A PSEUDOCODE OF ALGORITHM 1

In this section, we present the pseudocode of FQE with CNN function approximation, which we have introduced in Section 3.

Algorithm 1 Neural Fitted Q-Evaluation (Neural-FQE)

```
Input: Initial distribution $\xi$, target policy $\pi$, horizon $H$, effective sample size $K$, function class $\mathcal{F}$.
Init: $\widehat{Q}^{\pi}_{H+1} := 0$
for $h = H, H-1, \cdots, 1$ do
    Sample $\mathcal{D}_h = \{(s_{h,k}, a_{h,k}, s'_{h,k}, r_{h,k})\}_{k=1}^{K}$.
    Update $\widehat{Q}^{\pi}_h \leftarrow \widehat{\mathcal{T}}^{\pi}_h \left(\widehat{Q}^{\pi}_{h+1}\right)$ by (6).
end for
Output: $\widehat{v}^{\pi} := \int_{\mathcal{X}} \widehat{Q}^{\pi}_1(s,a)\xi(s)\pi(a \mid s)\,\mathrm{d}s\,\mathrm{d}a$.
```

B PROOF OF THEOREM 1

In this section, we provide a proof for the upper bound on the estimation error in Theorem 1. Recall that Assumption 2 does not require Bellman completeness with respect to $\mathcal{F}$; thus, the estimation error can be decomposed into a sum of statistical error and approximation error. A tradeoff exists about the network size: while a larger network reduces the approximation error, it leads to higher variance in the statistical error. Consequently, we choose the network size and architecture appropriately to balance the two types of error, which in turn minimizes the final estimation error.

Proof of Theorem 1. The goal is to bound

$$\mathbb{E}\left|\widehat{v}^{\pi} - v^{\pi}\right| = \mathbb{E}\left|\int_{\mathcal{X}} \left(Q^{\pi}_1 - \widehat{Q}^{\pi}_1\right)(s,a)\,\mathrm{d}q^{\pi}_1(s,a)\right| \le \mathbb{E}\left[\int_{\mathcal{X}} \left|Q^{\pi}_1 - \widehat{Q}^{\pi}_1\right|(s,a)\,\mathrm{d}q^{\pi}_1(s,a)\right].$$

To get an expression for that, we first expand it recursively. To illustrate the recursive relation, we examine the quantity at step h:

$$\begin{aligned}
&\mathbb{E}\left[\int_{\mathcal{X}} \left|Q^{\pi}_h - \widehat{Q}^{\pi}_h\right|(s,a)\,\mathrm{d}q^{\pi}_h(s,a)\right] \\
&= \mathbb{E}\left[\int_{\mathcal{X}} \left|\mathcal{T}^{\pi}_h Q^{\pi}_{h+1} - \widehat{\mathcal{T}}^{\pi}_h\left(\widehat{Q}^{\pi}_{h+1}\right)\right|(s,a)\,\mathrm{d}q^{\pi}_h(s,a)\right] \\
&\le \mathbb{E}\left[\int_{\mathcal{X}} \left|\mathcal{T}^{\pi}_h Q^{\pi}_{h+1} - \mathcal{T}^{\pi}_h \widehat{Q}^{\pi}_{h+1}\right|(s,a)\,\mathrm{d}q^{\pi}_h(s,a)\right] + \mathbb{E}\left[\int_{\mathcal{X}} \left|\mathcal{T}^{\pi}_h \widehat{Q}^{\pi}_{h+1} - \widehat{\mathcal{T}}^{\pi}_h\left(\widehat{Q}^{\pi}_{h+1}\right)\right|(s,a)\,\mathrm{d}q^{\pi}_h(s,a)\right] \\
&= \mathbb{E}\left[\int_{\mathcal{X}} \left|Q^{\pi}_{h+1} - \widehat{Q}^{\pi}_{h+1}\right|(s,a)\,\mathrm{d}q^{\pi}_{h+1}(s,a)\right] \\
&\quad + \mathbb{E}\left[\mathbb{E}\left[\int_{\mathcal{X}} \left|\mathcal{T}^{\pi}_h \widehat{Q}^{\pi}_{h+1} - \widehat{\mathcal{T}}^{\pi}_h\left(\widehat{Q}^{\pi}_{h+1}\right)\right|(s,a)\,\mathrm{d}q^{\pi}_h(s,a) \mid \mathcal{D}_{h+1}, \cdots, \mathcal{D}_H\right]\right] \\
&\overset{(a)}{\le} \mathbb{E}\left[\int_{\mathcal{X}} \left|Q^{\pi}_{h+1} - \widehat{Q}^{\pi}_{h+1}\right|(s,a)\,\mathrm{d}q^{\pi}_{h+1}(s,a)\right] \\
&\quad + \mathbb{E}\left[\mathbb{E}\left[\sqrt{\int_{\mathcal{X}} \left(\mathcal{T}^{\pi}_h \widehat{Q}^{\pi}_{h+1} - \widehat{\mathcal{T}}^{\pi}_h\left(\widehat{Q}^{\pi}_{h+1}\right)\right)^2(s,a)\,\mathrm{d}q^{\pi_0}_h(s,a)}\sqrt{\chi^2_{\mathcal{Q}}(q^{\pi}_h, q^{\pi_0}_h) + 1} \mid \mathcal{D}_{h+1}, \cdots, \mathcal{D}_H\right]\right] \\
&\overset{(b)}{\le} \mathbb{E}\left[\int_{\mathcal{X}} \left|Q^{\pi}_{h+1} - \widehat{Q}^{\pi}_{h+1}\right|(s,a)\,\mathrm{d}q^{\pi}_{h+1}(s,a)\right] \\
&\quad + \sqrt{\mathbb{E}\left[\mathbb{E}\left[\int_{\mathcal{X}} \left(\mathcal{T}^{\pi}_h \widehat{Q}^{\pi}_{h+1} - \widehat{\mathcal{T}}^{\pi}_h\left(\widehat{Q}^{\pi}_{h+1}\right)\right)^2(s,a)\,\mathrm{d}q^{\pi_0}_h(s,a) \mid \mathcal{D}_{h+1}, \cdots, \mathcal{D}_H\right]\right]}\sqrt{\chi^2_{\mathcal{Q}}(q^{\pi}_h, q^{\pi_0}_h) + 1}
\end{aligned}$$

$$
\begin{aligned}
&\overset{(c)}{\leq} \int_{\mathcal{X}} \left| Q^{\pi}_{h+1} - \widehat{Q}^{\pi}_{h+1} \right| (s,a) \, \mathrm{d}q^{\pi}_{h+1}(s,a) + \sqrt{C'(5H^2) K^{-\frac{2\alpha}{2\alpha+d}} \log^5 K} \sqrt{\chi^2_{\mathcal{Q}}(q^{\pi}_h, q^{\pi_0}_h) + 1} \\
&\leq \int_{\mathcal{X}} \left| Q^{\pi}_{h+1} - \widehat{Q}^{\pi}_{h+1} \right| (s,a) \, \mathrm{d}q^{\pi}_{h+1}(s,a) + CHK^{-\frac{\alpha}{2\alpha+d}} \log^{5/2} K \sqrt{\chi^2_{\mathcal{Q}}(q^{\pi}_h, q^{\pi_0}_h) + 1},
\end{aligned}
$$

where C denotes a (varying) constant depending on $D^{\frac{3\alpha}{2\alpha+d}}$, d, α, $\frac{d}{\alpha p - d}$, p, q, c_0, B, ω and the surface area of $\mathcal{X}$.

In (a), note $\mathcal{T}^{\pi}_h \widehat{Q}^{\pi}_{h+1} \in \mathcal{B}^{\alpha}_{p,q}(\mathcal{X})$ by Assumption 2 and $-\widehat{\mathcal{T}}^{\pi}_h \left(\widehat{Q}^{\pi}_{h+1} \right) \in \mathcal{F}$ by our algorithm, so $\mathcal{T}^{\pi}_h \widehat{Q}^{\pi}_{h+1} - \widehat{\mathcal{T}}^{\pi}_h \left(\widehat{Q}^{\pi}_{h+1} \right) \in \mathcal{Q}$. Then we obtain this inequality by invoking the following lemma.

Lemma 1. Given a function class $\mathcal{Q}$ that contains functions mapping from $\mathcal{X}$ to $\mathbb{R}$ and two probability distributions q_1 and q_2 supported on $\mathcal{X}$, for any $g \in \mathcal{Q}$,

$$
\mathbb{E}_{x \sim q_1} \left[g(x) \right] \leq \sqrt{\mathbb{E}_{x \sim q_2} [g^2(x)] (1 + \chi^2_{\mathcal{Q}}(q_1, q_2))}.
$$

Proof of Lemma 1.

$$
\begin{aligned}
\mathbb{E}_{x \sim q_1} \left[g(x) \right] &= \sqrt{\mathbb{E}_{x \sim q_2} [g^2(x)] \frac{\mathbb{E}_{x \sim q_1} [g(x)]^2}{\mathbb{E}_{x \sim q_2} [g^2(x)]}} \\
&\leq \sqrt{\mathbb{E}_{x \sim q_2} [g^2(x)] \sup_{f \in \mathcal{Q}} \frac{\mathbb{E}_{x \sim q_1} [f(x)]^2}{\mathbb{E}_{x \sim q_2} [f^2(x)]}} \\
&= \sqrt{\mathbb{E}_{x \sim q_2} [g^2(x)] (1 + \chi^2_{\mathcal{Q}}(q_1, q_2))},
\end{aligned}
$$

where the last step is by the definition of $\chi^2_{\mathcal{Q}}(q_1, q_2) := \sup_{f \in \mathcal{Q}} \frac{\mathbb{E}_{q_1}[f]^2}{\mathbb{E}_{q_2}[f^2]} - 1$. □

In (b), we use Jensen's inequality and the fact that square root is concave.

To obtain (c), we invoke Lemma 10, which provides an upper bound on the error of nonparametric regression at each step of the FQE algorithm.

Specifically, we will invoke Lemma 10 when conditioning on $\mathcal{D}_{h+1}, \cdots, \mathcal{D}_H$, i.e. the data from time step $h+1$ to time step H. Note that after conditioning, $\mathcal{T}^{\pi}_h \widehat{Q}^{\pi}_{h+1}$ becomes measurable and deterministic with respect to $\mathcal{D}_{h+1}, \cdots, \mathcal{D}_H$. Also, $\mathcal{D}_{h+1}, \cdots, \mathcal{D}_H$ are independent from $\mathcal{D}_h$, which we use in the regression at step h.

To justify our use of this theorem, we need to cast our problem into a regression problem described in the theorem. Since $\{(s_{h,k}, a_{h,k})\}_{k=1}^K$ are i.i.d. from $q^{\pi_0}_h$, we can view them as the samples x_i's in the lemma. We can view $\mathcal{T}^{\pi}_h \widehat{Q}^{\pi}_{h+1}$, which is measurable under our conditioning, as f_0 in the lemma. Furthermore, we let

$$
\zeta_{h,k} := r_{h,k} + \int_{\mathcal{A}} \widehat{Q}^{\pi}_{h+1}(s'_{h,k}, a) \pi(a \mid s'_{h,k}) \, \mathrm{d}a - \mathcal{T}^{\pi}_h \widehat{Q}^{\pi}_{h+1}(s_{h,k}, a_{h,k}).
$$

In order to invoke Lemma 10 under the conditioning on $\mathcal{D}_{h+1}, \cdots, \mathcal{D}_H$, we need to verify whether three conditions are satisfied (conditioning on $\mathcal{D}_{h+1}, \cdots, \mathcal{D}_H$):

1. Sample $\{(s_{h,k}, a_{h,k})\}_{k=1}^K$ are i.i.d;
2. Sample $\{(s_{h,k}, a_{h,k})\}_{k=1}^K$ and noise $\{\zeta_{h,k}\}_{k=1}^K$ are uncorrelated;
3. Noise $\{\zeta_{h,k}\}_{k=1}^K$ are independent, zero-mean, subgaussian random variables.

In our setting, $\{(s_{h,k}, a_{h,k})\}_{k=1}^K$ are i.i.d. from $q^{\pi_0}_h$. Due to the time-inhomogeneous setting, they are independent from $\mathcal{D}_{h+1}, \cdots, \mathcal{D}_H$, so $\{(s_{h,k}, a_{h,k})\}_{k=1}^K$ are still i.i.d. under our conditioning. Thus, Condition 1 is clearly satisfied.

We may observe that under our conditioning, the transition from $(s_{h,k}, a_{h,k})$ to $s'_{h,k}$ is the only source of randomness in $\zeta_{h,k}$, besides $(s_{h,k}, a_{h,k})$ itself. The distribution of $(s_{h,k}, a_{h,k}, s'_{h,k})$ is actually the product distribution between $P_h(\cdot|s_{h,k}, a_{h,k})$ and $q_h^{\pi_0}$, so a function of $s'_{h,k}$, generated from the transition distribution $P_h(\cdot|s_{h,k}, a_{h,k})$, is uncorrelated with $(s_{h,k}, a_{h,k})$. Thus, $(s_{h,k}, a_{h,k})$'s are uncorrelated with $\zeta_{h,k}$'s under our conditioning, and Condition 2 is satisfied.

Condition 3 can also be easily verified. Under our conditioning, the randomness in $\zeta_{h,k}$ only comes from $(s_{h,k}, a_{h,k}, s'_{h,k}, r_{h,k})$, which are independent from $(s_{h,k'}, a_{h,k'}, s'_{h,k'}, r_{h,k'})$ for any $k' \neq k$, so $\zeta_{h,k}$'s are independent from each other. As for the mean of $\zeta_{h,k}$,

$$
\begin{aligned}
&\mathbb{E}\left[\zeta_{h,k} \mid \mathcal{D}_{h+1}, \cdots, \mathcal{D}_H\right] \\
=&\, \mathbb{E}\left[r_{h,k} + \int_{\mathcal{A}} \widehat{Q}^{\pi}_{h+1}(s'_{h,k}, a)\pi(a \mid s'_{h,k})\,\mathrm{d}a - r_h(s_{h,k}, a_{h,k}) - \mathcal{P}^{\pi}_h \widehat{Q}^{\pi}_{h+1}(s_{h,k}, a_{h,k}) \mid \mathcal{D}_{h+1}, \cdots, \mathcal{D}_H\right] \\
=&\, \mathbb{E}\Bigg[r_{h,k} - r_h(s_{h,k}, a_{h,k}) + \int_{\mathcal{A}} \widehat{Q}^{\pi}_{h+1}(s'_{h,k}, a)\pi(a \mid s'_{h,k})\,\mathrm{d}a \\
&- \mathbb{E}_{s' \sim P_h(\cdot|s_{h,k}, a_{h,k})}\left[\int_{\mathcal{A}} \widehat{Q}^{\pi}_{h+1}(s', a)\pi(a \mid s')\,\mathrm{d}a \mid s_{h,k}, a_{h,k}, \mathcal{D}_{h+1}, \cdots, \mathcal{D}_H\right] \mid \mathcal{D}_{h+1}, \cdots, \mathcal{D}_H\Bigg] \\
=&\, 0 + 0 = 0.
\end{aligned}
$$

On the other hand, $\left\|\widehat{Q}^{\pi}_{h+1}\right\|_{\infty} \leq H$ almost surely, because it is a function in our CNN class $\mathcal{F}$. Thus, $\zeta_{h,k}$ is a bounded random variable with $\zeta_{h,k} \in [-2H, 2H]$ almost surely, so its variance is bounded by $4H^2$. Its boundedness also implies it is a subgaussian random variable. Thus, Condition 3 is also satisfied.

Hence, Lemma 10 proves, for step h in our algorithm,

$$
\begin{aligned}
&\mathbb{E}\left[\int_{\mathcal{X}} \left(\mathcal{T}^{\pi}_h \widehat{Q}^{\pi}_{h+1} - \widehat{\mathcal{T}}^{\pi}_h\left(\widehat{Q}^{\pi}_{h+1}\right)\right)^2 (s, a)\,\mathrm{d}q_h^{\pi_0}(s, a) \mid \mathcal{D}_{h+1}, \cdots, \mathcal{D}_H\right] \\
&\leq C'(H^2 + 4H^2) K^{-\frac{2\alpha}{2\alpha+d}} \log^5 K,
\end{aligned}
$$

where C' depends on $D^{\frac{6\alpha}{2\alpha+d}}$, d, α, $\frac{2d}{\alpha p - d}$, p, q, c_0, B, ω and the surface area of $\mathcal{X}$.

Note that this upper bound holds for any $\widehat{Q}^{\pi}_{h+1}$ or $\mathcal{D}_{h+1}, \cdots, \mathcal{D}_H$. The sole purpose of our conditioning is that we could view $\widehat{Q}^{\pi}_{h+1}$ as a measurable or deterministic function under the conditioning and then apply Lemma 10. Therefore,

$$
\begin{aligned}
&\mathbb{E}\left[\mathbb{E}\left[\int_{\mathcal{X}} \left(\mathcal{T}^{\pi}_h \widehat{Q}^{\pi}_{h+1} - \widehat{\mathcal{T}}^{\pi}_h\left(\widehat{Q}^{\pi}_{h+1}\right)\right)^2 (s, a)\,\mathrm{d}q_h^{\pi_0}(s, a) \mid \mathcal{D}_{h+1}, \cdots, \mathcal{D}_H\right]\right] \\
&\leq C'(H^2 + 4H^2) K^{-\frac{2\alpha}{2\alpha+d}} \log^5 K.
\end{aligned}
$$

Finally, we carry out the recursion from time step 1 to time step H, and the final result is

$$
\mathbb{E}\left|v^{\pi} - \widehat{v}^{\pi}\right| \leq CH^2 K^{-\frac{\alpha}{2\alpha+d}} \log^{5/2} K \left(\frac{1}{H}\sum_{h=1}^{H} \sqrt{\chi^2_{\mathcal{Q}}(q_h^{\pi}, q_h^{\pi_0}) + 1}\right).
$$

□

C PROOF OF THEOREM 2

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

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

as

$\mathcal{F}^{\mathrm{SCNN}}(L, J, I, \tau_1, \tau_2) = \{f \mid f(x)$ in the form of (3) with L layers. The number of filters per block

$$\text{is bounded by } L;\ \text{filter size is bounded by } I;\ \text{the number of channels is bounded by } J; \max_{m,l} \|\mathcal{W}_m^{(l)}\|_\infty \vee \|B_m^{(l)}\|_\infty \leq \tau_1, \|W\|_\infty \leq \tau_2 \big\}. \tag{11}$$

We will refer to CNNs in this form as "single-block CNNs" and use them as building blocks of our final CNN approximation for the ground truth Besov function.

C.1 PROOF OVERVIEW OF THEOREM 2

Theorem 2 serves as a building block for Theorem 1, which establishes the relation between network architecture and approximation error. It is proven in the following steps:

STEP 1: DECOMPOSE f AS SUM OF LOCALLY SUPPORTED FUNCTIONS OVER MANIFOLD

Since manifold $\mathcal{X}$ is assumed compact (Assumption 1), we can cover it with a finite set of D-dimensional open Euclidean balls $\{B_\beta(\mathbf{c}_i)\}_{i=1}^{C_\mathcal{X}}$, where $\mathbf{c}_i$ denotes the center of the i-th ball and β is its radius. We choose $\beta < \omega/2$, and define $U_i = B_\beta(\mathbf{c}_i) \cap \mathcal{X}$. Note that each U_i is diffeomorphic to an open subset of $\mathbb{R}^d$ (Lemma 5.4 in Niyogi et al. [40]); moreover, $\{U_i\}_{i=1}^{C_\mathcal{X}}$ forms an open cover for $\mathcal{X}$. There exists a carefully designed open cover with cardinality $C_\mathcal{X} \leq \lceil \frac{A(\mathcal{X})}{\beta^d} T_d \rceil$, where $A(\mathcal{X})$ denotes the surface area of $\mathcal{X}$ and T_d denotes the thickness of U_i's, i.e. the average number of U_i's that contain a given point on $\mathcal{X}$. T_d is $O(d \log d)$ (Conway et al. [5]).

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

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

where $a_i \in \mathbb{R}$ is the 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 $T_{\mathbf{c}_i}(\mathcal{X})$. Overall, the atlas $\{(\phi_i, U_i)\}_{i=1}^{C_\mathcal{X}}$ transforms each local neighborhood on the manifold to a d-dimensional cube.

Thus, we can decompose f_0 using this atlas as

$$f_0 = \sum_{i=1}^{C_\mathcal{X}} f_i \quad \text{with} \quad f_i = f \rho_i, \tag{12}$$

because there exists such a C^∞ partition of unity $\{\rho_i\}_{i=1}^{C_\mathcal{X}}$ with $\operatorname{supp}(\phi_i) \subset U_i$ (Proposition 1 in Liu et al. [32]). Since each f_i is only supported on U_i, we can further write

$$f_0 = \sum_{i=1}^{C_\mathcal{X}} \left(f_i \circ \phi_i^{-1}\right) \circ \phi_i \times \mathbb{1}_{U_i} \quad \text{with} \quad f_i = f \rho_i, \tag{13}$$

where $\mathbb{1}_{U_i}$ is the indicator for membership in U_i.

Lastly, we extend $f_i \circ \phi_i^{-1}$ to entire $[0,1]^d$ with 0, which is a function in $\mathcal{B}_{p,q}^\alpha([0,1]^d)$ with $\mathcal{B}_{p,q}^\alpha([0,1]^d)$ Besov norm at most Cc_0 (Lemma 4 in Liu et al. [32]), where C is a constant depending on α, p, q and d. This extended function is to be approximated with cardinal B-splines in the next step.

STEP 2: APPROXIMATE EACH LOCAL FUNCTION WITH CARDINAL B-SPLINES

With most things connected with the intrinsic dimension d in the last step, we proceed an approximation of f_0 on the low-dimensional manifold. With $\alpha \geq d/p + 1$ assumed in Assumption 2, we can invoke a classic result of using cardinal B-splines to approximate Besov functions (Lemma 5), by setting $r = +\infty$ and $m = \lceil \alpha \rceil + 1$ in the lemma. It states that there exists a weighted sum of cardinal B-splines $\widetilde{f}_i$ in the form

$$\widetilde{f}_i \equiv \sum_{j=1}^{N} \widetilde{f}_{i,j} \approx f_i \circ \phi_i^{-1} \ \text{ with } \ \widetilde{f}_{i,j} = c_{k,\mathbf{j}}^{(i)} \widetilde{g}_{k,\mathbf{j},m}^d \tag{14}$$

such that

$$\left\| \widetilde{f}_i - f_i \circ \phi_i^{-1} \right\|_{L^\infty} \le C c_0 N^{-\alpha/d}. \tag{15}$$

In (14), $c^{(i)}_{k,\mathbf{j}} \in \mathbb{R}$ is coefficient and $\widetilde{g}^d_{k,\mathbf{j},m} : [0,1]^d \to \mathbb{R}$ denotes a cardinal B-spline with index $k, m \in \mathbb{N}^+, \mathbf{j} \in \mathbb{R}^d$. k is a scaling factor, $\mathbf{j}$ is a shifting vector, m is the degree of the B-spline.

By (13) and (14), we now have a sum of cardinal B-splines

$$\widetilde{f} \equiv \sum_{i=1}^{C_\mathcal{X}} \widetilde{f}_i \circ \phi_i \times \mathbb{1}_{U_i} = \sum_{i=1}^{C_\mathcal{X}} \sum_{j=1}^{N} \widetilde{f}_{i,j} \circ \phi_i \times \mathbb{1}_{U_i}. \tag{16}$$

which can approximate our target Besov function f_0 with error

$$\left\| \widetilde{f} - f_0 \right\|_{L^\infty} \le C C_\mathcal{X} c_0 N^{-\alpha/d}. \tag{17}$$

STEP 3: APPROXIMATE EACH CARDINAL B-SPLINE WITH A COMPOSITION OF CNNS

Each summand in (16) is a composition of functions, each of which we can implement with a CNN. Specifically, we do so with a special class of CNNs defined in (11), which we refer to as ”single-block CNNs”.

The multiplication operation $\times$ can be approximated by a single-block CNN $\widehat{\times}$ with at most η error in the L^∞ sense (Proposition 1). $\widehat{\times}$ needs $O(\log \frac{1}{\eta})$ layers and 6 channels. All weight parameters are bounded by $(c_0^2 \vee 1)$.

We consider each $\widetilde{f}_i \circ \phi_i$ together, which we can approximate with a sum of N CNNs $\widehat{f}^{\mathrm{SCNN}}_{i,j} \circ \widehat{\phi}_i$ up to δ error, namely,

$$\left\| \sum_{j=1}^{N} \widehat{f}^{\mathrm{SCNN}}_{i,j} - \widetilde{f}_i \circ \phi_i^{-1} \right\|_{L^\infty} \le \delta.$$

In particular, we can use a single-block CNN $\widehat{f}^{\mathrm{SCNN}}_{i,j}$ to approximate the B-spline $\widetilde{f}_{i,j}$ up to δ/N error. Moreover, since ϕ_i is linear, it can be expressed with a single-layer perceptron $\widehat{\phi}_i$. The architecture and size of $\widehat{f}^{\mathrm{SCNN}}_{i,j}$ and $\widehat{\phi}_i$ are characterized in Proposition 2 as functions of δ.

$\mathbb{1}_{U_i}$ is an indicator for membership in U_i, so we need $\mathbb{1}_{U_i}(x) = 1$ if $d_i^2(x) = \|x - \mathbf{c}_i\|_2^2 \le \beta^2$ and $\mathbb{1}_{U_i}(x) = 0$ 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:

$$\mathbb{1}_{U_i}(x) = \mathbb{1}_{[0,\beta^2]} \circ d_i^2(x) \quad \text{for} \quad x \in \mathcal{X}. \tag{18}$$

Given $\theta \in (0,1)$ and $\Delta \ge 8DB^2\theta$, 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 (Proposition 3) such that

$$\left\| \widehat{d_i^2} - d_i^2 \right\|_{L^\infty} \le 4B^2 D\theta \tag{19}$$

and

$$\widehat{\mathbb{1}}_\Delta \circ \widehat{d_i^2}(x) = \begin{cases} 1, \text{ if } x \in U_i, d_i^2(x) \le \beta^2 - \Delta, \\ 0, \text{ if } x \notin U_i, \\ \text{some value between 0 and 1, otherwise.} \end{cases} \tag{20}$$

The architecture and size of $\widehat{\mathbb{1}}_\Delta$ and $\widehat{d_i^2}$ are characterized in Proposition 3 as functions of θ and Δ.

The above three approximations rely on the classic result of using CNN to approximate cardinal B-splines (Lemma 10 in Liu et al. [32]; Lemma 1 in Suzuki [45]). Putting the above together, we can develop a composition of single-block CNNs

$$\bar{f}_{i,j} \equiv \widehat{\times}\left(\widehat{f}^{\mathrm{SCNN}}_{i,j} \circ \widehat{\phi}_i, \widehat{\mathbb{1}}_\Delta \circ \widehat{d_i^2}\right) \tag{21}$$

as an approximation for $\widetilde{f}_{i,j} \circ \phi_i \times \mathbb{1}_{U_i}$. The overall approximation error of $\bar{f}_{i,j}$ can be written as a sum of the three types of approximation error above. Details are provided in Appendix C.2. Moreover, by Lemma 6, there exists a single-block CNN $\widehat{f}_{i,j}$ that can express $\bar{f}_{i,j}$.

STEP 4: EXPRESS THE SUM OF CNN COMPOSITIONS WITH A CNN

Finally, we can assemble everything into $\widehat{f}$

$$\widehat{f} \equiv \sum_{i=1}^{C_{\mathcal{X}}} \sum_{j=1}^{N} \widehat{f}_{i,j}, \tag{22}$$

which serves as an approximation for f_0. By choosing the appropriate network size in Lemma 2, which the tradeoff between the approximation error of $\widehat{f}_{i,j}$ and its size, we can ensure that

$$\left\| \widehat{f} - f_0 \right\|_{L^\infty} \le N^{-\alpha/d}. \tag{23}$$

By Lemma 7, for $\widetilde{M}, \widetilde{J} > 0$, we can write this sum of $N \cdot C_{\mathcal{X}}$ 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 2 to be more flexible with network architecture. By Lemma 4, this sum of $\widetilde{M}$ CNNs can be further expressed as one CNN in the CNN class (5). Finally, N will be chosen appropriately as a function of network architecture parameters, and the approximation theory of CNN is proven.

When Theorem 2 is applied in our problem setting, we will take the target function f above to be $\mathcal{T}_h^\pi \widehat{Q}_{h+1}^\pi$ at each time step h, which is the ground truth of the regression at each step of Algorithm 1. More details about the proof of Theorem 2 are in Appendix C.2.

C.2 PROOF OF THEOREM 2

In the following, we provide the proof details for Theorem 2, which quantifies the tradeoff between a CNN in the class of 11 and its approximation error for Besov functions on a low-dimensional manifold. We start from the decomposition of the approximation error of $\widehat{f}$, which is based on the decomposition of the approximation error of $\bar{f}_{i,j}$ in (21), and will proceed to the end of this proof.

Lemma 2. *Let η be the approximation error of the multiplication operator $\widehat{\times}(\cdot,\cdot)$ as defined in Step 3 of Appendix C.1 and Proposition 1, δ be defined as in Step 3 of Appendix C.1 and Proposition 2, Δ and θ be defined as in Step 3 of Appendix C.1 and Proposition 3. Assume N is chosen according to Proposition 2. For any $i = 1, ..., C_{\mathcal{X}}$, we have $\left\| \widehat{f} - f_0 \right\|_{L^\infty} \le \sum_{i=1}^{C_{\mathcal{X}}} (A_{i,1} + A_{i,2} + A_{i,3})$ with*

$$A_{i,1} = \sum_{j=1}^{N} \left\| \widehat{\times}(\widehat{f}_{i,j}^{\mathrm{SCNN}} \circ \widehat{\phi}_i, \widehat{\mathbb{1}}_\Delta \circ \widehat{d_i^2}) - \widehat{f}_{i,j}^{\mathrm{SCNN}} \circ \widehat{\phi}_i \times (\widehat{\mathbb{1}}_\Delta \circ \widehat{d_i^2}) \right\|_{L^\infty} \le C'' \delta^{-d/\alpha} \eta,$$

$$A_{i,2} = \left\| \left(\sum_{j=1}^{N} \left(\widehat{f}_{i,j}^{\mathrm{SCNN}} \circ \widehat{\phi}_i \right) \right) \times (\widehat{\mathbb{1}}_\Delta \circ \widehat{d_i^2}) - f_i \times (\widehat{\mathbb{1}}_\Delta \circ \widehat{d_i^2}) \right\|_{L^\infty} \le \delta,$$

$$A_{i,3} = \| f_i \times (\widehat{\mathbb{1}}_\Delta \circ \widehat{d_i^2}) - f_i \times \mathbb{1}_{U_i} \|_{L^\infty} \le \frac{c(\pi+1)}{\beta(1-\beta/\omega)} \Delta$$

for some constant C'' depending on d, α, p, q and some constant c. Furthermore, for any $\varepsilon \in (0,1)$, setting

$$\delta = \frac{N^{-\alpha/d}}{3C_{\mathcal{X}}}, \ \eta = \frac{1}{C''} \frac{N^{-1-\alpha/d}}{(3C_{\mathcal{X}})^{d/\alpha}}, \Delta = \frac{\beta(1-\beta/\omega) N^{-\alpha/d}}{3c(\pi+1) C_{\mathcal{X}}}, \ \theta = \frac{\Delta}{16 B^2 D} \tag{24}$$

gives rise to

$$\left\| \widehat{f} - f_0 \right\|_{L^\infty} \le N^{-\frac{\alpha}{d}}.$$

The choice in (24) satisfies the condition $\Delta > 8B^2 D\theta$ in Proposition 3.

Proof of Lemma 2. As in Proposition 1, $A_{i,1}$ measures the error from $\widehat{\times}$:

$$A_{i,1} = \sum_{j=1}^{N} \left\| \widehat{\times}(\widehat{f}_{i,j}^{\mathrm{SCNN}} \circ \widehat{\phi}_i, \widehat{\mathbb{1}}_\Delta \circ \widehat{d_i^2}) - \widehat{f}_{i,j}^{\mathrm{SCNN}} \circ \widehat{\phi}_i \times (\widehat{\mathbb{1}}_\Delta \circ \widehat{d_i^2}) \right\|_{L^\infty} \le N\eta \le C'' \delta^{-d/\alpha} \eta,$$

for some constant C'' depending on d, α, p, q. The last inequality is due to the choice of N in Proposition 2.

$A_{i,2}$ measures the error from CNN approximation of Besov functions. As in Proposition 2, $A_{i,2} \leq \delta$.

$A_{i,3}$ measures the error from CNN approximation of the chart determination function. The bound of $A_{i,3}$ can be derived using the proof of Lemma 4.5 in Chen et al. [4], since $f_i \circ \phi_i^{-1}$ is a Lipschitz function and its domain is in $[0,1]^d$. □

In order to attain the error desired in Lemma 2, we need each network in $\bar{f}_{i,j}$ with appropriate size. The network size of the components in $\bar{f}_{i,j}$ can be analyzed as follows:

- $\widehat{\mathbb{1}}_i$: The chart determination network $\widehat{\mathbb{1}}_i = \widehat{d_i^2} \circ \widehat{\mathbb{1}}_\Delta$ is the composition of $\widehat{d_i^2}$ and $\widehat{\mathbb{1}}_\Delta$. By Proposition 3, $\widehat{d_i^2}$ is a single-block CNN with $O(\log \frac{1}{\theta}) = O(\frac{\alpha}{d} \log N + D + \log D)$ layers and width $6D$; $\widehat{\mathbb{1}}_\Delta$ is a single-block CNN with $O(\log(\beta^2/\Delta)) = O(\frac{\alpha}{d} \log N)$ layers and width 2. In both subnetworks, all parameters are of $O(1)$. By Lemma 6, the chart determination network $\widehat{\mathbb{1}}_i$ is a single-block CNN with $O(\frac{\alpha}{d} \log N + D + \log D)$ layers, width $6D + 2$ and all weight parameters are of $O(1)$.
- $\widehat{\times}$: By Proposition 1, the multiplication network is a single-block CNN with $O(\log \frac{1}{\eta}) = O((1 + \frac{\alpha}{d}) \log N)$ layers and $O(1)$ width. All weight parameters are bounded by $(c_0^2 \vee 1)$.
- $\widehat{\phi}_i$: The projection ϕ_i is a linear one, so it can be expressed with a single-layer perceptron. By Lemma 8 in Liu et al. [32], this single-layer perceptron can be expressed with a single-block CNN with $2 + D$ layers and width d. All parameters are of $O(1)$.
- $\widehat{f}_{i,j}^{\mathrm{SCNN}}$: by Proposition 2, each $\widehat{f}_{i,j}^{\mathrm{SCNN}}$ is a single-block CNN with $O(\log \frac{1}{\delta}) = O(\frac{\alpha}{d} \log N)$ layers and $\lceil 24d(\alpha + 1)(\alpha + 3) + 8d \rceil$ channels. All weight parameters are in the order of $O\left(\delta^{-(\log 2)(\frac{2d}{\alpha p - d} + c_1 d^{-1})}\right) = O\left(N^{(\log 2)\frac{\alpha}{d}(\frac{2d}{\alpha p - d} + c_1 d^{-1})}\right)$.

Next, we want to show $\bar{f}_{i,j}$, a composition of the aforementioned single-block CNNs, can be simply expressed as a single-block CNN.

By Lemma 6, there exists a single-block CNN $g_{i,j}$ with $O(\log N + D)$ layers and $\lceil 24d(\alpha + 1)(\alpha + 3) + 9d \rceil$ width realizing $\widehat{f}_{i,j}^{\mathrm{SCNN}} \circ \widehat{\phi}_i$. All weight parameters in $g_{i,j}$ are in the order of $O\left(N^{(\log 2)\frac{\alpha}{d}(\frac{2d}{\alpha p - d} + c_1 d^{-1})}\right)$. 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 width $6D + 2$, whose weight parameters are of $O(1)$. By Lemma 14 in Liu et al. [32], one can construct a convolutional block, denoted by $\bar{g}_{i,j}$, such that

$$\bar{g}_{i,j}(x) = \begin{bmatrix} (g_{i,j}(x))_+ & (g_{i,j}(x))_- & (\widehat{\mathbb{1}}_i(x))_+ & (\widehat{\mathbb{1}}_i(x))_- \\ \star & \star & \star & \star \end{bmatrix}. \tag{25}$$

Here $\bar{g}_{i,j}$ has $\lceil 24d(\alpha + 1)(\alpha + 3) + 9d \rceil + 6D + 2$ channels.

Since the input of $\widehat{\times}$ is $\begin{bmatrix} g_{i,j} \\ \widehat{\mathbb{1}}_i \end{bmatrix}$, by Lemma 15 in Liu et al. [32], there exists a CNN $\mathring{g}_{i,j}$ which takes (25) as the input and outputs $\widehat{\times}(g_{i,j}, \widehat{\mathbb{1}}_i)$.

Note that $\bar{g}_{i,j}$ only contains convolutional layers. The composition $\mathring{g}_{i,j} \circ \bar{g}_{i,j}$, denoted by $\widehat{g}_{i,j}^{\mathrm{SCNN}}$, is a CNN and for any $x \in \mathcal{X}$, $\widehat{g}_{i,j}^{\mathrm{SCNN}}(x) = \bar{f}_{i,j}(x)$. We have $\widehat{g}_{i,j}^{\mathrm{SCNN}} \in \mathcal{F}^{\mathrm{SCNN}}(L, J, I, \tau, \tau)$ with

$$\begin{gathered} L = O\left(\log N + D + \log D\right), \ J = \lceil 48d(\alpha + 1)(\alpha + 3) + 18d \rceil + 12D + O(1), \\ \tau = O\left(N^{(\log 2)\frac{d}{\alpha}(\frac{2d}{\alpha p - d} + c_1 d^{-1})}\right), \end{gathered} \tag{26}$$

and I can be any integer in $[2, D]$.

Therefore, we have shown that $\widehat{g}_{i,j}^{\mathrm{SCNN}}$ is a single-block CNN that expresses $\bar{f}_{i,j}$, as we desired.

Furthermore, recall that $\widehat{f}$ can be written as a sum of $C_{\mathcal{X}} N$ such SCNNs. By Lemma 7, for any $\widetilde{M}, \widetilde{J}$ satisfying $\widetilde{M}\widetilde{J} = O(N)$, there exists a CNN architecture $\mathcal{F}^{\mathrm{SCNN}}(L, J, I, \tau, \tau)$ that gives rise to a set of single-block CNNs $\{\widehat{g}_i\}_{i=1}^{\widetilde{M}} \in \mathcal{F}^{\mathrm{SCNN}}(L, J, I, \tau, \tau)$ with

$$\widehat{f} = \sum_{i=1}^{\widetilde{M}} \widehat{g}_i \tag{27}$$

and

$$L = O\left(\log N + D + \log D\right), \ J = O(D\widetilde{J}), \ \tau = O\left(N^{(\log 2)\frac{d}{\alpha}(\frac{2d}{\alpha p - d} + c_1 d^{-1})}\right). \tag{28}$$

By Lemma 3 below, we slightly adjust the CNN architecture by re-balancing the weight parameter boundary 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. While this procedure does not change the approximation power of the CNN, it can make the CNN have a smaller covering number, which is conducive to a smaller variance.

Lemma 3 (Lemma 16 in Liu et al. [32])**.** *Let $\gamma \geq 1$. For any $g \in \mathcal{F}^{\mathrm{SCNN}}(L, J, I, \tau_1, \tau_2)$, there exists $f \in \mathcal{F}^{\mathrm{SCNN}}(L, J, I, \gamma^{-1}\tau, \gamma^L \tau)$ such that $g(x) = f(x)$.*

In this case, we set $\gamma = c' N^{(\log 2)\frac{d}{\alpha}(\frac{2d}{\alpha p - d} + c_1 d^{-1})}(8ID)\widetilde{M}^{\frac{1}{L}}$, where c' is a constant such that $\tau \leq c' N^{(\log 2)\frac{d}{\alpha}(\frac{2d}{\alpha p - d} + c_1 d^{-1})}$. With this γ, we have $\widehat{f}_i \in \mathcal{F}^{\mathrm{SCNN}}(L, J, I, \tau_1, \tau_2)$ with

$$\begin{gathered} L = O(\log N + D + \log D), \ J = O(D), \ \tau_1 = (8ID)^{-1}\widetilde{M}^{-\frac{1}{L}} = O(1), \\ \log \tau_2 = O\left(\log \widetilde{M} + \log^2 N + D \log N\right). \end{gathered}$$

Finally, we prove that it suffices to use one CNN to realize the sum of single-block CNNs in (27).

Lemma 4. *Let $\mathcal{F}^{\mathrm{SCNN}}(L, J, I, \tau_1, \tau_2)$ be any CNN architecture from $\mathbb{R}^D$ to $\mathbb{R}$. Assume the weight matrix in the fully connected layer of $\mathcal{F}^{\mathrm{SCNN}}(L, J, I, \tau_1, \tau_2)$ has nonzero entries only in the first row. For any positive integer M, there exists a ConvNet architecture $\mathcal{F}(M, L, J, I, \tau_1, \tau_2(1 \vee \tau_1^{-1}))$ such that for any $\{\widehat{f}_i(x)\}_{i=1}^{M} \subset \mathcal{F}^{\mathrm{SCNN}}(L, J, I, \tau_1, \tau_2)$, there exists $\widehat{f} \in \mathcal{F}(M, L, 4 + J, I, \tau_1, \tau_2(1 \vee \tau_1^{-1}))$ with*

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

Consequently, by Lemma 4, there exists a ConvNet that can express our sum of $\widetilde{M}$ single-block CNNs with architecture $\mathcal{F}(M, L, J, I, \tau_1, \tau_2)$ with

$$\begin{gathered} L = O(\log N + D + \log D), \ J = O(D\widetilde{J}), \ \tau_1 = (8ID)^{-1}\widetilde{M}^{-\frac{1}{L}} = O(1), \\ \log \tau_2 = O\left(\log \widetilde{M} + \log^2 N + D \log N\right), \ M = O(\widetilde{M}). \end{gathered} \tag{29}$$

and $\widetilde{J}, \widetilde{M}$ satisfying

$$\widetilde{M}\widetilde{J} = O(N), \tag{30}$$

which is a requirement inherited from Lemma 7. This CNN is our final approximation for f_0.

Applying this relation $N = O(\widetilde{M}\widetilde{J})$ to (29) gives

$$\left\| \widehat{f} - f_0 \right\|_{L^\infty} \leq (\widetilde{M}\widetilde{J})^{-\frac{\alpha}{d}} \tag{31}$$

and the network size

$$\begin{gathered} L = O\left(\log(\widetilde{M}\widetilde{J}) + D + \log D\right), \ J = O(D\widetilde{J}), \ \tau_1 = (8ID)^{-1}\widetilde{M}^{-\frac{1}{L}} = O(1), \\ \log \tau_2 = O\left(\log^2 \widetilde{M}\widetilde{J} + D \log \widetilde{M}\widetilde{J}\right), \ M = O(\widetilde{M}). \end{gathered}$$

C.3 PROOF OF LEMMA 4

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 $\mathcal{W}_m = \left\{\mathcal{W}_m^{(l)}\right\}_{l=1}^{L}, \mathcal{B}_m = \left\{B_m^{(l)}\right\}_{l=1}^{L}$. 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

$$Z = \left[(x)_+ \quad (x)_- \quad \star \quad \star\right], \tag{32}$$

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

$$\mathrm{Conv}_{\widehat{\mathcal{W}}_m,\widehat{\mathcal{B}}_m}(Z) = \begin{bmatrix} & & \frac{\tau_1}{\tau_2}(f_m(\boldsymbol{x}) \vee 0) & -\frac{\tau_1}{\tau_2}(f_m(\boldsymbol{x}) \wedge 0) \\ \mathbf{0} & \mathbf{0} & \star & \star \\ & & \vdots & \vdots \\ & & \star & \star \end{bmatrix} + Z \tag{33}$$

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

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

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 is a copy of Z. For the filters in later layers of $\widehat{f}_m$ and all biases, we simply set

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

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

$$\begin{aligned}
(\widehat{\mathcal{W}}_m^{(L)})_{3,:,:} &= \left[\mathbf{0} \quad \mathbf{0} \quad \mathbf{e}_1 \quad \mathbf{0} \quad \tfrac{\tau_1}{\tau_2}(W_m)_{1,:}\right], \\
(\widehat{\mathcal{W}}_m^{(L)})_{4,:,:} &= \left[\mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{e}_1 \quad -\tfrac{\tau_1}{\tau_2}(W_m)_{1,:}\right].
\end{aligned}$$

Under such choices, (33) is proved and all parameters in $\widehat{\mathcal{W}}_m, \widehat{\mathcal{B}}_m$ are bounded by τ_1.

By composing all convolutional blocks, we have

$$(\mathrm{Conv}_{\widehat{\mathcal{W}}_M, \widehat{\mathcal{B}}_M}) \circ \cdots \circ (\mathrm{Conv}_{\widehat{\mathcal{W}}_1, \widehat{\mathcal{B}}_1}) \circ P(x) = \begin{bmatrix} & & \frac{\tau_1}{\tau_2} \sum_{m=1}^{M} (\widehat{f}_m \vee 0) & -\frac{\tau_1}{\tau_2} \sum_{m=1}^{M} (\widehat{f}_m \wedge 0) \\ (x)_+ & (x)_- & \star & \star \\ & & \vdots & \vdots \\ & & \star & \star \end{bmatrix}.$$

Lastly, the fully connect layer can be set as

$$\widetilde{W} = \begin{bmatrix} 0 & 0 & \frac{\tau_2}{\tau_1} & -\frac{\tau_2}{\tau_1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \ \widetilde{b} = 0.$$

Note that the weights in the fully connected layer are bounded by $\tau_2(1 \vee \tau_1^{-1})$.

The above construction gives

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

C.4 SUPPORTING LEMMAE FOR THEOREM 2

Before stating Lemma 5, we provide a brief definition of cardinal B-splines.

Definition 5 (Cardinal B-spline)**.** *Let* $\psi(x) = \mathbb{1}_{[0,1]}(x)$ *be the indicator function for membership in* $[0, 1]$. *The cardinal B-spline of order* m *is defined by taking* $m + 1$*-times convolution of* ψ*:*

$$\psi_m(x) = (\underbrace{\psi * \psi * \cdots * \psi}_{m+1 \text{ times}})(x)$$

where $f * g(x) \equiv \int f(x - t)g(t)dt$.

Note that ψ_m is a piecewise polynomial with degree m and support $[0, m + 1]$. It can be expressed as [34]

$$\psi_m(x) = \frac{1}{m!} \sum_{j=0}^{m+1} (-1)^j \binom{m+1}{j} (x - j)_+^m.$$

For any $k, j \in \mathbb{N}$, let $\widetilde{g}_{k,j,m}(x) = \psi_m(2^k x - j)$, which is the rescaled and shifted cardinal B-spline with resolution 2^{-k} and support $2^{-k}[j, j + (m + 1)]$. For $\mathbf{k} = (k_1, \ldots, k_d) \in \mathbb{N}^d$ and $\mathbf{j} = (j_1, \ldots, j_d) \in \mathbb{N}^d$, we define the d dimensional cardinal B-spline as $\widetilde{g}^d_{\mathbf{k},\mathbf{j},m}(x) = \prod_{i=1}^{d} \psi_m(2^{k_i} x_i - j_i)$. When $k_1 = \ldots = k_d = k \in \mathbb{N}$, we denote $\widetilde{g}^d_{k,\mathbf{j},m}(x) = \prod_{i=1}^{d} \psi_m(2^k x_i - j_i)$.

C.4.1 APPROXIMATING BESOV FUNCTIONS WITH CARDINAL B-SPLINES

For any $m \in \mathbb{N}$, let $J(k) = \{-m, -m + 1, \ldots, 2^k - 1, 2^k\}^d$ and the quasi-norm of the coefficient $\{c_{k,j}\}$ for $k \in \mathbb{N}, \mathbf{j} \in J(k)$ be

$$\|\{c_{k,\mathbf{j}}\}\|_{b^{\alpha}_{p,q}} = \left(\sum_{k \in \mathbb{N}} \left[2^{k(\alpha - d/p)} \left(\sum_{\mathbf{j} \in J(k)} |c_{k,\mathbf{j}}|^p \right)^{1/p} \right]^q \right)^{1/q}. \tag{34}$$

We can state the following lemma, from DeVore & Popov [8], Dung [12], which provides an upper bound on the error of using cardinal B-splines to approximate functions in $\mathcal{B}^{\alpha}_{p,q}([0, 1]^d)$.

Lemma 5 (Lemma 2 in Suzuki [45]; DeVore & Popov [8], Dung [12])**.** Assume that $0 < p, q, r \leq \infty$ and $0 < \alpha < \infty$ satisfying $\alpha > d(1/p - 1/r)_+$. Let $m \in \mathbb{N}$ be the order of the cardinal B-spline basis such that $0 < \alpha < \min(m, m - 1 + 1/p)$. For any $f \in \mathcal{B}^{\alpha}_{p,q}([0, 1]^d)$, there exists f_N satisfying

$$\|f - f_N\|_{L^r([0,1]^d)} \leq C N^{-\alpha/d} \|f\|_{\mathcal{B}^{\alpha}_{p,q}([0,1]^d)}$$

for some constant C with $N \gg 1$. f is in the form of

$$f_N(x) = \sum_{k=0}^{H} \sum_{\mathbf{j} \in J(k)} c_{k,\mathbf{j}} \widetilde{g}^d_{k,\mathbf{j},m}(x) + \sum_{k=K+1}^{H^*} \sum_{i=1}^{n_k} c_{k,\mathbf{j}_i} \widetilde{g}^d_{k,\mathbf{j}_i,m}(x), \tag{35}$$

where $\{\mathbf{j}_i\}_{i=1}^{n_k} \subset J(k)$, $H = \lceil c_1 \log(N)/d \rceil$, $H^* = \lceil \nu^{-1} \log(\lambda N) \rceil + H + 1$, $n_k = \lceil \lambda N 2^{-\nu(k-H)} \rceil$ for $k = H+1, \ldots, H^*$, $u = d(1/p - 1/r)_+$ and $\nu = (\alpha - u)/(2u)$. The real numbers $c_1 > 0$ and $\lambda > 0$ are two absolute constants chosen to satisfy $\sum_{k=1}^{H} (2^k + m)^d + \sum_{k=H+1}^{H^*} n_k \leq N$, which are to N. Moreover, we can choose the coefficients $\{c_{k,\mathbf{j}}\}$ such that

$$\|\{c_{k,\mathbf{j}}\}\|_{b^{\alpha}_{p,q}} \leq C_1 \|f\|_{\mathcal{B}^{\alpha}_{p,q}([0,1]^d)}$$

for some constant C_1.

C.4.2 APPROXIMATING CARDINAL B-SPLINES AND OTHERS WITH SINGLE-BLOCK CNNS

The following Proposition 1 quantifies the tradeoff between the size of a single-block CNN and its approximation error for the multiplication operator.

Proposition 1. *Let $\times$ be defined as in (13). For any $\eta \in (0,1)$, there exists a single-block CNN $\widehat{\times}(\cdot,\cdot)$ such that*

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

where a, b are functions uniformly bounded by c_0.

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

Proof of Proposition 1. First, let us define a particular class of feed-forward ReLU networks of the form

$$f(x) = W_L \cdot \mathrm{ReLU}(W_{L-1} \cdots \mathrm{ReLU}(W_1 x + b_1) \cdots + b_{L-1}) + b_L, \tag{36}$$

as

$$\begin{aligned} \mathcal{F}(L, J, \tau) = \{ f \mid f(x) \text{ in the form (36) with } L \text{ layers and width at most } J, \\ \|W_i\|_{\infty,\infty} \leq \tau, \ \|b_i\|_\infty \leq \tau \text{ for } i = 1, \cdots, L \}. \end{aligned} \tag{37}$$

By Proposition 3 in Yarotsky, there exists a feed-forward ReLU network that can approximate the multiplication operation between values with magnitude bounded by c_0, with η error. Such feed-forward network has $O(\log 1/\eta)$ layers, whose width is all bounded by 6, and all its parameters are bounded by c_0^2. Therefore, such a feed-forward network is sufficient to approximate $\times$ with η error in L^∞-norm, because the arguments of $\times$ are uniformly bounded c_0 by Assumption 2.

Furthermore, by Lemma 8 in Liu et al. [32], we can express the aforementioned feed-forward network with a single-block CNN in $\mathcal{F}^{\mathrm{SCNN}}(L, J, I, \tau, \tau)$, where L, J, I, τ are as specified in the statement of the proposition. ☐

Proposition 2 quantifies the tradeoff between the size of a single-block CNN and its approximation error for the cardinal B-spline $f_i \circ \phi_i^{-1}$.

Proposition 2 (Proposition 3 in Liu et al. [32])**.** *Let $f_i \circ \phi_i^{-1}$ be defined as in (13). For any $\delta \in (0,1)$, set $N = C_1 \delta^{-d/\alpha}$. For any $2 \leq I \leq d$, there exists a set of single-block CNNs $\left\{ \widehat{f}^{\mathrm{SCNN}} \right\}_{j=1}^{N}$ such that*

$$\left\| \sum_{j=1}^{N} \widehat{f}^{\mathrm{SCNN}}_{i,j} - f_i \circ \phi_i^{-1} \right\|_{L^\infty} \leq \delta,$$

where C_1 is a constant depending on α, p, q and d.

$\widehat{f}_{i,j}^{\mathrm{SCNN}}$ is a single-block CNN approximation of $\widetilde{f}_{i,j}$ (defined in (14)) in $\mathcal{F}^{\mathrm{SCNN}}(L, J, I, \tau, \tau)$ with

$$L = O\left(\log(1/\delta)\right), J = \lceil 24d(\alpha+1)(\alpha+3) + 8d \rceil, \tau = O\left(\delta^{-(\log 2)(\frac{2d}{\alpha p - d} + c_1 d^{-1})}\right).$$

The constant hidden in $O(\cdot)$ depends on $d, \alpha, \frac{2d}{\alpha p - d}, p, q, c_0$.

Proposition 3 quantifies the tradeoff between the size of the sub-networks for the chart determination network and its approximation error for the chart determination indicators and the distance function d_i^2.

Proposition 3 (Lemma 9 in Liu et al. [32])**.** *Let d_i^2 and $\mathbb{1}_{[0,\beta^2]}$ be defined as in (18). For any $\theta \in (0,1)$ and $\Delta \geq 8B^2 D\theta$, there exists a single-block CNN $\widehat{d_i^2}$ approximating d_i^2 such that*

$$\|\widehat{d_i^2} - d_i^2\|_{L^\infty} \leq 4B^2 D\theta,$$

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

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

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

As a result, for any $x \in \mathcal{X}$, $\widehat{\mathbb{1}}_\Delta \circ \widehat{d_i^2}(x)$ gives an approximation of $\mathbb{1}_{U_i}$ satisfying

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

C.4.3 LEMMAE ABOUT SUMMATION AND COMPOSITION OF CNN

Lemma 6 states that the composition of two single-block CNNs can be expressed as one single-block CNN with augmented architecture.

Lemma 6. *Let $\mathcal{F}_1^{\mathrm{SCNN}}(L_1, J_1, I_1, \tau_1, \tau_1)$ be a CNN architecture from $\mathbb{R}^D \to \mathbb{R}$ and $\mathcal{F}_2^{\mathrm{SCNN}}(L_2, J_2, I_2, \tau_2, \tau_2)$ be a CNN architecture from $\mathbb{R} \to \mathbb{R}$. Assume the weight matrix in the fully connected layer of $\mathcal{F}_1^{\mathrm{SCNN}}(L_1, J_1, I_1, \tau_1, \tau_1)$ and $\mathcal{F}_2^{\mathrm{SCNN}}(L_2, J_2, I_2, \tau_2, \tau_2)$ has nonzero entries only in the first row. Then there exists a CNN architecture $\mathcal{F}^{\mathrm{SCNN}}(L, J, I, \tau, \tau)$ from $\mathbb{R}^D \to \mathbb{R}$ with*

$$L = L_1 + L_2,\ J = \max(J_1, J_2),\ I = \max(I_1, I_2), \tau = \max(\tau_1, \tau_2)$$

such that for any $f_1 \in \mathcal{F}^{\mathrm{SCNN}}(L_1, J_1, I_1, \tau_1, \tau_1)$ and $f_2 \in \mathcal{F}^{\mathrm{SCNN}}(L_2, J_2, I_2, \tau_2, \tau_2)$, there exists $f \in \mathcal{F}^{\mathrm{SCNN}}(L, J, I, \tau, \tau)$ such that $f(x) = f_2 \circ f_1(x)$. Furthermore, the weight matrix in the fully connected layer of $\mathcal{F}^{\mathrm{SCNN}}(L, J, I, \tau, \tau)$ has nonzero entries only in the first row.

Lemma 7 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 7 (Lemma 7 in Liu et al. [33])**.** *Let $\{f_i\}_{i=1}^{n_0}$ be a set of single-block CNNs with architecture $\mathcal{F}^{\mathrm{SCNN}}(L_0, J_0, I_0, \tau_0, \tau_0)$. For any integers $1 \leq n \leq n_0$ and $\widetilde{J}$ satisfying $n\widetilde{J} = O(n_0 J_0)$ and $\widetilde{J} \geq J_0$, there exists an architecture $\mathcal{F}^{\mathrm{SCNN}}(L, J, I, \tau, \tau)$ 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

$$L = O(L_0), J = O(\widetilde{J}), I = I_0, \tau = \tau_0.$$

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

D PROOF OF CONVNET CLASS COVERING NUMBER

In this section, we prove a bound on the covering number of the convolutional neural network class used in Algorithm 1.

Lemma 8. *Given $\delta > 0$, the δ-covering number of the neural network class $\mathcal{F}(M, L, J, I, \tau_1, \tau_2, V)$ satisfies*

$$\mathcal{N}(\delta, \mathcal{F}(M, L, J, I, \tau_1, \tau_2, V), \|\cdot\|_\infty) \leq \left(2(\tau_1 \vee \tau_2)\Lambda_1 \delta^{-1}\right)^{\Lambda_2}, \tag{38}$$

where

$$\Lambda_1 = (M+3)JD(1 \vee \tau_2)(1 \vee \tau_1)\widetilde{\rho}\widetilde{\rho}^+, \ \Lambda_2 = ML(J^2 I + J) + JD + 1$$

with $\widetilde{\rho} = \rho^M$, $\widetilde{\rho}^+ = 1 + ML\rho^+$, $\rho = (JI\tau_1)^L$ *and* $\rho^+ = (1 \vee JI\tau_1)^L$.

With a network architecture as stated in Theorem 2, we have

$$\log \mathcal{N}(\delta, \mathcal{F}(M, L, J, I, \tau_1, \tau_2, V) = O\left(\widetilde{M}\widetilde{J}^2 D^3 \log^5(\widetilde{M}\widetilde{J}) \log \frac{1}{\delta}\right),$$

where $O(\cdot)$ hides constant depending on d, α, $\frac{2d}{\alpha p - d}$, p, q, c_0, B, ω and the surface area of $\mathcal{X}$.

D.1 SUPPORTING LEMMAE AND PROOFS

Proposition 4 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. $Q_{[i,j]}$ is defined 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 4. *For $m = 1, 2, \cdots, M$ and $x \in [-1, 1]^D$, we have*

$$\left\|Q_{[1,m]}(x)\right\|_\infty \leq (1 \vee \tau_1)\left(\prod_{j=1}^{m}\prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} \tau_1\right)\left(1 + \sum_{k=1}^{m} L_k \prod_{i=1}^{L_k}(1 \vee J_k^{(i-1)} I_k^{(i)} \tau_1)\right).$$

Proof.

$$\begin{aligned}
&\left\|Q_{[1,m]}(x)\right\|_\infty \\
&= \left\|\mathrm{Conv}_{\mathcal{W}_m, \mathcal{B}_m}(Q_{[1,m-1]}(x))\right\|_\infty \\
&\leq \prod_{i=1}^{L_m} J_m^{(i-1)} I_m^{(i)} \tau_1 \left\|Q_{[1,m-1]}(x)\right\|_\infty + \tau_1 L_m \prod_{i=1}^{L_m}(1 \vee J_m^{(i-1)} I_m^{(i)} \tau_1) \\
&\leq \|P(x)\|_\infty \prod_{j=1}^{m}\prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} \tau_1 + \tau_1 \sum_{k=1}^{m} L_k \prod_{i=1}^{L_k}(1 \vee J_k^{(i-1)} I_k^{(i)} \tau_1) \prod_{l=j+1}^{m}\prod_{i=1}^{L_l} J_l^{(i-1)} I_l^{(i)} \tau_1 \\
&\leq \|x\|_\infty \prod_{j=1}^{m}\prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} \tau_1 + \tau_1 \sum_{k=1}^{m} L_k \prod_{i=1}^{L_k}(1 \vee J_k^{(i-1)} I_k^{(i)} \tau_1) \prod_{l=j+1}^{m}\prod_{i=1}^{L_l} J_l^{(i-1)} I_l^{(i)} \tau_1 \\
&\leq (1 \vee \tau_1)\left(\prod_{j=1}^{m}\prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} \tau_1\right)\left(1 + \sum_{k=1}^{m} L_k \prod_{i=1}^{L_k}(1 \vee J_k^{(i-1)} I_k^{(i)} \tau_1)\right),
\end{aligned}$$

where the first two inequalities are obtained by applying Proposition 9 from Oono & Suzuki [41] recursively. □

Lemma 9 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 9. *For $f, f' \in \mathcal{F}(M, L, J, I, \tau_1, \tau_2, V)$ such that for $\epsilon > 0$, $\|W - W'\|_\infty \leq \epsilon$, $\|b - b'\|_\infty \leq \epsilon$, $\left\|\mathcal{W}_m^{(l)} - \mathcal{W}_m^{(l)'}\right\|_\infty \leq \epsilon$ and $\left\|\mathcal{B}_m^{(l)} - \mathcal{B}_m^{(l)'}\right\|_\infty \leq \epsilon$ 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*

$$\|f - f'\|_\infty \leq \Lambda_1 \epsilon,$$

where Λ_1 is defined in Lemma 8.

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

$$\begin{aligned}
&|f(x) - f'(x)| \\
&= |W \otimes Q(x) + b - W' \otimes Q'(x) - b'| \\
&= |(W - W') \otimes Q(x) + b - b' + W' \otimes (Q(x) - Q'(x))| \\
&= |(W - W') \otimes Q(x) + b - b' + W' \otimes (Q(x) - \mathrm{Conv}_{\mathcal{W}_M, \mathcal{B}_M}(Q'(x)) + \mathrm{Conv}_{\mathcal{W}_M, \mathcal{B}_M}(Q'(x)) - Q'(x))| \\
&= \left|(W - W') \otimes Q(x) + b - b' + \sum_{m=1}^{M} W' \otimes Q_{[m+1,M]} \circ \left(\mathrm{Conv}_{\mathcal{W}_m, \mathcal{B}_m} - \mathrm{Conv}_{\mathcal{W}'_m, \mathcal{B}'_m}\right) \circ Q'_{[0,m-1]}\right| \\
&\leq |(W - W') \otimes Q(x;\theta) + b - b'| + \sum_{m=1}^{M} \left|W' \otimes Q_{[m+1,M]} \circ \left(\mathrm{Conv}_{\mathcal{W}_m, \mathcal{B}_m} - \mathrm{Conv}_{\mathcal{W}'_m, \mathcal{B}'_m}\right) \circ Q'_{[0,m-1]}\right| \\
&\overset{(a)}{\leq} (3 + M) JD (1 \vee \tau_1)(1 \vee \tau_2) \left(\prod_{j=1}^{M} \prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} \tau_1\right) \left(1 + \sum_{k=1}^{M} L_k \prod_{i=1}^{L_k} (1 \vee J_k^{(i-1)} I_k^{(i)} \tau_1)\right) \epsilon,
\end{aligned}$$

where (a) is obtained through the following reasoning.

The first term in (a) can be bounded as

$$\begin{aligned}
&|(W - W') \otimes Q(x) + b - b'| \\
&\leq (\|W\|_0 + \|W'\|_0) \|W - W'\|_\infty \|Q(x)\|_\infty + \|b - b'\|_\infty \\
&\leq 2JD\epsilon \|Q(x)\|_\infty + \epsilon \\
&\leq 3JD\epsilon \|Q(x)\|_\infty \\
&\leq 3JD \max\{1, \tau_1\} \left(\prod_{j=1}^{M} \prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} \tau_1\right) \left(1 + \sum_{k=1}^{M} L_k \prod_{i=1}^{L_k} (1 \vee J_k^{(i-1)} I_k^{(i)} \tau_1)\right) \epsilon,
\end{aligned}$$

where the first inequality uses Proposition 8 from Oono & Suzuki [41] and the last inequality is obtained by invoking Proposition 4.

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

$$\begin{aligned}
&\left|W' \otimes Q_{[m+1,M]} \circ \left(\mathrm{Conv}_{\mathcal{W}_m, \mathcal{B}_m} - \mathrm{Conv}_{\mathcal{W}'_m, \mathcal{B}'_m}\right) \circ Q'_{[1,m-1]}\right| \\
&\overset{(b)}{\leq} \|W'\|_0 \tau_2 \left\|Q_{[m+1,M]} \circ \left(\mathrm{Conv}_{\mathcal{W}_m, \mathcal{B}_m} - \mathrm{Conv}_{\mathcal{W}'_m, \mathcal{B}'_m}\right) \circ Q'_{[1,m-1]}\right\|_\infty \\
&\overset{(c)}{\leq} JD\tau_2 \left(\prod_{j=m+1}^{M} \prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} \tau_1\right) \left\|\left(\mathrm{Conv}_{\mathcal{W}_m, \mathcal{B}_m} - \mathrm{Conv}_{\mathcal{W}'_m, \mathcal{B}'_m}\right) \circ Q'_{[1,m-1]}\right\|_\infty \\
&\overset{(d)}{\leq} JD\tau_2 \left(\prod_{j=m+1}^{M} \prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} \tau_1\right) \left(\prod_{i=1}^{L_m} J_m^{(i-1)} I_m^{(i)} \tau_1 \left\|Q'_{[1,m-1]}\right\|_\infty \epsilon\right) \\
&\overset{(e)}{\leq} JD\tau_2 \left(\prod_{j=m+1}^{M} \prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} \tau_1\right) \left(\prod_{i=1}^{L_m} J_m^{(i-1)} I_m^{(i)} \tau_1\right)
\end{aligned}$$

$$
\begin{aligned}
&(1 \vee \tau_1) \left(\prod_{j=1}^{m} \prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} \tau_1 \right) \left(1 + \sum_{k=1}^{m} L_k \prod_{i=1}^{L_k} (1 \vee J_k^{(i-1)} I_k^{(i)} \tau_1) \right) \epsilon \\
\leq{}& J D \tau_2 \left(\prod_{j=1}^{M} \prod_{i=1}^{L_j} J_j^{(i-1)} I_j^{(i)} \tau_1 \right) (1 \vee \tau_1) \left(1 + \sum_{k=1}^{M} L_k \prod_{i=1}^{L_k} (1 \vee J_k^{(i-1)} I_k^{(i)} \tau_1) \right) \epsilon,
\end{aligned}
$$

where (b) is by Proposition 7 from Oono & Suzuki [41], (c) is by Proposition 2 and 4 from Oono & Suzuki [41], (d) is by Proposition 2 and 5 from Oono & Suzuki [41], and (e) is obtained by invoking Proposition 4. ☐

D.2 PROOF OF LEMMA 8

Proof of Lemma 8. We grid the range of each parameter into subsets with width $\Lambda_1^{-1}\delta$, so there are at most $2(\tau_1 \vee \tau_2)\Lambda_1\delta^{-1}$ different subsets for each parameter. In total, there are $\left(2(\tau_1 \vee \tau_2)\Lambda_1\delta^{-1}\right)^{\Lambda_2}$ bins in the grid. For any $f, f' \in \mathcal{F}(M, L, J, I, \tau_1, \tau_2, V)$ within the same grid, by Lemma 9, we have $\|f - f'\|_\infty \leq \delta$. We can construct the ϵ-covering with cardinality $\left(2(\tau_1 \vee \tau_2)\Lambda_1\delta^{-1}\right)^{\Lambda_2}$ by selecting one neural network from each bin in the grid.

Taking log and plugging in the network architecture parameters in Lemma 2, we have

$$
\begin{aligned}
\log \mathcal{N}(\delta, \mathcal{F}(M, L, J, I, \tau_1, \tau_2, V), \|\cdot\|_\infty) &= O\left(\Lambda_2 \log\left((\tau_1 \vee \tau_2)\, \Lambda_1 \delta^{-1}\right)\right) \\
&\leq O\left(\widetilde{M} D D^2 \widetilde{J}^2 \log(\widetilde{M}\widetilde{J}) \log^2(\widetilde{M}\widetilde{J}) \log^2(\widetilde{M}\widetilde{J}) \log \frac{1}{\delta} \right) \\
&= O\left(\widetilde{M}\widetilde{J}^2 D^3 \log^5(\widetilde{M}\widetilde{J}) \log \frac{1}{\delta} \right),
\end{aligned}
$$

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 τ_1, $\rho = (1/2)^L M^{-1} \leq M^{-1}$, so $\widetilde{\rho} = (1 + M^{-1})^M \leq e$. Moreover, $\widetilde{\rho}^+ = 1 + ML$. ☐

E STATISTICAL RESULT OF CNN-BESOV APPROXIMATION (LEMMA 10)

In this section, we derive the statistical estimation error for using a CNN empirical MSE minimizer to estimate a Besov 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 2 and variance. Thsi statistical estimation error can be decomposed into the error of using CNN to approximate Besov function (Theorem 2), terms that grow with the covering number of our CNN class, and the error of using the discrete covering to approximate our CNN class.

In Theorem 1, we expand the estimation error $\widehat{v}^\pi - v^\pi$ over time steps and upper-bound the amount of estimation error in each time step with Lemma 10. Details of Theorem 1 are in Appendix B.

Lemma 10. *Let $\mathcal{X}$ be a d-dimensional compact Riemannian manifold that satisfies Assumption 1. We are given a function $f_0 \in \mathcal{B}^\alpha_{p,q}(\mathcal{X})$, where s, p, q satisfies Assumption 2. We are also given samples $S_n = \{(x_i, y_i)\}_{i=1}^n$, where x_i are i.i.d. sampled from a distribution $\mathcal{P}_x$ on $\mathcal{X}$ and $y_i = f_0(x_i) + \zeta_i$. ζ_i's are i.i.d. sub-Gaussian random noise with variance σ^2, uncorrelated with x_i's. If we compute an estimator*

$$
\widehat{f}_n = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \left(f(x_i) - y_i\right)^2,
$$

with the neural network class $\mathcal{F} = \mathcal{F}(M, L, J, I, \tau_1, \tau_2, V)$ such that

$$
\begin{aligned}
L = O(\log n + D + \log D),\ J = O(D),\ \tau_1 = O(1),\ \log \tau_2 = O(\log^2 n + D \log n), \\
M = O(n^{\frac{d}{2\alpha + d}}),\ V = \|f_0\|_\infty, \qquad (39)
\end{aligned}
$$

with any integer $I \in [2, D]$ and $\widetilde{M}, \widetilde{J} > 0$ satisfying $\widetilde{M}\widetilde{J} = O(n^{\frac{d}{2\alpha + 2d}})$, then we have

$$
\mathbb{E}\left[\int_{\mathcal{X}} \left(\widehat{f}_n(x) - f_0(x) \right)^2 \mathrm{d}\mathcal{P}_x(x) \right] \leq c \left(V_{\mathcal{F}}^2 + \sigma^2 \right) n^{-\frac{2\alpha}{2\alpha + d}} \log^5 n, \qquad (40)
$$

where $V_{\mathcal{F}} = \|f_0\|_\infty$ *and the expectation is taken over the training sample* S_n*, and* c *is a constant depending on* $D^{\frac{6\alpha}{2\alpha+2d}}$*,* d*,* α*,* $\frac{2d}{\alpha p-d}$*,* p*,* q*,* c_0*,* B*,* ω *and the surface area of* $\mathcal{X}$*.* $O(\cdot)$ *hides constant depending on* d*,* α*,* $\frac{2d}{\alpha p-d}$*,* p*,* q*,* c_0*,* B*,* ω *and the surface area of* $\mathcal{X}$*.*

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

$$\mathbb{E}\left[\int_{\mathcal{X}}\left(\widehat{f}_n(x)-f_0(x)\right)^2 d\mathcal{D}_x(x)\right] = \underbrace{2\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(\widehat{f}_n(x_i)-f_0(x_i))^2\right]}_{T_1} + \underbrace{\mathbb{E}\left[\int_{\mathcal{X}}\left(\widehat{f}_n(x)-f_0(x)\right)^2 d\mathcal{D}_x(x)\right]-2\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(\widehat{f}_n(x_i)-f_0(x_i))^2\right]}_{T_2},$$

where T_1 reflects the squared bias of using neural networks to approximate ground truth f_0, which is related to Theorem 2, and T_2 is the variance term.

E.1 SUPPORTING LEMMAE

Lemma 11 (Lemma 5 in Chen et al. [4])**.** Fix the neural network class $\mathcal{F}(M,L,J,I,\tau_1,\tau_2,V)$. For any constant $\delta\in(0,2V)$, we have

$$\begin{aligned}T_1 \le 4 \inf_{f\in\mathcal{F}(M,L,J,I,\tau_1,\tau_2,V)} &\int_{\mathcal{X}}(f(x)-f_0(x))^2 d\mathcal{P}_x(x)\\ &+48\sigma^2\frac{\log\mathcal{N}(\delta,\mathcal{F}(M,L,J,I,\tau_1,\tau_2,V),\|\cdot\|_\infty)+2}{n}\\ &+(8\sqrt{6}\sqrt{\frac{\log\mathcal{N}(\delta,\mathcal{F}(M,L,J,I,\tau_1,\tau_2,V),\|\cdot\|_\infty)+2}{n}}+8)\sigma\delta,\end{aligned}$$

where $\mathcal{N}(\delta,\mathcal{F}(M,L,J,I,\tau_1,\tau_2,V),\|\cdot\|_\infty)$ denotes the δ-covering number of $\mathcal{F}(M,L,J,I,\tau_1,\tau_2,V)$ with respect to the ℓ_∞ norm, i.e., there exists a discretization of $\mathcal{F}(M,L,J,I,\tau_1,\tau_2,V)$ into $\mathcal{N}(\delta,\mathcal{F}(M,L,J,I,\tau_1,\tau_2,V),\|\cdot\|_\infty)$ distinct elements, such that for any $f\in\mathcal{F}$, there is $\bar{f}$ in the discretization satisfying $\left\|\bar{f}-f\right\|_\infty\le\epsilon$.

Lemma 12 (Lemma 6 in Chen et al. [4])**.** For any constant $\delta\in(0,2R)$, T_2 satisfies

$$T_2\le\frac{104V^2}{3n}\log\mathcal{N}(\delta/4V,\mathcal{F}(M,L,J,I,\tau_1,\tau_2,V),\|\cdot\|_\infty)+\left(4+\frac{1}{2V}\right)\delta.$$

E.2 PROOF OF LEMMA 10

Proof of Lemma 10. Recall that the bias and variance decomposition of $\mathbb{E}\left[\int_{\mathcal{X}}\left(\widehat{f}_n(x)-f_0(x)\right)^2 d\mathcal{P}_x(x)\right]$ as

$$\mathbb{E}\left[\int_{\mathcal{X}}\left(\widehat{f}_n(x)-f_0(x)\right)^2 d\mathcal{P}_x(x)\right] = \underbrace{\mathbb{E}\left[\frac{2}{n}\sum_{i=1}^{n}(\widehat{f}_n(x_i)-f_0(x_i))^2\right]}_{T_1} + \underbrace{\mathbb{E}\left[\int_{\mathcal{X}}\left(\widehat{f}_n(x)-f_0(x)\right)^2 d\mathcal{P}_x(x)\right]-\mathbb{E}\left[\frac{2}{n}\sum_{i=1}^{n}(\widehat{f}_n(x_i)-f_0(x_i))^2\right]}_{T_2}.$$

Applying the upper bounds of T_1 and T_2 in Lemmas 11 and 12 respectively, we can derive

$$\mathbb{E}\left[\int_{\mathcal{X}}\left(\widehat{f}_n(x)-f_0(x)\right)^2 d\mathcal{P}_x(x)\right]\le 4\inf_{f\in\mathcal{F}(M,L,J,I,\tau_1,\tau_2,V)}\int_{\mathcal{X}}(f(x)-f_0(x))^2 d\mathcal{P}_x(x)$$

$$
\begin{aligned}
&+ 48\sigma^2 \frac{\log \mathcal{N}(\delta, \mathcal{F}(M, L, J, I, \tau_1, \tau_2, V), \|\cdot\|_\infty) + 2}{n} \\
&+ 8\sqrt{6}\sqrt{\frac{\log \mathcal{N}(\delta, \mathcal{F}(M, L, J, I, \tau_1, \tau_2, V), \|\cdot\|_\infty) + 2}{n}}\sigma\delta \\
&+ \frac{104V_{\mathcal{F}}^2}{3n} \log \mathcal{N}(\delta/4V, \mathcal{F}(M, L, J, I, \tau_1, \tau_2, V), \|\cdot\|_\infty) \\
&+ \left(4 + \frac{1}{2V_{\mathcal{F}}} + 8\sigma\right)\delta.
\end{aligned}
$$

We need there to exist a network in $\mathcal{F}(M, L, J, I, \tau_1, \tau_2, V)$ which can yield a function f satisfying $\|f - f_0\|_\infty \le \epsilon$ for $\epsilon \in (0, 1)$. ϵ will be chosen later to balance the bias-variance tradeoff. In order to achieve such ϵ-error, we set $\widetilde{M}\widetilde{J} = \epsilon^{-d/\alpha}$, so we now have our network architecture as specified in Theorem 2 in terms of ϵ. Then, we can use the parameters in this architecture to invoke the upper bound of the covering number in Lemma 8:

$$
\begin{aligned}
\log \mathcal{N}(\delta, \mathcal{F}(M, L, J, I, \tau_1, \tau_2, V), \|\cdot\|_\infty) &= O\left(\Lambda_2 \log\left((\tau_1 \vee \tau_2)\, \Lambda_1 \delta^{-1}\right)\right) \\
&\le O\left(\widetilde{M}\widetilde{J}^2 D^3 \log^5(\widetilde{M}\widetilde{J}) \log \frac{1}{\delta}\right) \\
&= O\left(\epsilon^{-d/\alpha} D^3 \log^5 \epsilon \log \frac{1}{\delta}\right),
\end{aligned}
$$

where $O(\cdot)$ hides constant depending on $\log D$, d, α, $\frac{2d}{\alpha p - d}$, p, q, c_0, B, ω and the surface area of $\mathcal{X}$.

Plugging it in, we have

$$
\begin{aligned}
\mathbb{E}\left[\int_{\mathcal{X}} \left(\widehat{f}_n(x) - f_0(x)\right)^2 d\mathcal{D}_x(x)\right] &\le 4\epsilon^2 + \frac{48\sigma^2}{n}\left(c''\epsilon^{-d/\alpha} D^3 \log^5 \epsilon \log \frac{1}{\delta} + 2\right) \\
&\quad + 8\sqrt{6c''}\sqrt{\frac{\epsilon^{-d/\alpha} D^3 \log^5 \epsilon \log \frac{1}{\delta}}{n}}\sigma\delta \\
&\quad + \frac{104V^2}{3n}\epsilon^{-d/\alpha} D^3 \log^5 \epsilon \log \frac{1}{\delta} \\
&\quad + \left(4 + \frac{1}{2V_{\mathcal{F}}} + 8\sigma\right)\delta \\
&= \widetilde{O}\Bigg(\epsilon^2 + \frac{V_{\mathcal{F}}^2 + \sigma^2}{n}\epsilon^{-\frac{d}{\alpha}} D^3 \log^5 \epsilon \log \frac{1}{\delta} \\
&\qquad + \sigma\delta\sqrt{\frac{\epsilon^{-\frac{d}{\alpha}} D^3 \log^5 \epsilon \log \frac{1}{\delta}}{n}} + \sigma\delta + \frac{\sigma^2}{n}\Bigg). \qquad (41)
\end{aligned}
$$

Finally we choose ϵ to satisfy $\epsilon^2 = \frac{1}{n} D^3 \epsilon^{-\frac{d}{\alpha}}$, which gives $\epsilon = D^{\frac{3\alpha}{2\alpha+d}} n^{-\frac{\alpha}{2\alpha+d}}$. It suffices to pick $\delta = \frac{1}{n}$. Substituting both ϵ and δ into (41), we deduce the desired estimation error bound

$$
\mathbb{E}\left[\int_{\mathcal{X}} \left(\widehat{f}_n(x) - f_0(x)\right)^2 d\mathcal{D}_x(x)\right] \le c(V_{\mathcal{F}}^2 + \sigma^2) n^{-\frac{2\alpha}{2\alpha+d}} \log^5 n,
$$

where constant c depends on $D^{\frac{6\alpha}{2\alpha+d}}$, d, α, $\frac{2d}{\alpha p - d}$, p, q, c_0, B, ω and the surface area of $\mathcal{X}$. □

F A RESULT FOR FEED-FORWARD RELU NEURAL NETWORK

F.1 FEED-FORWARD RELU NEURAL NETWORK

We consider multi-layer ReLU (Rectified Linear Unit) neural networks [19]. ReLU activation is popular in computer vision, natural language processing, etc. because the vanishing gradient issue is less severe with it, which is nonetheless common with its counterparts like sigmoid or hyperbolic tangent activation [19, 21]. An L-layer ReLU neural network can be expressed as

$$
f(x) = W_L \cdot \mathrm{ReLU}(W_{L-1} \cdots \mathrm{ReLU}(W_1 x + b_1) \cdots + b_{L-1}) + b_L, \qquad (42)
$$

in which $W_1, \cdots, W_L$ and $b_1, \cdots, b_L$ are weight matrices and vectors and $\mathrm{ReLU}(\cdot)$ is the entrywise rectified linear unit, i.e. $\mathrm{ReLU}(a) = \max\{0, a\}$. The width of a neural network is defined as the number of neurons in its widest layer. For notational simplicity, we define a class of neural networks

$$\begin{aligned}\mathcal{F}(L,p,I,\tau,V) = \{f \mid\ & f(x) \text{ in the form (42) with } L \text{ layers and width at most } p,\\ & \|f\|_\infty \le V,\ \sum_{i=1}^{L} \|W_i\|_0 + \|b_i\|_0 \le I,\ \|W_i\|_{\infty,\infty} \le \tau,\ \|b_i\|_\infty \le \tau \text{ for } i = 1, \cdots, L\}.\end{aligned} \tag{43}$$

F.2 THEOREM 3 AND ITS PROOF

From this point, we denote the function class $\mathcal{F}(L,p,I,\tau,V)$, whose parameters L,p,I,τ,V are chosen according to Theorem 3, with the shorthand $\mathcal{F}$. In this section, this $\mathcal{F}$ is used in Algorithm 1, instead of the CNN class in (11).

Theorem 3. Suppose Assumption 1 and 2 hold. By choosing

$$\begin{gathered} L = O(\log K), \quad p = O\big(K^{\frac{d}{2\alpha+d}}\big), \quad I = O\big(K^{\frac{d}{2\alpha+d}} \log K\big), \\ \tau = \max\{B, H, \sqrt{d}, \omega^2\}, \quad V = H \end{gathered} \tag{44}$$

in Algorithm 1, in which $O(\cdot)$ hides factors depending on α, d and $\log D$, we have

$$\mathbb{E}\,|v^\pi - \widehat{v}^\pi| \le CH^2\kappa \left(K^{-\frac{\alpha}{2\alpha+d}} + \sqrt{D/K}\right) \log^{\frac{3}{2}} K, \tag{45}$$

in which the expectation is taken over the data, and C is a constant depending on $\log D$, α, B, d, ω, the surface area of $\mathcal{X}$ and c_0. The distributional mismatch is captured by

$$\kappa = \frac{1}{H}\sum_{h=1}^{H} \sqrt{\chi^2_{\mathcal{Q}}(q_h^\pi, q_h^{\pi_0}) + 1},$$

in which $\mathcal{Q}$ is the Minkowski sum between the ReLU function class and the Besov function class, i.e., $\mathcal{Q} = \{f + g \mid f \in \mathcal{B}^\alpha_{p,q}(\mathcal{X}), g \in \mathcal{F}\}$.

Proof of Theorem 3. The goal is to bound

$$\mathbb{E}\,|\widehat{v}^\pi - v^\pi| = \mathbb{E}\left|\int_{\mathcal{X}} \left(Q_1^\pi - \widehat{Q}_1^\pi\right)(s,a)\,\mathrm{d}q_1^\pi(s,a)\right| \le \mathbb{E}\left[\int_{\mathcal{X}} \left|Q_1^\pi - \widehat{Q}_1^\pi\right|(s,a)\,\mathrm{d}q_1^\pi(s,a)\right].$$

To get an expression for that, we first expand it recursively. To illustrate the recursive relation, we examine the quantity at step h:

$$\begin{aligned} & \mathbb{E}\left[\int_{\mathcal{X}} \left|Q_h^\pi - \widehat{Q}_h^\pi\right|(s,a)\,\mathrm{d}q_h^\pi(s,a)\right] \\ = & \mathbb{E}\left[\int_{\mathcal{X}} \left|\mathcal{T}_h^\pi Q_{h+1}^\pi - \widehat{\mathcal{T}}_h^\pi\left(\widehat{Q}_{h+1}^\pi\right)\right|(s,a)\,\mathrm{d}q_h^\pi(s,a)\right] \\ \le & \mathbb{E}\left[\int_{\mathcal{X}} \left|\mathcal{T}_h^\pi Q_{h+1}^\pi - \mathcal{T}_h^\pi \widehat{Q}_{h+1}^\pi\right|(s,a)\,\mathrm{d}q_h^\pi(s,a)\right] + \mathbb{E}\left[\int_{\mathcal{X}} \left|\mathcal{T}_h^\pi \widehat{Q}_{h+1}^\pi - \widehat{\mathcal{T}}_h^\pi\left(\widehat{Q}_{h+1}^\pi\right)\right|(s,a)\,\mathrm{d}q_h^\pi(s,a)\right] \\ = & \mathbb{E}\left[\int_{\mathcal{X}} \left|Q_{h+1}^\pi - \widehat{Q}_{h+1}^\pi\right|(s,a)\,\mathrm{d}q_{h+1}^\pi(s,a)\right] \\ & + \mathbb{E}\left[\mathbb{E}\left[\int_{\mathcal{X}} \left|\mathcal{T}_h^\pi \widehat{Q}_{h+1}^\pi - \widehat{\mathcal{T}}_h^\pi\left(\widehat{Q}_{h+1}^\pi\right)\right|(s,a)\,\mathrm{d}q_h^\pi(s,a) \mid \mathcal{D}_{h+1}, \cdots, \mathcal{D}_H\right]\right] \\ \overset{(a)}{\le} & \mathbb{E}\left[\int_{\mathcal{X}} \left|Q_{h+1}^\pi - \widehat{Q}_{h+1}^\pi\right|(s,a)\,\mathrm{d}q_{h+1}^\pi(s,a)\right] \\ & + \mathbb{E}\left[\mathbb{E}\left[\sqrt{\int_{\mathcal{X}} \left(\mathcal{T}_h^\pi \widehat{Q}_{h+1}^\pi - \widehat{\mathcal{T}}_h^\pi\left(\widehat{Q}_{h+1}^\pi\right)\right)^2(s,a)\,\mathrm{d}q_h^{\pi_0}(s,a)}\sqrt{\chi^2_{\mathcal{Q}}(q_h^\pi, q_h^{\pi_0}) + 1} \mid \mathcal{D}_{h+1}, \cdots, \mathcal{D}_H\right]\right] \end{aligned}$$

$$
\begin{aligned}
&\overset{(b)}{\leq} \mathbb{E}\left[\int_{\mathcal{X}} \left|Q^\pi_{h+1} - \widehat{Q}^\pi_{h+1}\right|(s,a)\,\mathrm{d}q^\pi_{h+1}(s,a)\right] \\
&\quad + \sqrt{\mathbb{E}\left[\mathbb{E}\left[\int_{\mathcal{X}} \left(\mathcal{T}^\pi_h \widehat{Q}^\pi_{h+1} - \widehat{\mathcal{T}}^\pi_h\left(\widehat{Q}^\pi_{h+1}\right)\right)^2(s,a)\,\mathrm{d}q^{\pi_0}_h(s,a) \mid \mathcal{D}_{h+1},\cdots,\mathcal{D}_H\right]\right]}\sqrt{\chi^2_{\mathcal{Q}}(q^\pi_h, q^{\pi_0}_h)+1} \\
&\overset{(c)}{\leq} \int_{\mathcal{X}} \left|Q^\pi_{h+1} - \widehat{Q}^\pi_{h+1}\right|(s,a)\,\mathrm{d}q^\pi_{h+1}(s,a) + \sqrt{c(5H^2)\left(K^{-\frac{2\alpha}{2\alpha+d}} + \frac{D}{K}\right)\log^3 K}\sqrt{\chi^2_{\mathcal{Q}}(q^\pi_h, q^{\pi_0}_h)+1} \\
&\leq \int_{\mathcal{X}} \left|Q^\pi_{h+1} - \widehat{Q}^\pi_{h+1}\right|(s,a)\,\mathrm{d}q^\pi_{h+1}(s,a) + CH\left(K^{-\frac{\alpha}{2\alpha+d}} + \sqrt{\frac{D}{K}}\right)\log^{3/2} K\sqrt{\chi^2_{\mathcal{Q}}(q^\pi_h, q^{\pi_0}_h)+1},
\end{aligned}
$$

where C denotes a (varying) constant depending on $\log D$, α, B, d, ω, the surface area of $\mathcal{X}$ and c_0.

In (a), note $\mathcal{T}^\pi_h \widehat{Q}^\pi_{h+1} \in \mathcal{B}^\alpha_{p,q}(\mathcal{X})$ by Assumption 2 and $-\widehat{\mathcal{T}}^\pi_h\left(\widehat{Q}^\pi_{h+1}\right) \in \mathcal{F}$ by our algorithm, so $\mathcal{T}^\pi_h \widehat{Q}^\pi_{h+1} - \widehat{\mathcal{T}}^\pi_h\left(\widehat{Q}^\pi_{h+1}\right) \in \mathcal{Q}$. Then we obtain this inequality by invoking the following lemma.

In (b), we use Jensen's inequality and the fact that square root is concave.

To obtain (c), we invoke the following lemma, which provides an upper bound on the regression error.

Specifically, we will use Lemma 13 when conditioning on $\mathcal{D}_{h+1},\cdots,\mathcal{D}_H$, i.e. the data from time step $h+1$ to time step H. Note that after conditioning, $\mathcal{T}^\pi_h \widehat{Q}^\pi_{h+1}$ becomes measurable and deterministic with respect to $\mathcal{D}_{h+1},\cdots,\mathcal{D}_H$. Also, $\mathcal{D}_{h+1},\cdots,\mathcal{D}_H$ are independent from $\mathcal{D}_h$, which we use in the regression at step h.

To justify our use of Lemma 13, we need to cast our problem into a regression problem described in the lemma. Since $\{(s_{h,k}, a_{h,k})\}_{k=1}^K$ are i.i.d. from $q^{\pi_0}_h$, we can view them as the samples x_i's in the lemma. We can view $\mathcal{T}^\pi_h \widehat{Q}^\pi_{h+1}$, which is measurable under our conditioning, as f_0 in the lemma. Furthermore, we let

$$
\zeta_{h,k} := r_{h,k} + \int_{\mathcal{A}} \widehat{Q}^\pi_{h+1}(s'_{h,k}, a)\pi(a \mid s'_{h,k})\,\mathrm{d}a - \mathcal{T}^\pi_h \widehat{Q}^\pi_{h+1}(s_{h,k}, a_{h,k}).
$$

In order to invoke Lemma 13 under the conditioning on $\mathcal{D}_{h+1},\cdots,\mathcal{D}_H$, we need to verify whether three conditions are satisfied (conditioning on $\mathcal{D}_{h+1},\cdots,\mathcal{D}_H$):

1. Sample $\{(s_{h,k}, a_{h,k})\}_{k=1}^K$ are i.i.d;
2. Sample $\{(s_{h,k}, a_{h,k})\}_{k=1}^K$ and noise $\{\zeta_{h,k}\}_{k=1}^K$ are uncorrelated;
3. Noise $\{\zeta_{h,k}\}_{k=1}^K$ are independent, zero-mean, subgaussian random variables.

In our setting, $\{(s_{h,k}, a_{h,k})\}_{k=1}^K$ are i.i.d. from $q^{\pi_0}_h$. Due to the time-inhomogeneous setting, they are independent from $\mathcal{D}_{h+1},\cdots,\mathcal{D}_H$, so $\{(s_{h,k}, a_{h,k})\}_{k=1}^K$ are still i.i.d. under our conditioning. Thus, Condition 1 is clearly satisfied.

We may observe that under our conditioning, the transition from $(s_{h,k}, a_{h,k})$ to $s'_{h,k}$ is the only source of randomness in $\zeta_{h,k}$, besides $(s_{h,k}, a_{h,k})$ itself. The distribution of $(s_{h,k}, a_{h,k}, s'_{h,k})$ is actually the product distribution between $P_h(\cdot|s_{h,k}, a_{h,k})$ and $q^{\pi_0}_h$, so a function of $s'_{h,k}$, generated from the transition distribution $P_h(\cdot|s_{h,k}, a_{h,k})$, is uncorrelated with $(s_{h,k}, a_{h,k})$. Thus, $(s_{h,k}, a_{h,k})$'s are uncorrelated with $\zeta_{h,k}$'s under our conditioning, and Condition 2 is satisfied.

Condition 3 can also be easily verified. Under our conditioning, the randomness in $\zeta_{h,k}$ only comes from $(s_{h,k}, a_{h,k}, s'_{h,k}, r_{h,k})$, which are independent from $(s_{h,k'}, a_{h,k'}, s'_{h,k'}, r_{h,k'})$ for any $k' \neq k$, so $\zeta_{h,k}$'s are independent from each other. As for the mean of $\zeta_{h,k}$,

$$
\begin{aligned}
&\mathbb{E}\left[\zeta_{h,k} \mid \mathcal{D}_{h+1},\cdots,\mathcal{D}_H\right] \\
&= \mathbb{E}\left[r_{h,k} + \int_{\mathcal{A}} \widehat{Q}^\pi_{h+1}(s'_{h,k}, a)\pi(a \mid s'_{h,k})\,\mathrm{d}a - r_h(s_{h,k}, a_{h,k}) - \mathcal{P}^\pi_h \widehat{Q}^\pi_{h+1}(s_{h,k}, a_{h,k}) \mid \mathcal{D}_{h+1},\cdots,\mathcal{D}_H\right]
\end{aligned}
$$

$$
\begin{aligned}
&= \mathbb{E}\Bigg[r_{h,k} - r_h(s_{h,k}, a_{h,k}) + \int_{\mathcal{A}} \widehat{Q}^{\pi}_{h+1}(s'_{h,k}, a)\pi(a \mid s'_{h,k})\,\mathrm{d}a \\
&\quad - \mathbb{E}_{s' \sim P_h(\cdot \mid s_{h,k}, a_{h,k})}\left[\int_{\mathcal{A}} \widehat{Q}^{\pi}_{h+1}(s', a)\pi(a \mid s')\,\mathrm{d}a \mid s_{h,k}, a_{h,k}, \mathcal{D}_{h+1}, \cdots, \mathcal{D}_H\right] \mid \mathcal{D}_{h+1}, \cdots, \mathcal{D}_H\Bigg] \\
&= 0 + 0 = 0.
\end{aligned}
$$

On the other hand, $\left\|\widehat{Q}^{\pi}_{h+1}\right\|_{\infty} \le H$ almost surely, because it is a function in our ReLU network class $\mathcal{F}$. Thus, $\zeta_{h,k}$ is a bounded random variable with $\zeta_{h,k} \in [-2H, 2H]$ almost surely, so its variance is bounded by $4H^2$. Its boundedness also implies it is a subgaussian random variable. Thus, Condition 3 is also satisfied.

Hence, Lemma 13 proves, for step h in our algorithm,

$$
\begin{aligned}
&\mathbb{E}\left[\int_{\mathcal{X}} \left(\mathcal{T}^{\pi}_h \widehat{Q}^{\pi}_{h+1} - \widehat{\mathcal{T}}^{\pi}_h\left(\widehat{Q}^{\pi}_{h+1}\right)\right)^2 (s, a)\,\mathrm{d}q^{\pi_0}_h(s, a) \mid \mathcal{D}_{h+1}, \cdots, \mathcal{D}_H\right] \\
&\le c(H^2 + 4H^2)\left(K^{-\frac{2\alpha}{2\alpha+d}} + \frac{D}{K}\right)\log^3 K.
\end{aligned}
$$

Note that this upper bound holds for any $\widehat{Q}^{\pi}_{h+1}$ or $\mathcal{D}_{h+1}, \cdots, \mathcal{D}_H$. The sole purpose of our conditioning is that we could view $\widehat{Q}^{\pi}_{h+1}$ as a measurable or deterministic function under the conditioning and then apply Lemma 13. Therefore,

$$
\begin{aligned}
&\mathbb{E}\left[\mathbb{E}\left[\int_{\mathcal{X}} \left(\mathcal{T}^{\pi}_h \widehat{Q}^{\pi}_{h+1} - \widehat{\mathcal{T}}^{\pi}_h\left(\widehat{Q}^{\pi}_{h+1}\right)\right)^2 (s, a)\,\mathrm{d}q^{\pi_0}_h(s, a) \mid \mathcal{D}_{h+1}, \cdots, \mathcal{D}_H\right]\right] \\
&\le c(H^2 + 4H^2)\left(K^{-\frac{2\alpha}{2\alpha+d}} + \frac{D}{K}\right)\log^3 K.
\end{aligned}
$$

Finally, we carry out the recursion from time step 1 to time step H, and the final result is

$$
\mathbb{E}\left|v^{\pi} - \widehat{v}^{\pi}\right| \le CH^2\left(K^{-\frac{\alpha}{2\alpha+d}} + \sqrt{\frac{D}{K}}\right)\log^{3/2} K\left(\frac{1}{H}\sum_{h=1}^{H}\sqrt{\chi^2_{\mathcal{Q}}(q^{\pi}_h, q^{\pi_0}_h) + 1}\right).
$$

□

F.3 LEMMA 13 AND ITS PROOF

Lemma 13. *Let $\mathcal{X}$ be a d-dimensional compact Riemannian manifold isometrically embedded in $\mathbb{R}^D$ with reach ω. There exists a constant $B > 0$ such that for any $x \in \mathcal{X}$, $|x_j| \le B$ for all $j = 1, \cdots, D$. We are given a function $f_0 \in \mathcal{B}^{\alpha}_{p,q}(\mathcal{X})$ and samples $S_n = \{(x_i, y_i)\}_{i=1}^n$, where x_i are i.i.d. sampled from a distribution $\mathcal{P}_x$ on $\mathcal{X}$ and $y_i = f_0(x_i) + \zeta_i$. ζ_i's are i.i.d. sub-Gaussian random noise with variance σ^2, uncorrelated with x_i's. If we compute an estimator*

$$
\widehat{f}_n = \arg\min_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\left(f(x_i) - y_i\right)^2,
$$

with the neural network class $\mathcal{F} = \mathcal{F}(L, p, I, \tau, V)$ such that

$$
\begin{aligned}
L = O\left(\log n\right), p = O\left(n^{\frac{d}{2\alpha+d}}\right), I = O\left(n^{\frac{d}{2\alpha+d}}\log n\right), \\
\tau = \max\{B, V_{\mathcal{F}}, \sqrt{d}, \omega^2\}, V = V_{\mathcal{F}}, \qquad (46)
\end{aligned}
$$

then we have

$$
\mathbb{E}\left[\int_{\mathcal{X}} \left(\widehat{f}_n(x) - f_0(x)\right)^2 \mathrm{d}\mathcal{P}_x(x)\right] \le c\left(V_{\mathcal{F}}^2 + \sigma^2\right)\left(n^{-\frac{2\alpha}{2\alpha+d}} + \frac{D}{n}\right)\log^3 n, \qquad (47)
$$

where $V_{\mathcal{F}} = \|f_0\|_{\infty}$ and the expectation is taken over the training sample S_n, and c is a constant depending on $\log D$, α, B, d, ω, the surface area of $\mathcal{X}$ and c_0.

Proof of Lemma 13. Recall that the bias and variance decomposition of $\mathbb{E}\left[\int_{\mathcal{X}} \left(\widehat{f}_n(x) - f_0(x)\right)^2 d\mathcal{P}_x(x)\right]$ as

$$\begin{aligned}\mathbb{E}\left[\int_{\mathcal{X}} \left(\widehat{f}_n(x) - f_0(x)\right)^2 d\mathcal{P}_x(x)\right] = &\underbrace{\mathbb{E}\left[\frac{2}{n}\sum_{i=1}^{n}(\widehat{f}_n(x_i) - f_0(x_i))^2\right]}_{T_1} \\ &+ \underbrace{\mathbb{E}\left[\int_{\mathcal{X}} \left(\widehat{f}_n(x) - f_0(x)\right)^2 d\mathcal{P}_x(x)\right] - \mathbb{E}\left[\frac{2}{n}\sum_{i=1}^{n}(\widehat{f}_n(x_i) - f_0(x_i))^2\right]}_{T_2}.\end{aligned}$$

Applying the upper bounds of T_1 and T_2 in Lemmas 11 and 12 respectively, we can derive

$$\begin{aligned}\mathbb{E}\left[\int_{\mathcal{X}} \left(\widehat{f}_n(x) - f_0(x)\right)^2 d\mathcal{P}_x(x)\right] \leq\ & 4 \inf_{f \in \mathcal{F}(L,p,I,\tau,V)} \int_{\mathcal{X}} (f(x) - f_0(x))^2 d\mathcal{P}_x(x) \\ &+ 48\sigma^2 \frac{\log \mathcal{N}(\delta, \mathcal{F}(L,p,I,\tau,V), \|\cdot\|_\infty) + 2}{n} \\ &+ 8\sqrt{6}\sqrt{\frac{\log \mathcal{N}(\delta, \mathcal{F}(L,p,I,\tau,V), \|\cdot\|_\infty) + 2}{n}}\sigma\delta \\ &+ \frac{104V_{\mathcal{F}}^2}{3n} \log \mathcal{N}(\delta/4V, \mathcal{F}(L,p,I,\tau,V), \|\cdot\|_\infty) \\ &+ \left(4 + \frac{1}{2V_{\mathcal{F}}} + 8\sigma\right)\delta.\end{aligned}$$

We need there to exist a network in $\mathcal{F}(L,p,I,\tau,V)$ which can yield a function f satisfying $\|f - f_0\|_\infty \leq \epsilon$ for $\epsilon \in (0,1)$. ϵ will be chosen later to balance the bias-variance tradeoff. By Lemma 2 of Nguyen-Tang et al. [38], in order to achieve such ϵ-error, we need

$$L = O\left(\log\frac{1}{\epsilon}\right), p = O\left(\epsilon^{-\frac{d}{\alpha}}\right), I = O\left(\epsilon^{-\frac{d}{\alpha}}\log\frac{1}{\epsilon}\right),$$
$$\tau = \max\{B, V_{\mathcal{F}}, \sqrt{d}, \omega^2\}, V = V_{\mathcal{F}},$$

where $O(\cdot)$ hides factors of $\log D$, α, d and the surface area of $\mathcal{X}$, so we now have our network architecture as specified in Theorem 2 in terms of ϵ. Then, we can use the architecture parameters in (13) to invoke the upper bound of the covering number in Lemma 7 of Chen et al. [4]:

$$\begin{aligned}\log \mathcal{N}(\delta, \mathcal{F}(L,p,I,\tau,V), \|\cdot\|_\infty) &= \log\left(\frac{2L^2(pB+2)\tau^L p^{L+1}}{\delta}\right)^I \\ &\leq c'' \epsilon^{-\frac{d}{\alpha}} \log^3\frac{1}{\epsilon}\log\frac{1}{\delta},\end{aligned}$$

where c'' is a constant depending on $\log B$, ω and $\log\log n$.

Plugging it in, we have

$$\begin{aligned}\mathbb{E}\left[\int_{\mathcal{X}} \left(\widehat{f}_n(x) - f_0(x)\right)^2 d\mathcal{D}_x(x)\right] \leq\ & 4\epsilon^2 + \frac{48\sigma^2}{n}\left(c''\epsilon^{-d/\alpha}\log^3\frac{1}{\epsilon}\log\frac{1}{\delta} + 2\right) \\ &+ 8\sqrt{6c''}\sqrt{\frac{\epsilon^{-d/\alpha}\log^3\frac{1}{\epsilon}\log\frac{1}{\delta}}{n}}\sigma\delta \\ &+ \frac{104V_{\mathcal{F}}^2}{3n}\epsilon^{-d/\alpha}\log^3\frac{1}{\epsilon}\log\frac{1}{\delta} \\ &+ \left(4 + \frac{1}{2V_{\mathcal{F}}} + 8\sigma\right)\delta \\ =\ & \widetilde{O}\left(\epsilon^2 + \frac{V_{\mathcal{F}}^2 + \sigma^2}{n}\epsilon^{-\frac{d}{\alpha}}\log^3\frac{1}{\epsilon}\log\frac{1}{\delta}\right.\end{aligned}$$

$$+ \sigma\delta\sqrt{\frac{\epsilon^{-\frac{d}{\alpha}}\log^3\frac{1}{\epsilon}\log\frac{1}{\delta}}{n}} + \sigma\delta + \frac{\sigma^2}{n}\Bigg). \tag{48}$$

Finally we choose ϵ to satisfy $\epsilon^2 = \frac{1}{n}\epsilon^{-\frac{d}{\alpha}}$, which gives $\epsilon = n^{-\frac{\alpha}{2\alpha+d}}$. It suffices to pick $\delta = \frac{1}{n}$. Substituting both ϵ and δ into (48), we deduce the desired estimation error bound

$$\mathbb{E}\left[\int_{\mathcal{X}}\left(\widehat{f}_n(x) - f_0(x)\right)^2 d\mathcal{D}_x(x)\right] \leq c(V_{\mathcal{F}}^2 + \sigma^2)\left(n^{-\frac{2\alpha}{2\alpha+d}} + \frac{D}{n}\right)\log^3 n,$$

where constant c depends on $\log D$, d, α, $\frac{2d}{\alpha p - d}$, p, q, c_0, B, ω and the surface area of $\mathcal{X}$. $\square$

G SUPPLEMENT FOR EXPERIMENTS

G.1 EXPERIMENT DETAILS

We use the CartPole environment from OpenAI gym. We consider it as a time-inhomogeneous finite-horizon MDP by setting a time limit of 100 steps. We turn the terminal states in the original CartPole into absorbing states, so if a trajectory terminates before 100 steps, the agent would keep receiving zero reward in its terminal state until the end. The target policy is a policy trained for 200 iterations using REINFORCE, in which each iteration samples for 100 trajectories with truncation after 150 time steps. The target policy value v^π is estimated to be 65.2117, which we obtain by Monte Carlo rollout from the initial state distribution.

For a given behavior policy, to obtain dataset $\mathcal{D}_h$ at time step h, we sample for K independent episodes under the behavior policy and only take the (s, a, s', r) tuple from the h-th transition in each episode. This is an excessive way to guarantee the independence among these K samples; in practice, we could directly sample from a sampling distribution. We sample for $\mathcal{D}_h$ for each $h = 1, \cdots, 100$.

We use the render() function in OpenAI gym for the visual display of CartPole. We downsample images to the desired resolution via cubic interpolation. A high-resolution image (see Figure 3) is represented as a $3 \times 40 \times 150$ RGB array; a low-resolution image (see Figure 4) is represented as a $3 \times 20 \times 75$ RGB array.

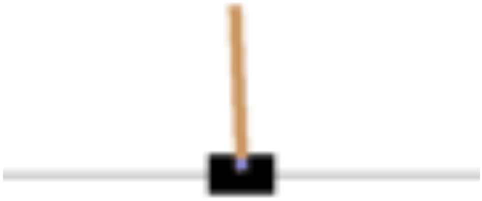

Figure 3: CartPole in high resolution.

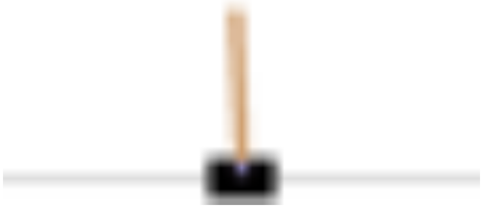

Figure 4: CartPole in low resolution.

For the function approximator in FQE, we use a neural network that comprises 3 convolutional layers each with output channel size 16, 32 and 32 and a final linear layer. These layers are interleaved with ReLU activation and batch norm layers for weight normalization. For high resolution input, we use kernel size 5 and stride 2; for low resolution input, we use kernel size 3 and stride 1. For experiments with high resolution, in each step of FQE, we solve the regression by training the network via stochastic gradient descent with batch size 256 for 20 epochs. In high-resolution experiments, we use 0.01 learning rate; in low-resolution experiments, we use 0.001 learning rate. We compute the average and standard deviation of FQE's result over 5 random seeds.