Besov Function Approximation and Binary Classification on Low-Dimensional Manifolds Using Convolutional Residual Networks

Hao Liu [1] Minshuo Chen [2] Tuo Zhao [2] Wenjing Liao [3]

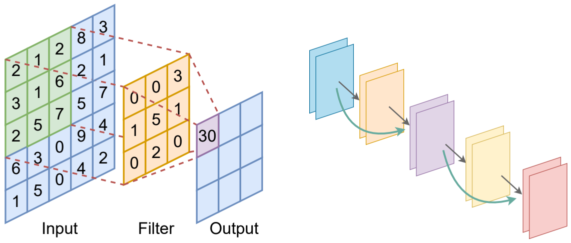
Abstract

Most of existing statistical theories on deep neural networks have sample complexities cursed by the data dimension and therefore cannot well explain the empirical success of deep learning on high-dimensional data. To bridge this gap, we propose to exploit low-dimensional geometric structures of the real world data sets. We establish theoretical guarantees of convolutional residual networks (ConvResNet) in terms of function approximation and statistical estimation for binary classification. Specifically, given the data lying on a d-dimensional manifold isometrically embedded in $\mathbb{R}^D$, we prove that if the network architecture is properly chosen, ConvResNets can (1) approximate *Besov functions* on manifolds with arbitrary accuracy, and (2) learn a classifier by minimizing the empirical logistic risk, which gives an *excess risk* in the order of $n^{-\frac{s}{2s+2(s\vee d)}}$, where s is a smoothness parameter. This implies that the sample complexity depends on the intrinsic dimension d, instead of the data dimension D. Our results demonstrate that ConvResNets are adaptive to low-dimensional structures of data sets.

1. Introduction

Deep learning has achieved significant success in various practical applications with high-dimensional data set, such as computer vision (Krizhevsky et al., 2012), natural language processing (Graves et al., 2013; Young et al., 2018; Wu et al., 2016), health care (Miotto et al., 2018; Jiang et al., 2017) and bioinformatics (Alipanahi et al., 2015; Zhou & Troyanskaya, 2015).

The success of deep learning clearly demonstrates the great power of neural networks in representing complex data. In the past decades, the representation power of neural networks has been extensively studied. The most commonly studied architecture is the feedforward neural network (FNN), as it has a simple composition form. The representation theory of FNNs has been developed with smooth activation functions (e.g., sigmoid) in Cybenko (1989); Barron (1993); McCaffrey & Gallant (1994); Hamers & Kohler (2006); Kohler & Krzyżak (2005); Kohler & Mehnert (2011) or nonsmooth activations (e.g., ReLU) in Lu et al. (2017); Yarotsky (2017); Lee et al. (2017); Suzuki (2019). These works show that if the network architecture is properly chosen, FNNs can approximate uniformly smooth functions (e.g., Hölder or Sobolev) with arbitrary accuracy.



(a) Convolution. (b) Skip-layer connection.

Figure 1. Illustration of (a) convolution and (b) skip-layer connection.

In real-world applications, convolutional neural networks (CNNs) are more popular than FNNs (LeCun et al., 1989; Krizhevsky et al., 2012; Sermanet et al., 2013; He et al., 2016; Chen et al., 2017; Long et al., 2015; Simonyan & Zisserman, 2014; Girshick, 2015). In a CNN, each layer consists of several filters (channels) which are convolved with the input, as demonstrated in Figure 1(a). Due to such complexity in the CNN architecture, there are limited works on the representation theory of CNNs (Zhou, 2020b;a; Fang et al., 2020; Petersen & Voigtlaender, 2020). The constructed CNNs in these works become extremely wide (in terms of the size of each layer's output) as the approximation error goes to 0. In most real-life applications, the network width does not exceed 2048 (Zagoruyko & Komodakis, 2016; Zhang et al., 2020).

Convolutional residual networks (ConvResNet) is a special

[1] Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong. [2] School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA. [3] School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332 USA.. Correspondence to: Wenjing Liao <wliao60@gatech.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

CNN architecture with skip-layer connections, as shown in Figure 1(b). Specifically, in addition to CNNs, ConvResNets have identity connections between inconsecutive layers. In many applications, ConvResNets outperform CNNs in terms of generalization performance and computational efficiency, and alleviate the vanishing gradient issue. Using this architecture, He et al. (2016) won the 1st place on the ImageNet classification task with a 3.57% top 5 error in 2015.

Recently, Oono & Suzuki (2019) develops the only representation and statistical estimation theory of ConvResNets. Oono & Suzuki (2019) proves that if the network architecture is properly set, ConvResNets with a fixed filter size and a fixed number of channels can universally approximate Hölder functions with arbitrary accuracy. However, the sample complexity in Oono & Suzuki (2019) grows exponentially with respect to the data dimension and therefore cannot well explain the empirical success of ConvResNets for high dimensional data. In order to estimate a C^s function in $\mathbb{R}^D$ with accuracy ε, the sample size required by Oono & Suzuki (2019) scales as $\varepsilon^{-\frac{2s+D}{s}}$, which is far beyond the sample size used in practical applications. For example, the ImageNet data set consists of 1.2 million labeled images of size $224 \times 224 \times 3$. According to this theory, to achieve a 0.1 error, the sample size is required to be in the order of $10^{224\times 224\times 3}$ which greatly exceeds 1.2 million. Due to the curse of dimensionality, there is a huge gap between theory and practice.

We bridge such a gap by taking low-dimensional geometric structures of data sets into consideration. It is commonly believed that real world data sets exhibit low-dimensional structures due to rich local regularities, global symmetries, or repetitive patterns (Hinton & Salakhutdinov, 2006; Osher et al., 2017; Tenenbaum et al., 2000). For example, the ImageNet data set contains many images of the same object with certain transformations, such as rotation, translation, projection and skeletonization. As a result, the degree of freedom of the ImageNet data set is significantly smaller than the data dimension (Gong et al., 2019).

The function space considered in Oono & Suzuki (2019) is the Hölder space in which functions are required to be differentiable everywhere up to certain order. In practice, the target function may not have high order derivatives. Function spaces with less restrictive conditions are more desirable. In this paper, we consider the Besov space $B^s_{p,q}$, which is more general than the Hölder space. In particular, the Hölder and Sobolev spaces are special cases of the Besov space:

$$W^{s+\alpha,\infty} = \mathcal{H}^{s,\alpha} \subseteq B^{s+\alpha}_{\infty,\infty} \subseteq B^{s+\alpha}_{p,q}$$

for any $0 < p, q \leq \infty$, $s \in \mathbb{N}$ and $\alpha \in (0, 1]$. For practical applications, it has been demonstrated in image processing that Besov norms can capture important features, such as edges (Jaffard et al., 2001). Due to the generality of the Besov space, it is shown in Suzuki & Nitanda (2019) that kernel ridge estimators have a sub-optimal rate when estimating Besov functions.

In this paper, we establish theoretical guarantees of ConvResNets for the approximation of Besov functions on a low-dimensional manifold, and a statistical theory on binary classification. Let $\mathcal{M}$ be a d-dimensional compact Riemannian manifold isometrically embedded in $\mathbb{R}^D$. Denote the Besov space on $\mathcal{M}$ as $B^s_{p,q}(\mathcal{M})$ for $0 < p, q \leq \infty$ and $0 < s < \infty$. Our function approximation theory is established for $B^s_{p,q}(\mathcal{M})$. For binary classification, we are given n i.i.d. samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathcal{M}$ and $y_i \in \{-1, 1\}$ is the label. The label y follows the Bernoulli-type distribution

$$\mathbb{P}(y = 1|\mathbf{x}) = \eta(\mathbf{x}),\ \mathbb{P}(y = -1|\mathbf{x}) = 1 - \eta(\mathbf{x})$$

for some $\eta : \mathcal{M} \to [0, 1]$. Our results (Theorem 1 and 2) are summarized as follows:

Theorem (informal)**.** *Assume $s \geq d/p + 1$.*

1. *Given $\varepsilon \in (0, 1)$, we construct a ConvResNet architecture such that, for any $f^* \in B^s_{p,q}(\mathcal{M})$, if the weight parameters of this ConvResNet are properly chosen, it gives rises to $\bar{f}$ satisfying*

$$\|\bar{f} - f^*\|_{L^\infty} \leq \varepsilon.$$

2. *Assume $\eta \in B^s_{p,q}(\mathcal{M})$. Let f^*_ϕ be the minimizer of the population logistic risk. If the ConvResNet architecture is properly chosen, minimizing the empirical logistic risk gives rise to $\widehat{f}_{\phi,n}$ with the following excess risk bound*

$$\mathbb{E}(\mathcal{E}_\phi(\widehat{f}_{\phi,n}, f^*_\phi)) \leq Cn^{-\frac{s}{2s+2(s\vee d)}} \log^4 n,$$

where $\mathcal{E}_\phi(\widehat{f}_{\phi,n}, f^*_\phi)$ denotes the excess logistic risk of $\widehat{f}_{\phi,n}$ against f^*_ϕ and C is a constant independent of n.

We remark that the first part of the theorem above requires the network size to depend on the intrinsic dimension d and only weakly depend on D. The second part is built upon the first part and shows a fast convergence rate of the excess risk in terms of n where the exponent depends on d instead of D. Our results demonstrate that ConvResNets are adaptive to low-dimensional structures of data sets.

Related work. Approximation theories of FNNs with the ReLU activation have been established for Sobolev (Yarotsky, 2017), Hölder (Schmidt-Hieber, 2017) and Besov (Suzuki, 2019) spaces. The networks in these works have certain cardinality constraint, i.e., the number of nonzero parameters is bounded by certain constant, which requires a lot of efforts for training.

Approximation theories of CNNs are developed in Zhou (2020b); Petersen & Voigtlaender (2020); Oono & Suzuki (2019). Among these works, Zhou (2020b) shows that

Table 1. Comparison of our approximation theory and existing theoretical results.

	Network type	Function class	Low dim. structure	Fixed width	Training
Yarotsky (2017)	FNN	Sobolev	✗	✗	difficult to train due to the cardinality constraint
Suzuki (2019)	FNN	Besov	✗	✗	
Chen et al. (2019a)	FNN	Hölder	✓	✗	
Petersen & Voigtlaender (2020)	CNN	FNN	✗	✗	
Zhou (2020b)	CNN	Sobolev	✗	✗	can be trained without the cardinality constraint
Oono & Suzuki (2019)	ConvResNet	Hölder	✗	✓	
Ours	ConvResNet	Besov	✓	✓	

CNNs can approximate Sobolev functions in $W^{s,2}$ for $s \geq D/2+2$ with an arbitrary accuracy $\varepsilon \in (0,1)$. The network in Zhou (2020b) has width increasing linearly with respect to depth and has depth growing in the order of ε^{-2} as ε decreases to 0. It is shown in Petersen & Voigtlaender (2020); Zhou (2020a) that any approximation error achieved by FNNs can be achieved by CNNs. Combining Zhou (2020a) and Yarotsky (2017), we can show that CNNs can approximate $W^{s,\infty}$ functions in $\mathbb{R}^D$ with arbitrary accuracy ε. Such CNNs have the number of channels in the order of $\varepsilon^{-D/s}$ and a cardinality constraint. The only theory on ConvResNet can be found in Oono & Suzuki (2019), where an approximation theory for Hölder functions is proved for ConvResNets with fixed width.

Statistical theories for binary classification by FNNs are established with the hinge loss (Ohn & Kim, 2019; Hu et al., 2020) and the logistic loss (Kim et al., 2018). Among these works, Hu et al. (2020) uses a parametric model given by a teacher-student network. The nonparametric results in Ohn & Kim (2019); Kim et al. (2018) are cursed by the data dimension, and therefore require a large number of samples for high-dimensional data.

Binary classification by CNNs has been studied in Kohler et al. (2020); Kohler & Langer (2020); Nitanda & Suzuki (2018); Huang et al. (2018). Image binary classification is studied in Kohler et al. (2020); Kohler & Langer (2020) in which the probability function is assumed to be in a hierarchical max-pooling model class. ResNet type classifiers are considered in Nitanda & Suzuki (2018); Huang et al. (2018) while the generalization error is not given explicitly.

Low-dimensional structures of data sets are explored for neural networks in Chui & Mhaskar (2018); Shaham et al. (2018); Chen et al. (2019a;b); Schmidt-Hieber (2019); Nakada & Imaizumi (2019); Cloninger & Klock (2020); Chen et al. (2020); Montanelli & Yang (2020). These works show that, if data are near a low-dimensional manifold, the performance of FNNs depends on the intrinsic dimension of the manifold and only weakly depends on the data dimension. Our work focuses on ConvResNets for practical applications.

The networks in many aforementioned works have a cardinality constraint. From the computational perspective, training such networks requires substantial efforts (Han et al., 2016; 2015; Blalock et al., 2020). In comparison, the ConvResNet in Oono & Suzuki (2019) and this paper does not require any cardinality constraint. Additionally, our constructed network has a fixed filter size and a fixed number of channels, which is desirable for practical applications.

As a summary, we compare our approximation theory and existing results in Table 1.

The rest of this paper is organized as follows: In Section 2, we briefly introduce manifolds, Besov functions on manifolds and convolution. Our main results are presented in Section 3. We give a proof sketch in Section 4 and conclude this paper in Section 5.

2. Preliminaries

Notations: We use bold lower-case letters to denote vectors, upper-case letters to denote matrices, calligraphic letters to denote tensors, sets and manifolds. For any $x > 0$, we use $\lceil x \rceil$ to denote the smallest integer that is no less than x and use $\lfloor x \rfloor$ to denote the largest integer that is no larger than x. For any $a, b \in \mathbb{R}$, we denote $a \vee b = \max(a,b)$. For a function $f : \mathbb{R}^d \to \mathbb{R}$ and a set $\Omega \subset \mathbb{R}^d$, we denote the restriction of f to Ω by $f|_\Omega$. We use $\|f\|_{L^p}$ to denote the L^p norm of f. We denote the Euclidean ball centered at $\mathbf{c}$ with radius ω by $B_\omega(\mathbf{c})$.

2.1. Low-dimensional manifolds

We first introduce some concepts on manifolds. We refer the readers to Tu (2010); Lee (2006) for details. Throughout this paper, we let $\mathcal{M}$ be a d-dimensional Riemannian manifold $\mathcal{M}$ isometrically embedded in $\mathbb{R}^D$ with $d \leq D$. We first introduce charts, an atlas and the partition of unity.

Definition 1 (Chart)**.** *A chart on $\mathcal{M}$ is a pair (U, ϕ) where $U \subset \mathcal{M}$ is open and $\phi : U \to \mathbb{R}^d$, is a homeomorphism (i.e., bijective, ϕ and ϕ^{-1} are both continuous).*

In a chart (U, ϕ), U is called a coordinate neighborhood and ϕ is a coordinate system on U. A collection of charts which covers $\mathcal{M}$ is called an atlas of $\mathcal{M}$.

Definition 2 (C^k Atlas)**.** *A C^k atlas for $\mathcal{M}$ is a collection of charts $\{(U_\alpha, \phi_\alpha)\}_{\alpha\in\mathcal{A}}$ which satisfies $\bigcup_{\alpha\in\mathcal{A}} U_\alpha = \mathcal{M}$,* *and are pairwise C^k compatible:*

$$\phi_\alpha \circ \phi_\beta^{-1} : \phi_\beta(U_\alpha \cap U_\beta) \to \phi_\alpha(U_\alpha \cap U_\beta) \quad \textit{and}$$
$$\phi_\beta \circ \phi_\alpha^{-1} : \phi_\alpha(U_\alpha \cap U_\beta) \to \phi_\beta(U_\alpha \cap U_\beta)$$

are both C^k for any $\alpha, \beta \in \mathcal{A}$. An atlas is called finite if it contains finitely many charts.

Definition 3 (Smooth Manifold)**.** *A smooth manifold is a manifold $\mathcal{M}$ together with a C^∞ atlas.*

The Euclidean space, the torus and the unit sphere are examples of smooth manifolds. C^s functions on a smooth manifold $\mathcal{M}$ are defined as follows:

Definition 4 (C^s functions on $\mathcal{M}$)**.** *Let $\mathcal{M}$ be a smooth manifold and $f : \mathcal{M} \to \mathbb{R}$ be a function on $\mathcal{M}$. We say f is a C^s function on $\mathcal{M}$, if for every chart (U, ϕ) on $\mathcal{M}$, the function $f \circ \phi^{-1} : \phi(U) \to \mathbb{R}$ is a C^s function.*

We next define the C^∞ partition of unity which is an important tool for the study of functions on manifolds.

Definition 5 (Partition of Unity)**.** *A C^∞ partition of unity on a manifold $\mathcal{M}$ is a collection of C^∞ functions $\{\rho_\alpha\}_{\alpha\in\mathcal{A}}$ with $\rho_\alpha : \mathcal{M} \to [0, 1]$ such that for any $\mathbf{x} \in \mathcal{M}$,*

1. *there is a neighbourhood of $\mathbf{x}$ where only a finite number of the functions in $\{\rho_\alpha\}_{\alpha\in\mathcal{A}}$ are nonzero, and*
2. $\sum_{\alpha\in\mathcal{A}} \rho_\alpha(\mathbf{x}) = 1.$

An open cover of a manifold $\mathcal{M}$ is called locally finite if every $\mathbf{x} \in \mathcal{M}$ has a neighbourhood which intersects with a finite number of sets in the cover. The following proposition shows that a C^∞ partition of unity for a smooth manifold always exists (Spivak, 1970, Chapter 2, Theorem 15).

Proposition 1 (Existence of a C^∞ partition of unity)**.** *Let $\{U_\alpha\}_{\alpha\in\mathcal{A}}$ be a locally finite cover of a smooth manifold $\mathcal{M}$. There is a C^∞ partition of unity $\{\rho_\alpha\}_{\alpha=1}^\infty$ such that* $\mathrm{supp}(\rho_\alpha) \subset U_\alpha$.

Let $\{(U_\alpha, \phi_\alpha)\}_{\alpha\in\mathcal{A}}$ be a C^∞ atlas of $\mathcal{M}$. Proposition 1 guarantees the existence of a partition of unity $\{\rho_\alpha\}_{\alpha\in\mathcal{A}}$ such that ρ_α is supported on U_α.

The reach of $\mathcal{M}$ introduced by Federer (Federer, 1959) is an important quantity defined below. Let $d(\mathbf{x}, \mathcal{M}) = \inf_{\mathbf{y}\in\mathcal{M}} \|\mathbf{x} - \mathbf{y}\|_2$ be the distance from $\mathbf{x}$ to $\mathcal{M}$.

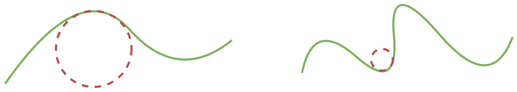
Definition 6 (Reach (Federer, 1959; Niyogi et al., 2008))**.** *Define the set*

$$G = \{\mathbf{x} \in \mathbb{R}^D : \exists \textit{ distinct } \mathbf{p}, \mathbf{q} \in \mathcal{M} \textit{ such that}$$
$$d(\mathbf{x}, \mathcal{M}) = \|\mathbf{x} - \mathbf{p}\|_2 = \|\mathbf{x} - \mathbf{q}\|_2\}.$$

The closure of G is called the medial axis of $\mathcal{M}$. The reach of $\mathcal{M}$ is defined as

$$\tau = \inf_{\mathbf{x}\in\mathcal{M}} \inf_{\mathbf{y}\in G} \|\mathbf{x} - \mathbf{y}\|_2.$$

We illustrate large and small reach in Figure 2.



Slow Change: Large τ

Rapid Change: Small τ

Figure 2. Illustration of manifolds with large and small reach.

2.2. Besov functions on a smooth manifold

We next define Besov function spaces on $\mathcal{M}$, which generalizes more elementary function spaces such as the Sobolev and Hölder spaces. To define Besov functions, we first introduce the modulus of smoothness.

Definition 7 (Modulus of Smoothness (DeVore & Lorentz, 1993; Suzuki, 2019))**.** *Let $\Omega \subset \mathbb{R}^D$. For a function $f : \mathbb{R}^D \to \mathbb{R}$ be in $L^p(\Omega)$ for $p > 0$, the r-th modulus of smoothness of f is defined by*

$$w_{r,p}(f, t) = \sup_{\|\mathbf{h}\|_2 \le t} \|\Delta_{\mathbf{h}}^r(f)\|_{L^p}, \textit{ where}$$

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

Definition 8 (Besov Space $B_{p,q}^s(\Omega)$)**.** *For $0 < p, q \le \infty, s > 0, r = \lfloor s \rfloor + 1$, define the seminorm $|\cdot|_{B_{p,q}^s}$ as*

$$|f|_{B_{p,q}^s(\Omega)} := \begin{cases} \left(\int_0^\infty (t^{-s} w_{r,p}(f, t))^q \frac{dt}{t}\right)^{\frac{1}{q}} & \textit{if } q < \infty, \\ \sup_{t>0} t^{-s} w_{r,p}(f, t) & \textit{if } q = \infty. \end{cases}$$

The norm of the Besov space $B_{p,q}^s(\Omega)$ is defined as $\|f\|_{B_{p,q}^s(\Omega)} := \|f\|_{L^p(\Omega)} + |f|_{B_{p,q}^s(\Omega)}$. The Besov space is $B_{p,q}^s(\Omega) = \{f \in L^p(\Omega) | \|f\|_{B_{p,q}^s} < \infty\}$.

We next define $B_{p,q}^s$ functions on $\mathcal{M}$ (Geller & Pesenson, 2011; Triebel, 1983; 1992).

Definition 9 ($B_{p,q}^s$ Functions on $\mathcal{M}$)**.** *Let $\mathcal{M}$ be a compact smooth manifold of dimension d. Let $\{(U_i, \phi_i)\}_{i=1}^{C_\mathcal{M}}$ be a finite atlas on $\mathcal{M}$ and $\{\rho_i\}_{i=1}^{C_\mathcal{M}}$ be a partition of unity on $\mathcal{M}$ such that* $\mathrm{supp}(\rho_i) \subset U_i$*. A function $f : \mathcal{M} \to \mathbb{R}$ is in $B_{p,q}^s(\mathcal{M})$ if*

$$\|f\|_{B_{p,q}^s(\mathcal{M})} := \sum_{i=1}^{C_\mathcal{M}} \|(f\rho_i) \circ \phi_i^{-1}\|_{B_{p,q}^s(\mathbb{R}^d)} < \infty. \quad (1)$$

Since ρ_i is supported on U_i, the function $(f\rho_i) \circ \phi_i^{-1}$ is supported on $\phi(U_i)$. We can extend $(f\rho_i) \circ \phi_i^{-1}$ from $\phi(U_i)$ to $\mathbb{R}^d$ by setting the function to be 0 on $\mathbb{R}^d \setminus \phi(U_i)$. The extended function lies in the Besov space $B_{p,q}^s(\mathbb{R}^d)$ (Triebel, 1992, Chapter 7).

2.3. Convolution and residual block

In this paper, we consider one-sided stride-one convolution in our network. Let $\mathcal{W} = \{\mathcal{W}_{j,k,l}\} \in \mathbb{R}^{C' \times K \times C}$ be a filter where C' is the output channel size, K is the filter size and C is the input channel size. For $z \in \mathbb{R}^{D\times C}$, the convolution of $\mathcal{W}$ with z gives $y \in \mathbb{R}^{D\times C'}$ such that

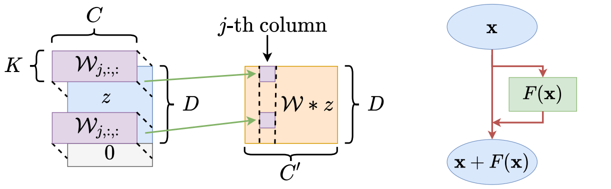
$$y = \mathcal{W} * z, \quad y_{i,j} = \sum_{k=1}^{K}\sum_{l=1}^{C} \mathcal{W}_{j,k,l} z_{i+k-1,l}, \tag{2}$$

where $1 \leq i \leq D, 1 \leq j \leq C'$ and we set $z_{i+k-1,l} = 0$ for $i+k-1 > D$, as demonstrated in Figure 3(a).

The building blocks of ConvResNets are residual blocks. For an input $\mathbf{x}$, each residual block computes

$$\mathbf{x} + F(\mathbf{x})$$

where F is a subnetwork consisting of convolutional layers (see more details in Section 3.1). A residual block is demonstrated in Figure 3(b).



(a) Convolution. (b) A residual block.

Figure 3. (a) Demonstration of $\mathcal{W} * z$, where the input is $z \in \mathbb{R}^{D\times C}$, and the output is $\mathcal{W} * z \in \mathbb{R}^{D\times C'}$. Here $\mathcal{W} = \{\mathcal{W}_{j,k,l}\} \in \mathbb{R}^{C'\times K\times C}$ is a filter where C' is the output channel size, K is the filter size and C is the input channel size. $\mathcal{W}_{j,:,:}$ is a $D\times C$ matrix for the j-th output channel. (b) Demonstration of a residual block.

3. Theory

In this section, we first introduce the ConvResNet architecture, and then present our main results.

3.1. Convolutional residual neural network

We study the ConvResNet with the rectified linear unit (ReLU) activation function: $\mathrm{ReLU}(z) = \max(z, 0)$. The ConvResNet we consider consists of a padding layer and several residual blocks followed by a fully connected feedforward layer.

We first define the padding layer. Given an input $A \in \mathbb{R}^{D\times C_1}$, the network first applies a padding operator $P : \mathbb{R}^{D\times C_1} \to \mathbb{R}^{D\times C_2}$ for some integer $C_2 \geq C_1$ such that

$$Z = P(A) = \begin{bmatrix} A & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{D\times C_2}.$$

Then the matrix Z is passed through M residual blocks.

In the m-th block, let $\mathcal{W}_m = \{\mathcal{W}_m^{(1)}, ..., \mathcal{W}_m^{(L_m)}\}$ and $\mathcal{B}_m = \{B_m^{(1)}, ..., B_m^{(L_m)}\}$ be a collection of filters and biases. The m-th residual block maps a matrix from $\mathbb{R}^{D\times C}$ to $\mathbb{R}^{D\times C}$ by

$$\mathrm{Conv}_{\mathcal{W}_m,\mathcal{B}_m} + \mathrm{id},$$

where id is the identity operator and

$$\mathrm{Conv}_{\mathcal{W}_m,\mathcal{B}_m}(Z) = \mathrm{ReLU}\Big(\mathcal{W}_m^{(L_m)} * \cdots \\ \cdots * \mathrm{ReLU}\left(\mathcal{W}_m^{(1)} * Z + B_m^{(1)}\right) \cdots + B_m^{(L_m)}\Big), \tag{3}$$

with ReLU applied entrywise. Denote

$$Q(\mathbf{x}) = (\mathrm{Conv}_{\mathcal{W}_M,\mathcal{B}_M} + \mathrm{id}) \circ \cdots \\ \circ (\mathrm{Conv}_{\mathcal{W}_1,\mathcal{B}_1} + \mathrm{id}) \circ P(\mathbf{x}). \tag{4}$$

For networks only consisting of residual blocks, we define the network class as

$$\mathcal{C}^{\mathrm{Conv}}(M, L, J, K, \kappa) = \\ \{Q | Q(\mathbf{x}) \text{ is in the form of (4) with } M \text{ residual blocks.} \\ \text{Each block has filter size bounded by } K, \text{ number of} \\ \text{channels bounded by } J, \max_m L_m \leq L, \\ \max_{m,l} \|\mathcal{W}_m^{(l)}\|_\infty \vee \|B_m^{(l)}\|_\infty \leq \kappa.\}. \tag{5}$$

where $\|\cdot\|_\infty$ denotes ℓ^∞ norm of a vector, and for a tensor $\mathcal{W}$, $\|\mathcal{W}\|_\infty = \max_{j,k,l} |\mathcal{W}_{j,k,l}|$.

Based on the network Q in (4), a ConvResNet has an additional fully connected layer and can be expressed as

$$f(\mathbf{x}) = WQ(\mathbf{x}) + \boldsymbol{b} \tag{6}$$

where W and $\boldsymbol{b}$ are the weight matrix and the bias in the fully connected layer. The class of ConvResNets is defined as

$$\mathcal{C}(M, L, J, K, \kappa_1, \kappa_2, R) = \\ \{f | f(\mathbf{x}) = WQ(\mathbf{x}) + \boldsymbol{b} \text{ with } Q \in \mathcal{C}^{\mathrm{Conv}}(M, L, J, K, \kappa_1), \\ \|W\|_\infty \vee |\boldsymbol{b}| \leq \kappa_2, \|f\|_{L^\infty} \leq R\}. \tag{7}$$

Sometimes we do not have restriction on the output, we omit the parameter R and denote the network class by $\mathcal{C}(M, L, J, K, \kappa_1, \kappa_2)$.

3.2. Approximation theory

Our approximation theory is based on the following assumptions of $\mathcal{M}$ and the object function $f^* : \mathcal{M} \to \mathbb{R}$.

Assumption 1. *$\mathcal{M}$ is a d-dimensional compact smooth Riemannian manifold isometrically embedded in $\mathbb{R}^D$. There is a constant B such that for any $\mathbf{x} \in \mathcal{M}$, $\|\mathbf{x}\|_\infty \leq B$.*

Assumption 2. *The reach of $\mathcal{M}$ is $\tau > 0$.*

Assumption 3. *Let $0 < p, q \leq \infty$, $d/p + 1 \leq s < \infty$. Assume $f^* \in B^s_{p,q}(\mathcal{M})$ and $\|f^*\|_{B^s_{p,q}(\mathcal{M})} \leq c_0$ for a constant $c_0 > 0$. Additionally, we assume $\|f^*\|_{L^\infty} \leq R$ for a constant $R > 0$.*

Assumption 3 implies that f^* is Lipschitz continuous (Triebel, 1983, Section 2.7.1 Remark 2 and Section 3.3.1).

Our first result is the following universal approximation error of ConvResNets for Besov functions on $\mathcal{M}$.

Theorem 1. *Assume Assumption 1-3. For any $\varepsilon \in (0, 1)$ and positive integer $K \in [2, D]$, there is a ConvResNet architecture $\mathcal{C}(M, L, J, K, \kappa_1, \kappa_2)$ such that, for any*

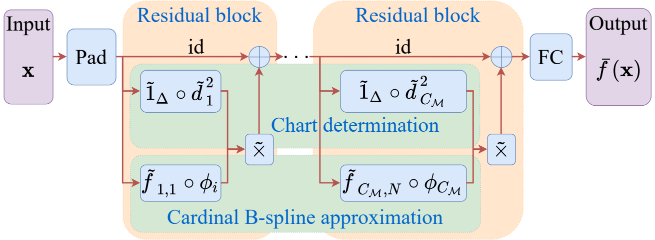

Input
$\mathbf{x}$
Pad
Residual block
id
$\tilde{\mathrm{I}}_\Delta \circ \tilde{d}_1^2$
$\tilde{\times}$
$\tilde{f}_{1,1} \circ \phi_i$
Chart determination
Cardinal B-spline approximation
Residual block
id
$\tilde{\mathrm{I}}_\Delta \circ \tilde{d}_{C_\mathcal{M}}^2$
$\tilde{\times}$
$\tilde{f}_{C_\mathcal{M},N} \circ \phi_{C_\mathcal{M}}$
FC
Output
$\bar{f}(\mathbf{x})$

Figure 4. The ConvResNet in Theorem 1 contains a padding layer, M residual blocks, and a fully connected (FC) layer.

$f^* \in B^s_{p,q}(\mathcal{M})$, if the weight parameters of this ConvResNet are properly chosen, the network yields a function $\bar{f} \in \mathcal{C}(M, L, J, K, \kappa_1, \kappa_2)$ satisfying

$$\|\bar{f} - f^*\|_{L^\infty} \le \varepsilon. \tag{8}$$

Such a network architecture has

$$M = O\left(\varepsilon^{-d/s}\right),\ L = O(\log(1/\varepsilon) + D + \log D),$$
$$J = O(D),\ \kappa_1 = O(1),\ \log \kappa_2 = O(\log^2(1/\varepsilon)). \tag{9}$$

The constant hidden in $O(\cdot)$ depend on d, s, $\frac{2d}{sp-d}$, p, q,c_0, τ and the surface area of $\mathcal{M}$.

The architecture of the ConvResNet in Theorem 1 is illustrated in Figure 4. It has the following properties:

- The network has a fixed filter size and a fixed number of channels.
- There is no cardinality constraint.
- The network size depends on the intrinsic dimension d, and only weakly depends on D.

Theorem 1 can be compared with Suzuki (2019) on the approximation theory for Besov functions in $\mathbb{R}^D$ by FNNs as follows: (1) To universally approximate Besov functions in $\mathbb{R}^D$ with ε error, the FNN constructed in Suzuki (2019) requires $O\left(\log(1/\varepsilon)\right)$ depth, $O\left(\varepsilon^{-D/s}\right)$ width and $O\left(\varepsilon^{-D/s}\log(1/\varepsilon)\right)$ nonzero parameters. By exploiting the manifold model, our network size depends on the intrinsic dimension d and weakly depends on D. (2) The ConvResNet in Theorem 1 does not require any cardinality constraint, while such a constraint is needed in Suzuki (2019).

3.3. Statistical theory

We next consider binary classification on $\mathcal{M}$. For any $\mathbf{x} \in \mathcal{M}$, denote its label by $y \in \{-1, 1\}$. The label y follows the following Bernoulli-type distribution

$$\mathbb{P}(y = 1|\mathbf{x}) = \eta(\mathbf{x}),\ \mathbb{P}(y = -1|\mathbf{x}) = 1 - \eta(\mathbf{x}) \tag{10}$$

for some $\eta : \mathcal{M} \to [0, 1]$.

We assume the following data model:

Assumption 4. *We are given i.i.d. sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathcal{M}$, and the y_i's are sampled according to (10).*

In binary classification, a classifier f predicts the label of $\mathbf{x}$ as $\mathrm{sign}(f(\mathbf{x}))$. To learn the optimal classifier, we consider the logistic loss $\phi(z) = \log(1 + \exp(-z))$. The logistic risk $\mathcal{E}_\phi(f)$ of a classifier f is defined as

$$\mathcal{E}_\phi(f) = \mathbb{E}(\phi(yf(\mathbf{x}))). \tag{11}$$

The minimizer of $\mathcal{E}_\phi(f)$ is denoted by f^*_ϕ, which satisfies

$$f^*_\phi(\mathbf{x}) = \log \frac{\eta(\mathbf{x})}{1 - \eta(\mathbf{x})}. \tag{12}$$

For any classifier f, we define its logistic excess risk as

$$\mathcal{E}_\phi(f, f^*_\phi) = \mathcal{E}_\phi(f) - \mathcal{E}_\phi(f^*_\phi). \tag{13}$$

In this paper, we consider ConvResNets with the following architecture:

$$\mathcal{C}^{(n)} = \{f | f = \bar{g}_2 \circ \bar{h} \circ \bar{g}_1 \circ \bar{\eta} \text{ where}$$
$$\bar{\eta} \in \mathcal{C}^{\mathrm{Conv}}\left(M_1, L_1, J_1, K, \kappa_1\right), \bar{g}_1 \in \mathcal{C}^{\mathrm{Conv}}\left(1, 4, 8, 1, \kappa_2\right),$$
$$\bar{h} \in \mathcal{C}^{\mathrm{Conv}}\left(M_2, L_2, J_2, 1, \kappa_1\right), \bar{g}_2 \in \mathcal{C}\left(1, 3, 8, 1, \kappa_3, 1, R\right)\} \tag{14}$$

where $M_1, M_2, L, J, K, \kappa_1, \kappa_2, \kappa_3$ are some parameters to be determined.

The empirical classifier is learned by minimizing the empirical logistic risk:

$$\widehat{f}_{\phi,n} = \underset{f \in \mathcal{C}^{(n)}}{\mathrm{argmin}}\ \frac{1}{n}\sum_{i=1}^n \phi(y_i f(\mathbf{x}_i)). \tag{15}$$

We establish an upper bound on the excess risk of $\widehat{f}_{\phi,n}$:

Theorem 2. *Assume Assumption 1, 2 and 4. Assume $0 < p, q \le \infty$, $0 < s < \infty$, $s \ge d/p + 1$ and $\eta \in B^s_{p,q}(\mathcal{M})$ with $\|\eta\|_{B^s_{p,q}} \le c_0$ for some constant c_0. For any $2 \le K \le D$, we set*

$$M_1 = O\left(n^{\frac{2d}{s+2(s\vee d)}}\right),\ M_2 = O\left(n^{\frac{2s}{s+2(s\vee d)}}\right),$$
$$L_1 = O(\log(1/\varepsilon) + D + \log D),\ L_2 = O(\log(1/\varepsilon)),$$
$$J_1 = O(D),\ J_2 = O(1),\ \kappa_1 = O(1),$$
$$\log \kappa_2 = O(\log^2 n),\ \kappa_3 = O(\log n),\ R = O(\log n)$$

for $\mathcal{C}^{(n)}$. Then

$$\mathbb{E}(\mathcal{E}_\phi(\widehat{f}_{\phi,n}, f^*_\phi)) \le Cn^{-\frac{s}{2s+2(s\vee d)}}\log^4 n \tag{16}$$

for some constant C. Here C is linear in $D\log D$ and additionally depends on $d, s, \frac{2d}{sp-d}, p, q, c_0, \tau$ and the surface area of $\mathcal{M}$. The constant hidden in $O(\cdot)$ depends on d, s, $\frac{2d}{sp-d}, p, q, c_0, \tau$ and the surface area of $\mathcal{M}$.

Theorem 2 shows that a properly designed ConvResNet gives rise to an empirical classifier, of which the excess risk converges at a fast rate with an exponent depending on the intrinsic dimension d, instead of D.

Theorem 2 is proved in Appendix A. Each building block of $\mathcal{C}^{(n)}$ is constructed for the following purpose:

- $\bar{g}_1 \circ \bar{\eta}$ is designed to approximate a truncated η on $\mathcal{M}$, which is realized by Theorem 1.
- $\bar{g}_2 \circ \bar{h}$ is designed to approximate a truncated univariate function $\log \frac{z}{1-z}$.

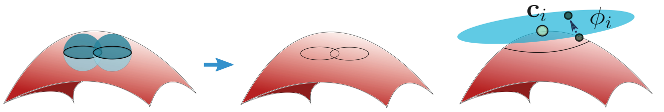


Figure 5. An atlas given by covering $\mathcal{M}$ using Euclidean balls.

4. Proof of Theorem 1

We provide a proof sketch of Theorem 1 in this section. More technical details are deferred to Appendix C.

We prove Theorem 1 in the following four steps:

1. Decompose $f^* = \sum_i f_i$ as a sum of locally supported functions according to the manifold structure.
2. Locally approximate each f_i using cardinal B-splines.
3. Implement the cardinal B-splines using CNNs.
4. Implement the sum of all CNNs by a ConvResNet for approximating f^*.

Step 1: Decomposition of f^*.
• **Construct an atlas on $\mathcal{M}$.** Since the manifold $\mathcal{M}$ is compact, we can cover $\mathcal{M}$ by a finite collection of open balls $B_\omega(\mathbf{c}_i)$ for $i = 1, \dots, C_\mathcal{M}$, where $\mathbf{c}_i$ is the center of the ball and ω is the radius to be chosen later. Accordingly, the manifold is partitioned as $\mathcal{M} = \bigcup_i U_i$ with $U_i = B_\omega(\mathbf{c}_i) \bigcap \mathcal{M}$. We choose $\omega < \tau/2$ such that U_i is diffeomorphic to an open subset of $\mathbb{R}^d$ (Niyogi et al., 2008, Lemma 5.4). The total number of partitions is then bounded by $C_\mathcal{M} \le \left\lceil \frac{\mathrm{SA}(\mathcal{M})}{\omega^d} T_d \right\rceil$, where $\mathrm{SA}(\mathcal{M})$ is the surface area of $\mathcal{M}$ and T_d is the average number of U_i's that contain a given point on $\mathcal{M}$ (Conway et al., 1987, Chapter 2 Equation (1)).

On each partition, we define a projection-based transformation ϕ_i as

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

where the scaling factor $a_i \in \mathbb{R}$ and the shifting vector $\mathbf{b}_i \in \mathbb{R}^d$ ensure $\phi_i(U_i) \subset [0,1]^d$, and the column vectors of $V_i \in \mathbb{R}^{D\times d}$ form an orthonormal basis of the tangent space $T_{\mathbf{c}_i}(\mathcal{M})$. The atlas on $\mathcal{M}$ is the collection (U_i, ϕ_i) for $i = 1, \dots, \mathcal{M}$. See Figure 5 for a graphical illustration of the atlas.

• **Decompose f^* according to the atlas.** We decompose f^* as

$$f^* = \sum_{i=1}^{C_\mathcal{M}} f_i \quad \text{with} \quad f_i = f\rho_i, \tag{17}$$

where $\{\rho_i\}_{i=1}^{C_\mathcal{M}}$ is a C^∞ partition of unity with $\mathrm{supp}(\phi_i) \subset U_i$. The existence of such a $\{\rho_i\}_{i=1}^{C_\mathcal{M}}$ is guaranteed by Proposition 1. As a result, each f_i is supported on a subset of U_i, and therefore, we can rewrite (17) as

$$f^* = \sum_{i=1}^{C_\mathcal{M}} (f_i \circ \phi_i^{-1}) \circ \phi_i \times \mathbb{1}_{U_i} \quad \text{with} \quad f_i = f\rho_i, \tag{18}$$

where $\mathbb{1}_{U_i}$ is the indicator function of U_i. Since ϕ_i is a bijection between U_i and $\phi_i(U_i)$, $f_i \circ \phi_i^{-1}$ is supported on $\phi_i(U_i) \subset [0,1]^d$. We extend $f_i \circ \phi_i^{-1}$ on $[0,1]^d \backslash \phi_i(U_i)$ by 0. The extended function is in $B^s_{p,q}([0,1]^d)$ (see Lemma 4 in Appendix C.1). This allows us to use cardinal B-splines to locally approximate each $f_i \circ \phi_i^{-1}$ as detailed in **Step 2**.

Step 2: Local cardinal B-spline approximation. We approximate $f_i \circ \phi_i^{-1}$ using cardinal B-splines $\widetilde{f}_i$ as

$$f_i \circ \phi_i^{-1} \approx \widetilde{f}_i \equiv \sum_{j=1}^{N} \widetilde{f}_{i,j} \;\; \text{with} \;\; \widetilde{f}_{i,j} = \alpha^{(i)}_{k,\mathbf{j}} M^d_{k,\mathbf{j},m}, \tag{19}$$

where $\alpha^{(i)}_{k,\mathbf{j}} \in \mathbb{R}$ is a coefficient and $M^d_{k,\mathbf{j},m} : [0,1]^d \to \mathbb{R}$ denotes a cardinal B-spline with indecies $k, m \in \mathbb{N}^+, \mathbf{j} \in \mathbb{R}^d$. Here k is a scaling factor, $\mathbf{j}$ is a shifting vector, m is the degree of the B-spline and d is the dimension (see a formal definition in Appendix C.2).

Since $s \ge d/p + 1$ (by Assumption 3), setting $r = +\infty, m = \lceil s \rceil + 1$ in Lemma 5 (see Appendix C.3) and applying Lemma 4 gives

$$\left\| \widetilde{f}_i - f_i \circ \phi_i^{-1} \right\|_{L^\infty} \le C c_0 N^{-s/d} \tag{20}$$

for some constant C depending on s, p, q and d.

Combining (18) and (19), we approximate f^* by

$$\widetilde{f}^* \equiv \sum_{i=1}^{C_\mathcal{M}} \widetilde{f}_i \circ \phi_i \times \mathbb{1}_{U_i} = \sum_{i=1}^{C_\mathcal{M}} \sum_{j=1}^{N} \widetilde{f}_{i,j} \circ \phi_i \times \mathbb{1}_{U_i}. \tag{21}$$

Such an approximation has error

$$\|\widetilde{f}^* - f^*\|_{L^\infty} \le C C_\mathcal{M} c_0 N^{-s/d}.$$

Step 3: Implement local approximations in Step 2 by CNNs. In **Step 2**, (21) gives a natural approximation of f^*. In the sequel, we aim to implement all ingredients of $\widetilde{f}_{i,j} \circ \phi_i \times \mathbb{1}_{U_i}$ using CNNs. In particular, we show that CNNs can implement the cardinal B-spline $\widetilde{f}_{i,j}$, the linear projection ϕ_i, the indicator function $\mathbb{1}_{U_i}$, and the multiplication operation.

• **Implement $\mathbb{1}_{U_i}$ by CNNs.** Recall our construction of U_i in **Step 1**. For any $\mathbf{x} \in \mathcal{M}$, we have $\mathbb{1}_{U_i}(\mathbf{x}) = 1$ if $d_i^2(\mathbf{x}) = \|\mathbf{x} - \mathbf{c}_i\|_2^2 \le \omega^2$; otherwise $\mathbb{1}_{U_i}(\mathbf{x}) = 0$.

To implement $\mathbb{1}_{U_i}$, we rewrite it as the composition of a univariate indicator function $\mathbb{1}_{[0,\omega^2]}$ and the distance function d_i^2:

$$\mathbb{1}_{U_i}(\mathbf{x}) = \mathbb{1}_{[0,\omega^2]} \circ d_i^2(\mathbf{x}) \quad \text{for} \quad \mathbf{x} \in \mathcal{M}. \tag{22}$$

We show that CNNs can efficiently implement both $\mathbb{1}_{[0,\omega^2]}$ and d_i^2. Specifically, given $\theta \in (0,1)$ and $\Delta \ge 8DB^2\theta$, there exist CNNs that yield functions $\widetilde{\mathbb{1}}_\Delta$ and $\widetilde{d}_i^2$ satisfying

$$\|\widetilde{d}_i^2 - d_i^2\|_{L^\infty} \le 4B^2 D\theta \tag{23}$$

and

$$\widetilde{\mathbb{1}}_\Delta \circ \widetilde{d}_i^2(\mathbf{x}) = \begin{cases} 1, \text{ if } \mathbf{x} \in U_i, d_i^2(\mathbf{x}) \le \omega^2 - \Delta, \\ 0, \text{ if } \mathbf{x} \notin U_i, \\ \text{between } 0 \text{ and } 1, \text{ otherwise.} \end{cases} \tag{24}$$

We also characterize the network sizes for realizing $\widetilde{\mathbb{1}}_\Delta$ and $\widetilde{d}_i^2$: The network for $\widetilde{\mathbb{1}}_\Delta$ has $O(\log(\omega^2/\Delta))$ layers, 2 channels and all weight parameters bounded by $\max(2, |\omega^2 - 4B^2D\theta|)$; the network for $\widetilde{d}_i^2$ has $O(\log(1/\theta) + D)$ layers, $6D$ channels and all weight parameters bounded by $4B^2$. More technical details are provided in Lemma 9 in Appendix C.6.

- **Implement $\widetilde{f}_{i,j} \circ \phi_i$ by CNNs.** Since ϕ_i is a linear projection, it can be realized by a single-layer perceptron. By Lemma 8 (see Appendix C.5), this single-layer perceptron can be realized by a CNN, denoted by ϕ_i^{CNN}.

For $\widetilde{f}_{i,j}$, Proposition 3 (see Appendix C.8) shows that for any $\delta \in (0,1)$ and $2 \le K \le d$, there exists a CNN $\widetilde{f}_{i,j}^{\text{CNN}} \in \mathcal{F}^{\text{CNN}}(L, J, K, \kappa, \kappa)$ with

$$L = O\left(\log \frac{1}{\delta}\right), J = O(1), \kappa = O\left(\delta^{-(\log 2)(\frac{2d}{sp-d} + \frac{c_1}{d})}\right)$$

such that when setting $N = C_1 \delta^{-d/s}$, we have

$$\Big\| \sum_{j=1}^N \widetilde{f}_{i,j}^{\text{CNN}} - f_i \circ \phi_i^{-1} \Big\|_{L^\infty(\phi_i(U_i))} \le \delta, \tag{25}$$

where C_1 is a constant depending on s, p, q and d. The constant hidden in $O(\cdot)$ depends on $d, s, \frac{2d}{sp-d}, p, q, c_0$. The CNN class $\mathcal{F}^{\text{CNN}}$ is defined in Appendix B.

- **Implement the multiplication $\times$ by a CNN.** According to Lemma 7 (see Appendix C.4) and Lemma 8, for any $\eta \in (0,1)$, the multiplication operation $\times$ can be approximated by a CNN $\widetilde{\times}$ with L^∞ error η:

$$\|a \times b - \widetilde{\times}(a, b)\|_{L^\infty} \le \eta. \tag{26}$$

Such a CNN has $O(\log 1/\eta)$ layers, 6 channels. All parameters are bounded by $\max(2c_0^2, 1)$.

Step 4: Implement $\widetilde{f}^*$ by a ConvResNet. We assemble all CNN approximations in **Step 3** together and show that the whole approximation can be realized by a ConvResNet.

- **Assemble all ingredients together.** Assembling all CNN approximations together gives an approximation of $\widetilde{f}_{i,j} \circ \phi_i \times \mathbb{1}_{U_i}$ as

$$\mathring{f}_{i,j} \equiv \widetilde{\times}\left(\widetilde{f}_{i,j}^{\text{CNN}} \circ \phi_i^{\text{CNN}}, \widetilde{\mathbb{1}}_\Delta \circ \widetilde{d}_i^2\right). \tag{27}$$

After substituting (27) into (21), we approximate the target function f^* by

$$\mathring{f} \equiv \sum_{i=1}^{C_{\mathcal{M}}} \sum_{j=1}^N \mathring{f}_{i,j}. \tag{28}$$

The approximation error of $\mathring{f}$ is analyzed in Lemma 12 (see Appendix C.9). According to Lemma 12, the approximation error can be bounded as follows:

$$\|\mathring{f} - f^*\|_{L^\infty} \le \sum_{i=1}^{C_{\mathcal{M}}} (A_{i,1} + A_{i,2} + A_{i,3}) \quad \text{with}$$

$$A_{i,1} = \sum_{j=1}^N \Big\| \widetilde{\times}(\widetilde{f}_{i,j}^{\text{CNN}} \circ \phi_i^{\text{CNN}}, \widetilde{\mathbb{1}}_\Delta \circ \widetilde{d}_i^2) - (\widetilde{f}_{i,j}^{\text{CNN}} \circ \phi_i^{\text{CNN}}) \times (\widetilde{\mathbb{1}}_\Delta \circ \widetilde{d}_i^2) \Big\|_{L^\infty} \le N\eta,$$

$$A_{i,2} = \Big\| \Big(\sum_{j=1}^N \left(\widetilde{f}_{i,j}^{\text{CNN}} \circ \phi_i^{\text{CNN}} \right) \Big) \times (\widetilde{\mathbb{1}}_\Delta \circ \widetilde{d}_i^2) - f_i \times (\widetilde{\mathbb{1}}_\Delta \circ \widetilde{d}_i^2) \Big\|_{L^\infty} \le \delta,$$

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

where δ, η, Δ and θ are defined in (25), (26), (24) and (23), respectively. For any $\varepsilon \in (0,1)$, with properly chosen δ, η, Δ and θ as in (53) in Lemma 12, one has

$$\|\mathring{f} - f^*\|_{L^\infty} \le \varepsilon. \tag{29}$$

With these choices, the network size of each CNN is quantified in Appendix C.10.

- **Realize $\mathring{f}$ by a ConvResNet.** Lemma 17 (see Appendix C.15) shows that for every $\mathring{f}_{i,j}$, there exists $\bar{f}_{i,j}^{\text{CNN}} \in \mathcal{F}^{\text{CNN}}(L, J, K, \kappa_1, \kappa_2)$ with $L = O(\log 1/\varepsilon + D + \log D), J = O(D), \kappa_1 = O(1), \log \kappa_2 = O\left(\log^2 1/\varepsilon\right)$ such that $\bar{f}_{i,j}^{\text{CNN}}(\mathbf{x}) = \mathring{f}_{i,j}(\mathbf{x})$ for any $\mathbf{x} \in \mathcal{M}$. As a result, the function $\mathring{f}$ in (28) can be expressed as a sum of CNNs:

$$\mathring{f} = \bar{f}^{\text{CNN}} \equiv \sum_{i=1}^{C_{\mathcal{M}}} \sum_{j=1}^N \bar{f}_{i,j}^{\text{CNN}}, \tag{30}$$

where N is chosen of $O\left(\varepsilon^{-d/s}\right)$ (see Proposition 3 and Lemma 12). Lemma 18 (see Appendix C.16) shows that $\bar{f}^{\text{CNN}}$ can be realized by $\bar{f} \in \mathcal{C}(M, L, J, \kappa_1, \kappa_2)$ with

$$M = O\left(\varepsilon^{-d/s}\right), L = O(\log(1/\varepsilon) + D + \log D),$$
$$J = O(D), \kappa_1 = O(1), \log \kappa_2 = O\left(\log^2(1/\varepsilon)\right).$$

5. Conclusion

Our results show that ConvResNets are adaptive to low-dimensional geometric structures of data sets. Specifically, we establish a universal approximation theory of ConvResNets for Besov functions on a d-dimensional manifold $\mathcal{M}$. Our network size depends on the intrinsic dimension d and only weakly depends on D. We also establish a statistical theory of ConvResNets for binary classification when the given data are located on $\mathcal{M}$. The classifier is learned by minimizing the empirical logistic loss. We prove that if the

ConvResNet architecture is properly chosen, the excess risk of the learned classifier decays at a fast rate depending on the intrinsic dimension of the manifold.

Our ConvResNet has many practical properties: it has a fixed filter size and a fixed number of channels. Moreover, it does not require any cardinality constraint, which is beneficial to training.

Our analysis can be extended to multinomial logistic regression for multi-class classification. In this case, the network will output a vector where each component represents the likelihood of an input belonging to certain class. By assuming that each likelihood function is in the Besov space, we can apply our analysis to approximate each function by a ConvResNet.

References

Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology*, 33:831–838, 2015.

Barron, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

Blalock, D., Ortiz, J. J. G., Frankle, J., and Guttag, J. What is the state of neural network pruning? *arXiv preprint arXiv:2003.03033*, 2020.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. DeepLAB: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

Chen, M., Jiang, H., Liao, W., and Zhao, T. Nonparametric regression on low-dimensional manifolds using deep ReLU networks. *arXiv preprint arXiv:1908.01842*, 2019a.

Chen, M., Jiang, H., Liao, W., and Zhao, T. Efficient approximation of deep ReLU networks for functions on low dimensional manifolds. In *Advances in Neural Information Processing Systems*, pp. 8172–8182, 2019b.

Chen, M., Liu, H., Liao, W., and Zhao, T. Doubly robust off-policy learning on low-dimensional manifolds by deep neural networks. *arXiv preprint arXiv:2011.01797*, 2020.

Chui, C. K. and Mhaskar, H. N. Deep nets for local manifold learning. *Frontiers in Applied Mathematics and Statistics*, 4:12, 2018.

Cloninger, A. and Klock, T. ReLU nets adapt to intrinsic dimensionality beyond the target domain. *arXiv preprint arXiv:2008.02545*, 2020.

Conway, J. H., Sloane, N. J. A., and Bannai, E. *Sphere-packings, Lattices, and Groups*. Springer-Verlag, Berlin, Heidelberg, 1987. ISBN 0-387-96617-X.

Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

DeVore, R. A. and Lorentz, G. G. *Constructive Approximation*, volume 303. Springer Science & Business Media, 1993.

DeVore, R. A. and Popov, V. A. Interpolation of Besov spaces. *Transactions of the American Mathematical Society*, 305(1):397–414, 1988.

Dispa, S. Intrinsic characterizations of Besov spaces on lipschitz domains. *Mathematische Nachrichten*, 260(1): 21–33, 2003.

Dũng, D. Optimal adaptive sampling recovery. *Advances in Computational Mathematics*, 34(1):1–41, 2011.

Fang, Z., Feng, H., Huang, S., and Zhou, D.-X. Theory of deep convolutional neural networks II: Spherical analysis. *Neural Networks*, 131:154–162, 2020.

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

Geer, S. A. and van de Geer, S. *Empirical Processes in M-estimation*, volume 6. Cambridge University press, 2000.

Geller, D. and Pesenson, I. Z. Band-limited localized parseval frames and Besov spaces on compact homogeneous manifolds. *Journal of Geometric Analysis*, 21(2):334–371, 2011.

Girshick, R. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, 2015.

Gong, S., Boddeti, V. N., and Jain, A. K. On the intrinsic dimensionality of image representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3987–3996, 2019.

Graves, A., Mohamed, A.-r., and Hinton, G. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649. IEEE, 2013.

Hamers, M. and Kohler, M. Nonasymptotic bounds on the L_2 error of neural network regression estimates. *Annals of the Institute of Statistical Mathematics*, 58(1):131–151, 2006.

Han, S., Pool, J., Tran, J., and Dally, W. J. Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*, 2015.

Han, S., Pool, J., Narang, S., Mao, H., Gong, E., Tang, S., Elsen, E., Vajda, P., Paluri, M., Tran, J., et al. Dsd: Dense-sparse-dense training for deep neural networks. *arXiv preprint arXiv:1607.04381*, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science*, 313 (5786):504–507, 2006.

Hu, T., Shang, Z., and Cheng, G. Sharp rate of convergence for deep neural network classifiers under the teacher-student setting. *arXiv preprint arXiv:2001.06892*, 2020.

Huang, F., Ash, J., Langford, J., and Schapire, R. Learning deep resnet blocks sequentially using boosting theory. In *International Conference on Machine Learning*, pp. 2058–2067, 2018.

Jaffard, S., Meyer, Y., and Ryan, R. D. *Wavelets: tools for science and technology*. SIAM, 2001.

Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., Wang, Y., Dong, Q., Shen, H., and Wang, Y. Artificial intelligence in healthcare: past, present and future. *Stroke and vascular neurology*, 2(4):230–243, 2017.

Kim, Y., Ohn, I., and Kim, D. Fast convergence rates of deep neural networks for classification. *arXiv preprint arXiv:1812.03599*, 2018.

Kohler, M. and Krzyżak, A. Adaptive regression estimation with multilayer feedforward neural networks. *Nonparametric Statistics*, 17(8):891–913, 2005.

Kohler, M. and Langer, S. Statistical theory for image classification using deep convolutional neural networks with cross-entropy loss. *arXiv preprint arXiv:2011.13602*, 2020.

Kohler, M. and Mehnert, J. Analysis of the rate of convergence of least squares neural network regression estimates in case of measurement errors. *Neural Networks*, 24(3): 273–279, 2011.

Kohler, M., Krzyzak, A., and Walter, B. On the rate of convergence of image classifiers based on convolutional neural networks. *arXiv preprint arXiv:2003.01526*, 2020.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

Lee, H., Ge, R., Ma, T., Risteski, A., and Arora, S. On the ability of neural nets to express distributions. In *Conference on Learning Theory*, pp. 1271–1296, 2017.

Lee, J. M. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width. In *Advances in Neural Information Processing Systems*, pp. 6231–6239, 2017.

McCaffrey, D. F. and Gallant, A. R. Convergence rates for single hidden layer feedforward networks. *Neural Networks*, 7(1):147–158, 1994.

Mhaskar, H. N. and Micchelli, C. A. Approximation by superposition of sigmoidal and radial basis functions. *Advances in Applied mathematics*, 13(3):350–373, 1992.

Miotto, R., Wang, F., Wang, S., Jiang, X., and Dudley, J. T. Deep learning for healthcare: review, opportunities and challenges. *Briefings in Bioinformatics*, 19(6):1236–1246, 2018.

Montanelli, H. and Yang, H. Error bounds for deep ReLU networks using the Kolmogorov–Arnold superposition theorem. *Neural Networks*, 129:1–6, 2020.

Nakada, R. and Imaizumi, M. Adaptive approximation and estimation of deep neural network with intrinsic dimensionality. *arXiv preprint arXiv:1907.02177*, 2019.

Nitanda, A. and Suzuki, T. Functional gradient boosting based on residual network perception. In *International Conference on Machine Learning*, pp. 3819–3828, 2018.

Niyogi, P., Smale, S., and Weinberger, S. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1-3): 419–441, 2008.

Ohn, I. and Kim, Y. Smooth function approximation by deep neural networks with general activation functions. *Entropy*, 21(7):627, 2019.

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

Osher, S., Shi, Z., and Zhu, W. Low dimensional manifold model for image processing. *SIAM Journal on Imaging Sciences*, 10(4):1669–1690, 2017.

Park, C. Convergence rates of generalization errors for margin-based classification. *Journal of Statistical Planning and Inference*, 139(8):2543–2551, 2009.

Petersen, P. and Voigtlaender, F. Equivalence of approximation by convolutional neural networks and fully-connected networks. *Proceedings of the American Mathematical Society*, 148(4):1567–1581, 2020.

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

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

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

Shaham, U., Cloninger, A., and Coifman, R. R. Provable approximation properties for deep neural networks. *Applied and Computational Harmonic Analysis*, 44(3):537–557, 2018.

Shen, X. and Wong, W. H. Convergence rate of sieve estimates. *The Annals of Statistics*, pp. 580–615, 1994.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Spivak, M. D. *A comprehensive introduction to differential geometry*. Publish or Perish, 1970.

Suzuki, T. Adaptivity of deep ReLU network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality. In *International Conference on Learning Representations*, 2019.

Suzuki, T. and Nitanda, A. Deep learning is adaptive to intrinsic dimensionality of model smoothness in anisotropic besov space. *arXiv preprint arXiv:1910.12799*, 2019.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Triebel, H. *Theory of Function Spaces*. Modern Birkhäuser Classics. Birkhäuser Basel, 1983.

Triebel, H. *Theory of function spaces II*. Monographs in Mathematics. Birkhäuser Basel, 1992.

Tu, L. *An Introduction to Manifolds*. Universitext. Springer New York, 2010. ISBN 9781441973993.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

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

Young, T., Hazarika, D., Poria, S., and Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75, 2018.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Zhang, Z., Lin, H., Sun, Y., He, T., Mueller, J., Manmatha, R., et al. ResNeSt: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.

Zhou, D.-X. Theory of deep convolutional neural networks: Downsampling. *Neural Networks*, 124:319–327, 2020a.

Zhou, D.-X. Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, 48(2):787–794, 2020b.

Zhou, J. and Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature Methods*, 12(10):931–934, 2015.

Supplementary Materials for Besov Function Approximation and Binary Classification on Low-Dimensional Manifolds Using Convolutional Residual Networks

Notations: Throughout our proofs, we define the following notations: For two functions $f : \Omega \to \mathbb{R}$ and $g : \Omega \to \mathbb{R}$ defined on some domain Ω, we denote $f \lesssim g$ if there is a constant C such that $f(\mathbf{x}) \leq Cg(\mathbf{x})$ for all $\mathbf{x} \in \Omega$. Similarly, we denote $f \gtrsim g$ if there is a constant C such that $f(\mathbf{x}) \geq Cg(\mathbf{x})$ for all $\mathbf{x} \in \Omega$. We denote $f \asymp g$ if $f \lesssim g$ and $f \gtrsim g$. We use $\mathbb{N}$ to denote the set of all nonnegative integers. For a real number a, we denote $a_+ = \max(a, 0)$ and $a_- = \min(a, 0)$.

The proof of Theorem 1 is sketched in Section 4. In this supplementary material, we prove Theorem 2 in Section A. We define convolutional network and multi-layer perceptrons classes in Section B, based on which the lemmas used in Section 4 are proved in Section C. The lemmas used in Section A are proved in Section D.

A. Proof of Theorem 2

A.1. Basic definitions and tools

We first define the bracketing entropy and covering number which are used in the proof of Theorem 2.

Definition 10 (Bracketing entropy)**.** *A set of function pairs $\{(f_i^L, f_i^U)\}_{i=1}^N$ is called a δ-bracketing of a function class $\mathcal{F}$ with respect to the norm $\|\cdot\|$ if for any i, $\|f_i^U - f_i^L\| \leq \delta$ and for any $f \in \mathcal{F}$, there exists a pair (f_i^L, f_i^U) such that $f_i^L \leq f \leq f_i^U$. The δ-bracketing number is defined as the cardinality of the minimal δ-bracketing set and is denoted by $\mathcal{N}_B(\delta, \mathcal{F}, \|\cdot\|)$. The δ-bracketing enropy, denoted by $\mathcal{H}_B(\delta, \mathcal{F}, \|\cdot\|)$, is defined as*

$$\mathcal{H}_B(\delta, \mathcal{F}, \|\cdot\|) = \log \mathcal{N}_B(\delta, \mathcal{F}, \|\cdot\|).$$

Definition 11 (Covering number)**.** *Let $\mathcal{F}$ be a set with metric ρ. A δ-cover of $\mathcal{F}$ is a set $\{f_1^*, ..., f_N^*\} \subset \mathcal{F}$ such that for any $f \in \mathcal{F}$, there exists f_k^* for some k such that $\rho(f, f_k^*) \leq \delta$. The δ-covering number of $\mathcal{F}$ is defined as*

$$\mathcal{N}(\delta, \mathcal{F}, \rho) = \inf\{N : \textit{ there exists a } \delta - \textit{cover } \{f_1^*, ..., f_N^*\} \textit{ of } \mathcal{F}\}.$$

It has been shown (Geer & van de Geer, 2000, Lemma 2.1) that for any $\delta > 0, p \geq 1$,

$$\mathcal{H}_B(\delta, \mathcal{F}, \|\cdot\|_{L^p}) \leq \log \mathcal{N}(\delta/2, \mathcal{F}, \|\cdot\|_{L^\infty}).$$

The proof of Theorem 2 relies on the following proposition which is a modified version of Kim et al. (2018, Theorem 5):

Proposition 2. *Let ϕ be a surrogate loss function for binary classification. Let $f_\phi^*, \mathcal{E}_\phi(f_n, f_\phi^*)$ be defined as in (12) and (13), respectively. Assume the following regularity conditions:*

(A1) ϕ is Lipschitz: $|\phi(z_1) - \phi(z_2)| \leq C_1|z_1 - z_2|$ for any z_1, z_2 and some constant C_1.

(A2) For a positive sequence $a_n = O(n^{-a_0})$ for some $a_0 > 0$, there exists a sequence of function classes $\{\mathcal{F}_n\}_{n\in\mathbb{N}}$ such that as $n \to \infty$,

$$\mathcal{E}_\phi(f_n, f_\phi^*) \leq a_n$$

for some $f_n \in \mathcal{F}_n$.

(A3) There exists a sequence $\{F_n\}_{n\in\mathbb{N}}$ with $F_n \gtrsim 1$ such that $\sup_{f\in\mathcal{F}_n} \|f\|_{L^\infty} \leq F_n$.

(A4) There exists a constant $\nu \in (0, 1]$ such that for any $f \in \mathcal{F}_n$ and any $n \in \mathbb{N}$,

$$\mathbb{E}\left(\phi(yf(\mathbf{x})) - \phi(yf_\phi^*(\mathbf{x}))\right)^2 \leq C_2 F_n^{2-\nu} e^{F_n} \left(\mathcal{E}_\phi(f, f_\phi^*)\right)^\nu$$

for some constant $C_2 > 0$ only depending on ϕ and η.

(A5) For a positive constant $C_3 > 0$, there exists a sequence $\{\delta_n\}_{n\in\mathbb{N}}$ such that

$$\mathcal{H}_B(\delta_n, \mathcal{F}_n, \|\cdot\|_{L^2}) \leq C_3 e^{-F_n} n \left(\frac{\delta_n}{F_n}\right)^{2-\nu}$$

for $\{\mathcal{F}_n\}_{n\in\mathbb{N}}$ in (A2), $\{F_n\}_{n\in\mathbb{N}}$ in (A3) and ν in (A4).

Let $\epsilon_n^2 \asymp \max(a_n, \delta_n)$. Then the empirical ϕ-risk minimizer $\widehat{f}_{\phi,n}$ over $\mathcal{F}_n$ satisfies

$$\mathbb{P}\left(\mathcal{E}_\phi(\widehat{f}_{\phi,n}, f_\phi^*) \geq \epsilon_n\right) \leq C_5 \exp\left(-C_4 e^{-F_n} n \left(\epsilon_n^2 / (F_n)\right)^{2-\nu}\right) \tag{31}$$

for some constants $C_4, C_5 > 0$.

Proposition 2 is proved in Appendix D.1. In Proposition 2, condition (A1) requires the surrogate loss function ϕ to be Lipschitz. This condition is satisfied in Theorem 2 since ϕ is the logistic loss. (A2) is a condition on the bias of $\widehat{f}_{\phi,n}$. Take n as the number of samples. (A2) requires the bias to decrease in the order of $O(n^{-a_0})$ for some a_0. (A3) requires all functions in the class $\mathcal{F}_n$ to be bounded. (A4) and (A5) are conditions relate to the variance of $\widehat{f}_{\phi,n}$. Condition (A4) for logistic loss can be verified using the following lemma:

Lemma 1 (Lemma 6.1 in Park (2009))**.** *Let ϕ be the logistic loss. Given a function class $\mathcal{F}$ which is uniformly bounded by F, for any function $f \in \mathcal{F}$, we have*

$$\mathbb{E}\left[\phi(yf) - \phi(yf_\phi^*)\right]^2 \leq Ce^F \mathcal{E}_\phi(f, f_\phi^*)$$

for some constant C.

According to Lemma 1, (A4) is verified with $\nu = 1$. Now we are ready to prove Theorem 2.

A.2. Proof of Theorem 2

Proof of Theorem 2. The main idea of the proof is to construct a sequence of network architectures, depending on n, such that Condition (A1)-(A5) in Proposition 2 are satisfied. The excess risk is then derived from (31). In particular, we choose

$$\mathcal{F}_n = \mathcal{C}^{(n)}, a_n = n^{-\frac{s}{2s+2(s\vee d)}} \log^2 n, F_n = \frac{s}{2s + 2(s \vee d)} \log n, \delta_n = n^{-\frac{s}{2s+2(s\vee d)}} \log^4 n \tag{32}$$

where $\mathcal{C}^{(n)}$ is the network architecture in Theorem 2.

We first prove the probability bound of $\mathcal{E}_\phi(\widehat{f}_{\phi,n}, f_\phi^*)$ by checking conditions (A1)-(A5) in Proposition 2. Note that ϕ is the logistic loss which is Lipschitz continuous with Lipschitz constant 1. Thus (A1) is verified. According to Lemma 1, (A4) is verified with $\nu = 1$. We next verify (A2), (A3) and (A5).

A truncation technique. Recall that $f^* = \log \frac{\eta}{1-\eta}$. As η goes to 0 (resp. 1), f^* goes to ∞ (resp. $-\infty$). Note that (A3) requires the function class $\mathcal{F}_n$ to be bounded by F_n. To study the approximation error of $\mathcal{F}_n$ with respect to f^*, we consider a truncated version of f^* defined as

$$f_{\phi,n}^* = \begin{cases} F_n, & \text{if } f_\phi^* > F_n, \\ f_\phi^*, & \text{if } -F_n \leq f_\phi^* \leq F_n, \\ -F_n, & \text{if } f_\phi^* < -F_n. \end{cases} \tag{33}$$

Verification of (A2) and (A3). The following lemma is a very important lemma on approximating $f_{\phi,n}^*$ by ConvResNets. It also provides the covering number of the network class which will be used to verify (A5).

Lemma 2. *Assume Assumption 1 and 2. Assume $0 < p, q \leq \infty$, $0 < s < \infty$, $s \geq d/p + 1$. For any $\varepsilon \in (0, 1)$ and any $K \leq D$, there exists a ConvResNet architecture*

$$\mathcal{C}^{(F_n)} = \big\{ f | f = \bar{g}_2 \circ \bar{h} \circ \bar{g}_1 \circ \bar{\eta} \text{ where } \bar{\eta} \in \mathcal{C}^{\mathrm{Conv}}(M_1, L_1, J_1, K, \kappa_1),\ \bar{g}_1 \in \mathcal{C}^{\mathrm{Conv}}(1, 4, 8, 1, \kappa_2), \\ \bar{h} \in \mathcal{C}^{\mathrm{Conv}}(M_2, L_2, J_2, 1, \kappa_1),\ \bar{g}_2 \in \mathcal{C}(1, 3, 8, 1, \kappa_3, 1, R) \big\}$$

with

$$M_1 = O\left(\varepsilon^{-d/s}\right),\ M_2 = O\left(e^{-F_n}\varepsilon^{-1}\right), L_1 = O(\log(1/\varepsilon) + D + \log D),\ L_2 = O(\log(1/\varepsilon)),$$

$$J_1 = O(D),\ J_2 = O(1),\ \kappa_1 = O(1),\ \log \kappa_2 = O(\log^2(1/\varepsilon)),\ \kappa_3 = O(\log(F_n/\varepsilon) + F_n),\ R = F_n,$$

such that for any $\eta \in B_{p,q}^s(\mathcal{M})$ with $\|\eta\|_{B_{p,q}^s(\mathcal{M})} \leq c_0$ for some constant c_0, and $f_{\phi,n}^*$ be defined as in (33), there exists $\bar{f}_{\phi,n} \in \mathcal{C}^{(F_n)}$ with

$$\|\bar{f}_{\phi,n} - f_{\phi,n}^*\|_{L^\infty} \leq 4e^{F_n}\varepsilon.$$

Moreover, the covering number of $\mathcal{C}^{(F_n)}$ is bounded by

$$\mathcal{N}(\delta, \mathcal{C}^{(F_n)}, \|\cdot\|_{L^\infty}) = O\left(D^3 \varepsilon^{-\left(\frac{d}{s} \vee 1\right)} \log(1/\varepsilon)\left(\log^2(1/\varepsilon) + \log D + F_n + \log(1/\delta)\right)\right).$$

The constant hidden in $O(\cdot)$ depends on $d, s, \frac{2d}{sp-d}, p, q, c_0, \tau$ and the surface area of $\mathcal{M}$.

Lemma 2 is proved in Section D.2. By Lemma 2, fix the network architecture $\mathcal{C}^{(F_n)}$, for $\varepsilon_1 \in (0,1)$, there exists a ConvResNet $\bar{f}_{\phi,n} \in \mathcal{C}^{(F_n)}$ such that $\|\bar{f}_{\phi,n} - f^*_{\phi,n}\|_{L^\infty} \le 4e^{F_n}\varepsilon_1$. In the following, we choose $\varepsilon_1 = n^{-\frac{2s}{2s+2(s\vee d)}}\log n$.

Next we check conditions (A2) and (A3) by estimating $\mathcal{E}_\phi(\bar{f}_{\phi,n}, f^*_\phi)$. Denote

$$A_n = \{\mathbf{x} \in \mathcal{M} : |f^*_\phi| \le F_n\},\ A_n^{\complement} = \{\mathbf{x} \in \mathcal{M} : |f^*_\phi| > F_n\}.$$

We have

$$\begin{aligned}
\mathcal{E}_\phi(\bar{f}_{\phi,n}, f^*_\phi) &= \int_{\mathcal{M}} \eta\left(\phi(\bar{f}_{\phi,n}) - \phi(f^*_\phi)\right) + (1-\eta)\left(\phi(-\bar{f}_{\phi,n}) - \phi(-f^*_\phi)\right)\mu(d\mathbf{x}) \\
&= \underbrace{\int_{A_n} \eta\left(\phi(\bar{f}_{\phi,n}) - \phi(f^*_{\phi,n})\right) + (1-\eta)\left(\phi(-\bar{f}_{\phi,n}) - \phi(-f^*_{\phi,n})\right)\mu(d\mathbf{x})}_{\mathrm{T}_1} \\
&\quad + \underbrace{\int_{A_n^{\complement}} \eta\left(\phi(\bar{f}_{\phi,n}) - \phi(f^*_\phi)\right) + (1-\eta)\left(\phi(-\bar{f}_{\phi,n}) - \phi(-f^*_\phi)\right)\mu(d\mathbf{x})}_{\mathrm{T}_2}, \qquad (34)
\end{aligned}$$

where we used $f^*_\phi = f^*_{\phi,n}$ on A_n. In (34), T_1 represents the approximation error of $\bar{f}_{\phi,n}$, and T_2 is the truncation error. Since $\|\bar{f}_{\phi,n} - f^*_{\phi,n}\|_{L^\infty} \le 4e^{F_n}\varepsilon_1$,

$$\begin{aligned}
\mathrm{T}_1 &\le \int_{A_n} \eta|\phi(\bar{f}_{\phi,n}) - \phi(f^*_{\phi,n})| + (1-\eta)|\phi(-\bar{f}_{\phi,n}) - \phi(-f^*_{\phi,n})|\mu(d\mathbf{x}) \\
&\le \|\phi(\bar{f}_{\phi,n}) - \phi(f^*_{\phi,n})\|_{L^\infty} \le 4e^{F_n}\varepsilon_1. \qquad (35)
\end{aligned}$$

A bound of T_2 is provided by the following lemma (see a proof in Appendix D.3):

Lemma 3. *Assume Assumption 1 and 2. Assume* $0 < p, q \le \infty$, $0 < s < \infty$, $s \ge d/p + 1$, $\eta \in B^s_{p,q}(\mathcal{M})$ *with* $\|\eta\|_{B^s_{p,q}(\mathcal{M})} \le c_0$ *for some constant* c_0. *Let* T_2 *be defined as in (34). If* $4e^{F_n}\varepsilon_1 < 1$, *the following bound holds:*

$$\mathrm{T}_2 \le 8F_n e^{-F_n}. \qquad (36)$$

According to our choices of ε_1 and F_n, $4e^{F_n}\varepsilon_1 < 1$ is satisfied. Combining (35) and (36) gives

$$\mathcal{E}_\phi(\bar{f}_{\phi,n}, f^*_\phi) \le \mathrm{T}_1 + \mathrm{T}_2 \le 4e^{F_n}\varepsilon_1 + 8F_n e^{-F_n}.$$

Substituting $\varepsilon_1 = n^{-\frac{2s}{2s+2(s\vee d)}}\log n$, $F_n = \frac{s}{2s+2(s\vee d)}\log n$ gives

$$\mathcal{E}_\phi(\bar{f}_{\phi,n}, f^*_\phi) \le C_6 n^{-\frac{s}{2s+2(s\vee d)}}\log^2 n$$

and $\mathcal{C}^{(F_n)} = \mathcal{C}^{(n)}$, where $\mathcal{C}^{(n)}$ is defined in Theorem 2. Here C_6 is a constant depending on s and d. Thus (A2) and (A3) are satisfied with $a_n = n^{-\frac{s}{2s+2(s\vee d)}}\log^2 n$, $F_n = \frac{s}{2s+2(s\vee d)}\log n$.

Verification of (A5). For (A5), we only need to check that $\log\mathcal{N}(\delta_n, \mathcal{C}^{(n)}, \|\cdot\|_{L^\infty}) \le C_3 n F_n^{-1}\delta_n$ for some constant C_3 . According to Lemma 2 with our choices of ε_1 and F_n, we have

$$\log\mathcal{N}(\delta, \mathcal{C}^{(n)}, \|\cdot\|_{L^\infty}) = O\left(D^3 n^{\frac{2(s\vee d)}{2s+2(s\vee d)}}\log(n)\left(\log^2 n + \log n + \log D + \log(1/\delta)\right)\right).$$

Substituting our choice $\delta_n = n^{-\frac{s}{2s+2(s\vee d)}}\log^4 n$ gives rise to

$$\log\mathcal{N}(\delta_n, \mathcal{C}^{(n)}, \|\cdot\|_{L^\infty}) = O\left((D^3\log D) n^{\frac{2(s\vee d)}{2s+2(s\vee d)}}\log^2 n\right) \le C_3 n F_n^{-1} e^{-F_n}\delta_n \qquad (37)$$

for some C_3 depending on $d, D^3\log D, s, \frac{d}{sp-d}, p, q, c_0, \tau$ and the surface area of $\mathcal{M}$. Therefore (A5) is satisfied.

Estimate the excess risk. Since (A1)-(A5) are satisfied, Proposition 2 gives

$$\mathbb{P}\left(\mathcal{E}_\phi(\widehat{f}_{\phi,n}, f^*_\phi) \ge \epsilon_n\right) \le C_5\exp\left(-C_4\frac{2s+2(s\vee d)}{s}\frac{n^{\frac{s+2(s\vee d)}{2s+2(s\vee d)}}\epsilon_n^2}{\log n}\right) \qquad (38)$$

with $\epsilon_n^2 \asymp \max(a_n, \delta_n) = C_7 n^{-\frac{s}{2s+2(s\vee d)}} \log^4 n$ and $\widehat{f}_{\phi,n} \in \mathcal{F}_n$ being the minimizer of the empirical risk in (11). Here C_7 is a constant depending on $d, D, \log D, s, \frac{d}{sp-d}, p, q, c_0, \tau$ and the surface area of $\mathcal{M}$.

Note that (A5) is also satisfied for any $\delta_n \geq C_7 n^{-\frac{s}{2s+2(s\vee d)}} \log^4 n$. Thus

$$\mathbb{P}\left(\mathcal{E}_\phi(\widehat{f}_{\phi,n}, f_\phi^*) \geq t\right) \leq C_5 \exp\left(-C_4 \frac{2s+2(s\vee d)}{s} \frac{n^{\frac{s+2(s\vee d)}{2s+2(s\vee d)}} t}{\log n}\right) \tag{39}$$

for any $t \geq C_7 n^{-\frac{s}{2s+2(s\vee d)}} \log^4 n$. Integrating (39), we estimate the expected excess risk as

$$\begin{aligned}
\mathbb{E}(\mathcal{E}_\phi(\widehat{f}_{\phi,n}, f_\phi^*)) &= \int_{\mathcal{M}} \mathcal{E}_\phi(\widehat{f}_{\phi,n}, f_\phi^*) \mu(d\mathbf{x}) \\
&\leq C_7 \mathbb{P}\left(\mathcal{E}_\phi(\widehat{f}_{\phi,n}, f_\phi^*) \leq C_7 n^{-\frac{s}{2s+2(s\vee d)}} \log^4 n\right) n^{-\frac{s}{2s+2(s\vee d)}} \log^4 n \\
&\quad + C_5 \int_{C_7 n^{-\frac{s}{2s+2(s\vee d)}} \log^4 n}^{\infty} \exp\left(-C_4 \frac{2s+2(s\vee d)}{s} \frac{n^{\frac{s+2(s\vee d)}{2s+2(s\vee d)}} t}{\log n}\right) dt \\
&\leq C_8 n^{-\frac{s}{2s+2(s\vee d)}} \log^4 n
\end{aligned} \tag{40}$$

for some constants C_7, C_8 depending on $d, D, \log D, s, \frac{2d}{sp-d}, p, q, c_0, \tau$ and the surface area of $\mathcal{M}$. □

B. Convolutional neural networks and muli-layer perceptrons

The proofs of the main results utilize properties convolutional neural networks (CNN) and multi-layer perceptrons (MLP) with the ReLU activation. We consider CNNs in the form of

$$f(\mathbf{x}) = W \cdot \mathrm{Conv}_{\mathcal{W},\mathcal{B}}(\mathbf{x}) \tag{41}$$

where $\mathrm{Conv}_{\mathcal{W},\mathcal{B}}(Z)$ is defined in (3), W is the weight matrix of the fully connected layer, $\mathcal{W}, \mathcal{B}$ are sets of filters and biases, respectively. We define the class of CNNs as

$$\begin{aligned}
\mathcal{F}^{\mathrm{CNN}}(L, J, K, \kappa_1, \kappa_2) = \big\{f \,|\, & f(\mathbf{x}) \text{ in the form (41) with } L \text{ layers.} \\
& \text{Each convolutional layer has filter size bounded by } K. \\
& \text{The number of channels of each layer is bounded by } J. \\
& \max_l \|\mathcal{W}^{(l)}\|_\infty \vee \|B^{(l)}\|_\infty \leq \kappa_1,\ \|W\|_\infty \leq \kappa_2\big\}.
\end{aligned} \tag{42}$$

For MLP, we consider the following form

$$f(\mathbf{x}) = W_L \cdot \mathrm{ReLU}(W_{L-1} \cdots \mathrm{ReLU}(W_1 \mathbf{x} + \mathbf{b}_1) \cdots + \mathbf{b}_{L-1}) + \mathbf{b}_L, \tag{43}$$

where $W_1, \ldots, W_L$ and $\mathbf{b}_1, \ldots, \mathbf{b}_L$ are weight matrices and bias vectors of proper sizes, respectively. The class of MLP is defined as

$$\begin{aligned}
\mathcal{F}^{\mathrm{MLP}}(L, J, \kappa) = \big\{f \,|\, & f(\mathbf{x}) \text{ in the form (43) with } L\text{-layers and width bounded by } J. \\
& \|W_i\|_{\infty,\infty} \leq \kappa, \|\mathbf{b}_i\|_\infty \leq \kappa \text{ for } i = 1, \ldots, L\big\}.
\end{aligned} \tag{44}$$

In some cases it is necessary to enforce the output of the MLP to be bounded. We define such a class as

$$\mathcal{F}^{\mathrm{MLP}}(L, J, \kappa, R) = \left\{f \,|\, f(\mathbf{x}) \in \mathcal{F}^{\mathrm{MLP}}(L, J, \kappa) \text{ and } \|f\|_\infty \leq R\right\}.$$

In some case we do not need the constraint on the output, we denote such MLP class as $\mathcal{F}^{\mathrm{MLP}}(L, J, \kappa)$.

C. Lemmas and proofs in Section 4

C.1. Lemma 4 and its proof

Lemma 4. *Define f_i, ϕ_i as in (17). We extend $f_i \circ \phi_i^{-1}$ by 0 on $[0,1]^d \backslash \phi_i(U_i)$ and denote the extended function by $f_i \circ \phi_i^{-1}|_{[0,1]^d}$. Under Assumption 3, we have $f_i \circ \phi_i^{-1}|_{[0,1]^d} \in B_{p,q}^s([0,1]^d)$ with*

$$\|f_i \circ \phi_i^{-1}\|_{B_{p,q}^s([0,1]^d)} < Cc_0$$

where C is a constant depending on s, p, q and d.

To prove Lemma 4, we first give an equivalent definition of Besov functions:

Definition 12. *Let Ω be a Lipschitz domain in $\mathbb{R}^d$. For $0 < p, q \leq \infty$ and $s > 0$, $\mathcal{B}^s_{p,q}(\Omega)$ is the set of functions*

$$\mathcal{B}^s_{p,q}(\Omega) = \{f : \Omega \to \mathbb{R} | \exists g \in B^s_{p,q}(\mathbb{R}^d) \text{ with } g|_\Omega = f\},$$

where $g|_\Omega$ denotes the restriction of g on Ω. The norm is defined as $\|f\|_{\mathcal{B}^s_{p,q}(\Omega)} = \inf_g \|g\|_{\mathcal{B}^s_{p,q}(\mathbb{R}^d)}$.

According to Dispa (2003, Theorem 3.18), for any Lipschitz domain $\Omega \subset \mathbb{R}^d$, the norm $\|\cdot\|_{B^s_{p,q}(\Omega)}$ in Definition 8 is equivalent to $\|\cdot\|_{\mathcal{B}^s_{p,q}(\Omega)}$ in Definition 12. Thus $B^s_{p,q}(\Omega) = \mathcal{B}^s_{p,q}(\Omega)$.

Proof of Lemma 4. Since $f \in B^s_{p,q}(\mathcal{M})$, according to Definition 9, $f_i \circ \phi_i^{-1} \in B^s_{p,q}(\mathbb{R}^d)$ in the sense of extending $f_i \circ \phi_i^{-1}$ by zero on $\mathbb{R}^d \backslash \phi_i(U_i)$, see Triebel (1983, Section 3.2.3) for details. From Assumption 3, $\|f_i \circ \phi_i^{-1}\|_{B^s_{p,q}(\mathbb{R}^d)} < c_0$. We next restrict $f_i \circ \phi_i^{-1}$ on $[0,1]^d$ and denote the restriction by $f_i \circ \phi_i^{-1}|_{[0,1]^d}$.

Using Definition 12 and Assumption , we have

$$\|f_i \circ \phi_i^{-1}|_{[0,1]^d}\|_{\mathcal{B}^s_{p,q}([0,1]^d))} \leq \|f_i \circ \phi_i^{-1}\|_{B^s_{p,q}(\mathbb{R}^d)} < c_0,$$

and we next show $f_i \circ \phi_i^{-1}|_{[0,1]^d} \in U(B^s_{p,q}([0,1]^d))$. Since $[0,1]^d$ is a Lipschitz domain, Dispa (2003, Theorem 3.18) implies $B^s_{p,q}([0,1]^d) = \mathcal{B}^s_{p,q}([0,1]^d)$. Therefore, there exists a constant C depending on s, p, q and d such that

$$\|f_i \circ \phi_i^{-1}|_{[0,1]^d}\|_{B^s_{p,q}([0,1]^d)} \leq \|f_i \circ \phi_i^{-1}\|_{\mathcal{B}^s_{p,q}(\mathbb{R}^d)} \leq Cc_0.$$

□

C.2. Cardinal B-splines

We give a brief introduction of cardinal B-splines.

Definition 13 (Cardinal B-spline)**.** *Let $\psi(x) = \mathbb{1}_{[0,1]}(x)$ be the indicator function of $[0,1]$. 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 (Mhaskar & Micchelli, 1992)

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

For any $k, j \in \mathbb{N}$, let $M_{k,j,m}(x) = \psi_m(2^k x - j)$, 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 $M^d_{\mathbf{k},\mathbf{j},m}(\mathbf{x}) = \prod_{i=1}^d \psi_m(2^{k_i} x_i - j_i)$. When $k_1 = \ldots = k_d = k \in \mathbb{N}$, we denote $M^d_{k,\mathbf{j},m}(\mathbf{x}) = \prod_{i=1}^d \psi_m(2^k x_i - j_i)$.

C.3. Lemma 5

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 $\{\alpha_{k,j}\}$ for $k \in \mathbb{N}, \mathbf{j} \in J(k)$ be

$$\|\{\alpha_{k,\mathbf{j}}\}\|_{b^s_{p,q}} = \left(\sum_{k \in \mathbb{N}} \left[2^{k(s-d/p)} \left(\sum_{\mathbf{j} \in J(k)} |\alpha_{k,\mathbf{j}}|^p \right)^{1/p} \right]^q \right)^{1/q}. \tag{45}$$

The following lemma, resulted from DeVore & Popov (1988); Dũng (2011), gives an error bound for the approximation of functions in $B^s_{p,q}([0,1]^d)$ by cardinal B-splines.

Lemma 5 (Lemma 2 in Suzuki (2019); DeVore & Popov (1988); Dũng (2011))**.** *Assume that $0 < p, q, r \leq \infty$ and $0 < s < \infty$ satisfying $s > d(1/p - 1/r)_+$. Let $m \in \mathbb{N}$ be the order of the Cardinal B-spline basis such that $0 < s < \min(m, m-1+1/p)$. For any $f \in B^s_{p,q}([0,1]^d)$, there exists f_N satisfying*

$$\|f - f_N\|_{L^r([0,1]^d)} \leq CN^{-s/d}\|f\|_{B^s_{p,q}([0,1]^d)}$$

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

$$f_N(\mathbf{x}) = \sum_{k=0}^{H}\sum_{\mathbf{j}\in J(k)} \alpha_{k,\mathbf{j}} M^d_{k,\mathbf{j},m}(\mathbf{x}) + \sum_{k=K+1}^{H^*}\sum_{i=1}^{n_k} \alpha_{k,\mathbf{j}_i} M^d_{k,\mathbf{j}_i,m}(\mathbf{x}), \tag{46}$$

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 $\nu = (s-u)/(2u)$. 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 $\{\alpha_{k,\mathbf{j}}\}$ such that

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

for some constant C_1.

Lemma 6. *Let $\alpha^{(i)}_{k,\mathbf{j}}$ be defined as in (19). Under Assumption 3, for any $i, k, \mathbf{j}$, we have*

$$|\alpha_{k,\mathbf{j}}| \leq Cc_0 N^{(\log 2)(\nu^{-1}+c_1 d^{-1})(d/p-s)_+} \tag{47}$$

for some C depending on $(d/p - s)_+\nu^{-1}$, s and d, where ν, c_1 are defined in Lemma 5.

Proof of Lemma 6. According to (45) and Lemma 5,

$$2^{k(s-d/p)}|\alpha_{k,\mathbf{j}}| \leq \|\{\alpha_{k,\mathbf{j}}\}\|_{b^s_{p,q}} \leq C_1\|f_i \circ \phi_i^{-1}\|_{B^s_{p,q}([0,1]^d)}.$$

Using Lemma 4 and since $k \leq H^*$ (from Lemma 5), we have

$$|\alpha^{(i)}_{k,\mathbf{j}}| \leq C_1 2^{k(d/p-s)_+}\|f_i \circ \phi_i^{-1}\|_{B^s_{p,q}([0,1]^d)} \leq C_2 2^{H^*(d/p-s)_+} c_1 c_0 \tag{48}$$

for some C_2 depending on s and d. From the expression of H^*, we can compute

$$2^{H^*} \leq C_3 N^{(\log 2)(\nu^{-1}+c_1 d^{-1})} \tag{49}$$

for some C_3 depending on $(d/p - s)_+\nu^{-1}$. Substituting (49) into (48) finishes the proof.

□

C.4. Lemma 7

Lemma 7 (Proposition 3 in (Yarotsky, 2017))**.** *For any $C > 0$ and $0 < \eta < 1$. If $|x| \leq C, |y| \leq C$, there is an MLP, denoted by $\widetilde{\times}(\cdot, \cdot)$, such that*

$$|\widetilde{\times}(x, y) - xy| < \eta, \ \widetilde{\times}(x, 0) = \widetilde{\times}(y, 0) = 0.$$

Such a network has $O\left(\log \frac{1}{\eta}\right)$ layers and parameters. The width of each layer is bounded by 6 and all parameters are bounded by C^2.

C.5. Lemma 8

The following lemma is a special case of Oono & Suzuki (2019, Theorem 1). It shows that each MLP can be realized by a CNN:

Lemma 8 (Theorem 1 in (Oono & Suzuki, 2019))**.** *Let D be the dimension of the input. Let L, J be positive integers and $\kappa > 0$. For any $2 \leq K' \leq D$, any MLP architectures $\mathcal{F}^{\mathrm{MLP}}(L, J, \kappa)$ can be realized by a CNN architecture $\mathcal{F}^{\mathrm{CNN}}(L', J', K', \kappa_1', \kappa_2')$ with*

$$L' = L + D, J' = 4J, \kappa_1' = \kappa_2' = \kappa.$$

Specifically, any $\bar{f}^{\mathrm{MLP}} \in \mathcal{F}^{\mathrm{MLP}}(L, J, \kappa)$ can be realized by a CNN $\bar{f}^{\mathrm{CNN}} \in \mathcal{F}^{\mathrm{CNN}}(L', J', K', \kappa_1', \kappa_2')$. Furthermore, the weight matrix in the fully connected layer of $\bar{f}^{\mathrm{CNN}}$ has nonzero entries only in the first row.

C.6. Lemma 9 and its proof

Lemma 9. *Let d_i^2 and $\mathbb{1}_{[0,\omega^2]}$ be defined as in (22). For any $\theta \in (0,1)$ and $\Delta \geq 8B^2D\theta$, there exists a CNN $\widetilde{d}_i^2$ approximating d_i^2 such that*

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

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

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

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

As a result, for any $\mathbf{x} \in \mathcal{M}$, $\widetilde{\mathbb{1}}_\Delta \circ \widetilde{d}_i^2(\mathbf{x})$ gives an approximation of $\mathbb{1}_{U_i}$ satisfying

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

Proof. We first show the existence of $\widetilde{d}_i^2$. Here $d_i^2(\mathbf{x})$ is the sum of D univariate quadratic functions. Each quadratic function can be approximated by an multi-layer perceptron (MLP, see Appendix B for the definition) according to Lemma 7. Let $\mathring{h}(x)$ be an MLP approximation of x^2 for $x \in [0,1]$ with error θ, i.e., $\|\mathring{h}(x) - x^2\|_\infty \leq \theta$. We define

$$\mathring{d}_i^2(\mathbf{x}) = 4B^2 \sum_{j=1}^{D} \mathring{h}\left(\left|\frac{x_j - c_{i,j}}{2B}\right|\right)$$

as an approximation of $d_i^2(\mathbf{x})$, which gives rise to the approximation error $\|\mathring{d}_i^2 - d_i^2\|_\infty \leq 4B^2D\theta$. Such a MLP has $O(\log 1/\theta)$ layers, and width $6D$. All weight parameters are bounded by $4B^2$. According to Lemma 8, $\mathring{d}_i^2$ can be realized by a CNN, which is denoted by $\widetilde{d}_i^2$. Such a CNN has $O(\log 1/\theta)$ layers, $6D$ channels. All weight parameters are bounded by $4B^2$.

To show the existence of $\widetilde{\mathbb{1}}_\Delta$, we use the following function to approximate $\mathbb{1}_{[0,\omega^2]}$:

$$\widetilde{\mathbb{1}}_\Delta(a) = \begin{cases} 1, & \text{if } a \leq \omega^2 - \Delta + 4B^2D\theta, \\ 0, & \text{if } a \geq \omega^2 - 4B^2Dv, \\ -\frac{1}{\Delta - 8B^2D\theta}a + \frac{r^2 - 4B^2D\theta}{\Delta - 8B^2D\theta}, & \text{otherwise}. \end{cases}$$

We implement $\widetilde{\mathbb{1}}_\Delta(a)$ based on the basic step function defined as: $g(a) = 2\mathrm{ReLU}(a - 0.5(\omega^2 - 4B^2D\theta)) - 2\mathrm{ReLU}(a - \omega^2 + 4B^2D\theta)$. Define

$$\begin{aligned} g_k(a) &= \underbrace{g \circ \cdots \circ g}_{k}(a) \\ &= \begin{cases} 0, & \text{if } a \leq (1-2^{-k})(\omega^2 - 4B^2D\theta), \\ \omega^2 - 4B^2D\theta, & \text{if } a \geq \omega^2 - 4B^2D\theta, \\ 2^k(a - \omega^2 + 4B^2D\theta) + \omega^2 - 4B^2D\theta, & \text{otherwise}. \end{cases} \end{aligned}$$

We set $\widetilde{\mathbb{1}}_\Delta = 1 - (\omega^2 - 4B^2D\theta)^{-1}g_k$ which can be realized by a CNN (according to Lemma 8). Such a CNN has k layers, 2 channels. All weight parameters are bounded by $\max(2, |\omega^2 - 4B^2D\theta|)$. The number of compositions k is chosen to satisfy $(1-2^{-k})(\omega^2 - 4B^2D\theta) \geq \omega^2 - \Delta + 4B^2D\theta$ which gives $k = \lceil \log(\omega^2/\Delta) \rceil$.

□

C.7. Lemma 10 and its proof

Lemma 10 shows that each cardinal B-spline can be approximated by a CNN with arbitrary accuracy. This lemma is used to prove Proposition 3.

Lemma 10. *Let k be any number in $\mathbb{N}$ and $\mathbf{j}$ be any element in $\mathbb{N}^d$. There exists a constant C depending only on d and m such that, for and $\varepsilon \in (0,1)$ and $2 \leq K \leq d$, there exists a CNN $\widetilde{M}^d_{k,\mathbf{j},m} \in \mathcal{F}^{\mathrm{CNN}}(L,J,K,\kappa,\kappa)$ with $L = 3 + 2\lceil \log_2\left(\frac{3\vee m}{C\varepsilon}\right) + 5\rceil \lceil \log_2(d \vee m)\rceil + d$, $J = 24dm(m+2) + 8d$ and $\kappa = 2(m+1)^m \vee 2^k$ such that for any $k \in \mathbb{N}$ and $\mathbf{j} \in \mathbb{N}^d$,*

$$\|M^d_{k,\mathbf{j},m} - \widetilde{M}^d_{k,\mathbf{j},m}\|_{L^\infty([0,1]^d)} \leq \varepsilon,$$

and $\widetilde{M}^d_{k,\mathbf{j},m}(\mathbf{x}) = 0$ for all $\mathbf{x} \notin 2^{-k}[0, m+1]^d$.

The proof of Lemma 10 is based on the following lemma:

Lemma 11 (Lemma 1 in Suzuki (2019))**.** *Let k be any number in $\mathbb{N}$ and $\mathbf{j}$ be any element in $\mathbb{N}^d$. There exists a constant C depending only on d and m such that, for all $\varepsilon > 0$, there exists an MLP $\bar{M}^d_{k,\mathbf{j},m} \in \mathcal{F}^{\mathrm{MLP}}(L,J,\kappa,1)$ with $L = 3 + 2\lceil \log_2\left(\frac{3\vee m}{C\varepsilon}\right) + 5\rceil \lceil \log_2(d \vee m)\rceil$, $J = 6dm(m+2) + 2d$ and $\kappa = 2(m+1)^m \vee 2^k$ such that for any $k \in \mathbb{N}$ and $\mathbf{j} \in \mathbb{N}^d$,*

$$\|M^d_{k,\mathbf{j},m} - \bar{M}^d_{k,\mathbf{j},m}\|_{L^\infty([0,1]^d)} \leq \varepsilon,$$

and $\bar{M}^d_{k,\mathbf{j},m}(\mathbf{x}) = 0$ for all $\mathbf{x} \notin 2^{-k}[0, m+1]^d$.

Proof of Lemma 10. According to Lemma 11, there exists an MLP $\bar{M}^d_{k,\mathbf{j},m} \in \mathcal{F}^{\mathrm{MLP}}(L',J',\kappa',1)$ with $L' = 3 + 2\lceil \log_2\left(\frac{3\vee m}{C\varepsilon}\right) + 5\rceil \lceil \log_2(d \vee m)\rceil$, $J' = 6dm(m+2) + 2d$ and $\kappa' = 2(m+1)^m \vee 2^k$ such that

$$\|M^d_{k,\mathbf{j},m} - \bar{M}^d_{k,\mathbf{j},m}\|_{L^\infty([0,1]^d)} \leq \varepsilon,$$

and $\bar{M}^d_{k,\mathbf{j},m}(\mathbf{x}) = 0$ for all $\mathbf{x} \notin 2^{-k}[0, m+1]^d$.

Lemma 8 shows that such an MLP can be realized by a CNN $\widetilde{M}^d_{k,\mathbf{j},m} \in \mathcal{F}^{\mathrm{CNN}}(L,J,K,\kappa,\kappa)$. □

C.8. Proposition 3 and its proof

Proposition 3 shows that if N and ε_1 are properly chosen, $\sum_{j=1}^N \widetilde{f}^{\mathrm{CNN}}_{i,j}$ can approximate $f_i \circ \phi_i^{-1}$ with arbitrary accuracy.

Proposition 3. *Let $f_i \circ \phi_i^{-1}$ be defined as in (18). For any $\delta \in (0,1)$, set $N = C_1 \delta^{-d/s}$. Suppose Assumption 3. For any $2 \leq K \leq d$, there exists a set of CNNs $\left\{\widetilde{f}^{\mathrm{CNN}}_{i,j}\right\}_{j=1}^N$ such that*

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

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

$\widetilde{f}^{\mathrm{CNN}}_{i,j}$ is a CNN approximation of $\widetilde{f}_{i,j}$ (defined in (19)) and is in $\mathcal{F}^{\mathrm{CNN}}(L,J,K,\kappa,\kappa)$ with

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

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

Proof of Proposition 3. Based on the approximation (19), for each $\widetilde{f}_{i,j}$, we construct CNN $\widetilde{f}^{\mathrm{CNN}}_{i,j}$ to approximate it.

Note that $\widetilde{f}_{i,j} = \alpha^{(i)}_{k,\mathbf{j}} M^d_{k,\mathbf{j},m}$ with some coefficient $\alpha^{(i)}_{k,\mathbf{j}}$ and index $k, \mathbf{j}, m$ where $M^d_{k,\mathbf{j},m}$ is a d-dimensional cardinal B-spline. Lemma 10 shows that $M^d_{k,\mathbf{j},m}$ can be approximated by a CNN $\widetilde{M}^d_{k,\mathbf{j},m}$ with arbitrary accuracy. Therefor $\widetilde{f}_{i,j}$ can be approximated by a CNN $\widetilde{f}^{\mathrm{CNN}}_{i,j}$ with arbitrary accuracy. Assume $\|\widetilde{M}^d_{k,\mathbf{j},m} - M^d_{k,\mathbf{j},m}\|_{L^\infty} \leq \varepsilon_1$ for some $\varepsilon_1 \in (0,1)$. Then $\widetilde{f}^{\mathrm{CNN}}_{i,j} \in \mathcal{F}^{\mathrm{CNN}}(L,J,K,\kappa,\kappa)$ with

$$L = O\left(\log(1/\varepsilon_1)\right), J = 24dm(m+2) + 8d, \kappa = \max\left(|\alpha^{(i)}_{k,\mathbf{j}}|, 2^k\right) = O\left(N^{(\log 2)(\nu^{-1} + c_1 d^{-1})(1 \vee (d/p - s)_+)}\right), \tag{50}$$

where the value of κ comes from Lemma 6 and (49).

The rest proof follows that of Suzuki (2019, Proposition 1) in which we show that with properly chosen N and ε_1, $\sum_{j=1}^N \widetilde{f}^{\mathrm{CNN}}_{i,j}$ can approximate $f_i \circ \phi_i^{-1}$ with arbitrary accuracy.

We decompose the error as

$$\left\|\sum_{j=1}^{N} \widetilde{f}_{i,j}^{\mathrm{CNN}} - f_i \circ \phi_i^{-1}\right\|_{L^\infty} \leq \left\|\widetilde{f}_i - f_i \circ \phi_i^{-1}\right\|_{L^\infty} + \left\|\sum_{j=1}^{N} \widetilde{f}_{i,j}^{\mathrm{CNN}} - \widetilde{f}_i\right\|_{L^\infty}. \tag{51}$$

where $\widetilde{f}_i$ is defined in (19). We next derive an error bound for each term.

Let m be the order of the Cardianl B-spline basis. Set $m = \lceil s \rceil + 1$. According to (20) and Lemma 5,

$$\left\|\widetilde{f}_i - f_i \circ \phi_i^{-1}\right\|_{L^\infty} \leq C c_0 N^{-s/d}, \quad \|\{\alpha_{k,\mathbf{j}}^{(i)}\}\|_{b_{p,q}^s} \leq C_1 \|f\|_{B_{p,q}^s}, \tag{52}$$

for some constant C depending on s, p, q and d, some universal constant C_1 with $\{\mathbf{j}_i\}_{i=1}^{n_k} \subset J(k)$, $H = \lceil c_1 \log(N)/d \rceil$, $H^* = \lceil \nu^{-1} \log(\lambda N) \rceil + H + 1$, $n_k = \lceil \lambda N 2^{-\nu(k-H)} \rceil$ for $k = H+1, \ldots, H^*$, $u = d(1/p - 1/r)_+$, $\nu = (s-u)/(2u)$ and $\sum_{k=1}^{H} (2^k + m)^d + \sum_{k=H+1}^{H^*} n_k \leq N$. By setting $N = \left\lceil \left(\frac{\delta}{2Cc_0}\right)^{-d/s} \right\rceil$, we have $\|\widetilde{f}_i - f_i \circ \phi_i^{-1}\|_\infty \leq \delta/2$.

Next we consider the second term in (51). For any $\mathbf{x} \in [0,1]^d$, we have

$$\begin{aligned}
&\left|\sum_{j=1}^{N} \widetilde{f}_{i,j}^{\mathrm{CNN}}(\mathbf{x}) - \widetilde{f}_i(\mathbf{x})\right| \leq \sum_{(k,\mathbf{j}) \in \mathcal{S}_N} |\alpha_{k,\mathbf{j}}^{(i)}| |M_{k,\mathbf{j},m}^d(\mathbf{x}) - \widetilde{M}_{k,\mathbf{j},m}^d(\mathbf{x})| \\
&\leq \varepsilon_1 \sum_{(k,\mathbf{j}) \in \mathcal{S}_N} |\alpha_{k,\mathbf{j}}^{(i)}| \mathbf{1}_{M_{k,\mathbf{j},m}^d(\mathbf{x}) \neq 0} \leq \varepsilon_1 (m+1)^d (1 + H^*) 2^{H^*(d/p-s)_+} \|f\|_{B_{p,q}^s} \\
&\leq \varepsilon_1 (m+1)^d \left(1 + \log(\lambda N)\nu^{-1} + c_1 \log(N)/d + 3\right) \left(e^3 (\lambda N)^{\nu^{-1}} N^{c_1/d}\right)^{(\log 2)(d/p-s)_+} \|f\|_{B_{p,q}^s} \\
&\leq C_2 c_0 \log(N) N^{(\log 2)(\nu^{-1} + c_1 d^{-1})(d/p-s)_+} \varepsilon_1 \\
&\leq C_2 c_0 \log\left(\frac{2}{\varepsilon}\right) \left(\frac{\varepsilon}{2}\right)^{-(\log 2)(\nu^{-1} + c_1 d^{-1})\left(\frac{d^2}{sp} - d\right)_+} \varepsilon_1
\end{aligned}$$

with C_2 being some constant depending on m, d, s, p, q, ν^{-1}. In the second inequality, $\mathbf{1}_{M_{k,\mathbf{j},m}^d(\mathbf{x}) \neq 0} = 1$ if $M_{k,\mathbf{j},m}^d(\mathbf{x}) \neq 0$ and it equals to 0 otherwise. The third inequality follows from the fact that for each k, there are $(m+1)^d$ basis functions which are non-zero at $\mathbf{x}$ and $2^{k(s-d/p)} \left|\alpha_{k,\mathbf{j}}^{(i)}\right| \leq \left\|\{\alpha_{k,\mathbf{j}}^{(i)}\}\right\|_{b_{p,q}^s} \leq C_1 \|f\|_{B_{p,q}^s}$. In the fourth inequality we use $H^* = \left\lceil \log(\lambda N)\nu^{-1} \right\rceil + H + 1$ and $H = \lceil c_1 \log(N)/d \rceil$, and the last inequality follows from $N = \left\lceil \left(\frac{\varepsilon}{2C}\right)^{-d/s} \right\rceil$. Setting $\varepsilon_1 = \frac{1}{C_2 c_0 \log(2/\delta)} \left(\frac{\delta}{2}\right)^{\frac{1}{2} + (\log 2)(\nu^{-1} + c_1 d^{-1})\left(\frac{d^2}{sp} - d\right)_+}$ proves the error bound.

Under Assumption 3, $s \geq d/p + 1$. Therefore $\left(\frac{d^2}{sp} - d\right)_+ = 0$ and $\nu = \frac{sp-d}{2d}$. Substituting these expressions into (50) gives the network architectures. □

C.9. Lemma 12

Lemma 12 estimates the approximation error of $\bar{f}$.

Lemma 12. *Let η be the approximation error of the multiplication operator $\widetilde{\times}(\cdot,\cdot)$, δ be defined as in Proposition 3, Δ and θ be defined as in Lemma 9. Assume N is chosen according to Proposition 3. For any $i = 1, ..., C_{\mathcal{M}}$, we have $\|\mathring{f} - f^*\|_{L^\infty} \leq \sum_{i=1}^{C_{\mathcal{M}}} (A_{i,1} + A_{i,2} + A_{i,3})$ with*

$$\begin{aligned}
A_{i,1} &= \sum_{j=1}^{N} \left\|\widetilde{\times}(\widetilde{f}_{i,j}^{\mathrm{CNN}} \circ \phi_i^{\mathrm{CNN}}, \widetilde{\mathbb{1}}_\Delta \circ \widetilde{d}_i^2) - \widetilde{f}_{i,j}^{\mathrm{CNN}} \circ \phi_i^{\mathrm{CNN}} \times (\widetilde{\mathbb{1}}_\Delta \circ \widetilde{d}_i^2)\right\|_{L^\infty} \leq C\delta^{-d/s}\eta, \\
A_{i,2} &= \left\|\left(\sum_{j=1}^{N} \left(\widetilde{f}_{i,j}^{\mathrm{CNN}} \circ \phi_i^{\mathrm{CNN}}\right)\right) \times (\widetilde{\mathbb{1}}_\Delta \circ \widetilde{d}_i^2) - f_i \times (\widetilde{\mathbb{1}}_\Delta \circ \widetilde{d}_i^2)\right\|_{L^\infty} \leq \delta, \\
A_{i,3} &= \|f_i \times (\widetilde{\mathbb{1}}_\Delta \circ \widetilde{d}_i^2) - f_i \times \mathbb{1}_{U_i}\|_{L^\infty} \leq \frac{c(\pi+1)}{\omega(1-\omega/\tau)}\Delta
\end{aligned}$$

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

$$\delta = \frac{\varepsilon}{3C_{\mathcal{M}}},\ \eta = \frac{1}{C}\left(\frac{\varepsilon}{3C_{\mathcal{M}}}\right)^{\frac{d}{s}+1}, \Delta = \frac{\omega(1-\omega/\tau)\varepsilon}{3c(\pi+1)C_{\mathcal{M}}},\ \theta = \frac{\Delta}{16B^2D} \tag{53}$$

gives rise to

$$\|\mathring{f} - f^*\|_{L^\infty} \le \varepsilon.$$

The choice in (53) satisfies the condition $\Delta > 8B^2D\theta$ in Lemma 9.

Proof of Lemma 12. In the error decomposition, $A_{i,1}$ measures the error from $\widetilde{\times}$:

$$A_{i,1} = \sum_{j=1}^{N} \left\| \widetilde{\times}(\widetilde{f}_{i,j}^{\rm CNN} \circ \phi_i^{\rm CNN}, \widetilde{\mathbb{1}}_\Delta \circ \widetilde{d}_i^2) - \widetilde{f}_{i,j}^{\rm CNN} \circ \phi_i^{\rm CNN} \times (\widetilde{\mathbb{1}}_\Delta \circ \widetilde{d}_i^2) \right\|_{L^\infty} \le N\eta \le C\delta^{-d/s}\eta,$$

for some constant C depending on d, s, p, q.

$A_{i,2}$ measures the error from CNN approximation of Besov functions. According to Proposition 3, $A_{i,2} \le \delta$.

$A_{i,3}$ measures the error from CNN approximation of the chart determination function. The bound of $A_{i,3}$ can be derived using Chen et al. (2019a, Proof of Lemma 4.5) since $f_i \circ \phi_i^{-1}$ is a Lipschitz function and its domain is in $[0,1]^d$. □

C.10. CNN size quantification of $\mathring{f}_{i,j}$

Let $\mathring{f}_{i,j}$ be defined as in (28). Under the choices of $\delta, \eta, \Delta, \theta$ in Lemma 12, we quantify the size of each CNN in $\mathring{f}_{i,j}$ as follows:

- $\widetilde{d}_i^2$ has $O(\log(1/\varepsilon) + D + \log D)$ layers, $6D$ channels and all weights parameters are bounded by $4B^2$.
- $\widetilde{\mathbb{1}}_\Delta$ has $O(\log(1/\varepsilon))$ layers with 2 channels. All weights are bounded by $\max(2, \omega^2)$.
- $\widetilde{\times}$ has $O(\log 1/\varepsilon)$ layers with 6 channels. All weights are of $\max(c_0^2, 1)$.
- $\widetilde{f}_{i,j}^{\rm CNN}$ has $O(\log 1/\varepsilon)$ layers with $\lceil 24d(s+1)(s+3) + 8d \rceil$ channels. All weights are in the order of $O\left(\varepsilon^{-(\log 2)\frac{d}{s}(\frac{2d}{sp-d}+c_1 d^{-1})}\right)$ where c_1 is defined in Lemma 5.
- $\phi_i^{\rm CNN}$ has $2 + D$ layers and d channels. All weights are bounded by $2B$.

In the above network architectures, the constant hidden in $O(\cdot)$ depend on d, s, $\frac{2d}{sp-d}, p, q, c_0, \tau$ and the surface area of $\mathcal{M}$. In particular, the constant depends on $D \log D$ linearly.

C.11. Lemma 13 and its proof

Lemma 13 shows that the composition of two CNNs can be realized by another CNN. Lemma 13 is used to prove Lemma 17.

Lemma 13. *Let $\mathcal{F}_1^{\rm CNN}(L_1, J_1, K_1, \kappa_1, \kappa_1)$ be a CNN architecture from $\mathbb{R}^D \to \mathbb{R}$ and $\mathcal{F}_2^{\rm CNN}(L_2, J_2, K_2, \kappa_2, \kappa_2)$ be a CNN architecture from $\mathbb{R} \to \mathbb{R}$. Assume the weight matrix in the fully connected layer of $\mathcal{F}_1^{\rm CNN}(L_1, J_1, K_1, \kappa_1, \kappa_1)$ and $\mathcal{F}_2^{\rm CNN}(L_2, J_2, K_2, \kappa_2, \kappa_2)$ has nonzero entries only in the first row. Then there exists a CNN architecture $\mathcal{F}^{\rm CNN}(L, J, K, \kappa, \kappa)$ from $\mathbb{R}^D \to \mathbb{R}$ with*

$$L = L_1 + L_2,\ J = \max(J_1, J_2),\ K = \max(K_1, K_2), \kappa = \max(\kappa_1, \kappa_2)$$

such that for any $f_1 \in \mathcal{F}^{\rm CNN}(L_1, J_1, K_1, \kappa_1, \kappa_1)$ and $f_2 \in \mathcal{F}^{\rm CNN}(L_2, J_2, K_2, \kappa_2, \kappa_2)$, there exists $f \in \mathcal{F}^{\rm CNN}(L, J, K, \kappa, \kappa)$ such that $f(\mathbf{x}) = f_2 \circ f_1(\mathbf{x})$. Furthermore, the weight matrix in the fully connected layer of $\mathcal{F}^{\rm CNN}(L, J, K, \kappa, \kappa)$ has nonzero entries only in the first row.

In Lemma 13 and the following lemmas, the subscript of $\mathcal{F}^{\rm CNN}$ are used to distinguish different network architectures.

Proof of Lemma 13. Compared to a CNN, directly composing f_1 and f_2 gives a network with an additional intermediate fully connected layer. In our network construction, we will design two convolutaionl layers to replace and realize this fully connected layer.

Denote f_1 and f_2 by

$$f_1(\mathbf{x}) = W_1 \cdot \mathrm{Conv}_{\mathcal{W}_1, \mathcal{B}_1}(\mathbf{x}) \text{ and } f_2(\mathbf{x}) = W_2 \cdot \mathrm{Conv}_{\mathcal{W}_2, \mathcal{B}_2}(\mathbf{x}).$$

where $\mathcal{W}_1 = \left\{\mathcal{W}_1^{(l)}\right\}_{i=1}^{L_1}, \mathcal{B}_1 = \left\{B_1^{(l)}\right\}_{l=1}^{L_1}, \mathcal{W}_2 = \left\{\mathcal{W}_2^{(l)}\right\}_{i=1}^{L_2}, \mathcal{B}_2 = \left\{B_2^{(l)}\right\}_{l=1}^{L_2}$, are sets of filters and biases and $\mathrm{Conv}_{\mathcal{W}_1,\mathcal{B}_1}, \mathrm{Conv}_{\mathcal{W}_2,\mathcal{B}_2}$ are defined in (3). In the rest of this proof, we will choose proper weight parameters in $\mathcal{W}, \mathcal{B}$ and W such that $f(\mathbf{x}) \in \mathcal{F}^{\mathrm{CNN}}(L, J, K, \kappa, \kappa)$ is in the form of

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

and satisfies $f(\mathbf{x}) = f_2 \circ f_1(\mathbf{x})$.

For $1 \leq l \leq L_1 - 1$, we set $\mathcal{W}^{(l)} = \mathcal{W}_1^{(l)}, B^{(l)} = B_1^{(l)}$.

For $l = L_1$, to realize the fully connected layer of f_1 by a convolutional layer, we set

$$\mathcal{W}_{1,:,:}^{(L_1)} = (W_1)_{1,:}, \; \mathcal{W}_{2,:,:}^{(L_1)} = -(W_1)_{1,:}$$

and $B^{(L_1)} = \mathbf{0}$. Here $\mathcal{W}^{(L_1)} \in \mathbb{R}^{2\times 1\times M}$ is a size-one filter with two output channels, where M is the number of input channels of W_1. The output of the L_1-th layer of f has the form

$$\begin{bmatrix} (f_1(\mathbf{x}))_+ & (f_1(\mathbf{x}))_- \\ \star & \star \end{bmatrix}$$

where $\star$ denotes some elements that will not affect the result.

Since the input of f_2 is a real number, all filters of f_2 has size 1. The weight matrix in the fully connected layer and all biases only have one row. For the (L_1+1)-th layer, we set

$$\mathcal{W}_{i,:,:}^{(L_1+1)} = \begin{bmatrix} (\mathcal{W}_2^{(1)})_{i,:,:} & -(\mathcal{W}_2^{(1)})_{i,:,:} \end{bmatrix}, \; B^{L_1+1} = \begin{bmatrix} B_2^{(1))} \\ \mathbf{0} \end{bmatrix}$$

where i varies from 1 to the number of output channels of $\mathcal{W}_2^{(1)}$. Here $\mathcal{W}^{(L_1+1)}$ is a size-one filter whose number of output channels is the same as that of $\mathcal{W}_2^{(1)}$.

For $L_1 + 1 \leq l \leq L - 1$, we set

$$\mathcal{W}^{(l)} = \mathcal{W}_2^{(l-L_1)}, \; B^l = \begin{bmatrix} B_2^{(l-L_1))} \\ \mathbf{0} \end{bmatrix}, \quad \text{for } l = L_1 + 1, ..., L_1 + L_2 - 1.$$

For $l = L$, we set

$$W = \begin{bmatrix} W_2 \\ \mathbf{0} \end{bmatrix}.$$

With the above settings, the lemma is proved. □

C.12. Lemma 14 and its proof

Lemma 14 is used to prove Lemma 17.

Lemma 14. *Let $f_1 \in \mathcal{F}^{\mathrm{CNN}}(L_1, J_1, K_1, \kappa_1, \kappa_1)$ be a CNN from $\mathbb{R}^D \to \mathbb{R}$ and $f_2 \in \mathcal{F}^{\mathrm{CNN}}(L_2, J_2, K_2, \kappa_2, \kappa_2)$ be a CNN from $\mathbb{R}^D \to \mathbb{R}$. Assume the weight matrix in the fully connected layer of f_1 and f_2 have nonzero entries only in the first row. Then there exists a set of filters $\mathcal{W}$ and biases $\mathcal{B}$ such that*

$$\mathrm{Conv}_{\mathcal{W},\mathcal{B}}(\mathbf{x}) = \begin{bmatrix} (f_1(\mathbf{x}))_+ & (f_1(\mathbf{x}))_- & (f_2(\mathbf{x}))_+ & (f_2(\mathbf{x}))_- \\ \star & \star & \star & \star \end{bmatrix} \in \mathbb{R}^{D\times 4}.$$

Such a network has $\max(L_1, L_2)$ layers, each filter has size at most $\max(K_1, K_2)$ and at most $J_1 + J_2$ channels. All parameter are bounded by $\max(\kappa_1, \kappa_2)$.

Proof of Lemma 14. For simplicity, we assume all convolutional layers of f_1 have J_1 channels and all convolutional layers of f_2 have J_2 channels. If some filters in f_1 (or f_2) have channels less than J_1 (or J_2), we can add additional channels with zero filters and biases. Without loss of generality, we assume $L_1 > L_2$.

Denotef_1 and f_2 by

$$f_1(\mathbf{x}) = W_1 \cdot \mathrm{Conv}_{\mathcal{W}_1,\mathcal{B}_1}(\mathbf{x}) \text{ and } f_2(\mathbf{x}) = W_2 \cdot \mathrm{Conv}_{\mathcal{W}_2,\mathcal{B}_2}(\mathbf{x}),$$

where $\mathcal{W}_1 = \left\{\mathcal{W}_1^{(l)}\right\}_{i=1}^{L_1}, \mathcal{B}_1 = \left\{B_1^{(l)}\right\}_{l=1}^{L_1}, \mathcal{W}_2 = \left\{\mathcal{W}_2^{(l)}\right\}_{i=1}^{L_2}, \mathcal{B}_2 = \left\{B_2^{(l)}\right\}_{l=1}^{L_2}$, are sets of filters and biases and $\mathrm{Conv}_{\mathcal{W}_1,\mathcal{B}_1}, \mathrm{Conv}_{\mathcal{W}_2,\mathcal{B}_2}$ are defined in (3). In the rest of this proof, We will choose proper weight parameters in $\mathcal{W}, \mathcal{B}$ such that

$$\mathrm{Conv}_{\mathcal{W},\mathcal{B}}(\mathbf{x}) = \begin{bmatrix} (f_1(\mathbf{x}))_+ & (f_1(\mathbf{x}))_- & (f_2(\mathbf{x}))_+ & (f_2(\mathbf{x}))_- \\ \star & \star & \star & \star \end{bmatrix} \in \mathbb{R}^{D\times 4}.$$

For $1 \leq l \leq L_2 - 1$, we set

$$\begin{aligned}
\mathcal{W}^{(l)}_{i,:,:} &= \begin{bmatrix} (\mathcal{W}^{(l)}_1)_{i,:,:} & \mathbf{0} \end{bmatrix} \text{ for } i = 1, ..., J_1, \\
\mathcal{W}^{(l)}_{i,:,:} &= \begin{bmatrix} \mathbf{0} & (\mathcal{W}^{(l)}_2)_{i-J_1,:,:} \end{bmatrix} \text{ for } i = J_1 + 1, ..., J_1 + J_2, \\
B^{(l)} &= \begin{bmatrix} B^{(l)}_1 & B^{(l)}_2 \end{bmatrix}.
\end{aligned}$$

Each $\mathcal{W}^{(l)}$ is a filter with size $\max(K_1, K_2)$ and $J_1 + J_2$ output channels. When $K_1 \neq K_2$, we pad the smaller filter by zeros. For example when $K_1 < K_2$, we set

$$\mathcal{W}^{(l)}_{i,:,:} = \begin{bmatrix} (\mathcal{W}^{(l)}_1)_{i,:,:} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \text{ for } i = 1, ..., J_1,$$

such that $\mathcal{W}^{(l)}$ has size K_2 filters.

For the L_2-th layer, we set

$$\begin{aligned}
\mathcal{W}^{(L_2)}_{i,:,:} &= \begin{bmatrix} (\mathcal{W}^{(L_2)}_1)_{i,:,:} & \mathbf{0} \end{bmatrix} \text{ for } i = 1, ..., J_1, \\
\mathcal{W}^{(L_2)}_{J_1+1,:,:} &= \begin{bmatrix} \mathbf{0} & (W_2)_{1,:} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \ \mathcal{W}^{(L_2)}_{J_1+2,:,:} = \begin{bmatrix} \mathbf{0} & -(W_2)_{1,:} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \\
B^{(L_2)} &= \begin{bmatrix} B^{(L_2)}_1 & \mathbf{0} \end{bmatrix}.
\end{aligned}$$

$\mathcal{W}^{(L_2)}$ is a filter with size K_1 and $J_1 + 2$ output channels.

For $L_2 + 1 \leq l \leq L_1 - 1$, we set

$$\begin{aligned}
\mathcal{W}^{(l)}_{i,:,:} &= \begin{bmatrix} (\mathcal{W}^{(l)}_1)_{i,:,:} & \mathbf{0} \end{bmatrix} \text{ for } i = 1, ..., J_1, \\
\mathcal{W}^{(l)}_{J_1+1,:,:} &= \begin{bmatrix} \mathbf{0} & 1 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \ \mathcal{W}^{(l)}_{J_1+2,:,:} = \begin{bmatrix} \mathbf{0} & 0 & 1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \\
B^{(L_2)} &= \begin{bmatrix} B^{(L_2)}_1 & \mathbf{0} \end{bmatrix}.
\end{aligned}$$

For the L_1-th layer, we set

$$\mathcal{W}^{(L_1)}_{1,:,:} = \begin{bmatrix} (W_1)_{1,:} & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \ \mathcal{W}^{(L_1)}_{2,:,:} = \begin{bmatrix} -(W_1)_{1,:} & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \mathcal{W}^{(L_1)}_{3,:,:} = \begin{bmatrix} \mathbf{0} & 1 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \ \mathcal{W}^{(L_1)}_{4,:,:} = \begin{bmatrix} \mathbf{0} & 0 & 1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

and $B^{(L_1)} = \mathbf{0}$. □

C.13. Lemma 15 and its proof

Lemma 15 is used to prove Lemma 17.

Lemma 15. *Let M, N be positive integers. For any $\mathbf{x} = \begin{bmatrix} x_1 & \cdots & x_M \end{bmatrix}^\top \in \mathbb{R}^M$, define*

$$X = \begin{bmatrix} (x_1)_+ & (x_1)_- & \cdots & (x_M)_+ & (x_M)_- \\ \star & \star & \cdots & \star & \star \end{bmatrix} \in \mathbb{R}^{N \times (2M)}.$$

For any CNN architecture $\mathcal{F}^{\mathrm{CNN}}_1(L, J, K, \kappa, \kappa)$ from $\mathbb{R}^M \to \mathbb{R}$, there exists a CNN architecture $\mathcal{F}^{\mathrm{CNN}}(L, MJ, K, \kappa, \kappa)$ from $\mathbb{R}^{N \times (2M)} \to \mathbb{R}$ such that for any $f_1 \in \mathcal{F}^{\mathrm{CNN}}_1(L, J, K, \kappa, \kappa)$, there exists $f \in \mathcal{F}^{\mathrm{CNN}}(L, MJ, K, \kappa, \kappa)$ with $f(X) = f_1(\mathbf{x})$. Furthermore, the fully connected layer of $\mathcal{F}^{\mathrm{CNN}}(L, MJ, K, \kappa, \kappa)$ has nonzero entries only in the first row.

Proof of Lemma 15. Denote f_1 as

$$f_1 = W_1 \cdot \mathrm{Conv}_{\mathcal{W}_1, \mathcal{B}_1}(\mathbf{x})$$

where $\mathcal{W}_1 = \left\{ \mathcal{W}^{(l)}_1 \right\}_{i=1}^{L_1}, \mathcal{B}_1 = \left\{ B^{(l)}_1 \right\}_{l=1}^{L_1}$ are sets of filters and biases and $\mathrm{Conv}_{\mathcal{W}_1, \mathcal{B}_1}$ is defined in (3). For simplicity, we assume all convolutional layers of f_1 have J channels . If some filters in f have less than J channels, we can add additional channels with zero filters and biases.

We next choose proper weight parameters in $\mathcal{W}, \mathcal{B}$ and W such that

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

and $f(X) = f_1(\mathbf{x})$.

For the first layer, i.e., $l = 1$, we design $\mathcal{W}^{(1)}$ and $B^{(1)}$ since $\mathbf{x}$ is a vector in $\mathbb{R}^M$, the filter $\mathcal{W}_1^{(1)}$ has 1 input channel and J output channel. For $1 \leq i \leq J$ and $1 \leq j \leq M - K + 1$, we set

$$\mathcal{W}^{(1)}_{(i-1)M+j,:,:} = \begin{bmatrix} \mathbf{0} & (\mathcal{W}_1^{(1)})_{i,1,:} & -(\mathcal{W}_1^{(1)})_{i,1,:} & \cdots & (\mathcal{W}_1^{(1)})_{i,K,:} & -(\mathcal{W}_1^{(1)})_{i,K,:} & \widetilde{\mathbf{0}} \end{bmatrix}$$

where $\mathbf{0}$ is of size $1 \times (j-1)$, $\widetilde{\mathbf{0}}$ is a zero matrix of size $1 \times (M - j - K + 1)$.

For $1 \leq i \leq J$ and $M - K + 2 \leq j \leq M$, we set

$$\mathcal{W}^{(1)}_{(i-1)M+j,:,:} = \begin{bmatrix} \mathbf{0} & (\mathcal{W}_1^{(1)})_{i,1,:} & -(\mathcal{W}_1^{(1)})_{i,1,:} & \cdots & (\mathcal{W}_1^{(1)})_{i,K-j+1,:} & -(\mathcal{W}_1^{(1)})_{i,K-j+1,:} \end{bmatrix}$$

where $\mathbf{0}$ is of size $1 \times (j-1)$. The bias is set as

$$B^{(1)} = \begin{bmatrix} \left((B_1^{(1)})_{:,1}\right)^\top & \cdots & \left((B_1^{(1)})_{:,J}\right)^\top \\ \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}.$$

We next choose weight parameters for $2 \leq l \leq L_1 - 1$. For $1 \leq i \leq J$ and $1 \leq j \leq M - K + 1$, we set

$$\mathcal{W}^{(l)}_{(i-1)M+j,:,:} = \begin{bmatrix} \mathbf{0} & \left((\mathcal{W}_1^{(l)})_{i,:,1}\right)^\top & \widetilde{\mathbf{0}} & \cdots & \mathbf{0} & \left((\mathcal{W}_1^{(l)})_{i,:,J}\right)^\top & \widetilde{\mathbf{0}} \end{bmatrix}$$

where $\mathbf{0}$ is of size $1 \times (j-1)$, $\widetilde{\mathbf{0}}$ is a zero matrix of size $1 \times (M - j - K + 1)$.

For $1 \leq i \leq J$ and $M - K + 2 \leq j \leq M$, we set

$$\mathcal{W}^{(l)}_{(i-1)M+j,:,:} = \begin{bmatrix} \mathbf{0} & \left((\mathcal{W}_1^{(l)})_{i,1:K-j+1,1}\right)^\top & \cdots & \mathbf{0} & \left((\mathcal{W}_1^{(l)})_{i,1:K-j+1,J}\right)^\top \end{bmatrix}$$

where $\mathbf{0}$ is of size $1 \times (j-1)$. The bias is set as

$$B^{(l)} = \begin{bmatrix} \left((B_1^{(l)})_{:,1}\right)^\top & \cdots & \left((B_1^{(l)})_{:,J}\right)^\top \\ \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}.$$

For the fully connected layer, we set

$$W = \begin{bmatrix} ((W_1)_{:,1})^\top & \cdots & ((W_1)_{:,J})^\top \\ \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}.$$

With these choices, the lemma is proved. □

C.14. Lemma 16 and its proof

Lemma 16 shows that for any CNN, if we scale all weight parameters in convolutional layers by some factors and scale the weight parameters in the fully connected layer properly, the output will remain the same. Lemma 16 is used to prove Lemma 17.

Lemma 16. *Let $\alpha \geq 1$. For any $f \in \mathcal{F}^{\mathrm{CNN}}(L, J, K, \kappa_1, \kappa_2)$, there exists $\widetilde{f} \in \mathcal{F}^{\mathrm{CNN}}(L, J, K, \alpha^{-1}\kappa_1, \alpha^L \kappa_2)$ such that $\widetilde{f}(\mathbf{x}) = f(\mathbf{x})$.*

Proof of Lemma 16. This lemma is proved using the linear property of ReLU and convolution. Let f be any CNN in $\mathcal{F}^{\mathrm{CNN}}(L, J, K, \kappa_1, \kappa_2)$. Denote its architecture as

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

Define $\bar{W} = \alpha^L W$ and $\bar{\mathcal{W}}, \bar{\mathcal{B}}$ as

$$\bar{\mathcal{W}}^{(l)} = \alpha^{-1}\mathcal{W}^{(l)}, \bar{\mathcal{B}}^{(l)} = \alpha^{-l}\mathcal{B}^{(l)}$$

for any $l \in (1, L)$. Set

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

We have $\widetilde{f} \in \mathcal{F}^{\mathrm{CNN}}(L, J, K, \alpha^{-1}\kappa_1, \alpha^L\kappa_2)$ and $\widetilde{f}(\mathbf{x}) = f(\mathbf{x})$ since $\mathrm{ReLU}(c\mathbf{x}) = c\mathrm{ReLU}(\mathbf{x})$ for any $c > 0$. □

C.15. Lemma 17 and its proof

Lemma 17 shows that each $\mathring{f}_{i,j}$ defined in (27) can be realized by a CNN.

Lemma 17. *Let $\mathring{f}_{i,j}$ be defined as in (27). Assume each CNN in $\mathring{f}_{i,j}$ has architecture discussed in Appendix C.10. Then there exists a CNN $\bar{f}^{\mathrm{CNN}}_{i,j} \in \mathcal{F}^{\mathrm{CNN}}(L, J, K, \kappa_1, \kappa_2)$ with*

$$L = O(\log(1/\varepsilon) + D + \log D), J = \lceil 48d(s+1)(s+3) + 28d + 6D \rceil, \kappa_1 = 1, \log \kappa_2 = O\left(\log^2 \frac{1}{\varepsilon}\right)$$

such that $\bar{f}^{\mathrm{CNN}}_{i,j}(\mathbf{x}) = \mathring{f}_{i,j}(\mathbf{x})$ for any $\mathbf{x} \in \mathcal{M}$. The constants hidden in $O(\cdot)$ depend on $d, D, s, \frac{2d}{sp-d}, p, q, c_0, \tau$ and the surface area of $\mathcal{M}$.

Proof of Lemma 17. According to Lemma 13, there exists a CNN $g_{i,j}$ realizing $\widetilde{f}^{\mathrm{CNN}}_{i,j} \circ \phi_i^{\mathrm{CNN}}$ and a CNN $\widetilde{g}_i$ realizing $\widetilde{\mathbb{1}}_\Delta \circ \widetilde{d}_i^2$. Using Lemma 14, one can construct a CNN excluding the fully connected layer, denoted by $\bar{g}_{i,j}$, such that

$$\bar{g}_{i,j}(\mathbf{x}) = \begin{bmatrix} (g_{i,j}(\mathbf{x}))_+ & (g_{i,j}(\mathbf{x}))_- & (\widetilde{g}_i(\mathbf{x}))_+ & (\widetilde{g}_i(\mathbf{x}))_- \\ \star & \star & \star & \star \end{bmatrix}. \tag{54}$$

Here $\bar{g}_{i,j}$ has $\lceil 48d(s+1)(s+3) + 28 \rceil$ channels.

Since the input of $\widetilde{\times}$ is $\begin{bmatrix} g_{i,j} \\ \widetilde{g}_i \end{bmatrix}$, Lemma 15 shows that there exists a CNN $\mathring{g}^{\mathrm{CNN}}_{i,j}$ which takes (54) as the input and outputs $\widetilde{\times}(g_{i,j}, \widetilde{g}_i)$.

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

$$L = O\left(\log \frac{1}{\varepsilon} + D + \log D\right),\ J = \lceil 28d(s+1)(s+3) + 18d \rceil + 6D,\ \kappa = O\left(\varepsilon^{-(\log 2)\frac{d}{s}(\frac{2d}{sp-d} + c_1 d^{-1})}\right).$$

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

We next rescale all parameters in convolutional layers of $\breve{f}^{\mathrm{CNN}}_{i,j}$ to be no larger than 1. Using Lemma 16, we can realize $\breve{f}^{\mathrm{CNN}}_{i,j}$ by $\bar{f}^{\mathrm{CNN}}_{i,j} \in \mathcal{F}^{\mathrm{CNN}}(L, J, K, \alpha^{-1}\kappa, \alpha^L \kappa)$ for any $\alpha > 1$. Set $\alpha = C' \varepsilon^{-(\log 2)\frac{d}{s}(\frac{2d}{sp-d} + c_1 d^{-1})}(8KD)M^{\frac{1}{L}}$ where C' is a constant such that $\kappa \leq C' \varepsilon^{-(\log 2)\frac{d}{s}(\frac{2d}{sp-d} + c_1 d^{-1})}$. With this α, we have $\bar{f}^{\mathrm{CNN}}_{i,j} \in \mathcal{F}^{\mathrm{CNN}}(L, J, K, \kappa_1, \kappa_2)$ with

$$\begin{aligned} & L = O(\log(1/\varepsilon) + D + \log D),\ J = \lceil 28d(s+1)(s+3) + 18d \rceil + 6D, \\ & \kappa_1 = (8KD)^{-1} M^{-\frac{1}{L}} = O(1),\ \log \kappa_2 = O\left(\log^2 1/\varepsilon\right). \end{aligned}$$

□

C.16. Lemma 18 and its proof

Lemma 18 shows that the sum of a set of CNNs can be realized by a ConvResNet.

Lemma 18. *Let $\mathcal{F}^{\mathrm{CNN}}(L, J, K, \kappa_1, \kappa_2)$ be any CNN architecture from $\mathbb{R}^D$ to $\mathbb{R}$. Assume the weight matrix in the fully connected layer of $\mathcal{F}^{\mathrm{CNN}}(L, J, K, \kappa_1, \kappa_2)$ has nonzero entries only in the first row. Let M be a positive integer. There exists a ConvResNet architecture $\mathcal{C}(M, L, J, \kappa_1, \kappa_2(1 \vee \kappa_1^{-1}))$ such that for any $\{f_i(\mathbf{x})\}_{m=1}^M \subset \mathcal{F}^{\mathrm{CNN}}(L, J, K, \kappa_1, \kappa_2)$, there exists $\bar{f} \in \mathcal{C}(M, L, J, \kappa_1, \kappa_2(1 \vee \kappa_1^{-1}))$ with*

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

Proof of Lemma 18. Denote the architecture of f_m by

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

with $\mathcal{W}_m = \left\{\mathcal{W}_m^{(l)}\right\}_{l=1}^L, \mathcal{B}_m = \left\{B_m^{(l)}\right\}_{l=1}^L$. In $\bar{f}$, denote the weight matrix and bias in the fully connected layer by $\bar{W}, \bar{b}$ and the set of filters and biases in the m-th block by $\bar{\mathcal{W}}_m$ and $\bar{\mathcal{B}}_m$, respectively. The padding layer P in $\bar{f}$ pads the input from $\mathbb{R}^D$ to $\mathbb{R}^{D \times 3}$ by zeros. Here each column denotes a channel.

We first show that for each m, there exists a subnetowrk $\mathrm{Conv}_{\bar{\mathcal{W}}_m,\bar{\mathcal{B}}_m} : \mathbb{R}^{D\times 3} \to \mathbb{R}^{D\times 3}$ such that for any $Z \in \mathbb{R}^{D\times 3}$ in the form of

$$Z = \begin{bmatrix}\mathbf{x} & \star & \star\end{bmatrix}, \tag{55}$$

we have

$$\mathrm{Conv}_{\bar{\mathcal{W}}_m,\bar{\mathcal{B}}_m}(Z) = \begin{bmatrix} 0 & \frac{\kappa_1}{\kappa_2}(f_m)_+ & \frac{\kappa_1}{\kappa_2}(f_m)_- \\ \mathbf{0} & \star & \star \end{bmatrix} \tag{56}$$

where $\star$ denotes some entries that we do not care.

For any m, the first layer of $\widetilde{f}_m$ takes input in $\mathbb{R}^D$. As a result, the filters in $\mathcal{W}_m^{(1)}$ are in $\mathbb{R}^D$. We pad these filters by zeros to get filters in $\mathbb{R}^{D\times 3}$ and construct $\bar{\mathcal{W}}_m^{(1)}$ as

$$(\bar{\mathcal{W}}_m^{(1)})_{j,:,:} = \begin{bmatrix}(\mathcal{W}_m^{(1)})_{j,:,:} & \mathbf{0} & \mathbf{0}\end{bmatrix}.$$

For any Z in the form of (55), we have $\bar{\mathcal{W}}_m^{(1)} * Z = \mathcal{W}_m^{(1)} * \mathbf{x}$. For the filters in the following layers and all biases, we simply set

$$\bar{\mathcal{W}}_m^{(l)} = \mathcal{W}_m^{(1)} \qquad \text{for } l = 2, \dots, L-1,$$
$$\bar{\mathcal{B}}_m^{(l)} = \mathcal{B}_m^{(1)} \qquad \text{for } l = 1, \dots, L-1.$$

In $\mathrm{Conv}_{\bar{\mathcal{W}}_m,\bar{\mathcal{B}}_m}$, another convolutional layer is constructed to realize the fully connected layer in f_m. According to our assumption, only the first row of W_m has nonzero entries. We set $\bar{\mathcal{B}}_m^{(L)} = \mathbf{0}$ and $\bar{\mathcal{W}}_m^L$ as size one filters with three output channels in the form of

$$(\bar{\mathcal{W}}_m^L)_{1,:,:} = \mathbf{0},\ (\bar{\mathcal{W}}_m^L)_{2,:,:} = \frac{\kappa_1}{\kappa_2}(W_m)_{1,:},\ (\bar{\mathcal{W}}_m^L)_{3,:,:} = -\frac{\kappa_1}{\kappa_2}(W_m)_{1,:}.$$

Under such choices, (56) is proved and all parameters in $\bar{\mathcal{W}}_m, \bar{\mathcal{B}}_m$ are bounded by κ_1.

By composing all residual blocks, one has

$$(\mathrm{Conv}_{\bar{\mathcal{W}}_M,\bar{\mathcal{B}}_M} + \mathrm{id}) \circ \cdots \circ (\mathrm{Conv}_{\bar{\mathcal{W}}_1,\bar{\mathcal{B}}_1} + \mathrm{id}) \circ P(\mathbf{x}) = \begin{bmatrix} \mathbf{x} & \frac{\kappa_1}{\kappa_2}\sum_{m=1}^M (f_m)_+ & \frac{\kappa_1}{\kappa_2}\sum_{m=1}^M (f_m)_+ \\ & \star & \star \\ & \vdots & \vdots \end{bmatrix}.$$

The fully connect layer is set as

$$\bar{W} = \begin{bmatrix} 0 & \frac{\kappa_2}{\kappa_1} & -\frac{\kappa_2}{\kappa_1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},\ \bar{b} = 0.$$

The weights in the fully connected layer are bounded by $\kappa_2(1 \vee \kappa_1^{-1})$.

Such a ConvResNet gives

$$\bar{f}(\mathbf{x}) = \sum_{m=1}^M (f_m(\mathbf{x}))_+ - \left(\sum_{m=1}^M (f_m(\mathbf{x}))_-\right) = \sum_{m=1}^M f_m(\mathbf{x}).$$

□

D. Proof of Lemmas and Propositions in Section A

D.1. Proof of Proposition 2

The proof of Proposition 2 relies on the following large-deviation inequality:

Lemma 19 (Theorem 3 in Shen & Wong (1994))**.** *Let $\{\mathbf{x}_i\}_{i=1}^n$ be i.i.d. samples from some probability distribution. Let $\mathcal{F}$ be a class of functions whose magnitude is bounded by F. Let $v \geq \sup_{f\in\mathcal{F}} \mathrm{Var}(f(\mathbf{x})), v > 0$ be a constant. Assume there are constants $M > 0, 0 < \lambda < 1$ such that*

(B1) $\mathcal{H}_B(v^{1/2}, \mathcal{F}, \|\cdot\|_{L^2}) \leq \lambda n M^2/(8(4v + MF/3))$,

(B2) $M \leq \lambda v/(4F),\ v^{1/2} \leq F$,

(B3) if $\lambda M/8 \leq v^{1/2}$, then

$$M^{-1}\int_{\lambda M/32}^{v^{1/2}} \mathcal{H}_B(u, \mathcal{F}, \|\cdot\|_{L^2})^{1/2} du \leq \frac{n^{1/2}\lambda^{3/2}}{2^{10}}.$$

Then

$$\mathbb{P}\left(\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n (f(\mathbf{x}_i) - \mathbb{E}[f(\mathbf{x}_i)]) \geq M\right) \leq 3\exp\left(-(1-\lambda)\frac{nM^2}{2(4v + MF/3)}\right).$$

Proof of Proposition 2. Let C_1, C_2, C_3 be constants defined in Proposition 2. Set $\epsilon_n^2 = \max\{2a_n, 2^7\delta_n/C_1\}$. For each $\mathcal{F}_n$ defined in (A2), define

$$E_n(f) = \frac{1}{n}\sum_{i=1}^n \left[\phi(y_i f_n(\mathbf{x}_i)) - \phi(y_i f(\mathbf{x}_i)) - \mathbb{E}\left[\phi(y_i f_n(\mathbf{x}_i)) - \phi(y_i f(\mathbf{x}_i))\right]\right]. \tag{57}$$

Note that $\mathcal{E}_\phi(f_n, f_\phi^*) \leq a_n$ by (A2). Since $\widehat{f}_{\phi,n}$ is the minimizer of $\mathcal{E}_\phi(f)$, we have

$$\mathbb{P}\left(\mathcal{E}_\phi(\widehat{f}_{\phi,n}, f_\phi^*) \geq \epsilon_n^2\right) \leq \mathbb{P}\left(\left[\sup_{f\in\mathcal{F}_n:\mathcal{E}_\phi(f,f_\phi^*)\geq\epsilon_n^2} \frac{1}{n}\sum_{i=1}^n \left[\phi(y_i f_n(\mathbf{x}_i)) - \phi(y_i f(\mathbf{x}_i))\right]\right] \geq 0\right).$$

We decompose the set $\left\{f \in \mathcal{F}_n : \mathcal{E}_\phi(f, f_\phi^*) \geq \epsilon_n^2\right\}$ into disjoint subsets $\{\mathcal{F}_{n,i}\}$ for $i = 1, ..., i_n$ in the form of

$$\mathcal{F}_{n,i} = \left\{f \in \mathcal{F}_n : 2^{i-1}\epsilon_n^2 \leq \mathcal{E}_\phi(f, f_\phi^*) < 2^i\epsilon_n^2\right\}.$$

Note that $\|f\|_{L^\infty} \leq F_n$ and $\mathcal{E}_\phi(f_\phi^*) \leq \mathcal{E}_\phi(f)$ for all $f \in \mathcal{F}_n$. Therefore, for any $f \in \mathcal{F}_n$, we have

$$\mathcal{E}_\phi(f, f_\phi^*) \leq \mathcal{E}_\phi(f) = \mathbb{E}[\phi(yf(\mathbf{x}))] \leq C_1\mathbb{E}[|f(\mathbf{x})|] \leq C_1F_n,$$

which implies $\mathcal{F}_{n,i}$ is an empty set for $2^{i-1}\epsilon_n^2 > C_1F_n$. We set $i_n = \inf\{i \in \mathbb{N} : 2^{i-1}\epsilon_n^2 > C_1F_n\}$. Then we have $\left\{f \in \mathcal{F}_n : \mathcal{E}_\phi(f, f_\phi^*) \geq \epsilon_n^2\right\} = \bigcup_{i=1}^{i_n} \mathcal{F}_{n,i}$. Since $\mathcal{E}_\phi(f_n, f_\phi^*) \leq a_n \leq \epsilon_n^2/2$, we have

$$\inf_{f\in\mathcal{F}_{n,i}} \mathbb{E}\left[\phi(yf(\mathbf{x})) - \phi(yf_n(\mathbf{x}))\right] = \inf_{f\in\mathcal{F}_{n,i}} \left[\mathcal{E}_\phi(f, f_\phi^*) - \mathcal{E}_\phi(f_n, f_\phi^*)\right] \geq 2^{i-2}\epsilon_n^2.$$

Denote $M_{n,i} = 2^{i-2}\epsilon_n^2$. We have

$$\begin{aligned}\mathbb{P}\left(\mathcal{E}_\phi(\widehat{f}_{\phi,n}, f_\phi^*) \geq \epsilon_n^2\right) &\leq \sum_{i=1}^{i_n} \mathbb{P}\left(\sup_{f\in\mathcal{F}_{n,i}} \mathbb{E}_n(f) \geq \mathbb{E}\left[\phi(yf(\mathbf{x})) - \phi(yf_n(\mathbf{x}))\right]\right)\\ &\leq \sum_{i=1}^{i_n} \mathbb{P}\left(\sup_{f\in\mathcal{F}_{n,i}} \mathbb{E}_n(f) \geq M_{n,i}\right),\end{aligned}$$

where $\mathbb{E}_n(f)$ is defined in (57). Then we will bound each summand on the right-hand side using Lemma 19. First, using (A4), we have

$$\begin{aligned}&\sup_{f\in\mathcal{F}_{n,i}} \mathbb{E}\left[\left(\phi(yf(\mathbf{x})) - \phi(yf_n(\mathbf{x}))\right)^2\right]\\ \leq& 2\sup_{f\in\mathcal{F}_{n,i}} \mathbb{E}\left[\left(\phi(yf(\mathbf{x})) - \phi(yf_\phi^*(\mathbf{x}))\right)^2 + \left(\phi(yf_n(\mathbf{x})) - \phi(yf_\phi^*(\mathbf{x}))\right)^2\right]\\ \leq& 2C_2e^{F_n}F_n^{2-v}\left(\sup_{f\in\mathcal{F}_{n,i}} \mathcal{E}_\phi(f, f_\phi^*)^\nu + \mathcal{E}_\phi(f_n, f_\phi^*)^\nu\right)\\ \leq& 2C_2e^{F_n}F_n^{2-\nu}\left(\left(2^i\epsilon_n^2\right)^\nu + \left(\epsilon_n^2/2\right)^\nu\right) \leq 4^{\nu+1}C_2e^{F_n}F_n^{2-\nu}\left(\left(2^{i-2}\epsilon_n^2\right)^\nu\right)\\ =& 4^{\nu+1}C_2e^{F_n}F_n^{2-\nu}M_{n,i}^\nu.\end{aligned}$$

Define $\mathcal{G}_{n,i} = \{g = \phi(yf_n(\mathbf{x})) - \phi(yf(\mathbf{x})) : f \in \mathcal{F}_{n,i}\}$. For any $g \in \mathcal{G}_{n,i}$, $\mathrm{Var}(g) \leq 4^{\nu+1}C_2e^{F_n}F_n^{2-\nu}M_{n,i}^\nu$. Since $f_n, f \in \mathcal{F}_n$, we have $\|g\|_{L^\infty} \leq C_1|f_n - f| \leq 2C_1F_n$.

To apply Lemma 19 on $\mathcal{G}_{n,i}$, we set $\lambda = 1/2,\ F = D_1F_n,\ M = M_{n,i}$ and $v = v_{n,i} = D_2F_n^{2-\nu}M_{n,i}^\nu$ with

$$D_1 = \frac{1}{8(2C_1)^{1-\nu}}D_2, \quad D_2 = \max\left\{4^{1+\nu}C_2e^{F_n}, 64(2C_1)^{2-\nu}\right\}.$$

Since $D_1 \geq 2C_1$ and $D_2 \geq 4^{1+\nu} C_2 e^{F_n}$, we have $\sup_{g \in \mathcal{G}_{n,i}} \|g\|_{L^\infty} \leq F$ and $\sup_{g \in \mathcal{G}_{n,i}} \operatorname{Var}(g) \leq v_{n,i}$. We first check the validation of (B2). Since $M_{n,i} \leq 2C_1 F_n, D_2 \geq 64(2C_1)^{2-\nu}$,

$$\frac{v}{F^2} = \frac{v_{n,i}}{D_1^2 F_n^2} \leq \frac{64(2C_1)^{2-2\nu} D_2 F_n^{2-\nu} (2C_1 F_n)^\nu}{D_2^2 F_n^2} = \frac{64(2C_1)^{2-\nu}}{D_2} \leq 1,$$

$$M_{n,i} = M_{n,i}^{1-\nu} M_{n,i}^\nu \leq (2C_1 F_n)^{1-\nu} M_{n,i}^\nu = \frac{8(2C_1)^{1-\nu} D_2 F_n^{2-\nu} M_{n,i}^\nu}{8 D_2 F_n} = \frac{v_{n,i}}{8 D_2 F_n} \leq \frac{v_{n,i}}{8 F_n}$$

when F_n is large enough. Thus (B2) is satisfied.

For (B3), note that for $g_1 = \phi(y f_n(\mathbf{x})) - \phi(y f_1(\mathbf{x})), g_2 = \phi(y f_n(\mathbf{x})) - \phi(y f_2(\mathbf{x}))$ where $f_1, f_2 \in \mathcal{F}_{n,i}$, $|g_1 - g_2| \leq C_1 |f_1 - f_2|$. We have

$$\mathcal{H}_B(\delta, \mathcal{G}_{n,i}, \|\cdot\|_{L^2}) \leq \mathcal{H}_B(C_1 \delta, \mathcal{F}_{n,i}, \|\cdot\|_{L^2}) \leq \mathcal{H}_B(C_1 \delta, \mathcal{F}_n, \|\cdot\|_{L^2}),$$

where the second inequality comes from $\mathcal{F}_{n,i} \subset \mathcal{F}_n$. Since $M_{n,i}^{-1} \int_{\lambda M_{n,i}/32}^{v_{n,i}^{1/2}} \left(\mathcal{H}_B(\tau, \mathcal{G}_{n,i}, \|\cdot\|_{L^2})\right)^{1/2} d\tau$ is a non-increasing function of i,

$$\begin{aligned}
&M_{n,i}^{-1} \int_{\lambda M_{n,i}/32}^{v_{n,i}^{1/2}} \left(\mathcal{H}_B(\tau, \mathcal{G}_{n,i}, \|\cdot\|_{L^2})\right)^{1/2} d\tau \\
\leq& M_{n,1}^{-1} \int_{M_{n,1}/64}^{v_{n,1}^{1/2}} \left(\mathcal{H}_B(C_1 \tau, \mathcal{F}_n, \|\cdot\|_{L^2})\right)^{1/2} d\tau \\
\leq& M_{n,1}^{-1} v_{n,1}^{1/2} \left(\mathcal{H}_B(C_1 \epsilon_n^2/128, \mathcal{F}_n, \|\cdot\|_{L^2})\right)^{1/2} \\
\leq& (D_2 F_n^{2-\nu})^{1/2} M_{n,1}^{\nu/2-1} \left(C_3 e^{-F_n} n \left(2^{-7} C_1 \epsilon_n^2 F_n^{-1}\right)^{2-\nu}\right)^{1/2} \\
\leq& C_3^{1/2} C_1^{1-\nu/2} D_2^{1/2} F_n^{1-\nu/2} e^{-F_n/2} (\epsilon_n^2/2)^{\nu/2-1} \left(2^{7\nu/2-7} n^{1/2} \epsilon_n^{2-\nu} F_n^{\nu/2-1}\right) \\
=& \left(2^{6\nu-12} e^{-F_n} C_3 C_1^{2-\nu} D_2\right)^{1/2} n^{1/2},
\end{aligned}$$

where in the third inequality we used (A5). (B3) is satisfied when F_n is large enough.

To verify (B1), we use (B2) and (B3). From (B2), since $v_{n,i}^{1/2} \leq F$, we have $M_{n,i} \leq \lambda v_{n,i}/(4F) \leq \frac{1}{8} v_{n,i}^{1/2}$ which implies $v_{n,i}^{1/2} \geq 8 M_{n,i} > M_{n,i}/16 = \lambda M_{n,i}/8$. Thus the condition in (B3) is satisfied. From (B3), we derive

$$\begin{aligned}
&\left(\mathcal{H}_B\left(v_{n,i}^{1/2}, \mathcal{G}_{n,i}, \|\cdot\|_{L^2}\right)\right)^{1/2} \leq \frac{M_{n,i}}{v_{n,i}^{1/2} - M_{n,i}/64} M_{n.i}^{-1} \int_{M_{n,i}/64}^{v_{n,i}^{1/2}} \left(\mathcal{H}_B(\tau, \mathcal{G}_{n,i}, \|\cdot\|_{L^2})\right)^{1/2} d\tau \\
\leq& \frac{M_{n,i}}{v_{n,i}^{1/2} - M_{n,i}/64} \cdot 2^{-23/2} n^{1/2} \leq \frac{4}{3} \frac{M_{n,i}}{v_{n,i}^{1/2}} \cdot 2^{-23/2} n^{1/2} = \frac{1}{3 \cdot 2^{19/2}} \frac{M_{n,i}}{v_{n,i}^{1/2}} n^{1/2},
\end{aligned}$$

where in the third inequality we used $M_{n,i} \leq 16 v_{n,i}^{1/2}$. Again from (B2), $M_{n,i} \leq \lambda v_{n,i}/(4F) = v_{n,i}/(8F)$. Therefore

$$\begin{aligned}
\frac{\lambda M_{n,i}^2 n}{8(4 v_{n,i} + M_{n,i} F/3)} &= \frac{M_{n,i}^2 n}{16(4 v_{n,i} + M_{n,i} F/3)} \geq \frac{M_{n,i}^2 n}{(64 + 2/3) v_{n,i}} \\
&\geq \frac{M_{n,i}^2 n}{9 \cdot 2^{19} v_{n,i}} \geq \mathcal{H}_B\left(v_{n,i}^{1/2}, \mathcal{G}_{n,i}, \|\cdot\|_{L^2}\right)
\end{aligned}$$

and (B1) is verified.

Apply Lemma 19 to each $\mathcal{H}_{n,i}$, we get

$$\begin{aligned}
&\mathbb{P}\left(\mathcal{E}_\phi(\widehat{f}_{\phi,n}, f_\phi^*) \geq \epsilon_n^2\right) \leq \sum_{i=1}^{i_n} 3 \exp\left(-\frac{n M_{n,i}^2}{4\left(4 v_{n,i} + M_{n,i} F/3\right)}\right) \\
\leq& \sum_{i=1}^{\infty} 3 \exp\left(-C_4 n M_{n,i}^2 / v_{n,i}\right) \leq \sum_{i=1}^{\infty} 3 \exp\left(-C_5 \left(2^i\right)^{2-\nu} e^{-F_n} n \left(\epsilon_n^2/F_n\right)^{2-\nu}\right) \\
\leq& C_6 \exp\left(-C_5 e^{-F_n} n \left(\epsilon_n^2/F_n\right)^{2-\nu}\right).
\end{aligned}$$

□

D.2. Proof of Lemma 2

The proof of Lemma 2 consists of two steps. We first show that there exists a composition of networks $\widetilde{f}_{\phi,n}$ such that $\|\widetilde{f}_{\phi,n} - f^*_{\phi,n}\|_{L^\infty} \leq 4e^{F_n}\varepsilon$. Then we show that $\widetilde{f}_{\phi,n}$ can be realized by a ConvResNet $\bar{f}_{\phi,n}$.

Lemma 20 shows the existence of $\widetilde{f}_{\phi,n}$.

Lemma 20. *Assume Assumption 1 and 2. Assume $0 < p, q \leq \infty$, $0 < s < \infty$, $s \geq d/p + 1$. There exists a network composition architecture*

$$\widetilde{\mathcal{F}}^{(F_n)} = \{g_{F_n} \circ \widetilde{h}_n \circ g_n \circ \bar{\eta}\} \tag{58}$$

where

$\bar{\eta} \in \mathcal{C}(M, L, J, K, \kappa_1, \kappa_2)$ with

$$M = O(\varepsilon^{-d/s}),\ L = O(\log(1/\varepsilon) + D + \log D),\ J = O(D),\ \kappa_1 = O(1),\ \log \kappa_2 = O\left(\log^2(1/\varepsilon)\right),$$

$\widetilde{h}_n \in \mathcal{F}^{\mathrm{MLP}}(L_2, J_2, \kappa_2, F_n)$ with

$$L_2 = O(\log(e^{F_n}/\varepsilon)),\ J_2 = O(e^{F_n}\varepsilon^{-1}\log(e^{F_n}/\varepsilon)),\ \kappa_2 = O(e^{F_n}),$$

and

$$g_n(z) = \mathrm{ReLU}\left(-\mathrm{ReLU}\left(-z + \frac{e^{F_n}}{1+e^{F_n}}\right) + \frac{e^{F_n}-1}{1+e^{F_n}}\right) + \frac{1}{1+e^{F_n}},$$
$$g_{F_n} = \mathrm{ReLU}\left(-\mathrm{ReLU}\left(-z + F_n\right) + 2F_n\right) - F_n.$$

For any $\eta \in B^s_{p,q}(\mathcal{M})$ with $\|\eta\|_{B^s_{p,q}(\mathcal{M})} \leq c_0$ for some constant c_0, let $f^*_{\phi,n}$ be defined as in (33). For any n and $\varepsilon \in (0,1)$, there exists a composition of networks $\widetilde{f}_{\phi,n} \in \widetilde{\mathcal{F}}^{(F_n)}$ such that

$$\|\widetilde{f}_{\phi,n} - f^*_{\phi,n}\|_{L^\infty} \leq 4e^{F_n}\varepsilon.$$

Proof of Lemma 20. According to Theorem 1, for any $\varepsilon_1 \in (0,1)$ and $K \in [2, D]$, there is a ConvResNet architecture $\mathcal{C}(M, L, J, K, \kappa_1, \kappa_2)$ with

$$M = O(\varepsilon_1^{-d/s}),\ L = O(\log(1/\varepsilon_1) + D + \log D),\ J = O(D),\ \kappa_1 = O(1),\ \log \kappa_2 = O\left(\log^2(1/\varepsilon)\right),$$

such that there exists $\bar{\eta} \in \mathcal{C}(M, L, J, K, \kappa_1, \kappa_2)$ with $\|\bar{\eta} - \eta\|_{L^\infty} \leq \varepsilon_1$

Since $f^*_\phi = \log\frac{\eta}{1-\eta}$, we have $f^*_{\phi,n} = \log\frac{\eta_n}{1-\eta_n}$ with

$$\eta_n(\mathbf{x}) = \begin{cases} \frac{1}{1+e^{F_n}}, & \text{if } \eta(\mathbf{x}) < \frac{1}{1+e^{F_n}}, \\ \eta(\mathbf{x}), & \text{if } \frac{1}{1+e^{F_n}} \leq \eta(\mathbf{x}) \leq \frac{e^{F_n}}{1+e^{F_n}}, \\ \frac{e^{F_n}}{1+e^{F_n}}, & \text{if } \eta(\mathbf{x}) > \frac{e^{F_n}}{1+e^{F_n}}. \end{cases}$$

The function $\max(\min(z, \frac{e^{F_n}}{1+e^{F_n}}), \frac{1}{1+e^{F_n}})$ can be realized by $g_n(z) = \mathrm{ReLU}\left(-\mathrm{ReLU}\left(-z + \frac{e^{F_n}}{1+e^{F_n}}\right) + \frac{e^{F_n}-1}{1+e^{F_n}}\right) + \frac{1}{1+e^{F_n}}$ Then $g_n \circ \bar{\eta}$ is an approximation of η_n.

Define

$$h_n = \max\left(\min\left(\log\left(\frac{z}{1-z}\right), F_n\right), -F_n\right)$$

for $z \in [0,1]$ which is a Lipschitz function with Lipschitz constant $(1+e^{F_n})^2/e^{F_n}$. According to Chen et al. (2019a, Theorem 4.1), there exists an MLP $\widetilde{h} \in \mathcal{F}^{\mathrm{MLP}}(L, J, \kappa)$ such that $\|\widetilde{h} - h_n \cdot \frac{e^{F_n}}{(1+e^{F_n})^2}\|_{L^\infty} \leq \varepsilon_2\frac{e^{F_n}}{(1+e^{F_n})^2}$ with

$$L = O(\log(e^{F_n}/\varepsilon_2)),\ J = O(e^{F_n}\varepsilon_2^{-1}\log(e^{F_n}/\varepsilon_2)),\ \kappa = 1.$$

Let $\widetilde{h}_n = \frac{(1+e^{F_n})^2}{e^{F_n}}\widetilde{h}$. Then $\widetilde{h}_n \in \mathcal{F}(L_2, J_2, \kappa_2)$ such that $\|\widetilde{h}_n - h_n\|_{L^\infty} \leq \varepsilon_2$ with

$$L_2 = O(\log(e^{F_n}/\varepsilon_2)),\ J_2 = O(e^{F_n}\varepsilon_2^{-1}\log(e^{F_n}/\varepsilon_2)),\ \kappa_2 = O(e^{F_n}).$$

Let $g_{F_n} = \mathrm{ReLU}\left(-\mathrm{ReLU}\left(-z + F_n\right) + 2F_n\right) - F_n$. We define

$$\widetilde{f}_{\phi,n} = g_{F_n} \circ \widetilde{h}_n \circ g_n \circ \bar{\eta}$$

as an approximation of $f^*_{\phi,n}$. Then the error of $\widetilde{f}_{\phi,n}$ can be decomposed as

$$\begin{aligned}\|\widetilde{f}_{\phi,n} - f^*_{\phi,n}\|_{L^\infty} &= \|(g_{F_n} \circ \widetilde{h}_n) \circ (g_n \circ \bar{\eta}) - h_n \circ \eta_n\|_{L^\infty} \\ &\leq \|(g_{F_n} \circ \widetilde{h}_n) \circ (g_n \circ \bar{\eta}) - h_n \circ (g_n \circ \bar{\eta})\|_{L^\infty} + \|h_n \circ (g_n \circ \bar{\eta}) - h_n \circ \eta_n\|_{L^\infty} \\ &\leq \varepsilon_2 + \frac{(1+e^{F_n})^2}{e^{F_n}}\|\eta_n - g_n \circ \bar{\eta}\|_{L^\infty} \leq \frac{(1+e^{F_n})^2}{e^{F_n}}\varepsilon_1 + \varepsilon_2.\end{aligned}$$

Choosing $\varepsilon_2 = \frac{(1+e^{F_n})^2}{e^{F_n}}\varepsilon_1$ gives rise to $\|\widetilde{f}_{\phi,n} - f^*_{\phi,n}\|_{L^\infty} \leq 4e^{F_n}\varepsilon_1$. With this choice, we have

$$L_2 = O(\log(1/\varepsilon_1)),\ J_2 = O(\varepsilon_1^{-1}\log(1/\varepsilon_1)),\ \kappa_2 = O(e^{F_n}).$$

Setting $\varepsilon_1 = \varepsilon$ proves the lemma. □

To show that $\widetilde{\mathcal{F}}^{(F_n)}$ can be realized by a ConvResNet class and to derive its covering number, we need the following lemma to bound the covering number of ConvResNets.

Lemma 21. *Let $\mathcal{C}(M,L,J,K,\kappa_1,\kappa_2)$ be the ConvResNet structure defined in Theorem 1. Its covering number is bounded by*

$$\log \mathcal{N}\left(\delta, \mathcal{C}(M,L,J,K,\kappa_1,\kappa_2), \|\cdot\|_{L^\infty}\right) = O\left(D^3\varepsilon^{-d/s}\log(1/\varepsilon)(\log(1/\varepsilon) + \log D \log(1/\delta)\right).$$

Lemma 21 is proved based on the following lemma:

Lemma 22 (Lemma 4 of Oono & Suzuki (2019))**.** *Let $\mathcal{C}(M,L,J,K,\kappa_1,\kappa_2)$ be a class of ConvResNet architecture from $\mathbb{R}^D$ to $\mathbb{R}$. Let $\kappa = \kappa_1 \vee \kappa_2$. For $\delta > 0$, we have*

$$\mathcal{N}(\delta, \mathcal{C}(M,L,J,K,\kappa_1,\kappa_2), \|\cdot\|_{L^\infty}) \leq (2\kappa\Lambda_1/\delta)^{\Lambda_2},$$

where

$$\Lambda_1 = (8M+12)D^2(1\vee\kappa_2)(1\vee\kappa_1)\widetilde{\rho}\widetilde{\rho}^+,\ \Lambda_2 = ML(16D^2K+4D)+4D^2+1$$

with $\widetilde{\rho} = (1+\rho)^M, \widetilde{\rho}^+ = 1 + ML\rho^+, \rho = (4DK\kappa_1)^L$ and $\rho^+ = (1 \vee 4DK\kappa_1)^L$.

Proof of Lemma 21. According to Lemma 22,

$$\log \mathcal{N}\left(\delta, \mathcal{C}(M,L,J,K,\kappa_1,\kappa_2), \|\cdot\|_{L^\infty}\right) \leq \Lambda_2 \log(2\kappa\Lambda_1/\delta).$$

In the ConvResNet architecture defined in Theorem 1, $\kappa_1 = (8DK)^{-1}M^{-1/L}$, $\rho = (1/2)^L M^{-1} < M^{-1}$. We have $\widetilde{\rho} = (1+\rho)^M \leq (1+M^{-1})^M \leq e$. Moreover, we have $\rho^+ = 1, \widetilde{\rho}^+ = 1 + ML$. Since $\log\kappa_2 = O(\log^2(1/\varepsilon))$, substituting $M = O\left(\varepsilon^{-d/s}\right)$ and $L = O(\log(1/\varepsilon) + D + \log D)$ gives rise to $\log\Lambda_1 = O(\log^2(1/\varepsilon) + \log D)$ and $\Lambda_2 = O\left(D^3\varepsilon^{-d/s}\log(1/\varepsilon)\right)$. Therefore,

$$\log \mathcal{N}\left(\delta, \mathcal{C}(M,L,J,K,\kappa_1,\kappa_2), \|\cdot\|_{L^\infty}\right) = O\left(D^3\varepsilon^{-d/s}\log(1/\varepsilon)(\log^2(1/\varepsilon) + \log D + \log(1/\delta)\right).$$

The constants hidden in $O(\cdot)$ depend on d, s, $\frac{2d}{sp-d}, p, q, c_0, \tau$ and the surface area of $\mathcal{M}$. □

The following lemma shows that $\widetilde{\mathcal{F}}^{(F_n)}$ can be realized by a ConvResNet class $\mathcal{C}^{(F_n)}$ and estimates the covering number of the class of $\mathcal{C}^{(F_n)}$.

Lemma 23. *Let $\mathcal{C}^{(F_n)}$ be defined as in Lemma 2. The network composition class $\widetilde{\mathcal{F}}^{(F_n)}$ defined in Lemma 20 can be realized by a ConvResNet class $\mathcal{C}^{(F_n)}$. Moreover, the covering number of $\mathcal{C}^{(F_n)}$ is bounded by*

$$\log \mathcal{N}(\delta, \mathcal{C}^{(F_n)}, \|\cdot\|_{L^\infty}) = O\left(\varepsilon^{-\left(\frac{d}{s}\vee 1\right)}\left(\log^3(1/\varepsilon) + F_n + \log(1/\delta)\right)\right).$$

Proof of Lemma 23. In this proof, we show that each part of $\widetilde{f}_{\phi,n}$ in Lemma 20 can be realized by ConvResNet architectures. Specifically, we show that $\bar{\eta}, g_n, \widetilde{h}_n$ can be realized by residual blocks $\bar{\eta}, \bar{g}_n, \bar{h}$, and g_{F_n} can be realized by a ConvResNet $\bar{g}_{F_n}$. In the following, we show the existence of each ingredient.

Realize $\bar{\eta}$ by residual blocks. In Lemma 20, $\bar{\eta} \in \mathcal{C}(M^{(\eta)}, L^{(\eta)}, J^{(\eta)}, K^{(\eta)}, \kappa_1^{(\eta)}, \kappa_2^{(\eta)})$ with

$$M^{(\eta)} = O\left(\varepsilon^{-d/s}\right),\ L^{(\eta)} = O(\log(1/\varepsilon) + D + \log D),\ J^{(\eta)} = O(D),\ \kappa^{(\eta)} = O(1),\ \log\kappa_2^{(\eta)} = O(\log^2(1/\varepsilon)).$$

By Lemma 21, the covering number of this architecture is bounded by

$$\begin{aligned}&\log\mathcal{N}\left(\delta, \mathcal{C}(M^{(\eta)}, L^{(\eta)}, J^{(\eta)}, K^{(\eta)}, \kappa_1^{(\eta)}, \kappa_2^{(\eta)}), \|\cdot\|_{L^\infty}\right)\\ &= O\left(\varepsilon^{-(\frac{d}{s}\vee 1))}\log(1/\varepsilon)\left(\log^2(1/\varepsilon) + \log D + \log(1/\delta)\right)\right).\end{aligned}$$

Excluding the final fully connected layer, denote all of the residual blocks of $\bar{\eta}$ by $\bar{\eta}^{(\mathrm{Conv})}$, then $\bar{\eta} \in \mathcal{C}^{(\eta)}$ with $\mathcal{C}^{(\eta)} = \mathcal{C}^{\mathrm{Conv}}(M^{(\eta)}, L^{(\eta)}, J^{(\eta)}, K^{(\eta)}, \kappa_1^{(\eta)})$ and

$$\begin{aligned}\log\mathcal{N}\left(\delta, \mathcal{C}^{(\eta)}, \|\cdot\|_{L^\infty}\right) &\leq \log\mathcal{N}\left(\delta, \mathcal{C}(M^{(\eta)}, L^{(\eta)}, J^{(\eta)}, K^{(\eta)}, \kappa_1^{(\eta)}, \kappa_2^{(\eta)}), \|\cdot\|_{L^\infty}\right)\\ &= O\left(D^3\varepsilon^{-d/s}\log(1/\varepsilon)(\log^2(1/\varepsilon) + \log D + \log(1/\delta))\right).\end{aligned}$$

We denote the i-th row (the i-th element of all channels) of the output of the residual-blocks $\bar{\eta}^{(\mathrm{Conv})}$ by $(\bar{\eta}^{(\mathrm{Conv})})_{i,:}$. In the proof of Theorem 1, the input in $\mathbb{R}^D$ is padded into $\mathbb{R}^{D\times 3}$ by 0's. The output of $\bar{\eta}$ has the form $(\bar{\eta}^{(\mathrm{Conv})})_{1,:} = \frac{\kappa_1^{(\eta)}}{\kappa_2^{(\eta)}}\begin{bmatrix}\star & \bar{\eta}_+ & \bar{\eta}_-\end{bmatrix}$. Here $\star$ denotes some number that does not affect the result. In this proof, instead of padding the input into size $D \times 3$, we pad it into size $D \times 8$. The weights in the first M blocks of h is the same as that of $\bar{\eta}$ except we need to pad the filters and biases by 0 to be compatible with the additional channels. Then the output of $\bar{\eta}^{(\mathrm{Conv})}$ is $\frac{\kappa_1^{(\eta)}}{\kappa_2^{(\eta)}}\begin{bmatrix}\star & \bar{\eta}_+ & \bar{\eta}_- & 0 & 0 & 0 & 0 & 0\end{bmatrix}$.

Realize g_n by residual blocks. To realize g_n, we add another block with 4 layers with filters and biases $\left\{\mathcal{W}_{g_n}^{(l)}, B_{g_n}^{(l)}\right\}_{l=1}^{5}$ where $\mathcal{W}_{g_n}^{(l)} \in \mathbb{R}^{8\times 1\times 8}, B_{g_n}^{(l)} \in \mathbb{R}^{8\times 8}$. We set the parameters in the first layer as

$$\begin{aligned}\left(\mathcal{W}_{g_n}^{(1)}\right)_{2,1,:} &= \begin{bmatrix}0 & \frac{\kappa_2^{(\eta)}}{\kappa_1^{(\eta)}} & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix},\\ \left(\mathcal{W}_{g_n}^{(1)}\right)_{3,1,:} &= \begin{bmatrix}0 & 0 & \frac{\kappa_2^{(\eta)}}{\kappa_1^{(\eta)}} & 0 & 0 & 0 & 0 & 0\end{bmatrix},\\ \left(\mathcal{W}_{g_n}^{(1)}\right)_{i,1,:} &= \mathbf{0} \quad \text{for } i = 1, 4, 5, ..., 8,\end{aligned}$$

and $B_{g_n}^{(1)} = \mathbf{0}$. This layer scales the output of $\bar{\eta}$ back to $\begin{bmatrix}\star & \bar{\eta}_+ & \bar{\eta}_- & 0 & 0 & 0 & 0 & 0\end{bmatrix}$. Then we use the other 4 layers to realize g_n. The second layer is set as

$$\begin{aligned}\left(\mathcal{W}_{g_n}^{(2)}\right)_{4,1,:} &= \begin{bmatrix}0 & -1 & 1 & 0 & 0 & 0 & 0 & 0\end{bmatrix},\\ \left(\mathcal{W}_{g_n}^{(2)}\right)_{i,1,:} &= \mathbf{0} \quad \text{for } i = 1, 2, 3, 5, 6, 7, 8,\\ \left(B_{g_n}^{(2)}\right)_{1,:} &= \begin{bmatrix}0 & 0 & 0 & \frac{e^{F_n}}{1+e^{F_n}} & 0 & 0 & 0 & 0 & \end{bmatrix},\\ \left(B_{g_n}^{(2)}\right)_{i,:} &= \mathbf{0} \quad \text{for } i = 2, ..., 8.\end{aligned}$$

The third layer is set as

$$\begin{aligned}\left(\mathcal{W}_{g_n}^{(3)}\right)_{4,1,:} &= \begin{bmatrix}0 & 0 & 0 & -1 & 0 & 0 & 0 & 0\end{bmatrix},\\ \left(\mathcal{W}_{g_n}^{(3)}\right)_{i,1,:} &= \mathbf{0} \quad \text{for } i = 1, 2, 3, 5, 6, 7, 8,\\ \left(B_{g_n}^{(3)}\right)_{1,:} &= \begin{bmatrix}0 & 0 & 0 & \frac{e^{F_n}-1}{1+e^{F_n}} & 0 & 0 & 0 & 0 & \end{bmatrix},\\ \left(B_{g_n}^{(3)}\right)_{i,:} &= \mathbf{0} \quad \text{for } i = 2, ..., 8.\end{aligned}$$

The forth layer is set as

$$\begin{aligned}
\left(\mathcal{W}^{(4)}_{g_n}\right)_{4,1,:} &= \begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\end{bmatrix}, \\
\left(\mathcal{W}^{(4)}_{g_n}\right)_{i,1,:} &= \mathbf{0} \quad \text{for } i = 1,2,3,5,6,7,8, \\
\left(B^{(4)}_{g_n}\right)_{1,:} &= \begin{bmatrix}0 & 0 & 0 & \frac{1}{1+e^{F_n}} & 0 & 0 & 0 & 0 & \end{bmatrix}, \\
\left(B^{(4)}_{g_n}\right)_{i,:} &= \mathbf{0} \quad \text{for } i = 2, ..., 8.
\end{aligned}$$

The output of $g_n \circ \bar{\eta}$ is stored as the first element in the forth channel of the output of $\bar{g}_n \circ \bar{\eta}^{(\mathrm{Conv})}$:

$$(\bar{g}_n \circ \bar{\eta})_{1,:} = \begin{bmatrix}\star & \star & \star & g_n \circ \bar{\eta} & 0 & 0 & 0 & 0\end{bmatrix}.$$

We have $\bar{g}_n \in \mathcal{C}^{(g_n)}$ where $\mathcal{C}^{(g_n)} = \mathcal{C}^{(g_n)} = \mathcal{C}^{\mathrm{Conv}}(M^{(g_n)}, L^{(g_n)}, J^{(g_n)}, K^{(g_n)}, \kappa^{(g_n)})$ with

$$M^{(g_n)} = 1,\ L^{(g_n)} = 4,\ J^{(g_n)} = 8,\ K^{(g_n)} = 1,\ \kappa^{(g_n)} = O\left(\frac{\kappa_2^{(\eta)}}{\kappa_1^{(\eta)}} \vee \frac{e^{F_n}}{1+e^{F_n}}\right).$$

According to Lemma 22, the covering number of $\mathcal{C}^{(g_n)}$ is bounded as

$$\log \mathcal{N}(\delta, \mathcal{C}^{(g_n)}, \|\cdot\|_{L^\infty}) = O\left(\log\left(\frac{\kappa_2^{(\eta)}}{\kappa_1^{(\eta)}} \vee \frac{e^{F_n}}{1+e^{F_n}}\right) + \log(1/\delta)\right).$$

Substituting the expressions of $\kappa_1^{(\eta)}, \kappa_2^{(\eta)}$ into the expression above gives rise to $\log \kappa^{(g_n)} = O(\left(\log^2(1/\varepsilon)\right)$ and $\log \mathcal{N}(\delta, \mathcal{C}^{(g_n)}, \|\cdot\|_{L^\infty}) = O(\log^2(1/\varepsilon) + \log(1/\delta))$.

Realize $\widetilde{h}_n$ by residual blocks. To realize $\widetilde{h}_n$, from the construction of $\widetilde{h}_n$ and using Oono & Suzuki (2019, Corollary 4), we can realize $\widetilde{h}_n$ by $\bar{h}_n \in \mathcal{C}^{(h_n)}$ with

$$\mathcal{C}^{(h_n)} = \mathcal{C}(M^{(h_n)}, L^{(h_n)}, J^{(h_n)}, K^{(h_n)}, \kappa_1^{(h_n)} \kappa_2^{(h_n)}, F_n),$$

and

$$\begin{aligned}
&M^{(h_n)} = O\left(\varepsilon_2^{-1}\right),\ L^{(h_n)} = O\left(\log(1/\varepsilon_2)\right),\ J^{(h_n)} = O(1),\ K^{(h_n)} = 1, \\
&\kappa_1^{(h_n)} = O(1),\ \log \kappa_2^{(h_n)} = O(\log(L_h/\varepsilon_2)),
\end{aligned}$$

where $\varepsilon_2 = \frac{(1+e^{F_n})^2}{e^{F_n}} \varepsilon$, and L_h is the Lipschitz constant of h_n. According to Lemma 22, the covering number of this class is bounded by

$$\log \mathcal{N}\left(\delta, \mathcal{C}^{(h_n)}, \|\cdot\|_{L^\infty}\right) = O\left(D^2 \varepsilon_2^{-1} \log(1/\varepsilon_2)(\log(1/\varepsilon_2) + \log(L_h/\varepsilon_2) + \log(1/\delta)\right).$$

Note that such $\bar{h}$ is from $\mathbb{R}$ to $\mathbb{R}$. Since the information we need from the output of $\bar{g}_n \circ \bar{\eta}$ is only the first element in the forth channel, we can follow the proof of Oono & Suzuki (2019, Theorem 6) to construct $\bar{h}$ by padding the elements in the filters and biases by 0 so that all operations work on the forth channel and store results on the fifth and sixth channel. Substituting $\varepsilon_2 = \frac{(1+e^{F_n})^2}{e^{F_n}} \varepsilon$ and $L_{h_n} = (1+e^{F_n})^2/e^{F_n}$ yields

$$\begin{aligned}
&M^{(h_n)} = O\left(e^{-F_n}\varepsilon^{-1}\right), L^{(h_n)} = O\left(\log(1/\varepsilon)\right), J^{(h_n)} = O(1), \\
&\kappa_1^{(h_n)} = O(1), \log \kappa_2^{(h_n)} = O\left(\log\left(e^{F_n}/\varepsilon\right)\right)
\end{aligned}$$

and

$$\log \mathcal{N}\left(\delta, \mathcal{C}^{(h_n)}, \|\cdot\|_{L^\infty}\right) = O\left(D^2 e^{-F_n} \varepsilon^{-1} \log(1/\varepsilon)(\log(1/\varepsilon) + F_n + \log(1/\delta)\right).$$

Similar to $\bar{\eta}^{(\mathrm{Conv})}$, denote all residual blocks of $\bar{h}$ by $\bar{h}^{(\mathrm{Conv})}$. We have

$$(\bar{h}^{(\mathrm{Conv})})_{1,:} = \frac{\kappa_1^{(h_n)}}{\kappa_2^{(h_n)}} \begin{bmatrix}\star & \star & \star & \star & (\widetilde{h}_n)_+ & (\widetilde{h}_n)_- & 0 & 0\end{bmatrix}.$$

Realize g_{F_n} by a ConvResNet. We then add another residual block of 3 layers followed by a fully connected layer to realize g_{F_n}. Denote the parameters in this block and the fully connected layer by $\{\mathcal{W}^{(l)}_{g_{F_n}}, B^{(l)}_{g_{F_n}}\}_{l=1}^{3}$ and $\{W, b\}$, respectively. Here $\mathcal{W}^{(l)}_{g_{F_n}} \in \mathbb{R}^{8\times 1\times 8}, B^{(l)}_{g_{F_n}} \in \mathbb{R}^{8\times 8}, W \in \mathbb{R}^{8\times 8}$ and $b \in \mathbb{R}$.

The first layer is set as

$$\begin{aligned}
\left(\mathcal{W}^{(1)}_{g_{F_n}}\right)_{5,1,:} &= \begin{bmatrix}0 & 0 & 0 & 0 & \frac{\kappa_2^{(h_n)}}{\kappa_1^{(h_n)}} & 0 & 0 & 0\end{bmatrix},\\
\left(\mathcal{W}^{(1)}_{g_{F_n}}\right)_{6,1,:} &= \begin{bmatrix}0 & 0 & 0 & 0 & 0 & \frac{\kappa_2^{(h_n)}}{\kappa_1^{(h_n)}} & 0 & 0\end{bmatrix},\\
\left(\mathcal{W}^{(1)}_{g_{F_n}}\right)_{i,1,:} &= \mathbf{0} \quad \text{for } i = 1,2,3,4,7,8,\\
\left(B^{(1)}_{g_{F_n}}\right)_{i,:} &= \mathbf{0} \quad \text{for } i = 1,2,...,8.
\end{aligned}$$

This layer scales the output of $\bar{h}^{(\mathrm{Conv})}$ back to $\begin{bmatrix}\star & \star & \star & \star & (\widetilde{h}_n)_+ & (\widetilde{h}_n)- & 0 & 0\end{bmatrix}$. The rest layers are used to realize g_{F_n}.

The second layer is set as

$$\begin{aligned}
\left(\mathcal{W}^{(2)}_{g_{F_n}}\right)_{7,1,:} &= \begin{bmatrix}0 & 0 & 0 & 0 & -1 & 0 & 0 & 0\end{bmatrix},\\
\left(\mathcal{W}^{(2)}_{g_{F_n}}\right)_{8,1,:} &= \begin{bmatrix}0 & 0 & 0 & 0 & 0 & -1 & 0 & 0\end{bmatrix},\\
\left(\mathcal{W}^{(2)}_{g_{F_n}}\right)_{i,1,:} &= \mathbf{0} \quad \text{for } i = 1,...,6\\
\left(B^{(2)}_{g_{F_n}}\right)_{1,:} &= \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & F_n & F_n\end{bmatrix},\\
\left(B^{(2)}_{g_{F_n}}\right)_{i,:} &= \mathbf{0} \quad \text{for } i = 2,...,8.
\end{aligned}$$

The third layer is set as

$$\begin{aligned}
\left(\mathcal{W}^{(3)}_{g_{F_n}}\right)_{7,1,:} &= \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & -1 & 0\end{bmatrix},\\
\left(\mathcal{W}^{(3)}_{g_{F_n}}\right)_{8,1,:} &= \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 0 & -1\end{bmatrix},\\
\left(\mathcal{W}^{(3)}_{g_{F_n}}\right)_{i,1,:} &= \mathbf{0} \quad \text{for } i = 1,...,6\\
\left(B^{(3)}_{g_{F_n}}\right)_{1,:} &= \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & F_n & F_n\end{bmatrix},\\
\left(B^{(3)}_{g_{F_n}}\right)_{i,:} &= \mathbf{0} \quad \text{for } i = 2,...,8.
\end{aligned}$$

The first row of the output of the third layer is

$$\begin{bmatrix}\star & \star & \star & \star & \star & \star & \min((\widetilde{h}_n)_+, F_n) & \min((\widetilde{h}_n)_-, F_n)\end{bmatrix}.$$

Then the fully connected layer is set as

$$\begin{aligned}
W_{1,:} &= \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 & 1 & -1\end{bmatrix},\\
W_{i,:} &= \mathbf{0} \quad \text{for } i = 2,...,8
\end{aligned}$$

and $b = 0$.

Thus $\bar{g}_{F_n} \in \mathcal{C}^{(g_{F_n})}$ with $\mathcal{C}^{(g_{F_n})} = \mathcal{C}(M^{(g_{F_n})}, L^{(g_{F_n})}, J^{(g_{F_n})}, K^{(g_{F_n})}, \kappa_1^{(g_{F_n})}, \kappa_2^{(g_{F_n})})$ and

$$M^{(g_{F_n})} = 1,\ L^{(g_{F_n})} = 4,\ J^{(g_{F_n})} = 8,\ K^{(g_{F_n})} = 1,\ \kappa_1^{(g_{F_n})} = O\left(\frac{\kappa_2^{(h_n)}}{\kappa_1^{(h_n)}} \vee F_n\right),\ \kappa_2^{(g_{F_n})} = 1.$$

According to Lemma 22, substituting the expressions of $\kappa_1^{(h_n)}, \kappa_2^{(h_n)}$ gives rise to $\log \mathcal{N}(\delta, \mathcal{C}^{(g_{F_n})}, \|\cdot\|_{L^\infty}) = O\left(D^2\left(\log(1/\varepsilon) + \log F_n + \log(1/\delta)\right)\right)$.

The resulting network $\bar{f}_{\phi,n} \equiv \bar{g}_{F_n} \circ \bar{h}^{(\mathrm{Conv})} \circ \bar{g}_n \circ \bar{\eta}^{(\mathrm{Conv})}$ is a ConvResNet and

$$\bar{f}_{\phi,n}(\mathbf{x}) = \widetilde{f}_{\phi,n}(\mathbf{x})$$

for any $\mathbf{x} \in \mathcal{M}$. Denote the class of the architecture of $\bar{f}_{\phi,n}$ by $\mathcal{C}^{(F_n)}$. Its covering number is bounded by

$$\begin{aligned}
&\log \mathcal{N}(\delta, \mathcal{C}^{(F_n)}, \|\cdot\|_{L^\infty}) \\
\leq &\log \mathcal{N}\left(\delta, \mathcal{C}^{(\eta)}, \|\cdot\|_{L^\infty}\right) + \log \mathcal{N}\left(\delta, \mathcal{C}^{(g_n)}, \|\cdot\|_{L^\infty}\right) \\
&+ \log \mathcal{N}\left(\delta, \mathcal{C}^{(h_n)}, \|\cdot\|_{L^\infty}\right) + \log \mathcal{N}(\delta, \mathcal{C}^{(g_{F_n})}, \|\cdot\|_{L^\infty}) \\
=&O\left(D^3 \varepsilon^{-(\frac{d}{s}\vee 1))} \log(1/\varepsilon)\left(\log^2(1/\varepsilon) + \log D + F_n + \log(1/\delta)\right)\right).
\end{aligned}$$

The constants hidden in $O(\cdot)$ depend on $d, s, \frac{2d}{sp-d}, p, q, c_0, \tau$ and the surface area of $\mathcal{M}$. □

Proof of Lemma 2. Lemma 2 is a direct result of Lemma 20 and Lemma 23. □

D.3. Proof of Lemma 3

Proof of Lemma 3. We divide $A_n^{\complement}$ into two regions: $\{\mathbf{x} \in \mathcal{M} : f_\phi^* > F_n\}$ and $\{\mathbf{x} \in \mathcal{M} : f_\phi^* < -F_n\}$. A bound of T_2 is derived by bounding the integral on both regions.

Let us first consider the region $\{\mathbf{x} \in \mathcal{M} : f_\phi^* > F_n\}$. Since $\eta = e^{f_\phi^*}/(1+e^{f_\phi^*})$, we have

$$\begin{aligned}
\eta\phi(f_\phi^*) + (1-\eta)\phi(-f_\phi^*) &= \frac{e^{f_\phi^*}}{1+e^{f_\phi^*}}\phi(f_\phi^*) + \frac{1}{1+e^{f_\phi^*}}\phi(-f_\phi^*) \\
&\leq \phi(F_n) + \sup_{z \geq F_n} \frac{\log(1+e^z)}{1+e^z} \\
&\leq \log(1+e^{-F_n}) + \frac{\log(1+e^{F_n})}{1+e^{F_n}} \\
&\leq 2F_n e^{-F_n}.
\end{aligned} \tag{59}$$

On this region, $F_n - 1 \leq \bar{f}_{\phi,n} \leq F_n$. Thus

$$\begin{aligned}
\eta\phi(\bar{f}_{\phi,n}) + (1-\eta)\phi(-\bar{f}_{\phi,n}) &= \frac{e^{f_\phi^*}}{1+e^{f_\phi^*}}\phi(\bar{f}_{\phi,n}) + \frac{1}{1+e^{f_\phi^*}}\phi(-\bar{f}_{\phi,n}) \\
&\leq \phi(F_n - 1) + \frac{\log(1+e^{\bar{f}_{\phi,n}})}{1+e^{f_\phi^*}} \\
&\leq \log(1+e^{-(F_n-1)}) + \frac{\log(1+e^{F_n})}{1+e^{F_n}} \\
&\leq 2F_n e^{-F_n}.
\end{aligned} \tag{60}$$

Combining (59) and (60) gives

$$\left|\left[\eta\phi(f_\phi^*) + (1-\eta)\phi(-f_\phi^*)\right] - \left[\eta\phi(\bar{f}_{\phi,n}) + (1-\eta)\phi(-\bar{f}_{\phi,n})\right]\right| \leq 4F_n e^{-F_n}. \tag{61}$$

Now consider the region $\{\mathbf{x} \in \mathcal{M} : f_\phi^* < -F_n\}$, we have

$$\begin{aligned}
\eta\phi(f_\phi^*) + (1-\eta)\phi(-f_\phi^*) &= \frac{e^{f_\phi^*}}{1+e^{f_\phi^*}}\phi(f_\phi^*) + \frac{1}{1+e^{f_\phi^*}}\phi(-f_\phi^*) \\
&= \frac{1}{1+e^{-f_\phi^*}}\phi(f_\phi^*) + \frac{e^{-f_\phi^*}}{1+e^{-f_\phi^*}}\phi(-f_\phi^*) \\
&\leq \phi(F_n) + \sup_{z \leq -F_n} \frac{\log(1+e^{-z})}{1+e^{-z}} \\
&\leq \log(1+e^{-F_n}) + \frac{\log(1+e^{F_n})}{1+e^{F_n}} \\
&\leq 2F_n e^{-F_n}.
\end{aligned} \tag{62}$$

On this region, $-F_n \leq \bar{f}_{\phi,n} \leq -F_n + 1$. Thus

$$\begin{aligned}\eta\phi(\bar{f}_{\phi,n}) + (1-\eta)\phi(-\bar{f}_{\phi,n}) &= \frac{1}{1+e^{-f_\phi^*}}\phi(\bar{f}_{\phi,n}) + \frac{e^{-f_\phi^*}}{1+e^{-f_\phi^*}}\phi(-\bar{f}_{\phi,n}) \\ &\leq \phi(F_n - 1) + \frac{\log(1+e^{-\bar{f}_{\phi,n}})}{1+e^{-f_\phi^*}} \\ &\leq \log(1+e^{-(F_n-1)}) + \frac{\log(1+e^{F_n})}{1+e^{F_n}} \\ &\leq 2F_n e^{-F_n}. \end{aligned} \tag{63}$$

Combining (62) and (63) gives

$$\left|\left[\eta\phi(f_\phi^*) + (1-\eta)\phi(-f_\phi^*)\right] - \left[\eta\phi(\bar{f}_{\phi,n}) + (1-\eta)\phi(-\bar{f}_{\phi,n})\right]\right| \leq 4F_n e^{-F_n}. \tag{64}$$

Putting (61) and (64) together, we have

$$\mathrm{T}_2 \leq \int_{A_n^\complement} \left|\left[\eta\phi(f_\phi^*) + (1-\eta)\phi(-f_\phi^*)\right] - \left[\eta\phi(\bar{f}_{\phi,n}) + (1-\eta)\phi(-\bar{f}_{\phi,n})\right]\right| \mu(d\mathbf{x}) \leq 8F_n e^{-F_n}.$$

□