A Other related work.

Besides Parhi and Nowak [27] which we discussed earlier, Parhi and Nowak [28, 29] also leveraged the connections between NNs and splines. Parhi and Nowak [28] focused on characterizing the variational form of multi-layer NN. Parhi and Nowak [29] showed that two-layer ReLU activated NN achieves minimax rate for a BV class of order 1 but did not cover multilayer NNs nor BV class with order > 1, which is our focus.

The connection between weight-decay regularization with sparsity-inducing penalties in two-layer NNs is folklore and used by Neyshabur et al. [25], Savarese et al. [32], Ongie et al. [26], Ergen and Pilanci [12, 14], Parhi and Nowak [27, 29]. The key underlying technique — an application of the AM-GM inequality (which we used in this paper as well) — can be traced back to Srebro et al. [35] (see a recent exposition by Tibshirani [39]). Tibshirani [39] also generalized the result to multi-layered NNs, but with a simple (element-wise) connections.

The approximation-theoretic and estimation-theoretic research for neural network has a long history too [6, 4, 46, 33, 36]. Most existing work considered the Holder, Sobolev spaces and their extensions, which contain only homogeneously smooth functions and cannot demonstrate the advantage of NNs over kernels. The only exception is Suzuki [36] which, as we discussed earlier, requires modifications to NN architecture for each class. In contrast, we require tuning only the standard weight decay parameter.

B Two-layer Neural Network with Truncated Power Activation Functions

We start by recapping the result of Parhi and Nowak [27] and formalizing its implication in estimating BV functions. Parhi and Nowak [27] considered a two layer neural network with truncated power activation function. Let the neural network be

$$f(x) = \sum_{j=1}^{M} v_j \sigma^m(w_j x + b_j) + c(x), \tag{7}$$

where w_j, v_j denote the weight in the first and second layer respectively, b_j denote the bias in the first layer, $c(x)$ is a polynomial of order up to m, $\sigma^m(x) := \max(x, 0)^m$. Parhi and Nowak [27, Theorem 8] showed that when M is large enough, The optimization problem

$$\min_{\boldsymbol{w},\boldsymbol{v}} \hat{L}(f) + \frac{\lambda}{2} \sum_{j=1}^{M} (|v_j|^2 + |w_j|^{2m}) \tag{8}$$

is equivalent to the locally adaptive regression spline:

$$\min_f \hat{L}(f) + \lambda TV(f^{(m)}(x)), \tag{9}$$

which optimizes over arbitrary functions that is m-times weakly differentiable. The latter was studied in Mammen and van de Geer [22], which leads to the following MSE:

Theorem 9. *Let $M \geq n - m$, and $\hat{f}$ be the function (7) parameterized by the minimizer of* (8)*, then*

$$\mathrm{MSE}(\hat{f}) = O(n^{-(2m+2)(2m+3)}).$$

We show a simpler proof in the univariate case due to Tibshirani [40]:

Proof. As is shown in Parhi and Nowak [27, Theorem 8], the minimizer of (8) satisfy

$$|v_j| = |w_j|^m, \forall k$$

so the TV of the neural network f_{NN} is

$$\begin{aligned}
TV^{(m)}(f_{NN}) &= TV^{(m)}c(x) + \sum_{j=1}^{M}|v_j||w_j|^m TV^{(m)}(\sigma^{(m)}(x)) \\
&= \sum_{j=1}^{M}|v_j||w_j|^m \\
&= \frac{1}{2}\sum_{j=1}^{M}(|v_j|^2 + |w_j|^{2m})
\end{aligned}$$

which shown that (8) is equivalent to the locally adaptive regression spline (9) as long as the number of knots in (9) is no more than M. Furthermore, it is easy to check that any spline with knots no more than M can be expressed as a two layer neural network (8). It suffices to prove that the solution in (9) has no more than $n-m$ number of knots.

Mammen and van de Geer [22, Proposition 1] showed that there is a solution to (9) $\hat{f}(x)$ such that $\hat{f}(x)$ is a mth order spline with a finite number of knots but did not give a bound. Let the number of knots be M, we can represent $\hat{f}$ using the truncated power basis

$$\hat{f}(x) = \sum_{j=1}^{M} a_j(x-t_j)_+^m + c(x) := \sum_{j=1}^{M} a_j\sigma_j^{(m)}(x) + c(x)$$

where t_j are the knots, $c(x)$ is a polynomial of order up to m, and define $\sigma_j^{(m)}(x) = (x-t_j)_+^m$.

Mammen and van de Geer [22] however did not give a bound on M. Parhi and Nowak [27]'s Theorem 1 implies that $M \leq n-m$. Its proof is quite technical and applies more generally to a higher dimensional generalization of the BV class.

Tibshirani [40] communicated to us the following elegant argument to prove the same using elementary convex analysis and linear algebra, which we present below.

Define $\Pi_m(f)$ as the $L^2(P_n)$ projection of f onto polynomials of degree up to m, $\Pi_m^\perp(f) := f - \Pi_m(f)$. It is easy to see that

$$\Pi_m^\perp f(x) = \sum_{j=1}^{M} a_j \Pi_m^\perp \sigma_j^{(m)}(x)$$

Denote $f(x_{1:n}) := \{f(x_1), \ldots, f(x_n)\} \in \mathbb{R}^n$ as a vector of all the predictions at the sample points.

$$\Pi_m^\perp \hat{f}(x_{1:n}) = \sum_{j=1}^{M} a_j \Pi_m^\perp \sigma_j^{(m)}(x_{1:n}) \in \Pi_m^\perp \text{conv}\{\pm\sigma_j^{(m)}(x_{1:n})\} \cdot \sum_{j=1}^{M}|a_j| = \in \text{conv}\{\pm\Pi_m^\perp\sigma_j^{(m)}(x_{1:n})\} \cdot \sum_{j=1}^{M}|a_j|$$

where conv denotes the convex hull of a set. The convex hull $\text{conv}\{\pm\sigma_j^{(m)}(x_{1:n})\} \cdot \sum_{j=1}^{M}|a_j|$ is an n-dimensional space, and polynomials of order up to m is an $m+1$ dimensional space, so the set defined above has dimension $n-m-1$. By Carathéodory's theorem, there is a subset of points in this space

$$\{\Pi_m^\perp \sigma_{j_k}^{(m)}(x_{1:n})\} \subseteq \{\Pi_m^\perp \sigma_j^{(m)}(x_{1:n})\}, 1 \leq k \leq n-m$$

such that

$$\Pi_m^\perp f(x) = \sum_{k=1}^{n-m} \tilde{a}_k \Pi_m^\perp \sigma_{j_k}^{(m)}(x), \sum_{k=1}^{n-m}|a_k| \leq 1$$

In other word, there exist a subset of knots $\{\tilde{t}_j, j \in [n-m]\}$ that perfectly recovers $\Pi_m^\perp \hat{f}(x)$ at all the sample points, and the TV of this function is no larger than $\hat{f}$.

This shows that

$$\tilde{f}(x) = \sum_{j=1}^{n-m} \tilde{a}_j(x-t_j)_+^m, s.t. \tilde{f}(x_i) = f(x_i)$$

for all x_i in n onbservation points.

The MSE of locally adaptivity regressive spline (9) was studied in Mammen and van de Geer [22, Section 3], which equals the error rate given in Theorem 9. $\square$

This indicates that the neural network (7) is minimax optimal for $BV(m)$.

Let us explain a few the key observations behind this equivalence. (a) The truncated power functions (together with an mth order polynomial) spans the space of an mth order spline. (b) The neural network in (7) is equivalent to a free-knot spline with M knots (up to reparameterization). (c) A solution to (9) is a spline with at most $n - m$ knots [27, Theorem 8]. (d) Finally, by the AM-GM inequality

$$|v_j|^2 + |w_j|^{2m} \geq 2|v_j||w_j|^m = 2|c_j|$$

where $c_j = v_j|w_j|^m$ is the coefficient of the corresponding jth truncated power basis. The mth order total variation of a spline is equal to $\sum_j |c_j|$. It is not hard to check that the loss function depends only on c_j, thus the optimal solution will always take "=" in the AM-GM inequality.

C Introduction To Common Function Classes

In the following definition define Ω be the domain of the function classes, which will be omitted in the definition.

C.1 Besov Class

Definition 1. *Modulus of smoothness: For a function $f \in L^p(\Omega)$ for some $1 \leq p \leq \infty$, the r-th modulus of smoothness is defined by*

$$w_{r,p}(f,t) = \sup_{h \in \mathbb{R}^d : \|h\|_2 \leq t} \|\Delta_h^r(f)\|_p,$$

$$\Delta_h^r(f) := \begin{cases} \sum_{j=0}^{r} \binom{r}{j}(-1)^{r-j} f(x+jh), & \textit{if } x \in \Omega, x+rh \in \Omega, \\ 0, & \textit{otherwise}. \end{cases}$$

Definition 2. *Besov space: For $1 \leq p, q \leq \infty, \alpha > 0, r := \lceil \alpha \rceil + 1$, define*

$$|f|_{B^\alpha_{p,q}} = \begin{cases} \Big(\int_{t=0}^{\infty} (t^{-\alpha} w_{r,p}(f,t))^q \frac{dt}{t}\Big)^{\frac{1}{q}}, & q < \infty \\ \sup_{t>0} t^{-\alpha} w_{r,p}(f,t), & q = \infty, \end{cases}$$

and define the norm of Besov space as:

$$\|f\|_{B^\alpha_{p,q}} = \|f\|_p + |f|_{B^\alpha_{p,q}}.$$

A function f is in the Besov space $B^\alpha_{p,q}$ if $\|f\|_{B^\alpha_{p,q}}$ is finite.

Note that the Besov space for $0 < p, q < 1$ is also defined, but in this case it is a quasi-Banach space instead of a Banach space and will not be covered in this paper.

Functions in Besov space can be decomposed using B-spline basis functions. Any function f in Besov space $B^\alpha_{p,q}, \alpha > d/p$ can be decomposed using B-spline of order $m, m > \alpha$: let $\boldsymbol{x} \in \mathbb{R}^d$,

$$f(\boldsymbol{x}) = \sum_{k=0}^{\infty} \sum_{\boldsymbol{s} \in J(k)} c_{k,\boldsymbol{s}}(f) M_{m,k,\boldsymbol{s}}(\boldsymbol{x}) \tag{10}$$

where $J(k) := \{2^{-k}\boldsymbol{s} : \boldsymbol{s} \in [-m, 2^k+m]^d \subset \mathbb{Z}^d\}$, $M_{m,k,\boldsymbol{s}}(\boldsymbol{x}) := M_m(2^k(\boldsymbol{x}-\boldsymbol{s}))$, and $M_k(\boldsymbol{x}) = \prod_{i=1}^d M_k(x_i)$ is the cardinal B-spline basis function which can be expressed as a polynomial:

$$M_m(x) = \frac{1}{m!} \sum_{j=1}^{m+1} (-1)^j \binom{m+1}{j} (x-j)_+^m = ((m+1)/2)^m \frac{1}{m!} \sum_{j=1}^{m+1} (-1)^j \binom{m+1}{j} \Big(\frac{x-j}{(m+1)/2}\Big)_+^m,$$

Furthermore, the norm of Besov space is equivalent to the sequence norm:

$$\|\{c_{k,\boldsymbol{s}}\}\|_{b^{\alpha}_{p,q}} := \Big(\sum_{k=0}^{\infty}(2^{(\alpha-d/p)k}\|\{c_{k,\boldsymbol{s}}(f)\}_{\boldsymbol{s}}\|_p)^q\Big)^{1/q} \eqsim \|f\|_{B^{\alpha}_{p,q}}.$$

See e.g. Dũng [11, Theorem 2.2] for the proof.

C.2 Other Function Spaces

Definition 3. *Hölder space: let $m \in \mathbb{N}$, the m-th order Holder class is defined as*

$$\mathcal{C}^m = \left\{ f : \max_{|a|=k} \frac{|D^a f(x) - D^a f(z)|}{\|x-z\|_2} < \infty, \forall x, z \in \Omega \right\}$$

where D^a denotes the weak derivative.

Note that fraction order of Hölder space can also be defined. For simplicity, we will not cover that case in this paper.

Definition 4. *Sobolev space: let $m \in \mathcal{N}, 1 \leq p \leq \infty$, the Sobolev norm is defined as*

$$\|f\|_{W^m_p} := \left(\sum_{|a|\leq m} \|D^a f\|_p^p\right)^{1/p},$$

the Sobolev space is the set of functions with finite Sobolev norm:

$$W^m_p := \{f : \|f\|_{W^m_p} < \infty\}.$$

Definition 5. *Total Variation (TV): The total variation (TV) of a function f on an interval $[a, b]$ is defined as*

$$TV(f) = \sup_{\mathcal{P}} \sum_{i=1}^{n_{\mathcal{P}}-1} |f(x_{i+1}) - f(x_i)|$$

where the $\mathcal{P}$ is taken among all the partitions of the interval $[a, b]$.

In many applications, functions with stronger smoothness conditions are needed, which can be measured by high order total variation.

Definition 6. *High order total variation: the m-th order total variation is the total variation of the $(m-1)$-th order derivative*

$$TV^{(m)}(f) = TV(f^{(m-1)})$$

Definition 7. *Bounded variation (BV): The m-th order bounded variation class is the set of functions whose total variation (TV) is bounded.*

$$BV(m) := \{f : TV(f^{(m)}) < \infty\}.$$

D Proof of Estimation Error

D.1 Equivalence Between Parallel Neural Networks and p-norm Penalized Problems

Proposition 3. *Fix the input dataset $\mathcal{D}_n$ and a constant $c_1 > 0$. For every λ, there exists $P' > 0$ such that (2) is equivalent to the following problem:*

$$\begin{aligned}
\underset{\{\bar{\mathbf{W}}_j^{(\ell)}, \bar{\boldsymbol{b}}_j^{(\ell)}, a_j\}}{\arg\min}\ & \hat{L}\Big(\sum_{j=1}^{M} a_j \bar{f}_j\Big) = \frac{1}{n}\sum_i (y_i - \bar{f}_{1:M}(\boldsymbol{x}_i)^T \boldsymbol{a})^2 \\
s.t.\ & \|\bar{\mathbf{W}}_j^{(1)}\|_F \leq c_1\sqrt{d}, \forall j \in [M], \\
& \|\bar{\mathbf{W}}_j^{(\ell)}\|_F \leq c_1\sqrt{w}, \forall j \in [M], 2 \leq \ell \leq L, \quad \|\{a_j\}\|_{2/L}^{2/L} \leq P'
\end{aligned}$$

where $\bar{f}_j(\cdot)$ is a subnetwork with parameters $\bar{\mathbf{W}}_j^{(\ell)}, \bar{\boldsymbol{b}}_j^{(\ell)}$.

Proof. Using Langrange's method, one can easily find (2) is equivalent to a constrained optimization problem:

$$\underset{\{\mathbf{W}_j^{(\ell)}, \boldsymbol{b}_j^{(\ell)}\}}{\arg\min}\ \hat{L}\Big(\sum_{j=1}^{M} f_j\Big), \quad s.t. \sum_{j=1}^{M}\sum_{\ell=1}^{L} \|\mathbf{W}_j^{(\ell)}\|_F^2 \le P \tag{11}$$

for some constant P that depends on λ and the dataset $\mathcal{D}$.

We make use of the property from (4) to minimize the constraint term in (11) while keeping this neural network equivalent to the original one. Specifically, let $\mathbf{W}^{(1)}, \boldsymbol{b}^{(1)}, \dots \mathbf{W}^{(L)}, \boldsymbol{b}^{(L)}$ be the parameters of an L-layer neural network.

$$f(x) = \mathbf{W}^{(L)}\sigma(\mathbf{W}^{(L-1)}\sigma(\dots\sigma(\mathbf{W}^{(1)}x + \boldsymbol{b}^{(1)})\dots) + \boldsymbol{b}^{(L-1)}) + \boldsymbol{b}^{(L)},$$

which is equivalent to

$$f(x) = \alpha_L\tilde{\mathbf{W}}^{(L)}\sigma(\alpha_{L-1}\tilde{\mathbf{W}}^{(L-1)}\sigma(\dots\sigma(\alpha_1\tilde{\mathbf{W}}^{(1)}x + \tilde{\boldsymbol{b}}^{(1)})\dots) + \tilde{\boldsymbol{b}}^{(L-1)}) + \tilde{\boldsymbol{b}}^{(L)},$$

as long as $\alpha_\ell > 0, \prod_{\ell=1}^{L}\alpha^L = \prod_{\ell=1}^{L}\|\mathbf{W}^{(\ell)}\|_F$, where $\tilde{\mathbf{W}}^{(\ell)} := \frac{\mathbf{W}^{(\ell)}}{\|\mathbf{W}^{(\ell)}\|_F}$. By the AM-GM inequality, the ℓ_2 regularizer of the latter neural network is

$$\sum_{\ell=1}^{L}\|\alpha_\ell\tilde{\mathbf{W}}^{(\ell)}\|_F^2 = \sum_{\ell=1}^{L}\alpha_\ell^2 \ge L\left(\prod_{\ell=1}^{L}a_\ell\right)^{2/L} = L\left(\prod_{\ell=1}^{L}\|\mathbf{W}^{(\ell)}\|_F\right)^{2/L}$$

and equality is reached when $\alpha_1 = \alpha_2 = \dots = \alpha_L$. In other word, in the problem (2), it suffices to consider the network that satisfies

$$\|\mathbf{W}_j^{(1)}\|_F = \|\mathbf{W}_j^{(2)}\|_F = \dots = \|\mathbf{W}_j^{(L)}\|_F, \forall j \in [M], \ell \in [L]. \tag{12}$$

Using (4) again, one can find that the neural network is also equivalent to

$$f(x) = \sum_{j=1}^{M} a_j\bar{\mathbf{W}}^{(L)}\sigma(\bar{\mathbf{W}}_j^{(L-1)}\sigma(\dots\sigma(\bar{\mathbf{W}}_j^{(1)}x + \bar{\boldsymbol{b}}_j^{(1)})\dots) + \bar{\boldsymbol{b}}_j^{(L-1)}) + \bar{\boldsymbol{b}}_j^{(L)},$$

where

$$\|\bar{\mathbf{W}}_j^{(\ell)}\|_F \le \beta^{(\ell)}, a_j = \frac{\prod_{\ell=1}^{L}\|\mathbf{W}_j^{(\ell)}\|_F}{\prod_{\ell=1}^{L}\beta^{(\ell)}} = \frac{\|\mathbf{W}_j^{(1)}\|_F^L}{\prod_{\ell=1}^{L}\beta^{(\ell)}} = \frac{(\sum_{\ell=1}^{L}\|\mathbf{W}_j^{(\ell)}\|_F^2/L)^{L/2}}{\prod_{\ell=1}^{L}\beta^{(\ell)}}, \tag{13}$$

where the last two equality comes from the assumption (12). Choosing $\beta^{(\ell)} = c_1\sqrt{w}$ expect $\ell = 1$ where $\beta^{(1)} = c_1\sqrt{d}$, and scaling $\bar{\boldsymbol{b}}^{(\ell)}$ accordingly and taking the constraint in (11) into (13) finishes the proof. □

D.2 Covering Number of Parallel Neural Networks

Theorem 4. *The covering number of the model defined in (5) apart from the bias in the last layer satisfies*

$$\log\mathcal{N}(\mathcal{F},\delta) \lesssim w^{2+2/(1-2/L)}L^2\sqrt{d}P'^{\frac{1}{1-2/L}}\delta^{-\frac{2/L}{1-2/L}}\log(wP'/\delta).$$

The proof relies on the covering number of each subnetwork in a parallel neural network (Lemma 10), observing that $|f(x)| \le 2^{L-1}w^{L-1}\sqrt{d}$ under the condition in Lemma 10, and then apply Lemma 5. We argue that our choice of condition on $\|\boldsymbol{b}^{(\ell)}\|_2$ in Lemma 10 is sufficient to analyzing the model apart from the bias in the last layer, because it guarantees that $\sqrt{w}\|\mathbf{W}^{(\ell)}\mathcal{A}_{\ell-1}(x)\|_2 \le \|\boldsymbol{b}^{(\ell)}\|_2$. This leads to

$$\|\mathbf{W}^{(\ell)}\mathcal{A}_{\ell-1}(\boldsymbol{x})\|_\infty \le \|\mathbf{W}^{(\ell)}\mathcal{A}_{\ell-1}(\boldsymbol{x})\|_2 \le \sqrt{w}\|\boldsymbol{b}^{(\ell)}\|_2 \le \|\boldsymbol{b}^{(\ell)}\|_\infty$$

If this condition is not met, $\mathbf{W}^{(\ell)}\mathcal{A}_{\ell-1}(\boldsymbol{x}) + b^{(\ell)}$ is either always positive or always negative for all feasible $\boldsymbol{x}$ along at least one dimension. If $(\mathbf{W}^{(\ell)}\mathcal{A}_{\ell-1}(\boldsymbol{x}) + b^{(\ell)})_i$ is always negative, one can replace $b^{(\ell)})_i$ with $-\max_{\boldsymbol{x}} \|\mathbf{W}^{(\ell)}\mathcal{A}_{\ell-1}(\boldsymbol{x})\|_\infty$ without changing the output of this model for any feasible $\boldsymbol{x}$. If $(\mathbf{W}^{(\ell)}\mathcal{A}_{\ell-1}(\boldsymbol{x}) + b^{(\ell)})_i$ is always positive, one can replace $b^{(\ell)})_i$ with $\max_{\boldsymbol{x}} \|\mathbf{W}^{(\ell)}\mathcal{A}_{\ell-1}(\boldsymbol{x})\|_\infty$, and adjust the bias in the next layer such that the output of this model is not changed for any feasible $\boldsymbol{x}$. In either cases, one can replace the bias $\boldsymbol{b}^{(\ell)}$ with another one with smaller norm while keeping the model equivalent except the bias in the last layer.

Lemma 10. *Let $\mathcal{F} \subseteq \{f : R^d \to \mathbb{R}\}$ denote the set of L-layer neural network (or a subnetwork in a parallel neural network) with width w in each hidden layer. It has the form*

$$\begin{aligned}
&f(x) = \mathbf{W}^{(L)}\sigma(\mathbf{W}^{(L-1)}\sigma(\dots\sigma(\mathbf{W}^{(1)}x + \boldsymbol{b}^{(1)})\dots) + \boldsymbol{b}^{(L-1)}) + \boldsymbol{b}^{(L)},\\
&\mathbf{W}^{(1)} \in \mathbb{R}^{w\times d}, \|\mathbf{W}^{(1)}\|_F \le \sqrt{d}, \boldsymbol{b}^{(1)} \in \mathbb{R}^w, \|\boldsymbol{b}^{(1)}\|_2 \le \sqrt{dw},\\
&\mathbf{W}^{(\ell)} \in \mathbb{R}^{w\times w}\|\mathbf{W}^{(\ell)}\|_F \le \sqrt{w}, \boldsymbol{b}^{(\ell)} \in \mathbb{R}^w, \|\boldsymbol{b}^{(\ell)}\|_2 \le 2^{\ell-1}w^{\ell-1}\sqrt{dw}, \quad \forall \ell = 2,\dots L-1,\\
&\mathbf{W}^{(L)} \in \mathbb{R}^{1\times w}, \|\mathbf{W}^{(L)}\|_F \le \sqrt{w}, b^{(L)} = 0
\end{aligned} \tag{14}$$

and $\sigma(\cdot)$ is the ReLU activation function, the input satisfy $\|x\|_2 \le 1$, then the supremum norm δ-covering number of $\mathcal{F}$ obeys

$$\log \mathcal{N}(\mathcal{F}, \delta) \le c_7 L w^2 \log(1/\delta) + c_8$$

where c_7 is a constant depending only on d, and c_8 is a constant that depend on d, w and L.

Proof. First study two neural networks which differ by only one layer. Let g_ℓ, g'_ℓ be two neural networks satisfying (14) with parameters $\mathbf{W}_1, \boldsymbol{b}_1, \dots, \mathbf{W}_L, \boldsymbol{b}_L$ and $\mathbf{W}'_1, \boldsymbol{b}'_1, \dots, \mathbf{W}'_L, \boldsymbol{b}'_L$ respectively. Furthermore, the parameters in these two models are the same except the ℓ-th layer, which satisfy

$$\|\mathbf{W}_\ell - \mathbf{W}'_\ell\|_F \le \epsilon, \|\boldsymbol{b}_\ell - \boldsymbol{b}'_\ell\|_2 \le \tilde{\epsilon}.$$

Denote the model as

$$g_\ell(x) = \mathcal{B}_\ell(\mathbf{W}_\ell\mathcal{A}_\ell(\boldsymbol{x}) + \boldsymbol{b}_\ell), g'_\ell(x) = \mathcal{B}_\ell(\mathbf{W}'_\ell\mathcal{A}_\ell(\boldsymbol{x}) + \boldsymbol{b}'_\ell)$$

where $\mathcal{A}_\ell(\boldsymbol{x}) = \sigma(\mathbf{W}_{\ell-1}\sigma(\dots\sigma(\mathbf{W}_1 x + \boldsymbol{b}_1)\dots) + \boldsymbol{b}_{\ell-1})$ denotes the first $\ell-1$ layers in the neural network, and $\mathcal{A}_\ell(x) = \mathbf{W}_L\sigma(\dots\sigma(\mathbf{W}_{\ell+1}\sigma(x) + \boldsymbol{b}_{\ell+1})\dots) + \boldsymbol{b}_L)$ denotes the last $L-\ell-1$ layers, with definition $\mathcal{A}_1(\boldsymbol{x}) = \boldsymbol{x}, \mathcal{B}_L(\boldsymbol{x}) = \boldsymbol{x}$.

Now focus on bounding $\|\mathcal{A}(\boldsymbol{x})\|$. Let $\mathbf{W} \in \mathbb{R}^{m\times m'}, \|\mathbf{W}\|_F \le \sqrt{m'}, \boldsymbol{x} \in \mathbb{R}^{m'}, \boldsymbol{b} \in \mathbb{R}^m, \|\boldsymbol{b}\|_2 \le \sqrt{m}$

$$\begin{aligned}
\|\sigma(\mathbf{W}x + \boldsymbol{b})\|_2 &\le \|\mathbf{W}\boldsymbol{x} + \boldsymbol{b}\|_2\\
&\le \|\mathbf{W}\|_2\|\boldsymbol{x}\|_2 + \|\boldsymbol{b}\|_2\\
&\le \|\mathbf{W}\|_F\|\boldsymbol{x}\|_2 + \|\boldsymbol{b}\|_2\\
&\le \sqrt{m'}\|\boldsymbol{x}\|_2 + \sqrt{m}
\end{aligned}$$

where we make use of $\|\cdot\|_2 \le \|\cdot\|_F$. Because of that,

$$\begin{aligned}
&\|\mathcal{A}_2(\boldsymbol{x})\|_2 \le \sqrt{d} + \sqrt{dw} \le 2\sqrt{dw},\\
&\|\mathcal{A}_3(\boldsymbol{x})\|_2 \le \sqrt{w}\|\mathcal{A}_2(\boldsymbol{x})\|_2 + 2w\sqrt{dw} \le 4w\sqrt{dw},\\
&\quad\cdots\\
&\|\mathcal{A}_\ell(\boldsymbol{x})\|_2 \le \sqrt{w}\|\mathcal{A}_{\ell-1}(\boldsymbol{x})\|_2 \le 2\sqrt{dw}(2w)^{\ell-2}.
\end{aligned} \tag{15}$$

Then focus on $\mathcal{B}(\boldsymbol{x})$. Let $\mathbf{W} \in \mathbb{R}^{m\times m'}, \|\mathbf{W}\|_F \le \sqrt{m'}, \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{m'}, \boldsymbol{b} \in \mathbb{R}^m, \|\boldsymbol{b}\|_2 \le \sqrt{m}$. Furthermore, $\|\boldsymbol{x} - \boldsymbol{x}'\|_2 \le \epsilon$, then

$$\|\sigma(\mathbf{W}\boldsymbol{x} + \boldsymbol{b}) - \sigma(\mathbf{W}\boldsymbol{x}' + \boldsymbol{b})\|_2 \le \|\mathbf{W}(\boldsymbol{x} - \boldsymbol{x}')\|_2 \le \|\mathbf{W}\|_F\|\boldsymbol{x} - \boldsymbol{x}'\|_2$$

which indicates that $\|\mathcal{B}(\boldsymbol{x}) - \mathcal{B}(\boldsymbol{x})'\|_2 \le (\sqrt{w})^{L-\ell}\|\boldsymbol{x} - \boldsymbol{x}'\|_2$

Finally, for any $\mathbf{W}, \mathbf{W}' \in \mathbb{R}^{m\times m'}, \boldsymbol{x} \in \mathbb{R}^{m'}, \boldsymbol{b}, \boldsymbol{b}' \in \mathbb{R}^m$, one have

$$\begin{aligned}
\|(\mathbf{W}\boldsymbol{x} + \boldsymbol{b}) - (\mathbf{W}'\boldsymbol{x} + \boldsymbol{b}')\|_2 &= \|(\mathbf{W} - \mathbf{W}')\boldsymbol{x} + (\boldsymbol{b} - \boldsymbol{b}')\|_2\\
&\le \|\mathbf{W} - \mathbf{W}'\|_2\|\boldsymbol{x}\|_2 + \|\boldsymbol{b} - \boldsymbol{b}'\|_2.\\
&\le \|\mathbf{W} - \mathbf{W}'\|_F\|\boldsymbol{x}\|_2 + \sqrt{m}\|\boldsymbol{b} - \boldsymbol{b}'\|_\infty.
\end{aligned}$$

In summary,

$$\begin{aligned}|g_\ell(\boldsymbol{x}) - g'_\ell(\boldsymbol{x})| &= |\mathcal{B}_\ell(\mathbf{W}_\ell \mathcal{A}_\ell(\boldsymbol{x}) + \boldsymbol{b}_\ell) - \mathcal{B}_\ell(\mathbf{W}'_\ell \mathcal{A}_\ell(\boldsymbol{x}) + \boldsymbol{b}'_\ell)| \\ &\le (\sqrt{w})^{L-\ell} \|(\mathbf{W}_\ell \mathcal{A}_\ell(\boldsymbol{x}) + \boldsymbol{b}_\ell) - (\mathbf{W}'_\ell \mathcal{A}_\ell(\boldsymbol{x}) + \boldsymbol{b}'_\ell)\|_2 \\ &\le (\sqrt{w})^{L-\ell} (\|\mathbf{W}_\ell - \mathbf{W}'_\ell\|_F \|\mathcal{A}_\ell(\boldsymbol{x})\|_2 + \|\boldsymbol{b}_\ell - \boldsymbol{b}'_\ell\|_2) \\ &\le 2^{(\ell-1)} w^{(L+\ell-3)/2} d^{1/2} \epsilon + w^{(L-\ell)/2} \bar{\epsilon}\end{aligned}$$

Let $f(x), f'(x)$ be two neural networks satisfying (14) with parameters $W_1, b_1, \ldots, W_L, b_L$ and $W'_1, b'_1, \ldots, W'_L, b'_L$ respectively, and $\|W_\ell - W'_\ell\|_F \le \epsilon_\ell, \|b_\ell - b'_\ell\|_F \le \tilde{\epsilon}_\ell$. Further define f_ℓ be the neural network with parameters $W_1, b_1, \ldots, W_\ell, b_\ell, W'_{\ell+1}, b'_{\ell+1}, \ldots, W'_L, b'_L$, then

$$\begin{aligned}|f(x) - f'(x)| &\le |f(x) - f_1(x)| + |f_1(x) - f_2(x)| + \cdots + |f_{L-1}(x) - f'(x)| \\ &\le \sum_{\ell=1}^{L} 2^{(\ell-2)} d^{1/2} w^{(L+\ell-3)/2} \epsilon + w^{(L-\ell)/2} \bar{\epsilon}\end{aligned}$$

For any $\delta > 0$, one can choose

$$\epsilon_\ell = \frac{\delta}{2^\ell w^{(L+\ell-3)/2} d^{1/2}}, \tilde{\epsilon}_\ell = \frac{\delta}{2 w^{(L-\ell)/2}}$$

such that $|f(x) - f'(x)| \le \delta$.

On the other hand, the ϵ-covering number of $\{\mathbf{W} \in \mathbb{R}^{m \times m'} : \|\mathbf{W}\|_F \le \sqrt{m'}\}$ on Frobenius norm is no larger than $(2\sqrt{m'}/\epsilon + 1)^{m \times m'}$, and the $\bar{\epsilon}$-covering number of $\{\boldsymbol{b} \in \mathbb{R}^m : \|\boldsymbol{b}\|_2 \le 1\}$ on infinity norm is no larger than $(2/\bar{\epsilon} + 1)^m$. The entropy of this neural network can be bounded by

$$\log \mathcal{N}(f; \delta) \le w^2 L \log(2^{L+1} w^{L-1}/\delta + 1) + wL \log(2^{L-1} w^{(L-1)/2} d^{1/2}/\delta + 1)$$

□

D.3 Covering Number of p-Norm Constrained Linear Combination

Lemma 5. *$\log \mathcal{N}(\mathcal{G}, \delta) \lesssim k \log(1/\delta)$ for some finite c_3, and for any $g \in \mathcal{G}, |a| \le 1$, we have $ag \in \mathcal{G}$. The covering number of $\mathcal{F} = \left\{\sum_{i=1}^{M} a_i g_i \middle| g_i \in \mathcal{G}, \|a\|_p^p \le P, 0 < p < 1\right\}$ for any $P > 0$ satisfies*

$$\log \mathcal{N}(\mathcal{F}, \epsilon) \lesssim k P^{\frac{1}{1-p}} (\delta/c_3)^{-\frac{p}{1-p}} \log(c_3 P/\delta)$$

up to a double logarithmic factor.

Proof. Let ϵ be a positive constant. Without the loss of generality, we can sort the coefficients in descending order in terms of their absolute values. There exists a positive integer $\mathcal{M}$ (as a function of ϵ), such that $|a_i| \ge \epsilon$ for $i \le \mathcal{M}$, and $|a_i| < \epsilon$ for $i > \mathcal{M}$.

By definition, $\mathcal{M}\epsilon^p \le \sum_{i=1}^{\mathcal{M}} |a_i|^p \le P$ so $\mathcal{M} \le P/\epsilon^p$, and $|a_i|^p \le P, |a_i| \le P^{1/p}$ for all i. Furthermore,

$$\sum_{i>m} |a_i| = \sum_{i>\mathcal{M}} |a_i|^p |a_i|^{1-p} < \sum_{i>\mathcal{M}} |a_i|^p \epsilon^{1-p} \le P\epsilon^{1-p}$$

Let $\tilde{g}_i = \arg\min_{g \in \tilde{\mathcal{G}}} \|g - \frac{a_i}{P^{1/p}} g_i\|_\infty$ where $\tilde{\mathcal{G}}$ is the δ'-convering set of $\mathcal{G}$. By definition of the covering set,

$$\begin{aligned}\left\|\sum_{i=1}^{M} a_i g_i(x) - \sum_{i=1}^{\mathcal{M}} P^{1/p} \tilde{g}_i(x)\right\|_\infty &\le \left\|\sum_{i=1}^{\mathcal{M}} (a_i g_i(x) - P^{1/p} \tilde{g}_i(x))\right\|_\infty + \left\|\sum_{i=\mathcal{M}+1}^{M} a_i g_i(x)\right\|_\infty \\ &\le \mathcal{M} P^{1/p} \delta' + c_3 P \epsilon^{1-p}.\end{aligned} \tag{16}$$

Choosing

$$\epsilon = (\delta/2c_3P)^{\frac{1}{1-p}}, \delta' \approx P^{-\frac{1}{p(1-p)}} (\delta/2c_3)^{\frac{1}{1-p}}/2, \tag{17}$$

we have $\mathcal{M} \le P^{\frac{1}{1-p}} (\delta/2c_3)^{-\frac{p}{1-p}}, \mathcal{M} P^{1/p} \delta' \le \delta/2, c_3 P \epsilon^{1-p} \le \delta/2$, so (16) $\le \delta$. One can compute the covering number of $\mathcal{F}$ by

$$\log \mathcal{N}(\mathcal{F}, \delta) \le \mathcal{M} \log \mathcal{N}(\mathcal{G}, \delta') \lesssim k\mathcal{M} \log(1/\delta') \tag{18}$$

Taking (17) into (18) finishes the proof. □

E Proof of Approximation Error

E.1 Approximation of Neural Networks to B-spline Basis Functions

Proposition 6. *There exists a parallel neural network that has the structure and satisfy the constraint in Proposition 3 for d-dimensional input and one output, containing $M = O(m^d)$ subnetworks, each of which has width $w = O(d)$ and depth $L = O(\log(c(m,d)/\epsilon))$ for some constant w, c that depends only on m and d, denoted as $\tilde{M}_m(\boldsymbol{x}), \boldsymbol{x} \in \mathbb{R}^d$, such that*

- $|\tilde{M}_{m,k,\boldsymbol{s}}(\boldsymbol{x}) - M_{m,k,\boldsymbol{s}}(\boldsymbol{x})| \leq \epsilon$, *if* $0 \leq 2^k(x_i - s_i) \leq m+1, \forall i \in [d]$,
- $\tilde{M}_{m,k,\boldsymbol{s}}(\boldsymbol{x}) = 0$, *otherwise.*
- *The weights in the last layer satisfy* $\|a\|_{2/L}^{2/L} \lesssim 2^k m^d e^{2md/L}$.

We follow the method developed in Yarotsky [46], Suzuki [36], while putting our attention on bounding the Frobenius norm of the weights.

Lemma 11 (Yarotsky [46, Proposition 3])**.** *: There exists a neural network with two-dimensional input and one output $f_\times(x,y)$, with constant width and depth $O(\log(1/\delta))$, and the weight in each layer is bounded by a global constant c_1, such that*

- $|f_\times(x,y) - xy| \leq \delta, \forall\, 0 \leq x, y \leq 1$,
- $f_\times(x,y) = 0, \forall\, x = 0$ *or* $y = 0$.

We first prove a special case of Proposition 6 on the unscaled, unshifted B-spline basis function by fixing $k = 0, \boldsymbol{s} = 0$:

Proposition 12. *There exists a parallel neural network that has the structure and satisfy the constraint in Proposition 3 for d-dimensional input and one output, containing $M = \lceil (m+1)/2 \rceil^d = O(m^d)$ subnetworks, each of which has width $w = O(d)$ and depth $L = O(\log(c(m,d)/\epsilon))$ for some constant w, c that depends only on m and d, denoted as $\tilde{M}_m(\boldsymbol{x}), \boldsymbol{x} \in \mathbb{R}^d$, such that*

- $|\tilde{M}_m(\boldsymbol{x}) - M_m(\boldsymbol{x})| \leq \epsilon$, *if* $0 \leq x_i \leq m+1, \forall i \in [d]$, *while $M_m(\cdot)$ denote m-th order B-spline basis function, and c only depends on m and d.*
- $\tilde{M}_m(\boldsymbol{x}) = 0$, *if* $x_i \leq 0$ *or* $x_i \geq m+1$ *for any* $i \in [d]$.
- *The weights in the last layer satisfy* $\|a\|_{2/L}^{2/L} \lesssim m^d e^{2md/L}$.

Proof. We first show that one can use a neural network with constant width w_0, depth $L \eqsim \log(m/\epsilon_1)$ and bounded norm $\|W^{(1)}\|_F \leq O(\sqrt{d}), \|W^{(\ell)}\|_F \leq O(\sqrt{w}), \forall \ell = 2, \dots, L$ to approximate truncated power basis function up to accuracy ϵ_1 in the range $[0,1]$. Let $m = \sum_{i=0}^{\lceil \log_2 m \rceil} m_i 2^i, m_i \in \{0,1\}$ be the binary digits of m, and define $\bar{m}_j = \sum_{j=0}^{i} m_i, \gamma = \lceil \log_2 m \rceil$, then for any x

$$
\begin{aligned}
x_+^m &= x_+^{\bar{m}_\gamma} \times \left(x_+^{2^\gamma}\right)^{m_\gamma} \\
[x_+^{\bar{m}_\gamma}, x_+^{2^\gamma}] &= [x_+^{\bar{m}_{\gamma-1}} \times \left(x_+^{2^{\gamma-1}}\right)^{m_{\gamma-1}}, x_+^{2^{\gamma-1}} \times x_+^{2^{\gamma-1}}] \\
&\dots \\
[x_+^{\bar{m}_2}, x_+^4] &= [x_+^{\bar{m}_1} \times \left(x_+^2\right)^{m_1}, x_+^2 \times x_+^2] \\
[x_+^{\bar{m}_1}, x_+^2] &= [x_+^{\bar{m}_0} \times x_+^{m_0}, x_+ \times x_+]
\end{aligned}
\tag{19}
$$

Notice that each line of equation only depends on the line immediately below. Replacing the multiply operator $\times$ with the neural network approximation shown in Lemma 11 demonstrates the architecture of such neural network approximation. For any $x, y \in [0,1]$, let $|f_\times(x,y) - xy| \leq \delta, |x - \tilde{x}| \leq \delta_1, |y - \delta y| \leq \delta_2$, then $|f_\times(\tilde{x}, \tilde{y}) - xy| \leq \delta_1 + \delta_2 + \delta$. Taking this into (19) shows that $\epsilon_1 \eqsim 2^\gamma \delta \eqsim m\delta$, where ϵ_1 is the upper bound on the approximate error to truncated power basis of order m and δ is the approximation error to a single multiply operator as in Lemma 11.

A univariate B-spline basis can be expressed using truncated power basis, and observing that it is symmetric around $(m+1)/2$:

$$\begin{aligned}
M_m(x) &= \frac{1}{m!}\sum_{j=1}^{m+1}(-1)^j\binom{m+1}{j}(x-j)_+^m \\
&= \frac{1}{m!}\sum_{j=1}^{\lceil (m+1)/2\rceil}(-1)^j\binom{m+1}{j}(\min(x, m+1-x)-j)_+^m \\
&= \frac{((m+1)/2)^m}{m!}\sum_{j=1}^{\lceil (m+1)/2\rceil}(-1)^j\binom{m+1}{j}\left(\frac{\min(x, m+1-x)-j}{(m+1)/2}\right)_+^m,
\end{aligned}$$

A multivariate (d-dimensional) B-spline basis function can be expressed as the product of truncated power basis functions and thus can be decomposed as

$$\begin{aligned}
M_m(\boldsymbol{x}) &= \prod_{i=1}^{d} M_m(x_i) \\
&= \frac{((m+1)/2)^{md}}{(m!)^d}\prod_{i=1}^{d}\left(\sum_{j=1}^{\lceil (m+1)/2\rceil}(-1)^j\binom{m+1}{j}\left(\frac{\min(x_i, m+1-x)-j}{(m+1)/2}\right)_+^m\right) \\
&= \frac{((m+1)/2)^{md}}{(m!)^d}\sum_{j_1,\dots,j_d=1}^{\lceil (m+1)/2\rceil}\prod_{i=1}^{d}(-1)^{j_i}\binom{m+1}{j_i}\left(\frac{\min(x, m+1-x)-j_i}{(m+1)/2}\right)_+^m
\end{aligned} \tag{20}$$

Using Lemma 11, one can construct a parallel neural network containing $M = \lceil (m+1)/2\rceil^d = O(m^d)$ subnetworks, and each subnetwork corresponds to one polynomial term in (20). Using the results above, the approximation of this constructed neural network can be bounded by

$$\left(\sum_{i=1}^{m+1}\binom{m+1}{j}d(\epsilon_1+\delta)\right)^d \lesssim \frac{e^{2m}}{\sqrt{m}}d\epsilon_1 + d\delta$$

where we applied Stirling's approximation and δ and ϵ_1 has the same definition as above. Choosing $\delta = \frac{\epsilon}{d(e^{2m}\sqrt{m}+1)}$, and recall $\epsilon_1 \eqsim m\delta$ proves the approximation error.

To bound the norm of the factors $\|a\|_{2/L}^{2/L}$, first observe that

$$\begin{aligned}
|a_{j_1,\dots,j_d}| &= \frac{((m+1)/2)^{md}}{(m!)^d}\frac{1}{(m+1)/2}\prod_{i=1}^{d}\binom{m+1}{j_i} \\
&\le \frac{((m+1)/2)^{md}}{(m!)^d}\frac{2^{md}}{(m+1)/2} = O(e^{md})
\end{aligned}$$

where the first inequality is from $\binom{m+1}{j_i} \le 2^{m+1}$, the last equality is from Stirling's appropximation. Finally,

$$\|a\|_{2/L}^{2/L} \le m^d \max_j |a_j|^{2/L} \lesssim m^d e^{2md/L}$$

which finishes the proof. □

The proof of the Proposition 6 for general $k, \boldsymbol{s}$ follows by appending one more layer in the front, as we show below.

Proof of Proposition 6. Using the neural network proposed in Proposition 12, one can construct a neural network for appropximating $M_{m,k,\boldsymbol{s}}$ by adding one layer before the first layer:

$$\sigma(2^k\mathbf{I}_d\boldsymbol{x} - 2^k\boldsymbol{s})$$

The unused neurons in the first hidden layer is zero padded. The Frobenius norm of the weight is $2^k\|\mathbf{I}_d\|_F = 2^k\sqrt{d}$. Following the proof of Proposition 3, rescaling the weight in this layer by 2^{-k}, and the weight matrix in the last layer by 2^k, and scaling the bias properly, one can verify that this neural network satisfy the statement. □

E.2 Sparse approximation of Besov functions using B-spline wavelets

Proposition 7. *Let $\alpha - d/p > 1, r > 0$. Let $M_{m,k,\boldsymbol{s}}$ be the B-spline of order m with scale 2^{-k} in each dimension and position $\boldsymbol{s} \in \mathbb{R}^d$. For any function in Besov space $f_0 \in B^\alpha_{p,q}$ and any positive integer $\bar{M}$, there is an $\bar{M}$-sparse approximation using B-spline basis of order m satisfying $0 < \alpha < \min(m, m-1+1/p)$: $\check{f}_{\bar{M}} = \sum_{i=1}^{\bar{M}} a_{k_i,\boldsymbol{s}_i} M_{m,k_i,\boldsymbol{s}_i}$ for any positive integer $\bar{M}$ such that the approximation error is bounded as $\|\check{f}_{\bar{M}} - f_0\|_r \lesssim \bar{M}^{-\alpha/d}\|f_0\|_{B^\alpha_{p,q}}$, and the coefficients satisfy*

$$\|\{2^{k_i} a_{k_i,\boldsymbol{s}_i}\}_{k_i,\boldsymbol{s}_i}\|_p \lesssim \|f_0\|_{B^\alpha_{p,q}}.$$

The proof is divided into three steps:

1. Bound the 0-norm and the 1-norm of the coefficients of B-spline basis in order to approximate an arbitrary function in Besov space up to any $\epsilon > 0$.
2. Bound p-norm of the coefficients of B-spline basis functions where $0 < p < 1$ using the results above .
3. Add the approximation to neural network to B-spline basis computed in Section 4.3.1 into Step 2.

Proof. Dũng [11, Theorem 3.1] Suzuki [36, Lemma 2] proposed an adaptive sampling recovery method that approximates a function in Besov space. The method is divided into two cases: when $p \geq r$, and when $p < r$.

When $p \geq r$, there exists a sequence of scalars $\lambda_{\boldsymbol{j}}, \boldsymbol{j} \in P^d(\mu), P_d(\mu) := \{\boldsymbol{j} \in \mathbb{Z}^d : |j_i| \leq \mu, \forall i \in d\}$ for some positive μ, for arbitrary positive integer $\bar{k}$, the linear operator

$$Q_{\bar{k}}(f, \boldsymbol{x}) = \sum_{\boldsymbol{s} \in J(\bar{k},m,d)} a_{\bar{k},\boldsymbol{s}}(f) M_{\bar{k},\boldsymbol{s}}(\boldsymbol{x}), \quad a_{\bar{k},\boldsymbol{s}}(f) = \sum_{\boldsymbol{j} \in \mathbb{Z}^d, P^d(\mu)} \lambda_{\boldsymbol{j}} \bar{f}(\boldsymbol{s} + 2^{-\bar{k}}\boldsymbol{j})$$

has bounded approximation error

$$\|f - Q_{\bar{k}}(f, x)\|_r \leq C 2^{-\alpha\bar{k}} \|f\|_{B^\alpha_{p,q}},$$

where $\bar{f}$ is the extrapolation of f, $J(\bar{k}, m, d) := \{\boldsymbol{s} : 2^{\bar{k}}\boldsymbol{s} \in \mathbb{Z}^d, -m/2 \leq 2^{\bar{k}} s_i \leq 2^{\bar{k}} + m/2, \forall i \in [d]\}$. See Dũng [11, 2.6-2.7] for the detail of the extrapolation as well as references for options of sequence $\lambda_{\boldsymbol{j}}$.

Furthermore, $Q_{\bar{k}}(f) \in B^\alpha_{p,q}$ so it can be decomposed in the form (10) with $M = \sum_{k=0}^{\bar{k}} (2^k + m - 1)^d \lesssim 2^{\bar{k}d}$ components and $\|\{\tilde{c}_{k,\boldsymbol{s}}\}_{k,\boldsymbol{s}}\| \lesssim \|Q_{\bar{k}}(f)\|_{B^\alpha_{p,q}} \lesssim \|f\|_{B^\alpha_{p,q}}$ where $\tilde{c}_{k,\boldsymbol{s}}$ is the coefficients of the decomposition of $Q_{\bar{k}}(f)$. Choosing $\bar{k} \eqsim \log_2 M/d$ leads to the desired approximation error.

On the other hand, when $p < r$, there exists a greedy algorithm that constructs

$$G(f) = Q_{\bar{k}}(f) + \sum_{k=\bar{k}+1}^{k^*} \sum_{j=1}^{n_k} c_{k,\boldsymbol{s}_j}(f) M_{k,\boldsymbol{s}_j}$$

where $\bar{k} \eqsim \log_2(M), k^* = [\epsilon^{-1}\log(\lambda M)] + \bar{k} + 1, n_k = [\lambda M 2^{-\epsilon(k-\bar{k})}]$ for some $0 < \epsilon < \alpha/\delta - 1, \delta = d(1/p - 1/r), \lambda > 0$, such that

$$\|f - G(f)\|_r \leq \bar{M}^{-\alpha/d} \|f\|_{B^\alpha_{p,q}}$$

and

$$\sum_{k=0}^{\bar{k}} (2^k + m - 1)^d + \sum_{k=\bar{k}+1}^{k^*} n_k \leq \bar{M}.$$

See Dũng [11, Theorem 3.1] for the detail.

Finally, since $\alpha - d/p > 1$,

$$\begin{aligned}
\|\{2^{k_i} c_{k_i,\boldsymbol{s}_i}\}_{k_i,\boldsymbol{s}_i}\|_p &\le \sum_{k=0}^{\bar{k}} 2^k \|\{c_{k_i,\boldsymbol{s}_i}\}_{\boldsymbol{s}_i}\|_p \\
&= \sum_{k=0}^{\bar{k}} 2^{(1-(\alpha-d/p))k} (2^{(\alpha-d/p)k} \|\{c_{k_i,\boldsymbol{s}_i}\}_{\boldsymbol{s}_i}\|_p) \\
&\lesssim \sum_{k=0}^{\bar{k}} 2^{(1-(\alpha-d/p))k} \|f\|_{B^\alpha_{p,q}} \\
&\eqsim \|f\|_{B^\alpha_{p,q}}
\end{aligned} \tag{21}$$

where the first line is because for arbitrary vectors $\boldsymbol{a}_i, i \in [n]$, $\|\sum_{i=1}^n \boldsymbol{a}_i\|_p \le \sum_{i=1}^n \|\boldsymbol{a}_i\|_p$, the third line is because the sequence norm of B-spline decomposition is equivalent to the norm in Besov space (see Section C.1) . □

Note that when $\alpha - d/p = 1$, the sequence norm (21) is bounded (up to a factor of constant) by $k^* \|f\|_{B^\alpha_{p,q}}$, which can be proven by following (21) except the last line. This adds a logarithmic term with respect to $\bar{M}$ compared with the result in Proposition 7. This will add a logarithmic factor to the MSE. We will not focus on this case in this paper of simplicity.

E.3 Sparse approximation of Besov functions using Parallel Neural Networks

Theorem 8. *Under the same condition as Proposition 7, for any positive integer $\bar{M}$, any function in Besov space $f_0 \in B^\alpha_{p,q}$ can be approximated by a parallel neural network with no less than $O(m^d \bar{M})$ number of subnetworks satisfying:*

1. *Each subnetwork has width $w = O(d)$ and depth L.*
2. *The weights in each layer satisfy $\|\bar{\mathbf{W}}^{(\ell)}_k\|_F \le O(\sqrt{w})$ except the first layer $\|\bar{\mathbf{W}}^{(1)}_k\|_F \le O(\sqrt{d})$,*
3. *The scaling factors have bounded $2/L$-norm: $\|\{a_j\}\|^{2/L}_{2/L} \lesssim m^d e^{2md/L} \bar{M}^{1-2/(pL)}$.*
4. *The approximation error is bounded by*

$$\|\tilde{f} - f_0\|_r \le (c_4 \bar{M}^{-\alpha/d} + c_5 e^{-c_6 L}) \|f\|_{B^\alpha_{p,q}}$$

where c_4, c_5, c_6 are constants that depend only on m, d and p.

We first prove the following lemma.

Lemma 13. *For any $a \in \mathbb{R}^{\bar{M}}$, $0 < p' < p$, it holds that:*

$$\|a\|^{p'}_{p'} \le \bar{M}^{1-p'/p} \|a\|^{p'}_p.$$

Proof.

$$\sum_i |a_i|^{p'} = \langle \mathbf{1}, |\boldsymbol{a}|^{p'} \rangle \le \left(\sum_i 1 \right)^{1-\frac{p'}{p}} \left(\sum_i (|a_i|^{p'})^{\frac{p}{p'}} \right)^{\frac{p'}{p}} = \bar{M}^{1-\frac{p'}{p}} \|a\|^{p'}_p$$

The first inequality uses a Holder's inequality with conjugate pair $\frac{p}{p'}$ and $1/(1 - \frac{p'}{p})$. □

Proof of Theorem 8. Using Proposition 7, one can construct $\bar{M}$ number of PNN each $O(m^d)$ subnetworks according to Proposition 6, and in each PNN, such that each PNN represents one B-spline basis function.The weights in the last layer of each PNN is scaled to match the coefficients in Proposition 7. Taking p' in Lemma 13 as $2/L$ and combining with Proposition 6 finishes the proof. □

F Proof of the Main Theorem

Theorem 1 extended form. *For any fixed $\alpha - d/p > 1, r > 0, L \geq 3$, given an L-layer parallel neural network satisfying*

- *The width of each subnetwork is fixed and large enough: $w \gtrsim d$. See Theorem 8 for the detail.*
- *The number of subnetworks is large enough: $M \gtrsim m^d n^{\frac{1-2/L}{2\alpha/d+1-2/(pL)}}$.*

With proper choice of the parameter of weight decay λ, the solution $\hat{f}$ parameterized by (2) satisfies

$$\mathrm{MSE}(\hat{f}) = \tilde{O}\left(\Big(\frac{w^{4-4/L}L^{2-4/L}}{n^{1-2/L}}\Big)^{\frac{2\alpha/d}{2\alpha/d+1-2/(pL)}} + e^{-c_6 L}\right)$$

where $\tilde{O}$ shows the scale up to a logarithmic factor, and c_6 is the constant defined in Theorem 8.

Proof. First recall the relationship between covering number (entropy) and estimation error:

Proposition 14. *Let $\mathcal{F} \subseteq \{\mathbb{R}^d \to [-F, F]\}$ be a set of functions. Assume that $\mathcal{F}$ can be decomposed into two orthogonal spaces $\mathcal{F} = \mathcal{F}_{\|} \times \mathcal{F}_{\perp}$ where $\mathcal{F}_{\perp}$ is an affine space with dimension of N. Let $f_0 \in \{\mathbb{R}^d \to [-F, F]\}$ be the target function and $\hat{f}$ be the least squares estimator in $\mathcal{F}$:*

$$\hat{f} = \arg\min_{f\in\mathcal{F}} \sum_{i=1}^{n} (y_i - f(x_i))^2, y_i = f_0(x_i) + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, \sigma^2) i.i.d.,$$

then it holds that

$$\mathrm{MSE}(\hat{f}) \leq \tilde{O}\Big(\arg\min_{f\in\mathcal{F}} \mathrm{MSE}(f) + \frac{N + \log \mathcal{N}(\mathcal{F}_{\|}, \delta) + 2}{n} + (F + \sigma)\delta\Big).$$

The proof of Proposition 14 is defered to the section below. We choose $\mathcal{F}$ as the set of functions that can be represented by a parallel neural network as stated, the (null) space $\mathcal{F}_{\perp} = \{f : f(\boldsymbol{x}) = constant\}$ be the set of functions with constant output, which has dimension 1. This space captures the bias in the last layer, while the other parameters contributes to the projection in $\mathcal{F}_{\|}$. See Section D.2 for how we handle the bias in the other layers. One can find that $\mathcal{F}_{\|}$ is the set of functions that can be represented by a parallel neural network as stated, and further satisfy $\sum_{i=1}^{n} f(\boldsymbol{x}_i) = 0$. Because $\mathcal{F}_{\|} \subseteq \mathcal{F}, \mathcal{N}(\mathcal{F}_{\|}, \delta) \leq \mathcal{N}(\mathcal{F}, \delta)$ for all $\delta > 0$, and the latter is studied in Theorem 4.

In Theorem 1, the width of each subnetwork is no less than what is required in Theorem 8, while the depth and norm constraint are the same, so the approximation error is no more that that in Theorem 8. Choosing $r = 2, p = 2/L$, and taking Theorem 4 and Theorem 8 into this Proposition 14, one gets

$$\mathrm{MSE}(\hat{f}) \lesssim \bar{M}^{-2\alpha/d} + \frac{w^{2+2/(1-2/L)}L^2}{n} \bar{M}^{\frac{1-2/(pL)}{1-2/L}} \delta^{-\frac{2/L}{1-2/L}} (\log(\bar{M}/\delta) + 3) + \delta,$$

where $\|f\|_{B^{\alpha}_{p,q}}, m$ and d taken as constants. The stated MSE is obtained by choosing

$$\delta \asymp \frac{w^{4-4/L}L^{2-4/L}\bar{M}^{1-2/(pL)}}{n^{1-2/L}}, \bar{M} \asymp \Big(\frac{n^{1-2/L}}{w^{4-4/L}L^{2-4/L}}\Big)^{\frac{1}{2\alpha/d+1-2/(pL)}}$$

Note that there exists a weight decay parameter λ' such that the $(2/L)$-norm of the coefficients of the parallel neural network satisfy that $\|\{a_j\}\|_{2/L}^{2/L} = m^d e^{2md/L} \|\{\tilde{a}_{j,\bar{M}}\}\|_{2/L}^{2/L}$ where $\{\tilde{a}_{j,\bar{M}}\}$ is the coefficient of the particular $\bar{M}$-sparse approximation, although $\{a_j\}$ is not necessarily $\bar{M}$ sparse. Empirically, one only need to guarantee that during initialization, the number of subnetworks $M \geq \bar{M}$ such that the $\bar{M}$-sparse approximation is feasible, thus the approximation error bound from Theorem 8 can be applied. Theorem 8 also says that $\|\{a_j\}\|_{2/L}^{2/L} = m^d e^{2md/L} \|\{\tilde{a}_{j,\bar{M}}\}\|_{2/L}^{2/L} \lesssim \bar{M}^{1-2/pL}$, thus we can apply the covering number bound from Theorem 4 with $P' = \bar{M}^{1-2/pL}$. Finally, if λ is optimally chosen, then it achieves a smaller MSE than this particular λ', which has been proven to be no more than $O(\bar{M}^{-\alpha/d})$ and completes the proof.

□

Proof of Proposition 14. For any function $f \in \mathcal{F}$, define $f_\perp = \arg\min_{h\in\mathcal{F}_\perp}\sum_{i=1}^n (f(\boldsymbol{x}_i) - h(\boldsymbol{x}_i))^2$ be the projection of f to $\mathcal{F}_\perp$, and define $f_\| = f - f_\perp$ be the projection to the orthogonal complement. Note that $f_\|$ is not necessarily in $\mathcal{F}_\|$. However, if $f \in \mathcal{F}$, then $f_\| \in \mathcal{F}_\|$. $y_{i\perp}$ and $y_{i\|}$ are defined by creating a function f_y such that $f_y(\boldsymbol{x}_i) = y_i, \forall i$, e.g. via interpolation. Because $\mathcal{F}_\|$ and $\mathcal{F}_\perp$ are orthononal, the empirical loss and population loss can be decomposed in the same way:

$$L_\|(f) = \frac{1}{n}\sum_{i=1}^n (f_\|(\boldsymbol{x}) - f_{0\|}(\boldsymbol{x}))^2 + \frac{n-N}{n}\sigma^2, \qquad L_\perp(f) = \frac{1}{n}\sum_{i=1}^n (f_\perp(\boldsymbol{x}) - f_{0\perp}(\boldsymbol{x}))^2 + \frac{N}{n}\sigma^2,$$
$$\hat{L}_\|(f) = \frac{1}{n}\sum_{i=1}^n (f_\|(\boldsymbol{x}) - y_{i\|})^2, \qquad \hat{L}_\perp(f) = \frac{1}{n}\sum_{i=1}^n (f_\perp(\boldsymbol{x}) - y_{i\perp}(\boldsymbol{x}))^2,$$
$$MSE_\|(f) = \mathbb{E}_\mathcal{D}\Big[\frac{1}{n}\sum_{i=1}^n (f_\|(\boldsymbol{x}) - f_{0\|}(\boldsymbol{x}))^2\Big], \qquad MSE_\perp(f) = \mathbb{E}_\mathcal{D}\Big[\frac{1}{n}\sum_{i=1}^n (f_\perp(\boldsymbol{x}) - f_{0\perp}(\boldsymbol{x}))^2\Big],$$

such that $L(f) = L_\|(f) + L_\perp(f), \hat{L}(f) = \hat{L}_\|(f) + \hat{L}_\perp(f)$. This can be verified by decomposing $\hat{f}, f_0$ and y into two orthogonal components as shown above, and observing that $\sum_{i=1}^n f_{1\perp}(\boldsymbol{x}_i) f_{2\|}(\boldsymbol{x}_i) = 0, \forall f_1, f_2$.

First prove the following claim

Claim 15. *Assume that $\hat{f} = \arg\min_{f\in\mathcal{F}} \hat{L}(f)$ is the empirical risk minimizer. Then $\hat{f}_\perp = \arg\min_{f\in\mathcal{F}_\perp} \hat{L}_\perp(f), \hat{f}_\| = \arg\min_{f\in\mathcal{F}_\|} \hat{L}_\|(f)$, where $\hat{f}_\perp$ is the projections of $\hat{f}$ in $\mathcal{F}_\perp$, and $\hat{f}_\| = \hat{f} - \hat{f}_\perp$ respectively.*

Proof. Since $\hat{f} \in \mathcal{F}$, by definition $\hat{f}_\| \in \mathcal{F}_\|$. Assume that there exist $\hat{f}'_\perp, \hat{f}'_\|$, and either $\hat{L}_\perp(\hat{f}'_\perp) < \hat{L}_\perp(\hat{f}_\perp)$, or $\hat{L}_\|(\hat{f}'_\|) < \hat{L}_\|(\hat{f}_\|)$. Then

$$\begin{aligned}\hat{L}(\hat{f}') &= \hat{L}(\hat{f}'_\perp + \hat{f}'_\|) = \hat{L}_\|(\hat{f}'_\perp + \hat{f}'_\|) + \hat{L}_\perp(\hat{f}'_\perp + \hat{f}'_\|) = \hat{L}_\|(\hat{f}'_\|) + \hat{L}_\perp(\hat{f}'_\perp)\\ &< \hat{L}_\|(\hat{f}_\|) + \hat{L}_\perp(\hat{f}_\perp) = \hat{L}_\|(\hat{f}_\perp + \hat{f}_\|) + \hat{L}_\perp(\hat{f}_\perp + \hat{f}_\|) = \hat{L}(\hat{f})\end{aligned}$$

which shows that $\hat{f}$ is not the minimizer of $\hat{L}(f)$ and violates the assumption.

□

Then we bound $MSE_\perp(f)$. We convert this part into a finite dimension least square problem:

$$\begin{aligned}\hat{f}_\perp &= \arg\min_{f\in\mathcal{F}_\perp} \hat{L}_\perp(f)\\ &= \arg\min_{f\in\mathcal{F}_\perp} \frac{1}{n}\sum_{i=1}^n (f(\boldsymbol{x}_i) - f_{0\perp}(\boldsymbol{x}_i) - \epsilon_{i\perp})^2\\ &= \arg\min_{f\in\mathcal{F}_\perp} \frac{1}{n}\sum_{i=1}^n (f(\boldsymbol{x}_i) - f_{0\perp}(\boldsymbol{x}_i) - \epsilon_{i\perp})^2 + \epsilon_{i\|}^2\\ &= \arg\min_{f\in\mathcal{F}_\perp} \frac{1}{n}\sum_{i=1}^n (f(\boldsymbol{x}_i) - f_{0\perp}(\boldsymbol{x}_i) - \epsilon_{i\perp} - \epsilon_{i\|})^2\\ &= \arg\min_{f\in\mathcal{F}_\perp} \frac{1}{n}\sum_{i=1}^n (f(\boldsymbol{x}_i) - f_{0\perp}(\boldsymbol{x}_i) - \epsilon_i)^2\end{aligned}$$

The forth line comes from our assumption that $\mathcal{F}_\perp$ is orthogonal to $\mathcal{F}_\|$, so $\forall f \in \mathcal{F}_\perp, f + f_{0\perp} + \epsilon_\perp$ is orthogonal to $\epsilon_\|$.

Let the basis function of $\mathcal{F}_\perp$ be $h_1, h_2, \ldots, h_N$, the above problem can be reparameterized as

$$\arg\min_{\boldsymbol{\theta}\in\mathbb{R}^N} \frac{1}{n}\|\mathbf{X}\boldsymbol{\theta} - \boldsymbol{y}\|^2$$

where $\mathbf{X} \in \mathbb{R}^{n\times N} : X_i = h_j(\boldsymbol{x}_i), \boldsymbol{y} = \boldsymbol{y}_{0\perp} + \boldsymbol{\epsilon}, \boldsymbol{y}_{0\perp} = [f_{0\perp}(x_1), \dots, f_{0\perp}(x_n)], \boldsymbol{\epsilon} = [\epsilon_1, \dots, \epsilon_n]$. This problem has a closed-form solution

$$\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}$$

Observe that $f_{0\perp} \in \mathcal{F}_\perp$, let $\boldsymbol{y}_{0\perp} = \mathbf{X}\boldsymbol{\theta}^*$,The MSE of this problem can be computed by

$$\begin{aligned} L(\hat{f}_\perp) &= \frac{1}{n}\|\mathbf{X}\boldsymbol{\theta} - \boldsymbol{y}_{0\perp}\|^2 = \frac{1}{n}\|\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\boldsymbol{\theta}^* + \boldsymbol{\epsilon}) - \mathbf{X}\boldsymbol{\theta}^*\|^2 \\ &= \frac{1}{n}\|\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\epsilon}\|^2 \end{aligned}$$

Observing that $\Pi := \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is an idempotent and independent projection whose rank is N, and that $\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] = \sigma^2\mathbf{I}$, we get

$$\mathrm{MSE}_\perp(\hat{f}_\perp) = \mathbb{E}[L(\hat{f}_\perp)] = \frac{1}{n}\|\Pi\boldsymbol{\epsilon}\|^2 = \frac{1}{n}\mathrm{tr}(\Pi\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T) = \frac{\sigma^2}{n}\mathrm{tr}(\Pi)$$

which concludes that

$$\mathrm{MSE}_\perp(\hat{f}) = O\Big(\frac{N}{n}\sigma^2\Big). \tag{22}$$

See also [17, Proposition 1].

Next we study $\mathrm{MSE}_\|(\hat{f})$**.** Denote $\tilde{\sigma}_\|^2 = \frac{1}{n}\sum_{i=1}^n \epsilon_{i\|}^2, E = \max_i |\epsilon_i|$. Using Jensen's inequality and union bound, we have

$$\exp(t\mathbb{E}[E]) \le \mathbb{E}[\exp(tE)] = \mathbb{E}[\max \exp(t|\epsilon_i|)] \le \sum_{i=1}^n \mathbb{E}[\exp(t|\epsilon_i|)] \le 2n\exp(t^2\sigma^2/2)$$

Taking expectation over both sides, we get

$$\mathbb{E}[E] \le \frac{\log 2n}{t} + \frac{t\sigma^2}{2}$$

maximizing the right hand side over t yields

$$\mathbb{E}[E] \le \sigma\sqrt{2\log 2n}.$$

Let $\tilde{\mathcal{F}}_\|$ be the covering set of $\mathcal{F}_\| = \{f_\| : f \in \mathcal{F}\}$. For any $\tilde{f}_\| \in \tilde{\mathcal{F}}_\|$,

$$\begin{aligned} L_\|(f_j) - \hat{L}_\|(f_j) &= \frac{1}{n}\sum_{i=1}^n (f_{j\|}(\boldsymbol{x}_i) - f_{0\|}(\boldsymbol{x}_i))^2 - \frac{1}{n}\sum_{i=1}^n (\tilde{f}_\|(\boldsymbol{x}_i) - y_{i\|})^2 + \frac{n-N}{n}\sigma^2 \\ &= \frac{1}{n}\sum_{i=1}^n \epsilon_{i\|}(2\tilde{f}_\|(\boldsymbol{x}_i) - f_{0\|}(\boldsymbol{x}_i) - y_{i\|}) + \frac{n-N}{n}\sigma^2 \\ &= \frac{1}{n}\sum_{i=1}^n \epsilon_i(2\tilde{f}_\|(\boldsymbol{x}_i) - f_{0\|}(\boldsymbol{x}_i) - y_{i\|}) + \frac{n-N}{n}\sigma^2 \\ &= \frac{1}{n}\sum_{i=1}^n \epsilon_i(2\tilde{f}_\|(\boldsymbol{x}_i) - 2f_{0\|}(\boldsymbol{x}_i)) + \frac{n-N}{n}\sigma^2 - \tilde{\sigma}_\|^2 \end{aligned}$$

The first term can be bounded using Bernstein's inequality: let $h_i = \epsilon_i(f_{j\|}(\boldsymbol{x}_i) - f_{0\|}(\boldsymbol{x}_i))$, by definition $|h_i| \le 2EF$,

$$\begin{aligned} \mathrm{Var}[h_i] &= \mathbb{E}[\epsilon_i^2(\tilde{f}_\|(\boldsymbol{x}_i) - f_{0\|}(\boldsymbol{x}_i))^2] \\ &= (\tilde{f}_\|(\boldsymbol{x}_i) - f_{0\|}(\boldsymbol{x}_i))^2\mathbb{E}[\epsilon_i^2] \\ &= (\tilde{f}_\|(\boldsymbol{x}_i) - f_{0\|}(\boldsymbol{x}_i))^2\sigma^2 \end{aligned}$$

using Bernstein's inequality, for any $\tilde{f}_{\|} \in \tilde{\mathcal{F}}_{\|}$, with probably at least $1-\delta_p$,

$$
\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}\epsilon_i(2\tilde{f}_{\|}(\boldsymbol{x}_i) - 2f_{0\|}(\boldsymbol{x}_i)) &= \frac{2}{n}\sum_{i=1}^{n} h_i \\
&\le \frac{2}{n}\sqrt{2\sum_{i=1}^{n}\big(\tilde{f}_{\|}(\boldsymbol{x}_i) - f_{0\|}(\boldsymbol{x}_i)\big)^2\sigma^2\log(1/\delta_p)} + \frac{8EF\log(1/\delta_p)}{3n} \\
&= 2\sqrt{\Big(L_{\|}(\tilde{f}_{\|}) - \frac{n-N}{n}\sigma^2\Big)\frac{2\sigma^2\log(1/\delta_p)}{n}} + \frac{8EF\log(1/\delta_p)}{3n} \\
&\le \epsilon\Big(L_{\|}(\tilde{f}_{\|}) - \frac{n-N}{n}\sigma^2\Big) + \frac{8\sigma^2\log(1/\delta_p)}{n\epsilon} + \frac{8EF\log(1/\delta_p)}{3n}
\end{aligned}
$$

the last inequality holds true for all $\epsilon > 0$. The union bound shows that with probably at least $1-\delta$, for all $\tilde{f}_{\|} \in \tilde{\mathcal{F}}_{\|}$,

$$
\begin{aligned}
L_{\|}(\tilde{f}_{\|}) - \hat{L}_{\|}(\tilde{f}_{\|}) \le{}& \epsilon\Big(L_{\|}(\tilde{f}_{\|}) - \frac{n-N}{n}\sigma^2\Big) + \frac{8\sigma^2\log(\mathcal{N}(\mathcal{F}_{\|},\delta)/\delta_p)}{n\epsilon} + \frac{8EF\log(\mathcal{N}(\mathcal{F}_{\|},\delta)/\delta_p)}{3n} \\
&+ \frac{n-N}{n}\sigma^2 - \tilde{\sigma}_{\|}^2.
\end{aligned}
$$

By rearanging the terms and using the definition of $L(\tilde{f}_{\|})$, we get

$$
(1-\epsilon)\Big(L_{\|}(\tilde{f}_{\|}) - \frac{n-N}{n}\sigma^2\Big) \le \hat{L}_{\|}(\tilde{f}_{\|}) + \frac{8\sigma^2\log(\mathcal{N}(\mathcal{F}_{\|},\delta)/\delta_p)}{n\epsilon} + \frac{8EF\log(\mathcal{N}(\mathcal{F}_{\|},\delta)/\delta_p)}{3n} - \tilde{\sigma}_{\|}^2.
$$

Taking the expectation (over $\mathcal{D}$) on both sides, and notice that $\mathbb{E}[\tilde{\sigma}_{\|}^2] = \frac{n-N}{n}\sigma^2$. Furthermore, for any random variable $X, \mathbb{E}[X] = \int_{-\infty}^{\infty} x dP(X \le x)$, we get

$$
\begin{aligned}
&\max_{\tilde{f}_{\|}\in\tilde{\mathcal{F}}_{\|}}\Big((1-\epsilon)MSE_{\|}(\tilde{f}_{\|}) - \mathbb{E}[\hat{L}_{\|}(\tilde{f}_{\|})]\Big) \\
&\le \Big(\frac{8\sigma^2}{n\epsilon} + \frac{8F\sigma\sqrt{2\log 2n}}{3n}\Big)\Big(\log\mathcal{N}(\mathcal{F}_{\|},\delta) - \int_{\delta=0}^{1}\log(\delta_p)d\delta_p\Big) - \frac{n-N}{n}\sigma^2 \\
&= \Big(\frac{8\sigma^2}{n\epsilon} + \frac{8F\sigma\sqrt{2\log 2n}}{3n}\Big)(\log\mathcal{N}(\mathcal{F}_{\|},\delta) + 1) - \frac{n-N}{n}\sigma^2.
\end{aligned}
\tag{23}
$$

where the integration can be computed by replacing δ with e^x. Though it is not integrable under Riemann integral, it is integrable under Lebesgue integration.

Similarly, let $\check{f}_{\|} = \arg\min_{f\in\mathcal{F}_{\|}} L_{\|}(f)$,

$$
L_{\|}(\check{f}_{\|}) - \hat{L}_{\|}(\check{f}_{\|}) = \frac{1}{n}\sum_{i=1}^{n}\epsilon_i(2\check{f}_{\|}(\boldsymbol{x}_i) - 2f_{0\|}(\boldsymbol{x}_i)) + \frac{n-N}{n}\sigma^2 - \tilde{\sigma}_{\|}^2
$$

with probably at least $1-\delta_q$, for any $\epsilon > 0$,

$$
\begin{aligned}
&-\frac{1}{n}\sum_{i=1}^{n}\epsilon_i(2\check{f}_{\|}(\boldsymbol{x}_i) - 2f_{0\|}(\boldsymbol{x}_i)) \le \epsilon\Big(L_{\|}(\check{f}_{\|}) - \frac{n-N}{n}\sigma^2\Big) + \frac{8\sigma^2\log(1/\delta_p)}{n\epsilon} + \frac{8EF\log(1/\delta_p)}{3n}, \\
&\hat{L}_{\|}(\check{f}_{\|}) \le (1+\epsilon)\Big(L_{\|}(\check{f}_{\|}) - \frac{n-N}{n}\sigma^2\Big) + \frac{8\sigma^2\log(1/\delta_p)}{n\epsilon} + \frac{8EF\log(1/\delta_q)}{3n} + \tilde{\sigma}_{\|}^2.
\end{aligned}
$$

Taking the expectation on both sides,

$$
\mathbb{E}[\hat{L}_{\|}(\check{f}_{\|})] \le (1+\epsilon)\mathrm{MSE}_{\|}(\check{f}_{\|}) + \frac{8\sigma^2}{n\epsilon} + \frac{8F\sigma\sqrt{2\log 2n}}{3n} + \frac{n-N}{n}\sigma^2. \tag{24}
$$

Finally, let $\hat{f}_* := \arg\min_{f\in\tilde{\mathcal{F}}_\parallel} \sum_{i=1}^n (\hat{f}_\parallel(\boldsymbol{x}_i) - f(\boldsymbol{x}_i))^2$ be the projection of $\hat{f}_\parallel$ in its δ-covering space,

$$\begin{aligned}
\mathrm{MSE}_\parallel(\hat{f}_\parallel) &= \mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^n (\hat{f}_\parallel(\boldsymbol{x}_i) - f_{0\parallel}(\boldsymbol{x}_i))^2\Big] \\
&= \mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^n (\hat{f}_*(\boldsymbol{x}_i) - f_{0\parallel}(\boldsymbol{x}_i))^2 + \frac{1}{n}\sum_{i=1}^n (\hat{f}_\parallel(\boldsymbol{x}_i) - \hat{f}_*(\boldsymbol{x}_i))(\hat{f}_\parallel(\boldsymbol{x}_i) + \hat{f}_*(\boldsymbol{x}_i) - 2f_{0\parallel}(\boldsymbol{x}_i))\Big] \\
&\le \mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^n (\hat{f}_*(\boldsymbol{x}_i) - f_{0\parallel}(\boldsymbol{x}_i))^2\Big] + 4F\delta \\
&= \mathrm{MSE}_\parallel(\hat{f}_*(\boldsymbol{x}_i)) + 4F\delta,
\end{aligned}$$

and similarly

$$\hat{L}_\parallel(\hat{f}_*) \le \hat{L}_\parallel(\hat{f}_\parallel) + (4F + 2E)\delta. \tag{25}$$

We can conclude that

$$\begin{aligned}
\mathrm{MSE}_\parallel(\hat{f}_\parallel) \le{}& \frac{1}{1-\epsilon}\Big(\mathbb{E}[\hat{L}_\parallel(\hat{f}_*)] + \Big(\frac{8\sigma^2}{n\epsilon} + \frac{8F\sigma\sqrt{2\log 2n}}{3n}\Big)(\log\mathcal{N}(\mathcal{F}_\parallel,\delta)+1) - \frac{n-N}{n}\sigma^2\Big) \\
&+ 4F\delta \\
\le{}& \frac{1}{1-\epsilon}\Big(\mathbb{E}[\hat{L}_\parallel(\hat{f}_\parallel)] + (4F + \sigma\sqrt{8\log 2n})\delta \\
&+ \Big(\frac{8\sigma^2}{n\epsilon} + \frac{8F\sigma\sqrt{2\log 2n}}{3n}\Big)(\log\mathcal{N}(\mathcal{F}_\parallel,\delta)+1) - \frac{n-N}{n}\sigma^2\Big) + 4F\delta \\
\le{}& \frac{1}{1-\epsilon}\Big(\mathbb{E}[\hat{L}_\parallel(\check{f}_\parallel)] + (4F + \sigma\sqrt{8\log 2n})\delta \\
&+ \Big(\frac{8\sigma^2}{n\epsilon} + \frac{8F\sigma\sqrt{2\log 2n}}{3n}\Big)(\log\mathcal{N}(\mathcal{F}_\parallel,\delta)+1) - \frac{n-N}{n}\sigma^2\Big) + 4F\delta \\
\le{}& \frac{1+\epsilon}{1-\epsilon}\mathrm{MSE}_\parallel(\check{f}_\parallel) + \frac{1}{n}\Big(\frac{8\sigma^2}{\epsilon} + \frac{8F\sigma\sqrt{2\log 2n}}{3}\Big)\Big(\frac{\log\mathcal{N}(\mathcal{F}_\parallel,\delta)+2}{1-\epsilon}\Big) \\
&+ \Big(4F + \frac{4F + \sigma\sqrt{8\log 2n}}{1-\epsilon}\Big)\delta,
\end{aligned}$$

where the first line comes from (23), and second comes from (25), the thid line is because $\hat{f}_\parallel = \arg\min_{f\in\mathcal{F}_\parallel} \hat{L}_\parallel(f)$, and the last line comes from (24). We also use that fact that $\hat{L}_\parallel(\hat{f}) \le \hat{L}_\parallel(f), \forall f$. Noticing that $\mathrm{MSE}(\hat{f}) = \mathrm{MSE}_\parallel(\hat{f}) + \mathrm{MSE}_\perp(\hat{f})$, combining this with (22) finishes the proof. □

G Detailed experimental setup

G.1 Target Functions

The doppler function used in Figure 2(d)-(f) is

$$f(x) = \sin(4/(x + 0.01)) + 1.5.$$

The "vary" function used in Figure 2(g)-(i) is

$$\begin{aligned}
f(x) = {}& M_1(x/0.01) + M_1((x-0.02)/0.02) + M_1((x-0.06)/0.03) \\
&+ M_1((x-0.12)/0.04) + M_3((x-0.2)/0.02) + M_3((x-0.28)/0.04) \\
&+ M_3((x-0.44)/0.06) + M_3((x-0.68)/0.08),
\end{aligned}$$

where $x_+ := \max(x, 0)$. We uniformly take 256 samples from 0 to 1 in the piecewise cubic function experiment, and uniformly 1000 samples from 0 to 1 in the doppler function and "vary" function experiment. We add zero mean independent (white) Gaussian noise to the observations. The standard derivation of noise is 0.05 in the piecewise cubic function experiment, 0.4 in the doppler function experiment and 0.1 in the "vary" function experiment.

G.2 Training/Fitting Method

In the piecewise polynomial function ("vary") experiment, the depth of the PNN $L = 10$, the width of each subnetwork $w = 10$, and the model contains $M = 500$ subnetworks. The depth of NN is also 10, and the width is 200 such that the NN and PNN have almost the same number of parameters. In the doppler function experiment, the depth of the PNN $L = 12$, the width of each subnetwork $w = 10$, and the model contains $M = 2000$ subnetworks, because this problem requires a more complex model to fit. The depth of NN is 12, and the width is 400. We used Adam optimizer with learning rate of 10^{-3}. We first train the neural network layer by layer without weight decay. Specifically, we start with a two-layer neural network with the same number of subnetworks and the same width in each subnetwork, then train a three layer neural network by initializing the first layer using the trained two layer one, until the desired depth is reached. After that, we turn the weight decay parameter and train it until convergence. In both trend filtering and smoothing spline experiment, the order is 3, and in wavelet denoising experiment, we use sym4 wavelet with soft thresholding. We implement the trend filtering problem according to Tibshirani [37] using CVXPY, and use MOSEK to solve the convex optimization problem. We directly call R function $smooth.spline$ to solve smoothing spline.

G.3 Post Processing

The degree of freedom of smoothing spline is returned by the solver in R, which is rounded to the nearest integer when plotting. To estimate the degree of freedom of trend filtering, for each choice of λ, we repeated the experiment for 10 times and compute the average number of nonzero knots as estimated degree of freedom. For neural networks, we use the definition [38]:

$$2\sigma^2\mathrm{df} = \mathbb{E}\|\boldsymbol{y}' - \hat{\boldsymbol{y}}\|_2^2 - \mathbb{E}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 \tag{26}$$

where df denotes the degree of freedom, σ^2 is the variance of the noise, $\boldsymbol{y}$ are the labels, $\hat{\boldsymbol{y}}$ are the predictions and $\boldsymbol{y}'$ are independent copy of y. We find that estimating (26) directly by sampling leads to large error when the degree of freedom is small. Instead, we compute

$$2\sigma^2\hat{\mathrm{df}} = \hat{\mathbb{E}}\|\boldsymbol{y}_0 - \hat{\boldsymbol{y}}\|_2^2 - \hat{\mathbb{E}}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 + \hat{\mathbb{E}}\|\boldsymbol{y} - \bar{y}_0\|_2^2 - \|\boldsymbol{y}_0 - \bar{y}_0\|_2^2 \tag{27}$$

where $\hat{\mathrm{df}}$ is the estimated degree of freedom, $\mathbb{E}$ denotes the empirical average (sample mean), $\boldsymbol{y}_0$ is the target function and $\bar{y}_0$ is the mean of the target function in its domain.

Proposition 16. *The expectation of (27) over the dataset $\mathcal{D}$ equals (26).*

Proof.

$$\begin{aligned}
2\sigma^2\hat{\mathrm{df}} &= \mathbb{E}_{\mathcal{D}}[\hat{\mathbb{E}}\|\boldsymbol{y}_0 - \hat{\boldsymbol{y}}\|_2^2 - \hat{\mathbb{E}}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 + \hat{\mathbb{E}}\|\boldsymbol{y} - \bar{y}_0\|_2^2 - \|\boldsymbol{y}_0 - \bar{y}_0\|_2^2] \\
&= \mathbb{E}\|\boldsymbol{y}_0 - \hat{\boldsymbol{y}}\|_2^2 - \mathbb{E}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 + \mathbb{E}_{\mathcal{D}}[\hat{\mathbb{E}}[(\boldsymbol{y} - \boldsymbol{y}_0)(\boldsymbol{y} + \boldsymbol{y}_0 - 2\bar{y}_0)]] \\
&= \mathbb{E}\|\boldsymbol{y}_0 - \hat{\boldsymbol{y}}\|_2^2 - \mathbb{E}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 + \mathbb{E}\Big[\sum_{i=1}^n \epsilon_i(2y_i + \epsilon_i - 2\bar{y}_0)\Big] \\
&= \mathbb{E}\|\boldsymbol{y}_0 - \hat{\boldsymbol{y}}\|_2^2 - \mathbb{E}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 + n\sigma^2 \\
&= \mathbb{E}\|\boldsymbol{y}' - \hat{\boldsymbol{y}}\|_2^2 - \mathbb{E}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2
\end{aligned}$$

where $\mathcal{D}$ denotes the dataset. In the third line, we make use of the fact that $\mathbb{E}[\epsilon_i] = 0, \mathbb{E}[\epsilon_i^2] = \sigma^2$, and in the last line, we make use of $\mathbb{E}[\epsilon_i'] = 0, \mathbb{E}[{\epsilon_i'}^2] = \sigma^2$, and ϵ_i' are independent of y_i and $y_{0,i}$ □

One can easily check that a "zero predictor" (a predictor that always predict $\bar{y}_0$, and it always predicts 0 if the target function has zero mean) always has an estimated degree of freedom of 0.

G.4 More experimental results

G.4.1 Regularization weight vs degree-of-freedom

As we explained in the previous section, the degree of freedom is the exact information-theoretic measure of the generalization gap. A Larger degree-of-freedom implies more overfitting.

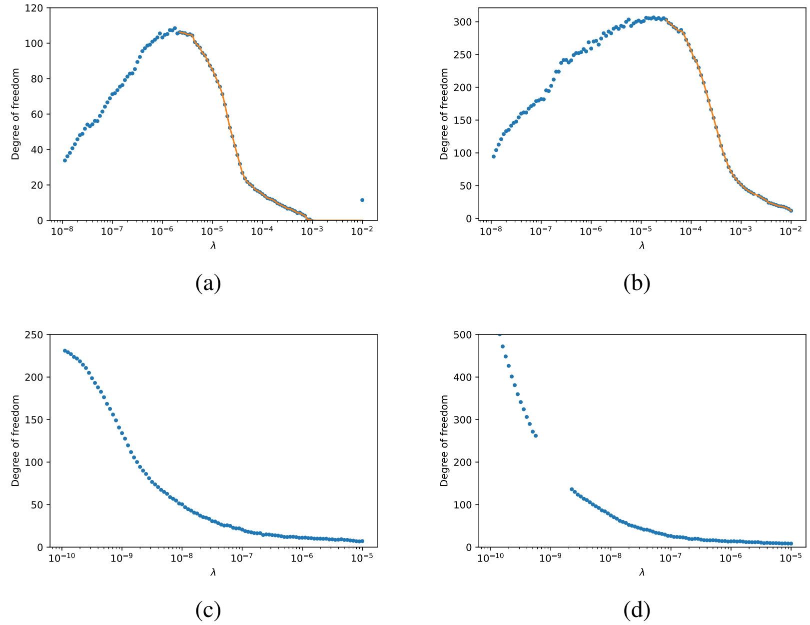

Figure 3: The relationship between degree of freedom and the scaling factor of the regularizer λ. The solid line shows the result after denoising. (a)(b)in a NN. (c)(d) In trend filtering. (a)(c): the piecewise cubic function. (b)(d) the doppler function.

In figure Figure 3, we show the relationship between the estimated degree of freedom and the scaling factor of the regularizer λ in a parallel neural network and in trend filtering. As is shown in the figure, generally speaking as λ decreases towards 0, the degree of freedom should increase too. However, for parallel neural networks, if λ is very close to 0, the estimated degree of freedom will not increase although the degree of freedom is much smaller than the number of parameters — actually even smaller than the number of subnetworks. Instead, it actually decreases a little. This effect has not been observed in other nonparametric regression methods, e.g. trend filtering, which overfits every noisy datapoint perfectly when $\lambda \to 0$. But for the neural networks, even if we do not regularize at all, the among of overfitting is still relatively mild $30/256$ vs $80/1000$. In our experiments using neural networks, when λ is small, we denoise the estimated degree of freedom using isotonic regression.

We do not know the exact reason of this curious observation. Our hypothesis is that it might be related to issues with optimization, i.e., the optimizer ends up at a local minimum that generalizes better than a global minimum; or it could be connected to the "double descent" behavior of DNN [24] under over-parameterization.

G.4.2 Detailed numerical results

In order to allow the readers to view our result in detail, we plot the numerical experiment results of each method separately in Figure 4 and Figure 5.

G.4.3 Practical equivalence between the weight-decayed two-layer NN and L1-Trend Filtering

In this section we investigate the equivalence of two-layer NN and the locally adaptive regression splines from Section B. In the special case when $m = 1$ the special regularization reduces to weight decay and the non-standard truncated power activation becomes ReLU. We compare L1 trend filtering [20] (shown to be equivalent to locally adaptive regression splines by Tibshirani [37]) and an overparameterized version of the neural network for all regularization parameter $\lambda > 0$, i.e., a regularization path. The results are shown in Figure 6. It is clear that as the weight decay in-

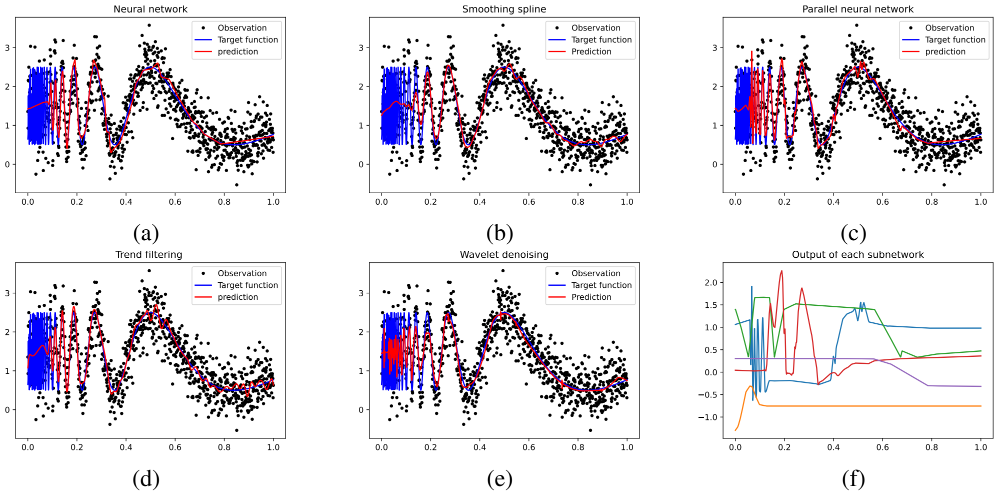

Figure 4: More experiments results of Doppler function.

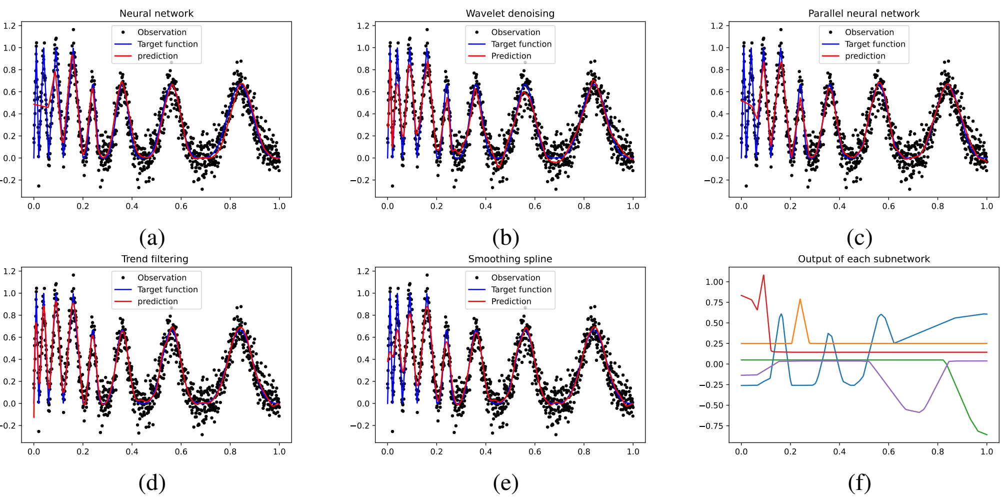

Figure 5: More experiments results of the “vary” function.

creases, it induces sparsity in the number of knots it selects similarly to L1-Trend Filtering, and the regularization path matches up nearly perfectly even though NNs are also learning knots locations.

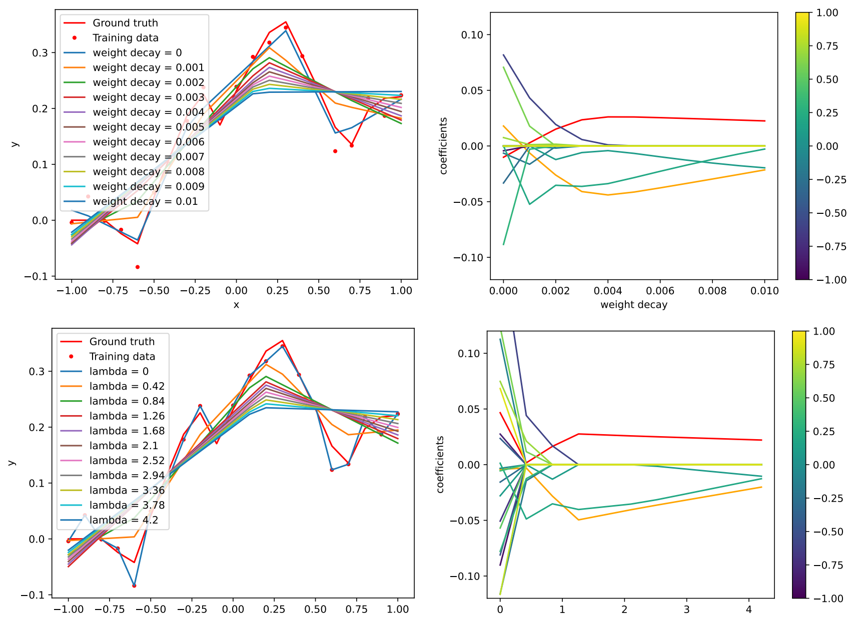

Figure 6: Comparison of the **weight decayed ReLU neural networks (Top row)** and **L1 Trend Filtering (Bottom row)** with different regularization parameters. The left column shows the fitted functions and the right column shows the *regularization path* (in the flavor of [15]) of the coefficients of the truncated power basis at individual data points (the free-knots learned by NN are snapped to the nearest input x to be comparable).