a|s) > 0]\pi'(a|s)\frac{\pi(a|s)}{\pi'(a|s)}\mathcal{P}(\cdot|s,a) \\
&\geq \vartheta^{-1}\mathbb{E}_{s\sim\rho} \sum_{a \in \mathcal{A}} \pi'(a|s)\mathcal{P}(\cdot|s,a) \\
&= \vartheta^{-1}\mathcal{P}_1^{\pi'}.
\end{aligned}$$

It follows immediately from induction that

$$\begin{aligned}\mathcal{P}_{t+1}^{\pi} &= \mathbb{E}_{s\sim\mathcal{P}_t^{\pi}} \sum_{a\in\mathcal{A}} \pi(a|s)\mathcal{P}(\cdot|s,a) \\ &\geq \mathbb{E}_{s\sim\mathcal{P}_t^{\pi'}} \frac{\mathrm{d}\mathcal{P}_t^{\pi}(s)}{\mathrm{d}\mathcal{P}_t^{\pi'}(s)} \sum_{a\in\mathcal{A}} \vartheta^{-1}\pi'(a|s)\mathcal{P}(\cdot|s,a) \\ &\geq \vartheta^{-(t+1)}\mathcal{P}_{t+1}^{\pi'}.\end{aligned}$$

Therefore, by definition of the visitation distribution (6) we have $\nu_\rho^{\pi} \gg \nu_\rho^{\pi'}$. ■

C.2 Supporting Lemmas for Iteration Complexity

We recall the performance difference lemma for the value function, which measures the difference between value functions under different policies.

Lemma 27 (Performance difference lemma) *For any pair of policies π and π' and any state s, we have*

$$V^{\pi'}(\rho) - V^{\pi}(\rho) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim\nu_\rho^{\pi'}}\left[\left\langle Q^{\pi}(s,\cdot), \pi'(\cdot|s) - \pi(\cdot|s)\right\rangle\right].$$

Proof The performance difference between π' and π is given by

$$\begin{aligned}&V^{\pi'}(\rho) - V^{\pi}(\rho) \\ =&V^{\pi'}(\mathcal{P}_0^{\pi'}) - V^{\pi}(\mathcal{P}_0^{\pi'}) \\ =&\mathbb{E}_{s\sim\mathcal{P}_0^{\pi'}} \sum_{a\in\mathcal{A}} \left(c(s,a)\left(\pi'(a|s) - \pi(a|s)\right) + \gamma \int_{\mathcal{S}} \left(V^{\pi'}(s')\pi'(a|s) - V^{\pi}(s')\pi(a|s)\right) \mathrm{d}\mathcal{P}(s'|s,a)\right) \\ =&\mathbb{E}_{s\sim\mathcal{P}_0^{\pi'}} \sum_{a\in\mathcal{A}} \left(c(s,a) + \gamma \int_{\mathcal{S}} V^{\pi}(s')\,\mathrm{d}\mathcal{P}(s'|s,a)\right)\left(\pi'(a|s) - \pi(a|s)\right) \\ &\quad + \gamma\mathbb{E}_{s\sim\mathcal{P}_0^{\pi'}} \sum_{a\in\mathcal{A}} \pi'(a|s) \int_{\mathcal{S}} \left(V^{\pi'}(s') - V^{\pi}(s')\right) \mathrm{d}\mathcal{P}(s'|s,a) \\ =&\mathbb{E}_{s\sim\mathcal{P}_0^{\pi'}} \left\langle Q^{\pi}(s,\cdot), \pi'(\cdot|s) - \pi(\cdot|s)\right\rangle + \gamma\mathbb{E}_{s\sim\mathcal{P}_0^{\pi}, a\sim\pi'(\cdot|s)} \int_{\mathcal{S}} \left(V^{\pi'}(s') - V^{\pi}(s')\right) \mathrm{d}\mathcal{P}(s'|s,a) \\ =&\mathbb{E}_{s\sim\mathcal{P}_0^{\pi'}} \left\langle Q^{\pi}(s,\cdot), \pi'(\cdot|s) - \pi(\cdot|s)\right\rangle + \gamma\left(V^{\pi'}(\mathcal{P}_1^{\pi'}) - V^{\pi}(\mathcal{P}_1^{\pi'})\right) \\ =&\sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{s\sim\mathcal{P}_t^{\pi'}} \left\langle Q^{\pi}(s,\cdot), \pi'(\cdot|s) - \pi(\cdot|s)\right\rangle.\end{aligned}$$

The first equality uses $\mathcal{P}_0^{\pi'} \equiv \rho$. The second equality is from (3) and (4). The fourth equality is from the opposite direction of (4). The last two lines are from the recursion (5) and the sequence converges because the value functions are bounded by (2). Plugging in the definition of visitation distribution completes the proof. ■

Our NPMD algorithm uses KL divergence and softmax neural policies. In this case, we have the following lemma that can replace the approximated state-action value function Q_{w_k} that appeared in the inner product by the difference of log-policies.

Lemma 28 *For any pair of policies π and π' and any state s, we have*

$$\begin{aligned}\left\langle Q_{w_k}(s,\cdot), \pi(\cdot|s) - \pi'(\cdot|s)\right\rangle &= -\frac{1}{\eta_k}\left\langle \log \pi^\star_{k+1}(s,\cdot) - \log \pi_k(s,\cdot), \pi(\cdot|s) - \pi'(\cdot|s)\right\rangle \\ &= -\frac{1}{\eta_k}\left\langle \nabla h^{\pi^\star_{k+1}}(s,\cdot) - \nabla h^{\pi_k}(s,\cdot), \pi(\cdot|s) - \pi'(\cdot|s)\right\rangle,\end{aligned}$$

where $h^\pi(s) \coloneqq \langle \log \pi(s,\cdot), \pi(\cdot|s)\rangle$ is the negative entropy of π at s and $\nabla h^\pi(s,\cdot) \in \mathbb{R}^{|\mathcal{A}|}$ is the gradient taken with respect to the actions, $\pi^\star_{k+1}$ is the exact solution of (17).

Proof For any policies π and π' and any state s we have $\langle \mathbf{1}, \pi(\cdot|s) - \pi'(\cdot|s)\rangle = 0$. By using Theorem 3 and the definition of the KL divergence, we obtain the result. ■

The next lemma bounds the expected error (measured in $\nu_\rho^{\pi_{k+1}}$ or $\nu_\rho^{\pi^\star}$) of replacing Q^{π_k} with Q_{w_k} by the critic loss.

Lemma 29 *Suppose Assumptions 2 and 3 hold. If π is π_{k+1} or $\pi^\star$, then for any pair of policies π' and π'', we have*

$$\left|\mathbb{E}_{s\sim\nu_\rho^\pi}\left[\left\langle Q^{\pi_k}(s,\cdot) - Q_{w_k}(s,\cdot), \pi'(\cdot|s) - \pi''(\cdot|s)\right\rangle\right]\right| \le 2\sqrt{C_\nu \mathcal{L}_{\text{critic}}(w_k;\pi_k)}.$$

Proof We have

$$\begin{aligned}&\left|\mathbb{E}_{s\sim\nu_\rho^\pi}\left[\left\langle Q^{\pi_k}(s,\cdot) - Q_{w_k}(s,\cdot), \pi'(\cdot|s) - \pi''(\cdot|s)\right\rangle\right]\right| \\ \le&\mathbb{E}_{s\sim\nu_\rho^\pi}\left|\left\langle Q^{\pi_k}(s,\cdot) - Q_{w_k}(s,\cdot), \pi'(\cdot|s) - \pi''(\cdot|s)\right\rangle\right| \\ \le&\mathbb{E}_{s\sim\nu_\rho^\pi}\left\|Q^{\pi_k}(s,\cdot) - Q_{w_k}(s,\cdot)\right\|_\infty \left\|\pi'(\cdot|s) - \pi''(\cdot|s)\right\|_1 \\ \le&2\mathbb{E}_{s\sim\nu_\rho^\pi}\left\|Q^{\pi_k}(s,\cdot) - Q_{w_k}(s,\cdot)\right\|_\infty,\end{aligned}$$

where the third line is by Hölder's inequality. By Theorem 6, $\frac{\mathrm{d}\nu_\rho^\pi(s)}{\mathrm{d}\nu_\rho^{\pi_k}(s)}$ exists. By using Cauchy–Schwarz inequality and Assumption 3, we can replace the error measure ν_ρ^π by $\nu_\rho^{\pi_k}$:

$$\begin{aligned}\mathbb{E}_{s\sim\nu_\rho^\pi}\left\|Q^{\pi_k}(s,\cdot) - Q_{w_k}(s,\cdot)\right\|_\infty =&\mathbb{E}_{s\sim\nu_\rho^{\pi_k}}\frac{\mathrm{d}\nu_\rho^\pi(s)}{\mathrm{d}\nu_\rho^{\pi_k}(s)}\left\|Q^{\pi_k}(s,\cdot) - Q_{w_k}(s,\cdot)\right\|_\infty \\ \le&\sqrt{\mathbb{E}_{s\sim\nu_\rho^{\pi_k}}\left[\frac{\mathrm{d}\nu_\rho^\pi(s)}{\mathrm{d}\nu_\rho^{\pi_k}(s)}\right]^2 \mathbb{E}_{s\sim\nu_\rho^{\pi_k}}\left[\left\|Q^{\pi_k}(s,\cdot) - Q_{w_k}(s,\cdot)\right\|_\infty^2\right]} \\ =&\sqrt{\left(\chi^2(\nu_\rho^\pi,\nu_\rho^{\pi_k})+1\right)\mathbb{E}_{s\sim\nu_\rho^{\pi_k}}\left\|Q^{\pi_k}(s,\cdot) - Q_{w_k}(s,\cdot)\right\|_\infty^2} \\ \le&\sqrt{C_\nu \mathcal{L}_{\text{critic}}(w_k;\pi_k)}.\end{aligned}$$

Plugging this back, we obtain the result. ■

Theorem 29 relates the expected error on the left-hand side to the critic loss (14), which measures the difference between Q^{π_k} and Q_{w_k} in $L^2(\nu_\rho^{\pi_k})$ norm on $\mathcal{S}$ and L^∞ norm on $\mathcal{A}$. This differs from Lan (2023) where the error is completely measured in L^∞ norm on $\mathcal{S}\times\mathcal{A}$. We note that for continuous state space $\mathcal{S}$, deriving L^∞ error bound is difficult as we cannot get samples from every state, thus the analysis in Lan (2023) does not directly apply.

The distribution change is measured in χ^2 divergence instead of the absolute density ratio, yielding a tighter result. Similar techniques appear in Yuan et al. (2023) and Alfano et al. (2023), where they are applied to the joint space $\mathcal{S}\times\mathcal{A}$. As a result, their concentrability assumptions involve state-action visitation rather than state visitation distributions as in Assumption 3, and their errors are measured in $L^2(\bar{\nu})$ norm where $\bar{\nu}$ is some distribution over $\mathcal{S}\times\mathcal{A}$. We note that when considering function approximation, $L^2(\bar{\nu})$ norm reduces to $L^2(\nu_\rho^{\pi_k})$. Indeed, since $\mathcal{A}$ is finite (discrete hence non-smooth), fitting a function over $\mathcal{S}\times\mathcal{A}$ reduces to fitting $|\mathcal{A}|$ functions on $\mathcal{S}$, for which $L^2(\nu_\rho^{\pi_k})$ error becomes a natural measure. This reduction would come with additional factors that complicate the analysis.

Analogous to Theorem 29, the next lemma bounds the expected error (measured in $\nu_\rho^{\pi_{k+1}}$ or $\nu_\rho^{\pi^\star}$) of replacing $\pi_{k+1}^\star$ with π_{k+1} by the actor loss.

Lemma 30 *Suppose Assumptions 2 and 3 hold. If π is π_{k+1} or $\pi^\star$, then for any pair of policies π' and π'', we have*

$$\left|\mathbb{E}_{s\sim\nu_\rho^\pi}\left[\left(D^{\pi'}_{\pi^\star_{k+1}}(s)-D^{\pi''}_{\pi^\star_{k+1}}(s)\right)-\left(D^{\pi'}_{\pi_{k+1}}(s)-D^{\pi''}_{\pi_{k+1}}(s)\right)\right]\right|\le 2\sqrt{C_\nu\mathcal{L}_{\text{actor}}(\theta_{k+1};\theta_k,w_k)},$$

where $D^\pi_{\pi'}(s)\coloneqq\langle\log\pi(s,\cdot)-\log\pi'(s,\cdot),\pi(\cdot|s)\rangle$ is the KL divergence between π and π' at s.

Proof By Theorem 3 and the definition of the KL divergence, we have

$$\begin{aligned}&\left(D^{\pi'}_{\pi^\star_{k+1}}(s)-D^{\pi''}_{\pi^\star_{k+1}}(s)\right)-\left(D^{\pi'}_{\pi_{k+1}}(s)-D^{\pi''}_{\pi_{k+1}}(s)\right)\\&=\left\langle\log\pi_{k+1}(s,\cdot)-\log\pi^\star_{k+1}(s,\cdot),\pi'(\cdot|s)-\pi''(\cdot|s)\right\rangle\\&=\left\langle\lambda_{k+1}^{-1}f_{\theta_{k+1}}(s,\cdot)-\lambda_k^{-1}f_{\theta_k}(s,\cdot)+\eta_kQ_{w_k}(s,\cdot),\pi'(\cdot|s)-\pi''(\cdot|s)\right\rangle.\end{aligned}$$

Similar to the proof of Theorem 29, by Hölder's inequality, we have

$$\begin{aligned}&\left|\mathbb{E}_{s\sim\nu_\rho^\pi}\left[\left(D^{\pi'}_{\pi^\star_{k+1}}(s)-D^{\pi''}_{\pi^\star_{k+1}}(s)\right)-\left(D^{\pi'}_{\pi_{k+1}}(s)-D^{\pi''}_{\pi_{k+1}}(s)\right)\right]\right|\\\le&\mathbb{E}_{s\sim\nu_\rho^\pi}\left|\left\langle\lambda_{k+1}^{-1}f_{\theta_{k+1}}(s,\cdot)-\lambda_k^{-1}f_{\theta_k}(s,\cdot)+\eta_kQ_{w_k}(s,\cdot),\pi'(\cdot|s)-\pi''(\cdot|s)\right\rangle\right|\\\le&2\mathbb{E}_{s\sim\nu_\rho^\pi}\left\|\lambda_{k+1}^{-1}f_{\theta_{k+1}}(s,\cdot)-\lambda_k^{-1}f_{\theta_k}(s,\cdot)+\eta_kQ_{w_k}(s,\cdot)\right\|_\infty.\end{aligned}$$

By Theorem 6 and Assumption 3 and the Cauchy–Schwarz inequality we have

$$\begin{aligned}&\mathbb{E}_{s\sim\nu_\rho^\pi}\left\|\lambda_{k+1}^{-1}f_{\theta_{k+1}}(s,\cdot)-\lambda_k^{-1}f_{\theta_k}(s,\cdot)+\eta_kQ_{w_k}(s,\cdot)\right\|_\infty\\\le&\sqrt{\left(\chi^2(\nu_\rho^\pi,\nu_\rho^{\pi_k})+1\right)\mathbb{E}_{s\sim\nu_\rho^{\pi_k}}\left\|\lambda_{k+1}^{-1}f_{\theta_{k+1}}(s,\cdot)-\lambda_k^{-1}f_{\theta_k}(s,\cdot)+\eta_kQ_{w_k}(s,\cdot)\right\|_\infty^2}\\\le&\sqrt{C_\nu\mathcal{L}_{\text{actor}}(\theta_{k+1};\theta_k,w_k)},\end{aligned}$$

and the result follows immediately. ■

Using the above auxiliary lemmas, we can now prove Theorem 7

C.3 Proof of Lemma 7

Proof By Theorem 27 we have the following relations:

$$V^{\pi_k}(\rho) - V^\star(\rho) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\left[\langle Q^{\pi_k}(s,\cdot), \pi_k(\cdot|s) - \pi^\star(\cdot|s)\rangle\right], \tag{32}$$

$$V^{\pi_{k+1}}(\rho) - V^{\pi_k}(\rho) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim\nu_\rho^{\pi_{k+1}}}\left[\langle Q^{\pi_k}(s,\cdot), \pi_{k+1}(\cdot|s) - \pi_k(\cdot|s)\rangle\right]. \tag{33}$$

Applying Theorem 29 to (32) and (33) gives

$$\begin{aligned} V^{\pi_k}(\rho) - V^\star(\rho) \le \frac{1}{1-\gamma}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}&\left[\langle Q_{w_k}(s,\cdot), \pi_k(\cdot|s) - \pi^\star(\cdot|s)\rangle\right] \\ &+\frac{2}{1-\gamma}\sqrt{C_\nu \mathcal{L}_{\text{critic}}(w_k;\pi_k)}, \end{aligned} \tag{34}$$

$$\begin{aligned} V^{\pi_{k+1}}(\rho) - V^{\pi_k}(\rho) \le \frac{1}{1-\gamma}\mathbb{E}_{s\sim\nu_\rho^{\pi_{k+1}}}&\left[\langle Q_{w_k}(s,\cdot), \pi_{k+1}(\cdot|s) - \pi_k(\cdot|s)\rangle\right] \\ &+\frac{2}{1-\gamma}\sqrt{C_\nu \mathcal{L}_{\text{critic}}(w_k;\pi_k)}. \end{aligned} \tag{35}$$

Applying Theorems 28 and 30 to (34), we have the following:

$$\begin{aligned} &\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\left[\langle Q_{w_k}(s,\cdot), \pi_k(\cdot|s) - \pi^\star(\cdot|s)\rangle\right] \\ =& -\frac{1}{\eta_k}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\left[\left\langle \nabla h^{\pi^\star_{k+1}}(s,\cdot) - \nabla h^{\pi_k}(s,\cdot), \pi_k(\cdot|s) - \pi^\star(\cdot|s)\right\rangle\right] \\ =& -\frac{1}{\eta_k}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\left[D^{\pi^\star}_{\pi^\star_{k+1}}(s) - D^{\pi^\star}_{\pi_k}(s) - D^{\pi_k}_{\pi^\star_{k+1}}(s)\right] \\ \le& -\frac{1}{\eta_k}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\left[D^{\pi^\star}_{\pi_{k+1}}(s) - D^{\pi^\star}_{\pi_k}(s) - D^{\pi_k}_{\pi_{k+1}}(s)\right] + \frac{2}{\eta_k}\sqrt{C_\nu \mathcal{L}_{\text{actor}}(\theta_{k+1};\theta_k,w_k)}, \end{aligned} \tag{36}$$

where the second equality uses the three-point identity of KL-divergence. Similarly, applying Theorems 28 and 30 to (35), we get

$$\begin{aligned} &\mathbb{E}_{s\sim\nu_\rho^{\pi_{k+1}}}\left[\langle Q_{w_k}(s,\cdot), \pi_{k+1}(\cdot|s) - \pi_k(\cdot|s)\rangle\right] \\ \le& \frac{1}{\eta_k}\mathbb{E}_{s\sim\nu_\rho^{\pi_{k+1}}}\left[-D^{\pi_{k+1}}_{\pi_k}(s) - D^{\pi_k}_{\pi_{k+1}}(s)\right] + \frac{2}{\eta_k}\sqrt{C_\nu \mathcal{L}_{\text{actor}}(\theta_{k+1};\theta_k,w_k)}. \end{aligned}$$

Note that $-D^{\pi_{k+1}}_{\pi_k}(s) - D^{\pi_k}_{\pi_{k+1}}(s) \le 0$. Hence by Assumption 2 we get

$$\begin{aligned} &\mathbb{E}_{s\sim\nu_\rho^{\pi_{k+1}}}\left[\langle Q_{w_k}(s,\cdot), \pi_{k+1}(\cdot|s) - \pi_k(\cdot|s)\rangle\right] \\ \le& -\frac{1}{\eta_k}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\left[\frac{\mathrm{d}\nu_\rho^{\pi_{k+1}}(s)}{\mathrm{d}\nu_\rho^{\pi^\star}(s)}\left(D^{\pi_{k+1}}_{\pi_k}(s) + D^{\pi_k}_{\pi_{k+1}}(s)\right)\right] + \frac{2}{\eta_k}\sqrt{C_\nu \mathcal{L}_{\text{actor}}(\theta_{k+1};\theta_k,w_k)} \\ \le& -\frac{1}{\eta_k\left\|\frac{\mathrm{d}\nu_\rho^{\pi^\star}}{\mathrm{d}\nu_\rho^{\pi_{k+1}}}\right\|_\infty}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\left[D^{\pi_{k+1}}_{\pi_k}(s) + D^{\pi_k}_{\pi_{k+1}}(s)\right] + \frac{2}{\eta_k}\sqrt{C_\nu \mathcal{L}_{\text{actor}}(\theta_{k+1};\theta_k,w_k)} \\ \le& -\frac{1}{\eta_k\kappa_\nu}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\left[D^{\pi_{k+1}}_{\pi_k}(s) + D^{\pi_k}_{\pi_{k+1}}(s)\right] + \frac{2}{\eta_k}\sqrt{C_\nu \mathcal{L}_{\text{actor}}(\theta_{k+1};\theta_k,w_k)}. \end{aligned} \tag{37}$$

We first multiply (34) by $1-\gamma_\rho = \frac{1}{\kappa_\nu}$ and sum it up with (35), and then apply (36) and (37). Rearranging terms gives us

$$\begin{aligned}
&V^{\pi_{k+1}}(\rho) - V^\star(\rho) + \frac{1}{(1-\gamma)\kappa_\nu\eta_k}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\left[D_{\pi_{k+1}}^{\pi^\star}(s)\right] \\
\leq&\gamma_\rho\left(V^{\pi_k}(\rho) - V^\star(\rho)\right) + \frac{1}{(1-\gamma)\kappa_\nu\eta_k}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\left[D_{\pi_k}^{\pi^\star}(s) - D_{\pi_k}^{\pi_{k+1}}(s)\right] \\
&+ \frac{2(2-\gamma_\rho)}{(1-\gamma)}\sqrt{C_\nu\mathcal{L}_{\text{critic}}(w_k;\pi_k)} + \frac{2(2-\gamma_\rho)}{(1-\gamma)\eta_k}\sqrt{C_\nu\mathcal{L}_{\text{actor}}(\theta_{k+1};\theta_k,w_k)} \\
\leq&\gamma_\rho\left(V^{\pi_k}(\rho) - V^\star(\rho) + \frac{1}{(1-\gamma)\kappa_\nu\gamma_\rho\eta_k}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\left[D_{\pi_k}^{\pi^\star}(s)\right]\right) \\
&+ \frac{4\sqrt{C_\nu}}{(1-\gamma)}\left(\sqrt{\mathcal{L}_{\text{critic}}(w_k;\pi_k)} + \frac{1}{\eta_k}\sqrt{\mathcal{L}_{\text{actor}}(\theta_{k+1};\theta_k,w_k)}\right),
\end{aligned}$$

where the last inequality is obtained by dropping some non-positive terms. ■

C.4 Proof of Theorem 8

Proof For $k=0$, we have $V^{\pi_0}(\rho) - V^\star(\rho) \leq \frac{C}{1-\gamma}$ from (2). Since $\theta_0 = 0$, we have $f_{\theta_0}(s,a)$ be constant for any s and a, thus $D_{\pi_0}^{\pi^\star}(s) = \log|\mathcal{A}|$.

By Theorem 7 and concavity of the square root function, taking the expectation with respect to the samples Ξ_k^Q and Ξ_k^Π conditioned on previous samples yields

$$\begin{aligned}
&\mathbb{E}\big[V^{\pi_{k+1}}(\rho) - V^\star(\rho)\big] \\
\leq&\mathbb{E}\Big(V^{\pi_{k+1}}(\rho) - V^\star(\rho) + \frac{1}{(1-\gamma)\kappa_\nu\gamma_\rho\eta_{k+1}}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\big[D_{\pi_{k+1}}^{\pi^\star}(s)\big]\Big) \\
=&\mathbb{E}\Big(V^{\pi_{k+1}}(\rho) - V^\star(\rho) + \frac{1}{(1-\gamma)\kappa_\nu\eta_k}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\big[D_{\pi_{k+1}}^{\pi^\star}(s)\big]\Big) \\
\leq&\gamma_\rho\Big(V^{\pi_k}(\rho) - V^\star(\rho) + \frac{1}{(1-\gamma)\kappa_\nu\gamma_\rho\eta_k}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\big[D_{\pi_k}^{\pi^\star}(s)\big]\Big) \\
&+ \frac{4\sqrt{C_\nu}}{(1-\gamma)}\Big(\sqrt{\mathbb{E}\big[\mathcal{L}_{\text{critic}}(w_k;\pi_k)\big]} + \frac{1}{\eta_k}\sqrt{\mathbb{E}\big[\mathcal{L}_{\text{actor}}(\theta_{k+1};\theta_k,w_k)\big]}\Big) \\
\leq&\gamma_\rho\Big(V^{\pi_k}(\rho) - V^\star(\rho) + \frac{1}{(1-\gamma)\kappa_\nu\gamma_\rho\eta_k}\mathbb{E}_{s\sim\nu_\rho^{\pi^\star}}\big[D_{\pi_k}^{\pi^\star}(s)\big]\Big) + \frac{8\sqrt{C_\nu}C}{(1-\gamma)}\gamma_\rho^{k+1}.
\end{aligned}$$

Dividing both sides by γ_ρ^{k+1}, telescoping from 0 to k and rearranging terms, we obtain

$$\mathbb{E}\big[V^{\pi_{k+1}}(\rho) - V^\star(\rho)\big] \leq \frac{C}{1-\gamma}\cdot\gamma_\rho^{k+1}\Big(\big(1+\log|\mathcal{A}|\big) + 8\sqrt{C_\nu}(k+1)\Big),$$

where the expectation is taken with respect to all samples from zeroth to k-th iteration.

Let $C_1 = 1+\log|\mathcal{A}|$, $C_2 = 8\sqrt{C_\nu}$. For any $\epsilon > 0$, it suffices to choose $k = \left\lceil a\left(\log\frac{b}{\epsilon} + \log\log\frac{b}{\epsilon}\right)\right\rceil$ with $a = \frac{1}{\log\frac{1}{\gamma_\rho}}$ and $b = \frac{2C(C_1+C_2)}{(1-\gamma)\log\frac{1}{\gamma_\rho}}$, since

$$\begin{aligned}\mathbb{E}\big[V^{\pi_k}(\rho) - V^\star(\rho)\big] &\le (C_1+C_2)k\gamma_\rho^k \cdot \frac{C}{1-\gamma} \\ &\le \frac{(C_1+C_2)a\left(\log\frac{b}{\epsilon} + \log\log\frac{b}{\epsilon}\right)}{\frac{b}{\epsilon}\log\frac{b}{\epsilon}} \cdot \frac{C}{1-\gamma} \\ &\le \frac{2(C_1+C_2)a\log\frac{b}{\epsilon}}{b\log\frac{b}{\epsilon}} \cdot \frac{C\epsilon}{1-\gamma} = \epsilon.\end{aligned}$$

In view of $\log x \ge 1 - \frac{1}{x}$, we obtain

$$\begin{aligned}k &\le \log_{\frac{1}{\gamma_\rho}}\left(\frac{2C(C_1+C_2)}{(1-\gamma)\epsilon\log\frac{1}{\gamma_\rho}}\log\left(\frac{2C(C_1+C_2)}{(1-\gamma)\epsilon\log\frac{1}{\gamma_\rho}}\right)\right) + 1 \\ &\le \log_{\frac{1}{\gamma_\rho}}\left(\frac{2C(C_1+C_2)}{(1-\gamma_\rho)(1-\gamma)\epsilon}\log\left(\frac{2C(C_1+C_2)}{(1-\gamma_\rho)(1-\gamma)\epsilon}\right)\right) + 1 \\ &= \widetilde{O}\left(\log_{\frac{1}{\gamma_\rho}}\left(\frac{C\left(\sqrt{C_\nu}+\log|\mathcal{A}|\right)}{(1-\gamma)(1-\gamma_\rho)\epsilon}\right)\right),\end{aligned}$$

where $\widetilde{O}(\cdot)$ hides the logarithm terms. ■

Appendix D. Proofs for Sample Complexity

In this section, we provide missing proofs for Theorems 22 and 23. The proofs are based on the statistical recovery result (Theorem 46) for CNN in Appendix G.

D.1 Proof of Theorem 22

Proof From (2) and Theorem 13 we know $Q^{\pi_k}(\cdot, a)$ is $\frac{C}{1-\gamma}$-bounded and (L_Q, α)-Lipschitz on $\mathcal{S}$ for all $a \in \mathcal{A}$. From (30) in Appendix A.2 we have

$$\mathbb{E}\left[\mathcal{L}_{\text{critic}}(w_k; \pi_k)\right] \le |\mathcal{A}| \max_{a\in\mathcal{A}} \mathbb{E}\left[\mathbb{E}_{s\sim\nu_\rho^{\pi_k}}|Q_{w_k}(s,a) - Q^{\pi_k}(s,a)|^2\right],$$

Thus to ensure $\mathbb{E}\big[\mathcal{L}_{\text{critic}}(w_k; \pi_k)\big] \le C^2\gamma_\rho^{2(k+1)}$, it suffices to use a sample size N such that for all $a \in \mathcal{A}$,

$$\mathbb{E}_{\Xi_k}\mathbb{E}_{s\sim\nu_\rho^{\pi_k}}|Q_{w_k}(s,a) - Q^{\pi_k}(s,a)|^2 \le \frac{C^2\gamma_\rho^{2(k+1)}}{|\mathcal{A}|}. \tag{38}$$

We set the following for $\mathcal{W}_k = \mathcal{W}_{\text{Lip}}(\frac{C}{1-\gamma}, L_Q, \alpha, \epsilon_Q)$:

$$\begin{aligned}&M = O(N^{\frac{d}{d+2\alpha}}),\ L = O(\log N + D + \log D),\ J = O(D),\ I \in [2, D],\ R_1 = O(1), \\ &\log R_2 = O(\log^2 N + D\log N),\ \epsilon_Q = (L_Q^2 + \tfrac{C^2}{(1-\gamma)^2})D^{\frac{3\alpha}{2\alpha+d}}N^{-\frac{\alpha}{2\alpha+d}},\end{aligned}$$

where $O(\cdot)$ hides some constant depending on $\log L_Q$, $\log \frac{C}{1-\gamma}$, d, α, ω, B, and the surface area Area($\mathcal{S}$). Then by Theorem 26 in Appendix A.2 and Theorem 46 in Appendix G, the following bound on the regression error holds:

$$\begin{aligned}&\mathbb{E}_{\Xi_k}\mathbb{E}_{s\sim\nu_\rho^{\pi_k}}|Q_{w_k}(s,a)-Q^{\pi_k}(s,a)|^2\\ \leq& C'\left((\bar{L}_Q+1)^2\frac{C^2}{(1-\gamma)^2}+\sigma^2\right)N^{-\frac{2\alpha}{2\alpha+d}}\log^6 N,\end{aligned}\tag{39}$$

where $\bar{L}_Q$ is defined as in (24), $\sigma^2=\frac{\gamma^2C^2}{4(1-\gamma)^2}\leq\frac{C^2}{4(1-\gamma)^2}$ is the variance proxy derived in Theorem 26 and C' is a constant depending on $D^{\frac{6\alpha}{2\alpha+d}}$, $\log L_Q$, $\log\frac{C}{1-\gamma}$, d, α, ω, B, and the surface area Area($\mathcal{S}$). In particular, the dependence on $D^{\frac{6\alpha}{2\alpha+d}}$ is linear.

By choosing $N=\left(\frac{1}{\delta}\log^3\frac{1}{\delta}\right)^{\frac{d}{\alpha}+2}$ where $\delta=\sqrt{\frac{1}{4096((\bar{L}_Q+1)^2+\frac{1}{4})C'|\mathcal{A}|(\frac{d}{\alpha}+2)^6}}(1-\gamma)\gamma_\rho^{k+1}$, we have

$$\begin{aligned}N^{-\frac{2\alpha}{2\alpha+d}}\log^6 N&=\frac{\left(\left(\frac{d}{\alpha}+2\right)\log\left(\frac{1}{\delta}\log^3\frac{1}{\delta}\right)\right)^6}{\frac{1}{\delta^2}\log^6\frac{1}{\delta}}\\&\leq\frac{\left(\frac{d}{\alpha}+2\right)^6\left(4\log\frac{1}{\delta}\right)^6}{\frac{1}{\delta^2}\log^6\frac{1}{\delta}}\\&=\frac{(1-\gamma)^2\gamma_\rho^{2(k+1)}}{((\bar{L}_Q+1)^2+\frac{1}{4})C'|\mathcal{A}|}.\end{aligned}$$

Plugging the result into (39), we obtain (38). In particular, the dependence of N on D is $O(D^{\frac{6\alpha}{2\alpha+d}\frac{d+2\alpha}{2\alpha}})=O(D^3)$

Denoting $C_3=\sqrt{4096((\bar{L}_Q+1)^2+\frac{1}{4})C'|\mathcal{A}|(\frac{d}{\alpha}+2)^6}=O(\sqrt{|\mathcal{A}|})$, we have the sample size $N=\widetilde{O}\left(\frac{C_3}{1-\gamma}\gamma_\rho^{-(k+1)}\right)^{\frac{d}{\alpha}+2}=\widetilde{O}\left(\frac{\sqrt{|\mathcal{A}|}}{1-\gamma}\gamma_\rho^{-(k+1)}\right)^{\frac{d}{\alpha}+2}$. When Assumptions 2 and 3 hold, we know from Theorem 8 that the total iteration number K satisfies

$$\gamma_\rho^{-K}=\widetilde{O}\left(\frac{C\left(\sqrt{C_\nu}+\log|\mathcal{A}|\right)}{(1-\gamma)(1-\gamma_\rho)\epsilon}\right).$$

Plugging K into our choice of N yields the result. ■

D.2 Proof of Theorem 23

Proof By (31), it suffices to specify the architecture $\mathcal{F}$ and restrictions so that for all $a\in\mathcal{A}$ and all $k\leq K$,

$$\mathbb{E}_{\Xi_k}\mathbb{E}_{s\sim\nu_\rho^{\pi_k}}\left|f_{\theta_{k+1}}(s,a)-\gamma_\rho f_{\theta_k}(s,a)+Q_{w_k}(s,a)\right|^2\leq\frac{C^2\gamma_\rho^{2(k+1)}}{|\mathcal{A}|}.\tag{40}$$

By Theorem 19, our choice of η_k, λ_k, $\mathcal{W}_k$ and Θ_k ensures that $\lambda_{k+1}g_{k+1}^\star(\cdot,a)=\gamma_\rho f_{\theta_k}(\cdot,a)-Q_{w_k}(\cdot,a)$ is $(\frac{L_Q}{1-\gamma_\rho},\alpha,\frac{\epsilon_Q}{1-\gamma_\rho})$-approximately Lipschitz and is uniformly bounded by $\frac{C}{(1-\gamma_\rho)(1-\gamma)}$.

We set the following for the underlying CNN architecture $\mathcal{F}$ and the restricted parameter spaces $\mathcal{W}_k$ and Θ_k:

$$M = O(N^{\frac{d}{d+2\alpha}}),\ L = O(\log N + D + \log D),\ J = O(D),\ I \in [2, D],\ R_1 = O(1),$$
$$\log R_2 = O(\log^2 N + D \log N),\ \epsilon_Q = (L_Q^2 + \tfrac{C^2}{(1-\gamma)^2}) D^{\frac{3\alpha}{2\alpha+d}} N^{-\frac{\alpha}{2\alpha+d}},$$

where $O(\cdot)$ hides some constant depending on $\log L_Q$, $\log \frac{C}{1-\gamma}$, d, α, ω, B, and the surface area $\mathrm{Area}(\mathcal{S})$. Then by Theorem 26 in Appendix A.2 and Theorem 46 in Appendix G, the following bound on the regression error holds:

$$\begin{aligned}&\mathbb{E}_{\Xi_k}\mathbb{E}_{s\sim\nu_\rho^{\pi_k}} \left| f_{\theta_{k+1}}(s,a) - \gamma_\rho f_{\theta_k}(s,a) + Q_{w_k}(s,a)\right|^2 \\ \leq & C' \frac{C^2(\bar{L}_Q+1)^2}{(1-\gamma_\rho)^2(1-\gamma)^2} N^{-\frac{2\alpha}{2\alpha+d}} \log^6 N, \end{aligned} \tag{41}$$

where $\bar{L}_Q$ is defined as in (24) and C' is a constant depending on $D^{\frac{6\alpha}{2\alpha+d}}$, $\log L_Q$, $\log \frac{C}{1-\gamma}$, d, α, ω, B, and the surface area $\mathrm{Area}(\mathcal{S})$.

We let $N = \left(\frac{1}{\delta}\log^3\frac{1}{\delta}\right)^{\frac{d}{\alpha}+2}$ with $\delta = \sqrt{\frac{1}{C''|\mathcal{A}|(\frac{d}{\alpha}+2)^6}}(1-\gamma)(1-\gamma_\rho)\gamma_\rho^{K+1}$, where C'' is some constant depending on C' and $\bar{L}_Q$ so that the right-hand side of (41) becomes $\frac{C^2\gamma_\rho^{2(K+1)}}{|\mathcal{A}|}$, then we have (40) satisfied. Denoting $C_4 = \sqrt{C''|\mathcal{A}|(\frac{d}{\alpha}+2)^6} = O(\sqrt{|\mathcal{A}|})$, we can write $N = \widetilde{O}\left(\frac{C_4}{(1-\gamma_\rho)(1-\gamma)}\gamma_\rho^{-(K+1)}\right)^{\frac{d}{\alpha}+2}$. Similar to Theorem 22, the hidden dependence on D is cubic.

For $\epsilon > 0$, the total iteration number K satisfies

$$\gamma_\rho^{-K} = \widetilde{O}\left(\frac{C\left(\sqrt{C_\nu} + \log|\mathcal{A}|\right)}{(1-\gamma)(1-\gamma_\rho)\epsilon}\right).$$

Plugging K into our choice of N yields the result. ■

Appendix E. Approximation Theory for CNN

In this section, we introduce the approximation theory for CNN. We first consider the case when target function $f_0 = Q^\pi(\cdot, a)$ is (L_Q, α)-Lipschitz for any $a \in \mathcal{A}$ (Theorem 16), and then proceed to the case when f_0 is a general $(L_f, \alpha, \epsilon_f)$-approximately Lipschitz function (Theorem 20). Theorem 21 follows immediately from Theorem 20.

Let us define a class of single-block CNNs in the form of

$$f(x) = W \cdot \mathrm{Conv}_{\mathcal{W},\mathcal{B}}(P(x)) \tag{42}$$

as

$$\begin{aligned}\mathcal{F}^{\mathrm{SCNN}}(L,J,I,R_1,R_2) = \big\{ & f \mid f(x) \text{ in the form (42) with } L \text{ layers; the filter size} \\ & \text{is bounded by } I; \text{ the number of channels is bounded by } J; \\ & \max_l \|\mathcal{W}^{(l)}\|_\infty \vee \|\mathcal{B}^{(l)}\|_\infty \leq R_1,\ \|W\|_\infty \leq R_2\big\}. \end{aligned} \tag{43}$$

We will use this class of single-block CNNs as the building blocks of our final CNN approximation for the ground truth Lipschitz function.

E.1 Proof of Theorem 16 Overview

Theorem 16 establishes the relation between network architecture and approximation error for $f_0 = Q^\pi$. We prove Theorem 16 for (L_f, α)-Lipschitz f_0 in the following steps:

Step 1: Decompose f_0 as a sum of locally supported functions over the manifold. Since manifold $\mathcal{S}$ is assumed compact (Assumption 1), we can cover it with a finite set of D-dimensional open Euclidean balls $\{B_\beta(\mathbf{c}_i)\}_{i=1}^{C_\mathcal{S}}$, where $\mathbf{c}_i$ denotes the center of the i-th ball and β is the radius. We choose $\beta < \frac{\omega}{4}$ and define $U_i = B_\beta(\mathbf{c}_i) \cap \mathcal{S}$. Note that each U_i is diffeomorphic to an open subset of $\mathbb{R}^d$ (Niyogi et al., 2008, Lemma 5.4). Moreover, the set $\{U_i\}_{i=1}^{C_\mathcal{S}}$ forms an open cover for $\mathcal{S}$. There exists a carefully designed open cover with cardinality $C_\mathcal{S} \leq \left\lceil \frac{\text{Area}(\mathcal{S})}{\beta^d} T_d \right\rceil$, where $\text{Area}(\mathcal{S})$ denotes the surface area of $\mathcal{S}$ and T_d denotes the thickness of U_i's, that is, the average number of U_i's that contain a given point on $\mathcal{S}$. It has been shown that $T_d = O(d \log d)$ (Conway and Sloane, 1988).

Moreover, for each U_i, we can define a linear transformation

$$\phi_i(x) = a_i V_i^\top (x - \mathbf{c}_i) + b_i, \tag{44}$$

where $a_i \in (0, 1]$ is a scaling factor and $b_i \in \mathbb{R}^d$ is the translation vector, both of which are chosen to ensure $\phi(U_i) \subset [0, 1]^d$, and the columns of $V_i \in \mathbb{R}^{D \times d}$ form an orthonormal basis for the tangent space $T_{\mathbf{c}_i}(\mathcal{S})$ at $\mathbf{c}_i$. Overall, the atlas $\{(\phi_i, U_i)\}_{i=1}^{C_\mathcal{S}}$ transforms each local neighborhood on the manifold to a d-dimensional cube.

Thus, we can decompose f_0 using this atlas as

$$f_0 = \sum_{i=1}^{C_\mathcal{S}} f_i, \quad \text{with} \quad f_i = f_0 \times \rho_i, \tag{45}$$

because there exists such a C^∞ partition of unity $\{\rho_i\}_{i=1}^{C_\mathcal{S}}$ with $\text{supp}(\rho_i) \subset U_i$ (Liu et al., 2021, Proposition 1). Since each f_i is only supported on a subset of U_i, we can further write

$$f_0 = \sum_{i=1}^{C_\mathcal{S}} [(f_i \circ \phi_i^{-1}) \circ \phi_i] \times \mathbb{1}_{U_i}, \tag{46}$$

where $\mathbb{1}_{U_i}$ is the indicator function of U_i.

Lastly, we extend $f_i \circ \phi_i^{-1}$ to the entire cube $[0, 1]^d$ with 0:

$$\overline{f}_i(x) = \begin{cases} f_i \circ \phi_i^{-1}(x), & x \in \text{supp}(f_i \circ \phi_i^{-1}), \\ 0, & x \in [0, 1]^d \setminus \text{supp}(f_i \circ \phi_i^{-1}). \end{cases} \tag{47}$$

By Theorem 33 in Appendix E.5, $\overline{f}_i$ is a Lipschitz function with Lipschitz constant at most $L_i \coloneqq C_i(L_f + \|f_0\|_\infty)$, where C_i is a constant depending on α, ω, ϕ_i and ρ_i. This extended function will be approximated with first-order B-splines in the next step.

Step 2: Approximate each local function with first-order B-splines. Since each local function $\overline{f}_i$ is Lipschitz on d-dimensional unit cube, a weighted sum of first-order B-splines can approximate it. The number of splines depends exponentially on the intrinsic dimension d, rather than the ambient dimension D. To be more precise, we partition the unit cube into $N = 2^{pd}$ small cubes with side lengths 2^{-p}, where $p \in \mathbb{N}$ is positive. We denote $J(p) = \{0, 1, \ldots, 2^p - 1\}^d$ as a vector index set. The first-order B-spline $M_{p,j}$ with shift vector $j \in J(p)$ is defined as

$$M_{p,j}(x) = \prod_{k=1}^{d} \psi(2^p x_k - j_k), \tag{48}$$

where $\psi \colon [0, 2] \to \mathbb{R}$ is a sawtooth function:

$$\psi(x) = \begin{cases} x, & 0 \le x \le 1, \\ 2 - x, & 1 < x \le 2, \\ 0, & \text{otherwise.} \end{cases}$$

Each B-spline $M_{p,j}$ is supported on the small cube $B_j = \{x \in \mathbb{R}^d \mid j_k \le x_k \le j_k + 2,\ \forall k \in [d]\}$. Then by Theorem 34, there exists a function $\widetilde{f}_i$ in the form

$$\widetilde{f}_i = \sum_{j \in J(p)} c_{i,j} M_{p,j},$$

such that

$$\left\| \widetilde{f}_i - \overline{f}_i \right\|_\infty \le 2 L_i d N^{-\alpha/d}. \tag{49}$$

By (46) and (49), we now have a sum of first-order B-splines

$$\widetilde{f} \coloneqq \sum_{i=1}^{C_{\mathcal{S}}} [\widetilde{f}_i \circ \phi_i] \times \mathbb{1}_{U_i} = \sum_{i=1}^{C_{\mathcal{S}}} \left(\sum_{j \in J(p)} c_{i,j} M_{p,j} \circ \phi_i \right) \times \mathbb{1}_{U_i}, \tag{50}$$

which can approximate the target Lipschitz function f_0 with error

$$\left\| \widetilde{f} - f_0 \right\|_\infty \le 2 C_{\mathcal{S}} d \max_{i=1,\ldots,C_{\mathcal{S}}} C_i (L_f + \|f_0\|_\infty) N^{-\alpha/d}. \tag{51}$$

Step 3: Approximate each first-order B-spline with a composition of CNNs. We now turn to approximate $\widetilde{f}$ defined in (50) with a composition of CNNs. We first approximate some building blocks with single-block CNNs defined as in (43) and then ensemble them together. The building blocks include the multiplication operator $\times$, chart mappings $\{\phi_i\}_{i=1}^{C_{\mathcal{S}}}$, indicator functions $\{\mathbb{1}_{U_i}\}_{i=1}^{C_{\mathcal{S}}}$, and first-order B-splines $\{M_{p,j}\}_{j \in J(p)}$.

The multiplication operator $\times$ can be approximated by a single-block CNN $\widehat{\times}$ with at most η error in the L^∞ sense (Theorem 37), which needs $O(\log \frac{1}{\eta})$ layers and 6 channels. All weight parameters are bounded by $(c_0^2 \vee 1)$, where c_0 is the uniform upper bound of the input functions to be multiplied.

The chart mapping ϕ_i, according to (44), is a linear transformation. Thus, it can be expressed with a single-layer perceptron $\widehat{\phi}_i$, which can be equivalently expressed by a CNN.

The indicator function $\mathbb{1}_{U_i}$ is equal to 1 if $d_i^2(x) = \|x - \mathbf{c}_i\|_2^2 \leq \beta^2$ and equal to 0 otherwise. By this definition, we can write $\mathbb{1}_{U_i}$ as a composition of a univariate indicator $\mathbb{1}_{[0,\beta^2]}$ and the distance function d_i^2:

$$\mathbb{1}_{U_i}(x) = \mathbb{1}_{[0,\beta^2]} \circ d_i^2(x). \tag{52}$$

Given $\theta \in (0,1)$ and $\Delta \geq 8DB^2\theta$, it turns out that $\mathbb{1}_{[0,\beta^2]}$ and d_i^2 can be approximated with two single-block CNNs $\widehat{\mathbb{1}}_\Delta$ and $\widehat{d}_i^2$ respectively (Theorem 38) such that

$$\left\| \widehat{d}_i^2 - d_i^2 \right\|_\infty \leq 4B^2 D\theta \tag{53}$$

and

$$\widehat{\mathbb{1}}_\Delta \circ \widehat{d}_i^2(x) = \begin{cases} 1, & \text{if } x \in U_i, d_i^2(x) \leq \beta^2 - \Delta, \\ 0, & \text{if } x \notin U_i, \\ \text{between 0 and 1}, & \text{otherwise.} \end{cases}$$

The architecture and size of $\widehat{\mathbb{1}}_\Delta$ and $\widehat{d}_i^2$ are characterized in Theorem 38 as functions of θ and Δ.

The first-order B-spline $M_{p,j}$ can be approximated by a single-block CNN $\widehat{M}_{p,j}$ up to arbitrarily chosen ϵ_1 error (Theorem 35). We can find a proper ϵ_1 a set of single-block CNNs $\{\widehat{f}_{i,j}\}_{j \in J(p)}$ such that the error matches (49):

$$\left\| \sum_{j \in J(p)} \widehat{f}_{i,j}^{\mathrm{SCNN}} - \widetilde{f}_i \right\|_\infty \leq 2L_i d N^{-\alpha/d}. \tag{54}$$

The architecture and size of $\widehat{f}_{i,j}^{\mathrm{SCNN}}$ are characterized in Theorem 36 as functions of N.

Putting the above results together, we can develop a composition of single-block CNNs, which can be further expressed by a single-block CNN (Theorem 39):

$$\widehat{g}_i^{\mathrm{SCNN}} = \widehat{\times} \left(\sum_{j \in J(p)} \widehat{f}_{i,j}^{\mathrm{SCNN}} \circ \widehat{\phi}_i, \widehat{\mathbb{1}}_\Delta \circ \widehat{d}_i^2 \right). \tag{55}$$

Details are provided in Appendix E.2.

Step 4: Express the sum of CNN compositions with a CNN. Finally, we can assemble everything into $\widehat{f}$ as

$$\widehat{f} = \sum_{i=1}^{C_{\mathcal{S}}} \widehat{g}_i^{\mathrm{SCNN}}, \tag{56}$$

which serves as an approximation of f_0. By choosing the appropriate network size in Theorem 31, we can ensure that

$$\left\| \widehat{f} - f_0 \right\|_\infty \leq c_0 (L_f + \|f_0\|_\infty) N^{-\alpha/d}$$

for some constant c_0 depending on d, α, ω, B, and the surface area $\mathrm{Area}(\mathcal{S})$.

By Theorem 40, for $\widetilde{M}, \widetilde{J} > 0$, we can write this sum of $C_{\mathcal{S}}$ single-block CNNs as a sum of $\widetilde{M}$ single-block CNNs with the same architecture, whose channel number upper bound J depends on $\widetilde{J}$. This allows Theorem 16 to be more flexible with network architecture. By Theorem 42, this sum of $\widetilde{M}$ single-block CNNs can be further expressed as one CNN in the CNN class (13). Finally, N (or equivalently, p) will be chosen appropriately as a function of network architecture parameters, and the approximation theory of CNN is proven by plugging in $f_0 = Q^\pi$, $L_f = L_Q$ in Theorem 13.

In the following, we provide the proof details for Theorem 16.

E.2 Proof of Theorem 16

Proof We start from the decomposition of the approximation error of $\widehat{f}$, which is based on the decomposition of the approximation error of $\widehat{g}_i^{\mathrm{SCNN}}$ in (55).

Lemma 31 *Let $\eta > 0$ be the approximation error of the multiplication operator $\widehat{\times}(\cdot,\cdot)$ as defined in Step 3 of Appendix E.1 and Theorem 37, Δ and θ be defined as in Step 3 of Appendix E.1 and Theorem 38. Assume $N = 2^{pd}$ is chosen according to Theorem 36. For any $i = 1, \ldots, C_{\mathcal{S}}$, we have $\left\|\widehat{f} - f_0\right\|_\infty \le \sum_{i=1}^{C_{\mathcal{S}}}(A_{i,1} + A_{i,2} + A_{i,3})$ with*

$$
\begin{aligned}
A_{i,1} &= \left\| \widehat{g}_i^{\mathrm{SCNN}} - \left(\sum_{j \in J(p)} \widehat{f}_{i,j}^{\mathrm{SCNN}} \circ \widehat{\phi}_i \right) \times (\widehat{\mathbb{1}}_\Delta \circ \widehat{d}_i^2) \right\|_\infty \le \eta, \\
A_{i,2} &= \left\| \left(\sum_{j \in J(p)} \widehat{f}_{i,j}^{\mathrm{SCNN}} \circ \widehat{\phi}_i \right) \times (\widehat{\mathbb{1}}_\Delta \circ \widehat{d}_i^2) - f_i \times (\widehat{\mathbb{1}}_\Delta \circ \widehat{d}_i^2) \right\|_\infty \le 4L_i d N^{-\alpha/d}, \\
A_{i,3} &= \left\| f_i \times (\widehat{\mathbb{1}}_\Delta \circ \widehat{d}_i^2) - f_i \times \mathbb{1}_{U_i} \right\|_\infty \le \frac{C'(\pi+1)}{\beta(1-\beta/\omega)} \Delta
\end{aligned}
$$

for some constant C' depending on ρ_i and ϕ_i. Furthermore, for any $\epsilon \in (0,1)$, setting

$$
\eta = \max_{i=1\ldots C_{\mathcal{S}}} L_i d N^{-\alpha/d}, \ \Delta = \max_{i=1\ldots C_{\mathcal{S}}} \frac{L_i d \beta (1-\beta/\omega) N^{-\alpha/d}}{C'(\pi+1)}, \ \theta = \frac{\Delta}{16 B^2 D} \tag{57}
$$

yields

$$
\left\|\widehat{f} - f_0\right\|_\infty \le C'' C_{\mathcal{S}} (L_f + \|f_0\|_\infty) d N^{-\frac{\alpha}{d}},
$$

where C'' is a constant depending on α, ρ_i and ϕ_i. The choice in (57) satisfies the condition $\Delta > 8B^2 D\theta$ in Theorem 38.

Proof of Theorem 31 As in Theorem 37, $A_{i,1}$ measures the approximation error from $\widehat{\times}$:

$$
A_{i,1} = \left\| \widehat{g}_i^{\mathrm{SCNN}} - \left(\sum_{j \in J(p)} \widehat{f}_{i,j}^{\mathrm{SCNN}} \circ \widehat{\phi}_i \right) \times (\widehat{\mathbb{1}}_\Delta \circ \widehat{d}_i^2) \right\|_\infty \le \eta.
$$

The term $A_{i,2}$ measures the error from CNN approximation of local Lipschitz functions. As in Theorem 36, $A_{i,2} \leq 4L_i d N^{-\alpha/d}$.

The term $A_{i,3}$ measures the error from the CNN approximation of the chart determination function. The bound of $A_{i,3}$ can be derived using Theorem 38 and the proof of Lemma 4 in (Chen et al., 2022), since $\bar{f}_i$ is a Lipschitz function on $[0,1]^d$.

Finally, by Theorem 33, we have $L_i = O(L_f + \|f\|_\infty)$, and the proof is complete. ■

In order to attain the error desired in Theorem 31, we need each network in (55) with appropriate size. The network size of the components can be analyzed as follows:

- $\widehat{\mathbb{1}}_i$: The chart determination network $\widehat{\mathbb{1}}_i \coloneqq \widehat{d}_i^2 \circ \widehat{\mathbb{1}}_\Delta$ is the composition of $\widehat{d}_i^2$ and $\widehat{\mathbb{1}}_\Delta$. By Theorem 38, $\widehat{d}_i^2$ is a single-block CNN with $O(\log \frac{1}{\theta} + D) = O(\frac{\alpha}{d} \log N + D + \log D)$ layers and $6D$ channels; $\mathbb{1}_\Delta$ is a single-block CNN with $O(\log(\beta^2/\Delta)) = O(\frac{\alpha}{d} \log N)$ layers and 2 channels. In both subnetworks, all parameters are bounded by $O(1)$. By Theorem 39, the chart determination network $\widehat{\mathbb{1}}_i$ is a single-block CNN with $O(\frac{\alpha}{d} \log N + D + \log D)$ layers, $6D+2$ channels and all weight parameters bounded by $O(1)$.
- $\widehat{\times}$: By Theorem 37, the multiplication network is a single-block CNN with $O(\log \frac{1}{\eta}) = O(\frac{\alpha}{d} \log N)$ layers and $O(1)$ channels. By construction of $\widehat{f}_{i,j}^{\mathrm{SCNN}}$ and $\widehat{\mathbb{1}}_\Delta$, all weight parameters are bounded by $(\|f\|_\infty^2 \vee 1)$.
- $\widehat{\phi}_i$: The projection ϕ_i is a linear mapping, so it can be expressed with a single-layer perceptron. By Lemma 8 in Liu et al. (2021), this single-layer perceptron can be expressed with a single-block CNN with $D+2$ layers and width d. All parameters are of order $O(1)$.
- $\widehat{f}_{i,j}^{\mathrm{SCNN}}$: By Theorem 36, each $\widehat{f}_{i,j}^{\mathrm{SCNN}}$ is a single-block CNN with $O(\log N)$ layers and $80d$ channels. All weight parameters are in the order of $O(N^{\frac{1}{d}})$. Moreover, the sum $\widehat{f}_i^{\mathrm{SCNN}} \coloneqq \sum_{j \in J(p)} \widehat{f}_{i,j}^{\mathrm{SCNN}}$ can be realized by a single-block CNN with $O(\log N)$ layers and $O(d)$ channels by Theorem 40, and the parameters are still in the order of $O(N^{\frac{1}{d}})$.

Next, we show that the composition $\widehat{\times}\left(\widehat{f}_i^{\mathrm{SCNN}} \circ \widehat{\phi}_i, \widehat{\mathbb{1}}_\Delta \circ \widehat{d}_i^2\right)$ can be expressed as a single-block CNN. By Theorem 39, there exists a single-block CNN g_i with $O(\log N + D)$ layers and $O(d)$ channels realizing $\widehat{f}_i^{\mathrm{SCNN}} \circ \widehat{\phi}_i$. All parameters in g_i are in the order of $O(N^{\frac{1}{d}})$. Moreover, recall that the chart determination network $\widehat{\mathbb{1}}_i$ is a single-block CNN with $O(\log N + D + \log D)$ layers and $6D+2$ channels, whose parameters are of $O(1)$. By Lemma 14 in Liu et al. (2021), one can construct a convolutional block, denoted by $\bar{g}_i$, such that

$$\bar{g}_i(x) = \begin{bmatrix} (g_i(x))_+ & (g_i(x))_- & (\widehat{\mathbb{1}}_i(x))_+ & (\widehat{\mathbb{1}}_i(x))_- \\ \star & \star & \star & \star \end{bmatrix}. \tag{58}$$

Here $\bar{g}_{i,j}$ has $O(d+D) = O(D)$ channels. Since the input of $\widehat{\times}$ is $\begin{bmatrix} g_i \\ \widehat{\mathbb{1}}_i \end{bmatrix}$, by Lemma 15 in Liu et al. (2021), there exists a CNN $\mathring{g}_i$ which takes (58) as the input and outputs $\widehat{\times}(g_i, \widehat{\mathbb{1}}_i)$.

Since $\bar{g}_i$ only contains convolutional layers, the composition $\mathring{g}_i \circ \bar{g}_i$, denoted by $\widehat{g}_i^{\mathrm{SCNN}}$, is a single-block CNN and for any $x \in \mathcal{S}$, $\widehat{g}_i^{\mathrm{SCNN}}(x) = \widehat{\times}\left(\widehat{f}_i^{\mathrm{SCNN}} \circ \widehat{\phi}_i(x), \widehat{\mathbb{1}}_\Delta \circ \widehat{d}_i^2(x)\right)$. We have $\widehat{g}_i^{\mathrm{SCNN}} \in \mathcal{F}^{\mathrm{SCNN}}(L, J, I, R, R)$ with

$$L = O(\log N + D + \log D),\ J = O(D),\ R = O(N^{\frac{1}{d}}),$$

and I can be any integer in $[2, D]$. Therefore, we have shown that $\widehat{g}_i^{\mathrm{SCNN}}$ is a single-block CNN that expresses the composition (55), as we desired.

Furthermore, recall that $\widehat{f}$ can be written as a sum of $C_{\mathcal{S}}N$ such single-block CNNs. By Theorem 40, for any $\widetilde{M}$ and $\widetilde{J}$ satisfying $\widetilde{M}\widetilde{J} = O(C_{\mathcal{S}}ND)$, $\widetilde{M} \leq C_{\mathcal{S}}N$, $\widetilde{J} = \Theta(D)$, there exists a CNN architecture $\mathcal{F}^{\mathrm{SCNN}}(L, J, I, R, R)$ that gives rise to a set of single-block CNNs $\{\widehat{g}_i\}_{i=1}^{\widetilde{M}} \subset \mathcal{F}^{\mathrm{SCNN}}(L, J, I, R, R)$ with

$$\widehat{f} = \sum_{i=1}^{\widetilde{M}} \widehat{g}_i \tag{59}$$

and

$$L = O(\log N + D + \log D),\ J = O(D),\ R = O(N^{\frac{1}{d}}).$$

By Theorem 41, we slightly adjust the CNN architecture by re-balancing the weight parameters of the convolutional blocks and that of the final fully connected layer. In particular, we rescale all parameters in convolutional layers of $\widehat{g}_i$ to be no larger than 1. This procedure preserves the approximation power of the CNN while reducing the covering number (see Appendix F) of the CNN class. We set $\lambda = c' N^{\frac{1}{d}} (8ID)\widetilde{M}^{\frac{1}{L}}$, where c' is a constant such that $R \leq c' N^{\frac{1}{d}}$. With this λ, we have $\widehat{f}_i \in \mathcal{F}^{\mathrm{SCNN}}(L, J, I, R_1, R_2)$ with

$$\begin{aligned} &L = O(\log N + D + \log D),\ J = O(D),\ R_1 = O((8ID)^{-1}\widetilde{M}^{-\frac{1}{L}}) = O(1), \\ &\log R_2 = O(\log \widetilde{M} + \log^2 N + D \log N) \end{aligned} \tag{60}$$

such that $\widehat{g}_i \equiv \widehat{f}_i$.

Finally, we prove that it suffices to use one CNN to realize the sum of single-block CNNs in (59). By Theorem 42, there exists a CNN that can express the sum of $\widetilde{M}$ single-block CNNs with architecture $\mathcal{F}(M, L, J, I, R_1, R_2)$, where

$$\begin{aligned} &M = O(\widetilde{M}),\ L = O(\log N + D + \log D),\ J = O(D\widetilde{J}), \\ &R_1 = O((8ID)^{-1}\widetilde{M}^{-\frac{1}{L}}) = O(1),\ \log R_2 = O(\log \widetilde{M} + \log^2 N + D \log N). \end{aligned}$$

Here, $\widetilde{M}$ and $\widetilde{J}$ satisfy $\widetilde{M}\widetilde{J} = O(C_{\mathcal{S}}N)$ (note that $\widetilde{J}$ has changed to be the factor after D), which is a requirement inherited from Theorem 40. Changing $\widetilde{M}$ and $\widetilde{J}$ to eliminate the factors in front of Theorem 31 gives our final approximation $\widehat{f}$ of f_0:

$$\left\|\widehat{f} - f_0\right\| \leq (L_f + \|f_0\|_\infty)(\widetilde{M}\widetilde{J})^{-\frac{\alpha}{d}}$$

where the network has parameters

$$M = O(\widetilde{M}),\ L = O(\log(\widetilde{M}\widetilde{J}) + D + \log D),\ J = O(D\widetilde{J}),$$
$$R_1 = O((8ID)^{-1}\widetilde{M}^{-\frac{1}{L}}) = O(1),\ \log R_2 = O(\log\widetilde{M} + \log^2(\widetilde{M}\widetilde{J}) + D\log(\widetilde{M}\widetilde{J})),$$

where O hides some constant depending on $\log L_f$, $\log\|f_0\|_\infty$, d, α, ω, B, and the surface area $\mathrm{Area}(\mathcal{S})$. In particular, the hidden constant is bounded by $O((C''C_{\mathcal{S}}d)^{\frac{d}{\alpha}}) \le O((\frac{C''d^2\log d}{\omega^d})^{\frac{d}{\alpha}}) \le \exp\big(O(d^2)\big)$. By (2) and Theorem 13, we have $L_f = L_Q = \frac{C}{1-\gamma}\overline{L}_Q$ and $\|f_0\|_\infty = \frac{C}{1-\gamma}$, which completes the proof of Theorem 16. ■

E.3 Proof of Lemma 13

Proof Let us recall a useful characterization for total variation distance between probability measures μ, ν on any measurable space $\mathcal{X}$:

$$d_{\mathrm{TV}}(\mu,\nu) = \frac{1}{2}\sup_{f:\,\mathcal{X}\to[-1,1]}\left|\int_{\mathcal{X}} f\,\mathrm{d}\mu - \int_{\mathcal{X}} f\,\mathrm{d}\nu\right|. \tag{61}$$

Define

$$f(s) = \frac{2(1-\gamma)}{C}\cdot V^\pi(s) - 1.$$

By (2) we have $-1 \le f(s) \le 1$ for any $s \in \mathcal{S}$. In view of (4), (61) and that $\mathcal{M}$ is $(L_{\mathcal{P}}, L_c)$-Lipschitz, we have

$$\begin{aligned}
\big|Q^\pi(s,a) - Q^\pi(s',a)\big| &\le \big|c(s,a) - c(s',a)\big| + \gamma\left|\int_{\mathcal{S}} V^\pi(s'')\,\mathrm{d}\big(\mathcal{P}(s''|s,a) - \mathcal{P}(s''|s',a)\big)\right| \\
&= \big|c(s,a) - c(s',a)\big| + \frac{\gamma C}{2(1-\gamma)}\left|\int_{\mathcal{S}} f(s'')\,\mathrm{d}\big(\mathcal{P}(s''|s,a) - \mathcal{P}(s''|s',a)\big)\right| \\
&\le \big|c(s,a) - c(s',a)\big| + \frac{\gamma C}{(1-\gamma)} d_{\mathrm{TV}}(\mathcal{P}(\cdot|s,a), \mathcal{P}(\cdot|s',a)) \\
&\le L_c\cdot d_{\mathcal{S}}^\alpha(s,s') + \frac{\gamma C}{1-\gamma}L_{\mathcal{P}}\cdot d_{\mathcal{S}}^\alpha(s,s') \\
&= L_Q\cdot d_{\mathcal{S}}^\alpha(s,s').
\end{aligned}$$

The first line is from (4) and the triangle inequality. The second line is from the definition of f and that $\mathcal{P}(\cdot|s,a)$ is a distribution. The third line is from (61), and the fourth line is from the Lipschitz assumption. ■

E.4 Proof of Theorem 20

Proof We first show in Theorem 32 that for any approximately Lipschitz function f_0, there exists a Lipschitz "reference function" that is not far from f_0 in the L^∞ sense and has the same Lipschitz constant.

Lemma 32 *Suppose Assumption 1 holds. If a function $f_0\colon \mathcal{S} \to \mathbb{R}$ is (L, α, ϵ)-approximately Lipschitz, then there exists an (L, α)-Lipschitz function $\bar{f}_0$ such that $\|\bar{f}_0\|_\infty \le \|f_0\|_\infty$ and $\|f_0 - \bar{f}_0\|_\infty \le 2\epsilon$.*

Proof of Theorem 32 Define an envelope function $f_L(x) \coloneqq \inf_{y\in\mathcal{S}} \{f_0(y) + L \cdot d_{\mathcal{S}}^\alpha(y, x)\}$. It follows immediately from the (L, α, ϵ)-approximate Lipschitzness of f that for any $x \in \mathcal{S}$,

$$\begin{aligned}
&f_L(x) \le f_0(x) + L \cdot d_{\mathcal{S}}^\alpha(x, x) = f_0(x),\\
&f_L(x) \ge \inf_{y\in\mathcal{S}} \{f_0(x) - L \cdot d_{\mathcal{S}}^\alpha(y, x) - 2\epsilon + L \cdot d_{\mathcal{S}}^\alpha(y, x)\} = f_0(x) - 2\epsilon,
\end{aligned}$$

hence $\|f_L - f_0\|_\infty \le 2\epsilon$. Furthermore, for any $x, y \in \mathcal{S}$,

$$\begin{aligned}
f_L(x) - f_L(y) &= \inf_{z\in\mathcal{S}} \{f_0(z) + L \cdot d_{\mathcal{S}}^\alpha(z, x)\} - \inf_{z\in\mathcal{S}} \{f_0(z) + L \cdot d_{\mathcal{S}}^\alpha(z, y)\}\\
&= \inf_{z\in\mathcal{S}} \{f_0(z) + L \cdot d_{\mathcal{S}}^\alpha(z, x)\} + \sup_{z\in\mathcal{S}} \{-f_0(z) - L \cdot d_{\mathcal{S}}^\alpha(z, y)\}\\
&= \sup_{z\in\mathcal{S}} \left\{ \inf_{z'\in\mathcal{S}} \left\{f_0(z') + L \cdot d_{\mathcal{S}}^\alpha(z', x)\right\} - f_0(z) - L \cdot d_{\mathcal{S}}^\alpha(z, y) \right\}\\
&\le \sup_{z\in\mathcal{S}} \{f_0(z) + L \cdot d_{\mathcal{S}}^\alpha(z, x) - f_0(z) - L \cdot d_{\mathcal{S}}^\alpha(z, y)\}\\
&\le \sup_{z\in\mathcal{S}} \{L \cdot (d_{\mathcal{S}}(z, y) + d_{\mathcal{S}}(y, x))^\alpha - L \cdot d_{\mathcal{S}}^\alpha(z, y)\}\\
&\le \sup_{z\in\mathcal{S}} \{L \cdot (d_{\mathcal{S}}^\alpha(z, y) + d_{\mathcal{S}}^\alpha(y, x)) - L \cdot d_{\mathcal{S}}^\alpha(z, y)\}\\
&\le L \cdot d_{\mathcal{S}}^\alpha(x, y).
\end{aligned}$$

The first inequality is from that $\inf_{z'\in\mathcal{S}} \{f_0(z') + L \cdot d_{\mathcal{S}}^\alpha(z', x)\} \le f_0(z) + L \cdot d_{\mathcal{S}}^\alpha(z, x)$ for any $z \in \mathcal{S}$. The second inequality is from the triangle inequality of $d_{\mathcal{S}}(\cdot, \cdot)$ and the third inequality is from the subadditivity of $h(x) = x^\alpha$ when $x \ge 0$ and $\alpha \in (0, 1]$. Similarly we have $f_L(y) - f_L(x) \le L \cdot d_{\mathcal{S}}^\alpha(y, x)$ and hence f_L is L-Lipschitz. By truncating the negative part of f_L by $-\|f_0\|_\infty$, we obtain $\bar{f}_0$ such that

$$\bar{f}_0(x) = \begin{cases} f_L(x), & f_L(x) \ge -\|f_0\|_\infty,\\ -\|f_0\|_\infty, & f_L(x) < -\|f_0\|_\infty. \end{cases}$$

We can easily verify that $\bar{f}_0$ is L-Lipschitz, $\|\bar{f}_0\|_\infty \le \|f_0\|_\infty$ and $\|f_0 - \bar{f}_0\|_\infty \le 2\epsilon$. ■

By Theorem 32, there exists an (L_f, α)-Lipschitz function $\bar{f}_0$ such that $\|\bar{f}_0\|_\infty \le \|f_0\|_\infty$ and

$$\|\bar{f}_0 - f_0\|_\infty \le 2\epsilon_f.$$

Therefore, similar to Theorem 16, for any integers $I \in [2, D]$, $\widetilde{M}, \widetilde{J} > 0$,

$$\begin{aligned}
&M = O(\widetilde{M}),\ L = O(\log(\widetilde{M}\widetilde{J}) + D + \log D),\ J = O(D\widetilde{J}),\\
&\log R_2 = O(\log^2(\widetilde{M}\widetilde{J}) + D\log(\widetilde{M}\widetilde{J})),\ R_1 = (8ID)^{-1}\widetilde{M}^{-\frac{1}{L}} = O(1),
\end{aligned}$$

there exists a CNN $f \in \mathcal{F}(M, L, J, I, R_1, R_2)$ such that

$$\|f - f_0\|_\infty \le \left\|f - \bar{f}_0\right\|_\infty + \left\|\bar{f}_0 - f_0\right\|_\infty \le (L_f + \|f_0\|_\infty)(\widetilde{M}\widetilde{J})^{-\frac{\alpha}{d}} + 2\epsilon_f.$$

where $O(\cdot)$ hides a constant depending on $\log L_f$, $\log \|f_0\|_\infty$, d, α, ω, B, and the surface area $\mathrm{Area}(\mathcal{S})$.

The rest of the proof is to show that f is uniformly bounded by $\|f_0\|_\infty$ and is $(L_f, \alpha, \widehat{\epsilon}_f)$-approximately Lipschitz with $\widehat{\epsilon}_f = (L_f + \|f_0\|_\infty)(\widetilde{M}\widetilde{J})^{-\frac{\alpha}{d}}$. To show the uniform upper bound, we can apply a truncation layer to the components of $\widehat{f}$ so that every output will not exceed the range $[-\|f_0\|_\infty, \|f_0\|_\infty]$. This can be realized by adding a two-layer ReLU network $g\colon \mathbb{R} \to \mathbb{R}$,

$$g(x) = \mathrm{ReLU}(2\|f_0\|_\infty - \mathrm{ReLU}(\|f_0\|_\infty - x)) - \|f_0\|_\infty.$$

By Theorem 1 in Oono and Suzuki (2019), such a ReLU network can be expressed by a CNN $\widehat{g}$ with constant parameters. By Theorem 39, applying this CNN to the output of f results in a new CNN with the same order of size. In this case, we simply replace $\widehat{f}$ with this new CNN. By Theorem 32, $\left\|\bar{f}_0\right\|_\infty \le \|f_0\|_\infty$, so the truncation layer would not affect the approximation error, and we complete the proof for the uniform upper bound.

By Theorem 18, we conclude that f is $(L_f, \alpha, \widehat{\epsilon}_f)$-approximately Lipschitz with $\widehat{\epsilon}_f = (L_f + \|f_0\|_\infty)(\widetilde{M}\widetilde{J})^{\frac{\alpha}{d}}$. ■

E.5 Supporting Lemmas for CNN Approximation

In this section, we provide some auxiliary lemmas for CNN approximation. Theorem 33 shows that each local function $\bar{f}_i$ defined in (47) is Lipschitz on the low-dimensional Euclidean unit cube $[0,1]^d$. The Lipschitz constant L_i is controlled by the L^∞ norm and the Lipschitz constant of the original function L_f

Lemma 33 *Let $\bar{f}_i$ be defined as in (47). Then each function $\bar{f}_i$ is uniformly bounded by $\|f\|_\infty$ and is (L_i, α)-Lipschitz on $[0,1]^d$ with $L_i = O(L_f + \|f\|_\infty)$, where $O(\cdot)$ hides some constant depending on α, ω, ϕ_i and ρ_i.*

Proof We only need to show the Lipschitz continuity on the support of $f_i \circ \phi_i^{-1}$. Otherwise, the Lipschitz condition holds trivially since $f_i \circ \phi_i^{-1}$ is bounded and extended to the whole unit cube with 0.

Suppose $x, y \in \mathrm{supp}(f_i \circ \phi_i^{-1})$ are two points in the support, then there exist $u, v \in \mathrm{supp}(\rho_i) \subset U_i$ such that $u = \phi_i^{-1}(x), v = \phi_i^{-1}(y)$. By definition of the chart (U_i, ϕ_i), we have $\|u - v\|_2 \le 2\beta < \frac{\omega}{2}$ and that

$$\|x - y\|_2 = \left\|a_i V_i^\top (u - v)\right\|_2 \le \|u - v\|_2 < \frac{\omega}{2},$$

since $a_i \le 1$ and V_i is orthonormal. According to Proposition 6.3 in Niyogi et al. (2008), the geodesic distance between u and v is upper bounded by the Euclidean distance in $\mathbb{R}^D$ up to a constant factor:

$$d_{\mathcal{S}}(u, v) \le \omega - \omega\sqrt{1 - \frac{2\|u - v\|_2}{\omega}} \le 2\|u - v\|_2.$$

By Lemma 2 in Chen et al. (2022), ϕ_i is a diffeomorphism as long as the Euclidean ball radius satisfies $\beta \leq \frac{\omega}{4}$. By our construction, $\beta < \frac{\omega}{4}$, thus ϕ_i^{-1} is differentiable with its Jacobian bounded. Therefore, there exists a constant $L_{i,1}$ such that

$$L_{i,1} \coloneqq \sup_{z\in[0,1]^d} \left\|\nabla\phi_i^{-1}(z)\right\|_{\text{op}} < +\infty.$$

where $\nabla\phi_i^{-1}(z)$ is the Jacobian of ϕ_i^{-1} at z and $\|\cdot\|_{\text{op}}$ denotes the operator norm. Also notice that ρ_i is C^∞, thus we conclude that there exist another constant $L_{i,2}$ such that

$$L_{i,2} \coloneqq \sup_{z\in[0,1]^d} \left\|\nabla(\rho_i \circ \phi_i^{-1})(z)\right\|_2 < +\infty.$$

Combine the results together, we have

$$\begin{aligned}
&\left|f_i \circ \phi_i^{-1}(x) - f_i \circ \phi_i^{-1}(y)\right| \\
=&|f_i(u) - f_i(v)| \\
=&|f(u)\rho_i(u) - f(v)\rho_i(v)| \\
\leq&|f(u)\rho_i(u) - f(u)\rho_i(v)| + |f(u)\rho_i(v) - f(v)\rho_i(v)| \\
\leq&\|f\|_\infty \left|\rho_i \circ \phi_i^{-1}(x) - \rho_i \circ \phi_i^{-1}(y)\right| + |f(u) - f(v)| \\
\leq&\|f\|_\infty \sup_{z\in[0,1]^d} \left\|\nabla(\rho_i \circ \phi_i^{-1})(z)\right\|_2 \|x-y\|_2 + L_f \cdot d_{\mathcal{S}}^\alpha(u,v) \\
\leq&\|f\|_\infty \left(\frac{\omega}{2}\right)^{1-\alpha} L_{i,2} \cdot \|x-y\|_2^\alpha + L_f 2^\alpha L_{i,1}^\alpha \cdot \|x-y\|_2^\alpha \\
\leq&C_i(\|f\|_\infty + L_f)\|x-y\|_2^\alpha
\end{aligned}$$

where $C_i = \max(2^\alpha L_{i,1}^\alpha, (\frac{\omega}{2})^{1-\alpha} L_{i,2})$ is a constant depending on α, ω, ϕ_i and ρ_i. Denote $L_i \coloneqq C_i(\|f\|_\infty + L_f)$, we conclude that $f_i \circ \phi_i^{-1}$ is (L_i, α)-Lipschitz. ■

Theorem 34 further shows that Lipschitz functions on the unit cube $[0,1]^d$ can be arbitrarily approximated by first-order B-splines. The approximation error is $O(N^{-\alpha/d})$.

Lemma 34 *Let f be an (L,α)-Lipschitz function on the unit cube $[0,1]^d$ and take nonzero value only in the interior of the cube. For any $p \in \mathbb{N}$, $p \geq 1$, $N = 2^{pd}$, there exists a function $\widetilde{f}_N$ in the form*

$$\widetilde{f}_N = \sum_{j\in J(p)} c_j M_{p,j}$$

such that

$$\left\|\widetilde{f}_N - f\right\|_\infty \leq 2LdN^{-\alpha/d},$$

where $\max_{j\in J(p)} c_j = \|f\|_\infty$.

Proof For any $p \in \mathbb{N}$, $p \geq 1$, the index set $J(p) = \{0, 1, \ldots, 2^p - 2\}^d$ as defined in Step 2 of Appendix E.1. We denote $G(p) = \{2^{-k} y \mid y \in \mathbb{N}^d, 0 \leq y_k \leq 2^p, \forall k \in [d]\}$ as the set of all grid points. Let

$$\widetilde{f}_N = \sum_{j \in J(p)} c_j M_{p,j},$$

where $c_j = f(2^{-p}(j_1 + 1), \ldots, 2^{-p}(j_d + 1))$ for all $j \in J(p)$. By definition of the first-order B-spline and that f takes nonzero value only in the interior of the cube, we have $\widetilde{f}_N(x) = 0$ if $x_k \in \{0, 1\}$ for some $k \in [d]$ and thus $\widetilde{f}_N(x) = f(x)$ for all $x \in G(p)$. Moreover, $\widetilde{f}_N$ is a coordinate-wise (L, α)-Lipschitz linear function.

For any point $x \in [0, 1]^d$, there exists a grid point $y \in G(p)$ such that $\|x - y\|_\infty \leq 2^{-p-1}$. Define a sequence of points $\{y^{(t)}\}_{t=0}^d$ as

$$y_k^{(t)} = \begin{cases} y_k, & k \leq t, \\ x_k, & k > t. \end{cases}$$

We have $y^{(0)} = x$ and $y^{(d)} = y$. We can move the point x to y by changing one coordinate at a time following the sequence $\{y^{(t)}\}$. Thus we have

$$\begin{aligned} \left|\widetilde{f}_N(x) - f(x)\right| =& \left|\widetilde{f}_N(y) - f(y) + f(y) - f(x) + \sum_{t=0}^{d-1} \left(\widetilde{f}_N(y^{(t)}) - \widetilde{f}_N(y^{(t+1)})\right)\right| \\ \overset{(a)}{\leq} & |f(y) - f(x)| + \sum_{t=0}^{d-1} \left|\widetilde{f}_N(y^{(t)}) - \widetilde{f}_N(y^{(t+1)})\right| \\ \overset{(b)}{\leq} & L\|x - y\|_2^\alpha + L \sum_{t=0}^{d-1} \left\|y^{(t)} - y^{(t+1)}\right\|_2^\alpha \\ \overset{(c)}{\leq} & 2^{1-(p+1)\alpha} L d, \end{aligned}$$

where (a) uses $\widetilde{f}_N(y) = f(y)$ and the triangle inequality, (b) uses the Lipschitz continuity, and (c) is from the upper bound for norms. Since x is arbitrarily chosen from the unit cube, we have

$$\left\|\widetilde{f}_N - f\right\|_\infty \leq 2^{1-(p+1)\alpha} L d.$$

Plugging in $N = 2^{pd}$ yields the result. ■

Theorem 35 is a special case of Lemma 10 in Liu et al. (2021) for first-order splines (first-order cardinal B-splines). It shows that a single-block CNN can approximate each first-order B-spline to arbitrary accuracy.

Proposition 35 (Liu et al. (2021, Lemma 10)) *Let $p \in \mathbb{N}$ and $j \in \mathbb{N}^d$. There exists a constant C depending only on d such that for any $\epsilon \in (0, 1)$ and any $2 \leq I \leq d$, there exists a single-block CNN $\widehat{M}_{p,j} \in \mathcal{F}^{\mathrm{SCNN}}(L, J, I, R, R)$ with $L = 3 + 2\lceil \log_2(\frac{3}{C\epsilon}) + 5 \rceil \lceil \log_2 d \rceil$, $J = 80d$ and $R = 4 \vee 2^p$ that satisfies*

$$\left\|M_{p,j} - \widehat{M}_{p,j}\right\|_{L^\infty([0,1]^d)} \leq \epsilon,$$

and $\widehat{M}_{p,j}(x) = 0$ *for all* $x \notin B_{p,j} \coloneqq \{x \in \mathbb{R}^d \mid 2^{-p} j_k \le x_k \le 2^{-p}(j_k+2)\}$.

As a result, we show in Theorem 36 that CNNs can approximate local functions.

Proposition 36 *Let* $\overline{f}_i$ *be defined as in* (47). *For* $N = 2^{pd}$ *and any* $2 \le I \le D$, *there exists a set of single-block CNNs* $\{\widehat{f}_{i,j}^{\mathrm{SCNN}}\}_{j\in J(p)}$ *such that*

$$\left\| \sum_{j\in J(p)} \widehat{f}_{i,j}^{\mathrm{SCNN}} - \overline{f}_i \right\|_{L^\infty([0,1]^d)} \le 4L_i d N^{-\alpha/d}.$$

Each single-block CNN $\widehat{f}_{i,j}^{\mathrm{SCNN}}$ *is in* $\mathcal{F}^{\mathrm{SCNN}}(L, J, I, R, R)$ *with*

$$L = O(\log N),\ J = 80d,\ R = O(N^{\frac{1}{d}}),$$

where $O(\cdot)$ *hides some constant depending on* d *and* α.

Proof By Theorem 34 and the (L_i, α)-Lipschitzness of $\overline{f}_i$, for $p \ge 1$, $N = 2^{pd}$, there exists a function $\widetilde{f}_i$ in the form

$$\widetilde{f}_i = \sum_{j\in J(p)} c_{i,j} M_{p,j}$$

such that

$$\left\| \widetilde{f}_i - \overline{f}_i \right\|_\infty \le 2L_i d N^{-\alpha/d}.$$

By Theorem 35, there exists a collection of single-block CNNs $\{\widehat{M}_{p,j}\}_{j\in J(p)}$ that approximates the first-order B-splines $\{M_{p,j}\}_{j\in J(p)}$. Suppose $\left\|\widehat{M}_{p,j} - M_{p,j}\right\|_{L^\infty} \le \epsilon_1$ for all $j \in J(p)$ and some $\epsilon_1 \in (0,1)$, we have

$$\begin{aligned}\left\| \sum_{j\in J(p)} c_{i,j}\widehat{M}_{p,j} - \overline{f}_i \right\|_\infty &\le |J(p)| \left\|\overline{f}_i\right\|_\infty \epsilon_1 + 2L_i d N^{-\alpha/d} \\ &\le N L_i \epsilon_1 + 2L_i d N^{-\alpha/d},\end{aligned}$$

where the second inequality is from $\left\|\overline{f}_i\right\|_\infty = \|f\|_\infty \le L_i$ in Theorem 33. By letting $\epsilon_1 = 2dN^{-\frac{d+\alpha}{d}}$, we obtain

$$\left\| \sum_{j\in J(p)} c_{i,j}\widehat{M}_{p,j} - \overline{f}_i \right\|_\infty \le 4L_i d N^{-\alpha/d}.$$

According to Theorem 35, for any $j \in J(p)$, the single-block CNN $\widehat{M}_{p,j} \in \mathcal{F}^{\mathrm{SCNN}}(L, J, I, R, R)$ with

$$\begin{aligned}&L = 3 + 2\left\lceil \frac{d+\alpha}{d} \log_2 \frac{3^{\frac{d}{d+\alpha}} N}{2C_0 d} \right\rceil \lceil \log_2 d \rceil,\ J = 80d, \\ &2 \le I \le d,\ R = 4 \vee N^{\frac{1}{d}}.\end{aligned}$$

By letting $\widehat{f}_{i,j}^{\mathrm{SCNN}} = c_{i,j}\widehat{M}_{p,j}$ we prove the proposition. ■

The rest is to show that the multiplication operator and the indicator function can be approximated by CNNs. Theorem 37 shows that CNN can approximate the multiplication of scalars.

Proposition 37 *Let $\times$ be the scalar multiplication operator. For any $\eta \in (0,1)$, there exists a single-block CNN $\widehat{\times}$ such that*

$$\left\|a \times b - \widehat{\times}(a,b)\right\|_\infty \leq \eta,$$

where a, b are functions uniformly bounded by c_0. The approximated single-block CNN $\widehat{\times}$ is in $\mathcal{F}^{\mathrm{SCNN}}(L, J, I, R, R)$ with $L = O(\log \frac{1}{\eta}) + D$ layers, $J = 24$ channels and any filter size I such that $2 \leq I \leq D$. All parameters are bounded by $R = (c_0^2 \vee 1)$. Furthermore, the weight matrix in the fully connected layer of $\widehat{\times}$ has nonzero entries only in the first row.

Proof By Proposition 3 in Yarotsky (2017), there exists a feed-forward ReLU network that can approximate the multiplication operator between values with magnitude bounded by c_0 with η error. Such a feed-forward network has $O(\log \frac{1}{\eta})$ layers, each layer has its width bounded by 6, and all parameters are bounded by c_0^2. Therefore, such a feed-forward neural network is sufficient to approximate $\times$ with η error in L^∞-norm, since the function a, b are uniformly bounded by c_0.

Furthermore, by Lemma 8 in Liu et al. (2021), we can express the aforementioned feed-forward network with a single-block CNN in $\mathcal{F}^{\mathrm{SCNN}}(L, J, I, R, R)$, where L, J, I, R are specified in the statement of the proposition. ■

The indicator function $\mathbb{1}_{U_i}$ can be written as the composition of the indicator function of the closed interval $[0, \beta^2]$ and the squared Euclidean distance function $d_i \colon \mathcal{S} \to \mathbb{R}_+$ to the ball center $\mathbf{c}_i$:

$$\mathbb{1}_{U_i}(x) = \mathbb{1}_{[0,\beta^2]} \circ d_i(x), \tag{62}$$

where $d_i(x) = \|x - \mathbf{c}_i\|_2^2$. As Theorem 38 shows, these components can be approximated by CNNs.

Proposition 38 (Liu et al. (2021, Lemma 9)) *Let d_i and $\mathbb{1}_{[0,\beta^2]}$ be defined as in (62). For any $\theta \in (0,1)$ and $\Delta \geq 8B^2D\theta$, there exists a single-block CNN $\widehat{d}_i$ approximating d_i such that*

$$\left\|\widehat{d}_i - d_i\right\|_\infty \leq 4B^2D\theta,$$

and a single-block CNN $\widehat{\mathbb{1}}_\Delta$ approximating $\mathbb{1}_{[0,\beta^2]}$ with

$$\widehat{\mathbb{1}}_\Delta(x) = \begin{cases} 1, & \text{if } x \leq (1-2^{-k})(\beta^2 - 4B^2D\theta), \\ 0, & \text{if } x \geq \beta^2 - 4B^2D\theta, \\ 2^k((\beta^2 - 4B^2D\theta)^{-1}x - 1), & \text{otherwise,} \end{cases}$$

for $x \in \mathcal{S}$. The single-block CNN for $\widehat{d}_i$ has $O(\log(1/\theta) + D)$ layers, $6D$ channels and all weight parameters are bounded by $4B^2$. The single-block CNN $\widehat{\mathbb{1}}_\Delta$ has $\lceil \log(\beta^2/\Delta) \rceil$ layers, 2 channels and all weight parameters are bounded by $\max\left(2, \left|\beta^2 - 4B^2 D\theta\right|\right)$.

As a result, for any $x \in \mathcal{S}$, $\widehat{\mathbb{1}}_\Delta \circ \widehat{d}_i(x)$ gives an approximation of $\mathbb{1}_{U_i}$ satisfying

$$\widehat{\mathbb{1}}_\Delta \circ \widehat{d}_i(x) = \begin{cases} 1, & \text{if } x \in U_i \text{ and } d_i(x) \le \beta^2 - \Delta, \\ 0, & \text{if } x \notin U_i, \\ \text{between } 0 \text{ and } 1, & \text{otherwise.} \end{cases}$$

E.6 Supporting Lemmas for CNN Architecture

In this section, we introduce several lemmas for (single-block) CNN architecture. Theorem 39 states that the composition of two single-block CNNs can be expressed as one single-block CNN with augmented architecture.

Lemma 39 (Liu et al. (2021, Lemma 13)) *Let $\mathcal{F}_1 = \mathcal{F}^{\mathrm{SCNN}}(L_1, J_1, I_1, R_1, R_1)$ be a CNN architecture from $\mathbb{R}^D \to \mathbb{R}$ and $\mathcal{F}_2 = \mathcal{F}^{\mathrm{SCNN}}(L_2, J_2, I_2, R_2, R_2)$ be a CNN architecture from $\mathbb{R} \to \mathbb{R}$. Assume the weight matrix in the fully connected layer of $\mathcal{F}_1$ and $\mathcal{F}_2$ has nonzero entries only in the first row. Then there exists a CNN architecture $\mathcal{F}_3 = \mathcal{F}^{\mathrm{SCNN}}(L, J, I, R, R)$ from $\mathbb{R}^D \to \mathbb{R}$ with*

$$L = L_1 + L_2,\ J = \max(J_1, J_2),\ I = \max(I_1, I_2),\ R = \max(R_1, R_2)$$

such that for any $f_1 \in \mathcal{F}_1$ and $f_2 \in \mathcal{F}_2$, there exists $f \in \mathcal{F}_3$ such that $f(x) = f_2 \circ f_1(x)$. Furthermore, the weight matrix in the fully connected layer of $\mathcal{F}_3$ has nonzero entries only in the first row.

Theorem 40 states that the sum of n_0 single-block CNNs with the same architecture can be expressed as the sum of n_1 single-block CNNs with modified width.

Lemma 40 (Liu et al. (2022, Lemma 7)) *Let $\{f_i\}_{i=1}^{n_0}$ be a set of single-block CNNs with architecture $\mathcal{F}^{\mathrm{SCNN}}(L_0, J_0, I_0, R_0, R_0)$. For any integers n and $\widetilde{J}$ satisfying $1 \le n \le n_0$, $n\widetilde{J} = O(n_0 J_0)$ and $\widetilde{J} \ge J_0$, there exists an architecture $\mathcal{F}^{\mathrm{SCNN}}(L, J, I, R, R)$ that gives a set of single-block CNNs $\{g_i\}_{i=1}^{n}$ such that*

$$\sum_{i=1}^{n} g_i(x) = \sum_{i=1}^{n_0} f_i(x).$$

Such an architecture has

$$L = O(L_0),\ J = O(\widetilde{J}),\ I = I_0,\ R = R_0.$$

Furthermore, the fully connected layer of f has nonzero elements only in the first row.

Theorem 41 implies that one can slightly adjust the CNN architecture by re-balancing the weight parameter boundary of the convolutional blocks and that of the final fully connected layer. While re-balancing the weight would not affect the approximation power of the CNN, it will change the covering number of the CNN class, which is conducive to a different variance.

Lemma 41 (Liu et al. (2021, Lemma 16)) *Let $\lambda \geq 1$. For any $g \in \mathcal{F}^{\mathrm{SCNN}}(L, J, I, R_1, R_2)$, there exists $f \in \mathcal{F}^{\mathrm{SCNN}}(L, J, I, \lambda^{-1}R_1, \lambda^L R_2)$ such that $g(x) \equiv f(x)$.*

Finally, we prove Theorem 42, which states that the sum of single-block CNNs can be realized by a CNN of the form (13).

Lemma 42 *Let $\mathcal{F}^{\mathrm{SCNN}}(L, J, I, R_1, R_2)$ be any CNN architecture from $\mathbb{R}^D \to \mathbb{R}$. Assume the weight matrix in the fully connected layer of $\mathcal{F}^{\mathrm{SCNN}}(L, J, I, R_1, R_2)$ has nonzero entries only in the first row. For any positive integer M, there exists a CNN architecture $\mathcal{F}(M, L, J + 4, I, R_1, R_2(1 \vee R_1^{-1}))$ such that for any $\{\widehat{f}_i(x)\}_{i=1}^M \subset \mathcal{F}^{\mathrm{SCNN}}(L, J, I, R_1, R_2)$, there exists $\widehat{f} \in \mathcal{F}(M, L, J + 4, I, R_1, R_2(1 \vee R_1^{-1}))$ with*

$$\widehat{f}(x) = \sum_{m=1}^{M} \widehat{f}_m(x).$$

Proof Denote the architecture of $\widehat{f}_m$ with

$$\widehat{f}_m(x) = W_m \cdot \mathrm{Conv}_{\mathcal{W}_m, \mathcal{B}_m}(x),$$

where $\mathcal{W}_m = \{\mathcal{W}_m^{(l)}\}_{l=1}^L$, $\mathcal{B}_m = \{B_m^{(l)}\}_{l=1}^L$. Furthermore, denote the weight matrix and bias in the fully connected layer of $\widehat{f}$ with $\widehat{W}$, $\widehat{b}$ and the set of filters and biases in the m-th block of $\widehat{f}$ with $\widehat{\mathcal{W}}_m$ and $\widehat{\mathcal{B}}_m$ respectively. The padding layer $\widehat{P}$ in $\widehat{f}$ pads the input x from $\mathbb{R}^D$ to $\mathbb{R}^{D\times 4}$ with zeros. Each column denotes a channel.

Let us first show that for each m, there exists some $\mathrm{Conv}_{\widehat{\mathcal{W}}_m, \widehat{\mathcal{B}}_m} : \mathbb{R}^{D\times 4} \to \mathbb{R}^{D\times 4}$ such that for any $Z \in \mathbb{R}^{D\times 4}$ with the form

$$Z = \begin{bmatrix} (x)_+ & (x)_- & \star & \star \end{bmatrix}, \tag{63}$$

where $(x)_+$ means applying $(\cdot \vee 0)$ to every entry of x and $(x)_-$ means applying $-(\cdot \wedge 0)$ to every entry of x, so all entries in Z are non-negative. We have

$$\mathrm{Conv}_{\widehat{\mathcal{W}}_m, \widehat{\mathcal{B}}_m}(Z) = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \frac{R_1}{R_2}(f_m(\boldsymbol{x}) \vee 0) & -\frac{R_1}{R_2}(f_m(\boldsymbol{x}) \wedge 0) \\ & & \star & \star \\ & & \vdots & \vdots \\ & & \star & \star \end{bmatrix} + Z \tag{64}$$

where $\star$'s denotes entries that do not affect this result and may take any different value.

For any m, the first layer of f_m takes input in $\mathbb{R}^D$. Thus, the filters in $\mathcal{W}_m^{(1)}$ are in $\mathbb{R}^D$. Again, we pad these filters with zeros to get filters in $\mathbb{R}^{D\times 4}$ and construct $\widehat{\mathcal{W}}_m^{(1)}$ such that

$$\begin{aligned}
(\widehat{\mathcal{W}}_m^{(1)})_{1,:,:} &= \begin{bmatrix} \mathbf{e}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \\
(\widehat{\mathcal{W}}_m^{(1)})_{2,:,:} &= \begin{bmatrix} \mathbf{0} & \mathbf{e}_1 & \mathbf{0} & \mathbf{0} \end{bmatrix}, \\
(\widehat{\mathcal{W}}_m^{(1)})_{3,:,:} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{e}_1 & \mathbf{0} \end{bmatrix}, \\
(\widehat{\mathcal{W}}_m^{(1)})_{4,:,:} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{e}_1 \end{bmatrix}, \\
(\widehat{\mathcal{W}}_m^{(1)})_{4+j,:,:} &= \begin{bmatrix} (\mathcal{W}_m^{(1)})_{j,:,:} & (-\mathcal{W}_m^{(1)})_{j,:,:} & \mathbf{0} & \mathbf{0} \end{bmatrix},
\end{aligned}$$

where we use the fact that $\mathcal{W}_m^{(1)} * (x)_+ - \mathcal{W}_m^{(1)} * (x)_- = \mathcal{W}_m^{(1)} * x$. The first four output channels at the end of this first layer are copies of Z. For the filters in later layers of $\widehat{f}_m$ and all biases, we simply set

$$
\begin{aligned}
(\widehat{\mathcal{W}}_m^{(l)})_{1,:,:} &= \begin{bmatrix} \mathbf{e}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} && \text{for } l = 2, \dots, L, \\
(\widehat{\mathcal{W}}_m^{(l)})_{2,:,:} &= \begin{bmatrix} \mathbf{0} & \mathbf{e}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} && \text{for } l = 2, \dots, L, \\
(\widehat{\mathcal{W}}_m^{(l)})_{3,:,:} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{e}_1 & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} && \text{for } l = 2, \dots, L-1, \\
(\widehat{\mathcal{W}}_m^{(l)})_{4,:,:} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{e}_1 & \cdots & \mathbf{0} \end{bmatrix} && \text{for } l = 2, \dots, L-1, \\
(\widehat{\mathcal{W}}_m^{(l)})_{4+j,:,:} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & (\mathcal{W}_m^{(l)})_{j,:,:} \end{bmatrix} && \text{for } l = 2, \dots, L-1, \\
(\widehat{\mathcal{B}}_m^{(l)})_{j,:,:} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & (\mathcal{B}_m^{(l)})_{j,:,:} \end{bmatrix} && \text{for } l = 1, \dots, L-1.
\end{aligned}
$$

In $\mathrm{Conv}_{\widehat{\mathcal{W}}_m, \widehat{\mathcal{B}}_m}$, an additional convolutional layer is constructed to realize the fully connected layer in $\widehat{f}_m$. By our assumption, only the first row of W_m is nonzero. Furthermore, we set $\widehat{\mathcal{B}}_m^{(L)} = \mathbf{0}$ and $\widehat{\mathcal{W}}_m^L$ as size-one filters with three output channels in the form of

$$
\begin{aligned}
(\widehat{\mathcal{W}}_m^{(L)})_{3,:,:} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{e}_1 & \mathbf{0} & \frac{R_1}{R_2}(W_m)_{1,:} \end{bmatrix}, \\
(\widehat{\mathcal{W}}_m^{(L)})_{4,:,:} &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{e}_1 & -\frac{R_1}{R_2}(W_m)_{1,:} \end{bmatrix}.
\end{aligned}
$$

Under such choices, (64) is proved, and all parameters in $\widehat{\mathcal{W}}_m, \widehat{\mathcal{B}}_m$ are bounded by R_1.

By composing all convolutional blocks, we have

$$
\begin{aligned}
&(\mathrm{Conv}_{\widehat{\mathcal{W}}_M, \widehat{\mathcal{B}}_M}) \circ \cdots \circ (\mathrm{Conv}_{\widehat{\mathcal{W}}_1, \widehat{\mathcal{B}}_1}) \circ P(x) \\
&= \begin{bmatrix} (x)_+ & (x)_- & \frac{R_1}{R_2}\sum_{m=1}^M (\widehat{f}_m \vee 0) & -\frac{R_1}{R_2}\sum_{m=1}^M (\widehat{f}_m \wedge 0) \\ & & \star & \star \\ & & \vdots & \vdots \\ & & \star & \star \end{bmatrix}.
\end{aligned}
$$

Lastly, the fully connected layer can be set as

$$
\widetilde{W} = \begin{bmatrix} 0 & 0 & \frac{R_2}{R_1} & -\frac{R_2}{R_1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \ \widetilde{b} = 0.
$$

Note that the weights in the fully connected layer are bounded by $R_2(1 \vee R_1^{-1})$.

The above construction gives

$$
\widehat{f}(x) = \sum_{m=1}^M (\widehat{f}_m(x) \vee 0) + \sum_{m=1}^M (\widehat{f}_m(x) \wedge 0) = \sum_{m=1}^M \widehat{f}_m(x).
$$

■

Appendix F. CNN Class Covering Number

In this section, we prove a bound on the covering number of the convolutional neural network class used in Algorithm 1. The supporting lemmas and their proofs are provided in Appendix F.1

Lemma 43 *Given $\delta > 0$, the δ-covering number of the CNN class $\mathcal{F}(M, L, J, I, R_1, R_2)$ satisfies*

$$\mathcal{N}(\delta, \mathcal{F}(M, L, J, I, R_1, R_2), \|\cdot\|_\infty) \leq \left(2(R_1 \vee R_2)\Lambda_1 \delta^{-1}\right)^{\Lambda_2},$$

where

$$\Lambda_1 = (M+3)JD(1 \vee R_2)(1 \vee R_1)\widetilde{\rho}\widetilde{\rho}^+, \ \Lambda_2 = ML(J^2 I + J) + JD + 1$$

with $\widetilde{\rho} = \rho^M, \widetilde{\rho}^+ = 1 + ML\rho^+, \rho = (JIR_1)^L$ and $\rho^+ = (1 \vee JIR_1)^L$.

With a network architecture as stated in Theorems 16 and 20, we have

$$\log \mathcal{N}(\delta, \mathcal{F}(M, L, J, I, R_1, R_2), \|\cdot\|_\infty) = O\left(\widetilde{M}\widetilde{J}^2 D^3 \log^5(\widetilde{M}\widetilde{J}) \log \frac{1}{\delta}\right),$$

where $O(\cdot)$ hides a constant depending on d, α, ω, B, and the surface area $\mathrm{Area}(\mathcal{S})$.

To show Theorem 43, we first prove a supporting lemma (Theorem 45) that relates the distance in the function space of CNNs (in the L^∞ sense) to the distance in the parameter space. In this way, we transform the covering of the CNN class into the covering of the parameter space, which is simpler to deal with. We then give a proof of Theorem 43 in Appendix F.2.

F.1 Supporting Lemmas for Lemma 43

Theorem 44 below provides an upper bound on the L^∞-norm of a series of convolutional neural network blocks in terms of its architecture parameters, e.g. number of layers, number of channels, etc.

Let $J_m^{(i)}$ be the number of channels in i-th layer of the m-th block, and let $I_m^{(i)}$ be the filter size of i-th layer in the m-th block. We define $Q_{[i,j]}$ as

$$Q_{[i,j]}(x) = \left(\mathrm{Conv}_{\mathcal{W}_j, \mathcal{B}_j}\right) \circ \cdots \circ \left(\mathrm{Conv}_{\mathcal{W}_i, \mathcal{B}_i}\right)(x).$$

Proposition 44 *For $m = 1, 2, \cdots, M$ and $x \in [-1, 1]^D$, we have*

$$\left\|Q_{[1,m]}(x)\right\|_\infty \leq (1 \vee R_1)\left(\prod_{j=1}^{m}\prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} R_1\right)\left(1 + \sum_{k=1}^{m} L_k \prod_{i=1}^{L_k}(1 \vee J_k^{(i-1)} I_k^{(i)} R_1)\right).$$

Proof

$$
\begin{aligned}
&\left\|Q_{[1,m]}(x)\right\|_\infty \\
&= \left\|\mathrm{Conv}_{\mathcal{W}_m,\mathcal{B}_m}(Q_{[1,m-1]}(x))\right\|_\infty \\
&\le \prod_{i=1}^{L_m} J_m^{(i-1)} I_m^{(i)} R_1 \left\|Q_{[1,m-1]}(x)\right\|_\infty + R_1 L_m \prod_{i=1}^{L_m}(1 \vee J_m^{(i-1)} I_m^{(i)} R_1) \\
&\le \|P(x)\|_\infty \prod_{j=1}^{m}\prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} R_1 + R_1 \sum_{k=1}^{m} L_k \prod_{i=1}^{L_k}(1 \vee J_k^{(i-1)} I_k^{(i)} R_1) \prod_{l=j+1}^{m}\prod_{i=1}^{L_l} J_l^{(i-1)} I_l^{(i)} R_1 \\
&\le \|x\|_\infty \prod_{j=1}^{m}\prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} R_1 + R_1 \sum_{k=1}^{m} L_k \prod_{i=1}^{L_k}(1 \vee J_k^{(i-1)} I_k^{(i)} R_1) \prod_{l=j+1}^{m}\prod_{i=1}^{L_l} J_l^{(i-1)} I_l^{(i)} R_1 \\
&\le (1 \vee R_1)\left(\prod_{j=1}^{m}\prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} R_1\right)\left(1 + \sum_{k=1}^{m} L_k \prod_{i=1}^{L_k}(1 \vee J_k^{(i-1)} I_k^{(i)} R_1)\right),
\end{aligned}
$$

where the first two inequalities are obtained by applying Proposition 9 from Oono and Suzuki (2019) recursively. ■

Theorem 45 quantifies the sensitivity of a CNN with respect to small changes in its weight parameters. This will be used to create a discrete covering for the CNN class.

Lemma 45 *Let $\epsilon > 0$. For any $f, f' \in \mathcal{F}(M, L, J, I, R_1, R_2)$ such that $\|W - W'\|_\infty \le \epsilon$, $\|b - b'\|_\infty \le \epsilon$, $\left\|\mathcal{W}_m^{(l)} - \mathcal{W}_m^{(l)'}\right\|_\infty \le \epsilon$ and $\left\|\mathcal{B}_m^{(l)} - \mathcal{B}_m^{(l)'}\right\|_\infty \le \epsilon$ for all m and l, where $(W, b, \{\{(\mathcal{W}_m^{(l)}, \mathcal{B}_m^{(l)})\}_{l=1}^{L_m}\}_{m=1}^{M})$ and $(W', b', \{\{(\mathcal{W}_m^{(l)'}, \mathcal{B}_m^{(l)'})\}_{l=1}^{L_m}\}_{m=1}^{M})$ are the parameters of f and f' respectively, we have*

$$\left\|f - f'\right\|_\infty \le \Lambda_1 \epsilon,$$

where Λ_1 is defined in Theorem 43.

Proof For any $x \in [-1, 1]^D$,

$$
\begin{aligned}
&\left|f(x) - f'(x)\right| \\
&= \left|W \otimes Q(x) + b - W' \otimes Q'(x) - b'\right| \\
&= \left|(W - W') \otimes Q(x) + b - b' + W' \otimes \big(Q(x) - Q'(x)\big)\right| \\
&= \left|(W - W') \otimes Q(x) + b - b' + W' \otimes \big(Q(x) - \mathrm{Conv}_{\mathcal{W}_M,\mathcal{B}_M}(Q'(x)) + \mathrm{Conv}_{\mathcal{W}_M,\mathcal{B}_M}(Q'(x)) - Q'(x)\big)\right| \\
&= \left|(W - W') \otimes Q(x) + b - b' + \sum_{m=1}^{M} W' \otimes Q_{[m+1,M]} \circ \big(\mathrm{Conv}_{\mathcal{W}_m,\mathcal{B}_m} - \mathrm{Conv}_{\mathcal{W}'_m,\mathcal{B}'_m}\big) \circ Q'_{[0,m-1]}\right| \\
&\le \left|(W - W') \otimes Q(x;\theta) + b - b'\right| + \sum_{m=1}^{M} \left|W' \otimes Q_{[m+1,M]} \circ \big(\mathrm{Conv}_{\mathcal{W}_m,\mathcal{B}_m} - \mathrm{Conv}_{\mathcal{W}'_m,\mathcal{B}'_m}\big) \circ Q'_{[0,m-1]}\right| \\
&\overset{(a)}{\le} (3 + M) J D (1 \vee R_1)(1 \vee R_2)\left(\prod_{j=1}^{M}\prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} R_1\right)\left(1 + \sum_{k=1}^{M} L_k \prod_{i=1}^{L_k}(1 \vee J_k^{(i-1)} I_k^{(i)} R_1)\right)\epsilon,
\end{aligned}
$$

where (a) is obtained through the following reasoning.

The first term in (a) can be bounded as

$$\begin{aligned}
&\left|(W-W')\otimes Q(x)+b-b'\right| \\
&\leq \left(\|W\|_0+\left\|W'\right\|_0\right)\left\|W-W'\right\|_\infty \|Q(x)\|_\infty+\left\|b-b'\right\|_\infty \\
&\leq 2JD\epsilon\|Q(x)\|_\infty+\epsilon \\
&\leq 3JD\epsilon\|Q(x)\|_\infty \\
&\leq 3JD\max\{1,R_1\}\left(\prod_{j=1}^{M}\prod_{i=1}^{L_j}J_j^{(i-1)}I_j^{(i)}R_1\right)\left(1+\sum_{k=1}^{M}L_k\prod_{i=1}^{L_k}(1\vee J_k^{(i-1)}I_k^{(i)}R_1)\right)\epsilon,
\end{aligned}$$

where the first inequality uses Proposition 8 from Oono and Suzuki (2019) and the last inequality is obtained by invoking Theorem 44.

For the second term in (a), it is true that for any $m=1,\cdots,M$, we have

$$\begin{aligned}
&\left|W'\otimes Q_{[m+1,M]}\circ\left(\mathrm{Conv}_{\mathcal{W}_m,\mathcal{B}_m}-\mathrm{Conv}_{\mathcal{W}'_m,\mathcal{B}'_m}\right)\circ Q'_{[1,m-1]}\right| \\
&\overset{(b)}{\leq} \left\|W'\right\|_0 R_2\left\|Q_{[m+1,M]}\circ\left(\mathrm{Conv}_{\mathcal{W}_m,\mathcal{B}_m}-\mathrm{Conv}_{\mathcal{W}'_m,\mathcal{B}'_m}\right)\circ Q'_{[1,m-1]}\right\|_\infty \\
&\overset{(c)}{\leq} JDR_2\left(\prod_{j=m+1}^{M}\prod_{i=1}^{L_j}J_j^{(i-1)}I_j^{(i)}R_1\right)\left\|\left(\mathrm{Conv}_{\mathcal{W}_m,\mathcal{B}_m}-\mathrm{Conv}_{\mathcal{W}'_m,\mathcal{B}'_m}\right)\circ Q'_{[1,m-1]}\right\|_\infty \\
&\overset{(d)}{\leq} JDR_2\left(\prod_{j=m+1}^{M}\prod_{i=1}^{L_j}J_j^{(i-1)}I_j^{(i)}R_1\right)\left(\prod_{i=1}^{L_m}J_m^{(i-1)}I_m^{(i)}R_1\left\|Q'_{[1,m-1]}\right\|_\infty\epsilon\right) \\
&\overset{(e)}{\leq} JDR_2\left(\prod_{j=m+1}^{M}\prod_{i=1}^{L_j}J_j^{(i-1)}I_j^{(i)}R_1\right)\left(\prod_{i=1}^{L_m}J_m^{(i-1)}I_m^{(i)}R_1\right) \\
&\qquad (1\vee R_1)\left(\prod_{j=1}^{m}\prod_{i=1}^{L_j}J_j^{(i-1)}I_j^{(i)}R_1\right)\left(1+\sum_{k=1}^{m}L_k\prod_{i=1}^{L_k}(1\vee J_k^{(i-1)}I_k^{(i)}R_1)\right)\epsilon \\
&\leq JDR_2\left(\prod_{j=1}^{M}\prod_{i=1}^{L_j}J_j^{(i-1)}I_j^{(i)}R_1\right)(1\vee R_1)\left(1+\sum_{k=1}^{M}L_k\prod_{i=1}^{L_k}(1\vee J_k^{(i-1)}I_k^{(i)}R_1)\right)\epsilon,
\end{aligned}$$

where (b) is by Proposition 7 from Oono and Suzuki (2019), (c) is by Proposition 2 and 4 from Oono and Suzuki (2019), (d) is by Proposition 2 and 5 from Oono and Suzuki (2019), and (e) is obtained by invoking Theorem 44. ■

F.2 Proof of Lemma 43

Proof We grid the range of each parameter into subsets with width $\Lambda_1^{-1}\delta$, so there are at most $2(R_1\vee R_2)\Lambda_1\delta^{-1}$ different subsets for each parameter. In total, there are $\left(2(R_1\vee R_2)\Lambda_1\delta^{-1}\right)^{\Lambda_2}$ bins in the grid. For any $f,f'\in\mathcal{F}(M,L,J,I,R_1,R_2)$ within the

same grid, by Theorem 45, we have $\|f - f'\|_\infty \le \delta$. We can construct an ϵ-covering with cardinality $\left(2(R_1 \vee R_2)\Lambda_1 \delta^{-1}\right)^{\Lambda_2}$ by selecting one neural network from each bin in the grid.

Taking log and plugging in the network architecture parameters in Theorems 16 and 20, we have

$$\begin{aligned}\log \mathcal{N}(\delta, \mathcal{F}(M, L, J, I, R_1, R_2), \|\cdot\|_\infty) &= O\left(\Lambda_2 \log\left((R_1 \vee R_2)\,\Lambda_1 \delta^{-1}\right)\right) \\ &\le O\left(\widetilde{M} D D^2 \widetilde{J}^2 \log(\widetilde{M}\widetilde{J}) \log^2(\widetilde{M}\widetilde{J}) \log^2(\widetilde{M}\widetilde{J}) \log\frac{1}{\delta}\right) \\ &= O\left(\widetilde{M}\widetilde{J}^2 D^3 \log^5(\widetilde{M}\widetilde{J}) \log\frac{1}{\delta}\right),\end{aligned}$$

where the inequality is due to $\Lambda_2 = O(\widetilde{M} D D^2 \widetilde{J}^2 \log(\widetilde{M}\widetilde{J}))$. By plugging in the choice of R_1 with sufficiently small integer $\widetilde{J}$, we have $\rho = (1/2)^L M^{-1} \le M^{-1}$ and thus $\widetilde{\rho} \le (1 + M^{-1})^M \le e$. Moreover, we have $\widetilde{\rho}^+ = 1 + ML$. ■

Appendix G. Statistical Result of CNN Approximation

In this section, we derive the statistical estimation error for using a CNN empirical risk minimizer to estimate an approximately Lipschitz ground truth function over an i.i.d. dataset. We need to choose the appropriate CNN architecture and size in order to balance the approximation error from Theorem 20 and the variance. This statistical estimation error can be decomposed into the error of using CNN to approximate the target function (Theorem 20), terms that grow with the covering number of our CNN class, and the error of using the discrete covering to approximate our CNN class.

Lemma 46 *Suppose Assumption 1 holds, $f_0 \colon \mathcal{S} \to \mathbb{R}$ is a bounded $(L_f, \alpha, \epsilon_f)$-approximately Lipschitz function. We are given samples $\Xi_N = \{x_i, y_i\}_{i=1}^N$ where x_i's are i.i.d. sampled from a distribution $\mathcal{D}_x$ on $\mathcal{S}$ and $y_i = f_0(x_i) + \zeta_i$. ζ_i's are i.i.d. sub-Gaussian random noise with variance proxy σ^2 and are uncorrelated with x_i's. If $\epsilon_f = \mu D^{\frac{3\alpha}{2\alpha+d}} N^{-\frac{\alpha}{2\alpha+d}}$ for some constant $\mu = O(L_f + \|f_0\|_\infty)$ and the estimator*

$$\widehat{f}_N = \operatorname*{argmin}_{f \in \mathcal{F}_0} \frac{1}{N} \sum_{i=1}^N (f(x_i) - y_i)^2$$

is computed with neural network function class $\mathcal{F}_0 = \mathcal{F}_{\mathrm{Lip}}(A, L_0, \alpha, \epsilon_0)$ such that

$$\begin{aligned}&M = O(N^{\frac{d}{d+2\alpha}}),\ L = O(\log N + D + \log D), J = O(D),\ I \in [2, D],\ A = \|f_0\|_\infty, \\ &R_1 = O(1),\ \log R_2 = O(\log^2 N + D \log N),\ L_0 = L_f,\ \epsilon_0 = D^{\frac{3\alpha}{2\alpha+d}} N^{-\frac{\alpha}{2\alpha+d}},\end{aligned}$$

then we have

$$\mathbb{E}_{\Xi_N}\left[\int_{\mathcal{S}} \left(\widehat{f}_N(x) - f_0(x)\right)^2 \mathrm{d}\mathcal{D}_x(x)\right] \le c_0((L_f + A)^2 + \sigma^2) N^{-\frac{2\alpha}{2\alpha+d}} \log^6 N,$$

where c_0 *is some constant depending on* $D^{\frac{6\alpha}{2\alpha+d}}$, $\log L_f$, $\log \|f_0\|_\infty$, d, α, ω, B, *and the surface area* $\mathrm{Area}(\mathcal{S})$. $O(\cdot)$ *hides some constant depending on* $\log L_f$, $\log \|f_0\|_\infty$, d, α, ω, B, *and the surface area* $\mathrm{Area}(\mathcal{S})$.

First, note that the nonparametric regression error can be decomposed into two terms:

$$\begin{aligned}
&\mathbb{E}_{\Xi_N}\left[\int_{\mathcal{S}}\left(\widehat{f}_N(x)-f_0(x)\right)^2 \mathrm{d}\mathcal{D}_x(x)\right] \\
=&\underbrace{2\mathbb{E}_{\Xi_N}\left[\frac{1}{N}\sum_{i=1}^{N}(\widehat{f}_N(x_i)-f_0(x_i))^2\right]}_{T_1} \\
&+\underbrace{\mathbb{E}_{\Xi_N}\left[\int_{\mathcal{S}}\left(\widehat{f}_N(x)-f_0(x)\right)^2 \mathrm{d}\mathcal{D}_x(x)\right]-2\mathbb{E}_{\Xi_N}\left[\frac{1}{N}\sum_{i=1}^{N}(\widehat{f}_N(x_i)-f_0(x_i))^2\right]}_{T_2},
\end{aligned}$$

where T_1 reflects the squared bias of using CNN to approximate f_0, and T_2 is the variance term.

G.1 Supporting Lemmas for Lemma 46

We introduce two supporting lemmas from Chen et al. (2022) that show upper bounds for T_1 and T_2 in terms of the approximation error and covering number.

Lemma 47 (Chen et al. (2022, Lemma 5)) *Fix the neural network class* $\mathcal{F}_{\mathrm{Lip}}(A, L_0, \alpha, \epsilon_0)$. *For any constant* $\delta \in (0, 2A)$, *we have*

$$\begin{aligned}
T_1 \leq& 4 \inf_{f\in\mathcal{F}_{\mathrm{Lip}}(A,L_0,\alpha,\epsilon_0)} \int_{\mathcal{S}} (f(x)-f_0(x))^2 \,\mathrm{d}\mathcal{D}_x(x) \\
&+ 48\sigma^2 \frac{\log \mathcal{N}(\delta, \mathcal{F}_{\mathrm{Lip}}(A, L_0, \alpha, \epsilon_0), \|\cdot\|_\infty) + 2}{N} \\
&+ \left(8\sqrt{6}\sqrt{\frac{\log \mathcal{N}(\delta, \mathcal{F}_{\mathrm{Lip}}(A, L_0, \alpha, \epsilon_0), \|\cdot\|_\infty) + 2}{N}} + 8\right)\sigma\delta,
\end{aligned}$$

where $\mathcal{N}(\delta, \mathcal{F}_{\mathrm{Lip}}(A, L_0, \alpha, \epsilon_0), \|\cdot\|_\infty)$ *denotes the* δ*-covering number of* $\mathcal{F}_{\mathrm{Lip}}(A, L_0, \alpha, \epsilon_0)$ *with respect to the* L^∞ *norm, that is, there exists a discretization of the class* $\mathcal{F}_{\mathrm{Lip}}(A, L_0, \alpha, \epsilon_0)$ *with* $\mathcal{N}(\delta, \mathcal{F}_{\mathrm{Lip}}(A, L_0, \alpha, \epsilon_0), \|\cdot\|_\infty)$ *distinct elements such that for any* $f \in \mathcal{F}$, *there is a* $\overline{f}$ *in the discretization satisfying* $\left\|\overline{f} - f\right\|_\infty \leq \delta$.

Lemma 48 (Chen et al. (2022, Lemma 6)) *Fix the neural network class* $\mathcal{F}_{\mathrm{Lip}}(A, L_0, \alpha, \epsilon_0)$. *For any constant* $\delta \in (0, 2A)$, *we have*

$$T_2 \leq \frac{104A^2}{3N} \log \mathcal{N}(\delta/4A, \mathcal{F}_{\mathrm{Lip}}(A, L_0, \alpha, \epsilon_0), \|\cdot\|_\infty) + \left(4 + \frac{1}{2A}\right)\delta.$$

With Theorems 47 and 48, we can immediately prove Theorem 46.

G.2 Proof of Lemma 46

Proof Applying Theorems 47 and 48 to the bias and variance decomposition, we derive

$$\begin{aligned}\mathbb{E}_{\Xi_N}\left[\int_{\mathcal{S}}\left(\widehat{f}_N(x)-f_0(x)\right)^2\mathrm{d}\mathcal{D}_x(x)\right] \le\ & 4\inf_{f\in\mathcal{F}_{\mathrm{Lip}}(A,L_0,\alpha,\epsilon_0)}\int_{\mathcal{S}}(f(x)-f_0(x))^2\,\mathrm{d}\mathcal{D}_x(x)\\ &+48\sigma^2\frac{\log\mathcal{N}(\delta,\mathcal{F}_{\mathrm{Lip}}(A,L_0,\alpha,\epsilon_0),\|\cdot\|_\infty)+2}{N}\\ &+8\sqrt{6}\sqrt{\frac{\log\mathcal{N}(\delta,\mathcal{F}_{\mathrm{Lip}}(A,L_0,\alpha,\epsilon_0),\|\cdot\|_\infty)+2}{N}}\sigma\delta\\ &+\frac{104A^2}{3N}\log\mathcal{N}(\delta/4A,\mathcal{F}_{\mathrm{Lip}}(A,L_0,\alpha,\epsilon_0),\|\cdot\|_\infty)\\ &+\left(4+\frac{1}{2A}+8\right)\delta. \end{aligned}\tag{65}$$

By Theorem 20, if we set $\widetilde{M}\widetilde{J}=\epsilon^{-\frac{d}{\alpha}}$ and choose M,L,J,I,R_1,R_2,A such that

$$\begin{aligned}&M=O(\epsilon^{-\frac{d}{\alpha}}),\ L=O(\log(\epsilon^{-\frac{d}{\alpha}})+D+\log D),J=O(D),\ I\in[2,D],\\ &R_1=O(1),\ \log R_2=O(\log^2(\epsilon^{-\frac{d}{\alpha}})+D\log(\epsilon^{-\frac{d}{\alpha}})),\ A=\|f_0\|_\infty,\end{aligned}$$

for some $\epsilon\in(0,1)$, then there exists an $f\in\mathcal{F}_{\mathrm{Lip}}(A,L_f,\alpha,\epsilon)$ such that

$$\|f-f_0\|_\infty\le(L_f+A)\epsilon+2\epsilon_f.$$

Since $\mathcal{F}_{\mathrm{Lip}}(A,L_f,\alpha,\epsilon)\subseteq\mathcal{F}(M,L,J,I,R_1,R_2)$, we have

$$\mathcal{M}\left(\delta,\mathcal{F}_{\mathrm{Lip}}(A,L_f,\alpha,\epsilon),\|\cdot\|_\infty\right)\le\mathcal{M}\left(\delta,\mathcal{F}(M,L,J,I,R_1,R_2),\|\cdot\|_\infty\right),$$

where $\mathcal{M}$ denotes the packing number. Combining the relation between covering and packing numbers that $\mathcal{N}(\delta,\mathcal{F})\le\mathcal{M}(\delta,\mathcal{F})\le\mathcal{N}(\delta/2,\mathcal{F})$, we have

$$\mathcal{N}\left(\delta,\mathcal{F}_{\mathrm{Lip}}(A,L_f,\alpha,\epsilon),\|\cdot\|_\infty\right)\le\mathcal{N}\left(\delta/2,\mathcal{F}(M,L,J,I,R_1,R_2),\|\cdot\|_\infty\right).$$

By Theorem 43, we have

$$\begin{aligned}\log\mathcal{N}(\delta',\mathcal{F}(M,L,J,I,R_1,R_2)&=O\left(\widetilde{M}\widetilde{J}^2D^3\log^5(\widetilde{M}\widetilde{J})\log\frac{1}{\delta'}\right)\\ &=O\left(\epsilon^{-\frac{d}{\alpha}}D^3\log^5(\epsilon^{-\frac{d}{\alpha}})\log\frac{1}{\delta'}\right).\end{aligned}$$

Plugging the results back in (65), we get

$$\begin{aligned}&\mathbb{E}_{\Xi_N}\left[\int_{\mathcal{S}}\left(\widehat{f}_N(x)-f_0(x)\right)^2\mathrm{d}\mathcal{D}_x(x)\right]\\ &\quad\le\widetilde{O}\Bigg(((L_f+A)\epsilon+2\epsilon_f)^2+\frac{A^2+\sigma^2}{N}\epsilon^{-\frac{d}{\alpha}}D^3\log^5(\epsilon^{-\frac{d}{\alpha}})\log\frac{1}{\delta}\\ &\qquad+\sqrt{\frac{\epsilon^{-\frac{d}{\alpha}}D^3\log^5(\epsilon^{-\frac{d}{\alpha}})\log\frac{1}{\delta}}{N}}\sigma\delta+\sigma\delta+\frac{\sigma^2}{N}\Bigg).\end{aligned}\tag{66}$$

Finally we choose ϵ to satisfy $\epsilon^2 = D^3 N^{-1} \epsilon^{-\frac{d}{\alpha}}$, which gives $\epsilon = D^{\frac{3\alpha}{2\alpha+d}} N^{-\frac{\alpha}{2\alpha+d}}$. It suffices to pick $\delta = \frac{1}{N}$. Since $\epsilon_f = \mu D^{\frac{3\alpha}{2\alpha+d}} N^{-\frac{\alpha}{2\alpha+d}}$ with $\mu = O(L_f + \|f_0\|_\infty)$, we have $T_1 = O(T_2)$, that is, the bias term is dominated by the variance term. Therefore, by substituting both ϵ and δ in (66), we get the estimation error bound

$$\mathbb{E}_{\Xi_N}\left[\int_{\mathcal{S}} \left(\widehat{f}_N(x) - f_0(x)\right)^2 \mathrm{d}\mathcal{D}_x(x)\right] \le c_0((L_f + A)^2 + \sigma^2) N^{-\frac{2\alpha}{2\alpha+d}} \log^6 N,$$

where c_0 is a constant depending on $D^{\frac{6\alpha}{2\alpha+d}}$, $\log L_f$, $\log \|f_0\|_\infty$, d, α, ω, B, and the surface area $\mathrm{Area}(\mathcal{S})$. In particular, the dependence on $D^{\frac{6\alpha}{2\alpha+d}}$ is linear. ■

Appendix H. Experiment Details

In this section, we provide more details of the experiment in Section 5.

H.1 Environment

We use a visualized CartPole environment built upon the original version in Barto et al. (1983). The original CartPole environment has 4-dimensional state space $\mathcal{S}_0 = [-4.8, 4.8] \times \mathbb{R} \times [-0.418, 0.418] \times \mathbb{R}$ where the coordinates correspond to the cart position, cart velocity, pole angle and pole angular velocity respectively. The initial distribution $\rho = \mathrm{Unif}([-0.05, 0.05]^4)$. We use the renderer implemented in OpenAI Gym (Brockman et al., 2016) to generate images of different resolutions. Since the internal state contains velocity information that cannot be captured by one image, we use the differences between time-consecutive images to form the state space $\mathcal{S}$.

The environment terminates when current state $s \notin \bar{\mathcal{S}}_0 \coloneqq [-2.4, 2.4] \times \mathbb{R} \times [-0.209, 0.209] \times \mathbb{R}$ or the episode is greater than 200. The reward (negative cost) is 1 before termination and becomes 0 after termination. We normalize the cost function c so that $C = 1 - \gamma$.

H.2 Parameters

We choose step size η_k and temperature λ_k specified in Theorem 24. Note that after normalizing the cost function, η_k and λ_k depend solely on γ_ρ, and the actor update of NPMD becomes

$$f_{\theta_{k+1}} \leftarrow \gamma_\rho f_{\theta_k} - Q_{w_k},$$

thus the only parameter we need to tune is the (estimation of) shifted discount factor. In our implementation, we assume $\kappa_\nu = \frac{1}{1-\gamma}$ as if $\rho = \nu_\rho^{\pi^\star}$ and set $\gamma_\rho = \gamma$. Accordingly, we choose the step size $\eta_k = \gamma^{-(k+1)}$ and temperature $\lambda_k = \gamma^k$ for the k-th iteration.

References

A. Agarwal, M. Henaff, S. Kakade, and W. Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. *Advances in Neural Information Processing Systems*, 33:13399–13412, 2020.

A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(1):4431–4506, 2021.

C. Alfano and P. Rebeschini. Linear convergence for natural policy gradient with log-linear policy parametrization. *arXiv preprint arXiv:2209.15382*, 2022.

C. Alfano, R. Yuan, and P. Rebeschini. A novel framework for policy mirror descent with general parametrization and linear convergence. *arXiv preprint arXiv:2301.13139*, 2023.

F. Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(1):629–681, 2017.

A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 834–846, 1983.

C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

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

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

S. Cen, C. Cheng, Y. Chen, Y. Wei, and Y. Chi. Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*, 70(4):2563–2578, 2022.

L. Chen and S. Xu. Deep neural tangent kernel and laplace kernel have the same rkhs. *arXiv preprint arXiv:2009.10683*, 2020.

M. Chen, H. Jiang, W. Liao, and T. Zhao. Efficient approximation of deep relu networks for functions on low dimensional manifolds. *Advances in Neural Information Processing Systems*, 32, 2019.

M. Chen, H. Jiang, W. Liao, and T. Zhao. Nonparametric regression on low-dimensional manifolds using deep relu networks: Function approximation and statistical recovery. *Information and Inference: A Journal of the IMA*, 11(4):1203–1253, 2022.

J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups*, volume 290. Springer, 1988.

M. P. Do Carmo and J. Flaherty Francis. *Riemannian Geometry*, volume 6. Springer, 1992.

S. S. Du, S. M. Kakade, R. Wang, and L. F. Yang. Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*, 2020.

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

H. Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93 (3):418–491, 1959.

M. Geist, B. Scherrer, and O. Pietquin. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169. PMLR, 2019.

F. Gogianu, T. Berariu, M. C. Rosca, C. Clopath, L. Busoniu, and R. Pascanu. Spectral normalisation for deep reinforcement learning: an optimisation perspective. In *International Conference on Machine Learning*, pages 3734–3744. PMLR, 2021.

H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110:393–416, 2021.

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. *Advances in Neural Information Processing Systems*, 30, 2017.

S. Guo, C. Shi, C. Yang, and C. Zacharias. An online mirror descent learning algorithm for multiproduct inventory systems. *Available at SSRN 4806687*, 2024.

D. Hsu, C. H. Sanford, R. Servedio, and E. V. Vlatakis-Gkaragkounis. On the approximation power of two-layer networks of random relus. In *Conference on Learning Theory*, pages 2423–2461. PMLR, 2021.

K. Huang, Y. Wang, M. Tao, and T. Zhao. Why do deep residual networks generalize better than deep feedforward networks?—a neural tangent kernel perspective. *Advances in Neural Information Processing Systems*, 33:2698–2709, 2020.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

X. Ji, M. Chen, M. Wang, and T. Zhao. Sample complexity of nonparametric off-policy evaluation on low-dimensional manifolds using deep networks. *arXiv preprint arXiv:2206.02887*, 2022.

C. Jin, Z. Yang, Z. Wang, and M. I. Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.

E. Johnson, C. Pike-Burke, and P. Rebeschini. Optimal convergence rate for exact policy mirror descent in discounted markov decision processes. *arXiv preprint arXiv:2302.11381*, 2023.

S. M. Kakade. A natural policy gradient. *Advances in Neural Information Processing Systems*, 14, 2001.

V. Konda and J. Tsitsiklis. Actor-critic algorithms. *Advances in Neural Information Processing Systems*, 12, 1999.

G. Lan. Policy optimization over general state and action spaces. *arXiv preprint arXiv:2211.16715*, 2022.

G. Lan. Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes. *Mathematical Programming*, 198(1):1059–1106, 2023.

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

B. Liu, Q. Cai, Z. Yang, and Z. Wang. Neural proximal/trust region policy optimization attains globally optimal policy. *Advances in Neural Information Processing Systems*, 32, 2019.

H. Liu, M. Chen, T. Zhao, and W. Liao. Besov function approximation and binary classification on low-dimensional manifolds using convolutional residual networks. In *International Conference on Machine Learning*, pages 6770–6780. PMLR, 2021.

H. Liu, M. Chen, S. Er, W. Liao, T. Zhang, and T. Zhao. Benefits of overparameterized convolutional residual networks: Function approximation under smoothness constraint. In *International Conference on Machine Learning*, pages 13669–13703. PMLR, 2022.

T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2018.

W. U. Mondal and V. Aggarwal. Improved sample complexity analysis of natural policy gradient algorithm with general parameterization for infinite horizon discounted reward markov decision processes. *arXiv preprint arXiv:2310.11677*, 2023.

R. Mulayoff, T. Michaeli, and D. Soudry. The implicit bias of minima stability: A view from function space. *Advances in Neural Information Processing Systems*, 34:17749–17761, 2021.

T. Nguyen-Tang, S. Gupta, H. Tran-The, and S. Venkatesh. On sample complexity of offline reinforcement learning with deep reLU networks in besov spaces. *Transactions of Machine Learning Research*, 2022. URL `https://openreview.net/forum?id=LdEm0umNcv`.

P. Niyogi, S. Smale, and S. Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39:419–441, 2008.

K. Oono and T. Suzuki. Approximation and non-parametric estimation of resnet-type convolutional neural networks. In *International Conference on Machine Learning*, pages 4922–4931. PMLR, 2019.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 20, 2007.

J. Schmidt-Hieber. Deep relu network approximation of functions on a manifold. *arXiv preprint arXiv:1908.00695*, 2019.

J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897. PMLR, 2015.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

L. Shani, Y. Efroni, and S. Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *AAAI Conference on Artificial Intelligence*, volume 34(04), pages 5668–5675, 2020.

M. Song, A. Montanari, and P. Nguyen. A mean field view of the landscape of two-layers neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12, 1999.

E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

L. Wang, Q. Cai, Z. Yang, and Z. Wang. Neural policy gradient methods: Global optimality and rates of convergence. In *International Conference on Learning Representations*, 2019.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement Learning*, pages 5–32, 1992.

L. Xiao. On the convergence rates of policy gradient methods. *Journal of Machine Learning Research*, 23(282):1–36, 2022.

Z. Yang, C. Jin, Z. Wang, M. Wang, and M. Jordan. Provably efficient reinforcement learning with kernel and neural function approximations. *Advances in Neural Information Processing Systems*, 33:13903–13916, 2020.

D. Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.

G. Yehudai and O. Shamir. On the power and limitations of random features for understanding neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

M. Yin, M. Wang, and Y.-X. Wang. Offline reinforcement learning with differentiable function approximation is provably efficient. *arXiv preprint arXiv:2210.00750*, 2022.

Y. Yoshida and T. Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

R. Yuan, S. S. Du, R. M. Gower, A. Lazaric, and L. Xiao. Linear convergence of natural policy gradient methods with log-linear policies. In *International Conference on Learning Representations*, 2023.

W. Zhan, S. Cen, B. Huang, Y. Chen, J. D. Lee, and Y. Chi. Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence. *SIAM Journal on Optimization*, 33(2):1061–1091, 2023.

K. Zhang and Y.-X. Wang. Deep learning meets nonparametric regression: Are weight-decayed dnns locally adaptive? In *International Conference on Learning Representations*, 2022.

D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.