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 \( n^{-s/2+2(s+d)} \), 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).

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; 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

Figure 1. Illustration of (a) convolution and (b) skip-layer connection.
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 $\varepsilon$, the sample size required by Oono & Suzuki (2019) scales as $\varepsilon^{-\frac{2sD}{2+sD}}$, 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:

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

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

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

for some $\eta : M \to [0, 1]$. 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 $f^* \in B^s_{p,q}(M)$, if the weight parameters of this ConvResNet are properly chosen, it gives rise to $\hat{f}$ satisfying

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

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

$$\mathbb{E}(\mathcal{E}_\eta(\hat{f}_{\eta,n}, f^*_\eta)) \leq Cn^{-\frac{2sD}{2+sD}} \log^4 n,$$

where $\mathcal{E}_\eta(\hat{f}_{\eta,n}, f^*_\eta)$ denotes the excess logistic risk of $\hat{f}_{\eta,n}$ against $f^*_\eta$ 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
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 $\varepsilon^{-2}$ as $\varepsilon$ decreases to 0. It is shown in Petersen & Vogtlaender (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 $\varepsilon$. 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.

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 $\lfloor x \rfloor$ to denote the smallest integer that is no less than $x$ and use $\lceil x \rceil$ 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 $\Omega$ by $f|\Omega$. We use $\|f\|_{L^p}$ to denote the $L^p$ norm of $f$. We denote the Euclidean ball centered at $c$ with radius $\omega$ by $B_\omega(c)$.

2.1. Low-dimensional manifolds

We first introduce some concepts on manifolds. We refer the readers to Tü (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 $(U, \phi)$ where $U \subset \mathcal{M}$ is open and $\phi : U \to \mathbb{R}^d$, is a homeomorphism (i.e., bijective, $\phi$ and $\phi^{-1}$ are both continuous).

In a chart $(U, \phi)$, $U$ is called a coordinate neighborhood and $\phi$ 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:
\[
\phi_\alpha \circ \phi_\beta^{-1} : \phi_\beta(U_\alpha \cap U_\beta) \to \phi_\alpha(U_\alpha \cap U_\beta) \quad \text{and} \quad \phi_\beta \circ \phi_\alpha^{-1} : \phi_\alpha(U_\alpha \cap U_\beta) \to \phi_\beta(U_\alpha \cap U_\beta)
\]
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 $M$ together with a $C^\infty$ atlas.

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

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

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

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

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

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

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

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

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

**Definition 6 (Reach (Federer, 1959; Niyogi et al., 2008)).** Define the set
\[
G = \{x \in \mathbb{R}^d : \exists \text{ distinct } p, q \in M \text{ such that } d(x, M) = ||x - p||_2 = ||x - q||_2\}.
\]
The closure of $G$ is called the medial axis of $M$. The reach of $M$ is defined as
\[
\tau = \inf_{x \in M} \inf_{y \in G} ||x - y||_2.
\]
We illustrate large and small reach in Figure 2.

2.2. Besov functions on a smooth manifold

We next define Besov function spaces on $M$, 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
\[
\Delta^r f(x) = \left\{ \sum_{j=0}^{r} \left( \begin{array}{c} r \end{array} \right) \frac{1}{r} f(x + jh) \right\} \quad \text{if } x \in \Omega, x + rh \in \Omega,
\]
\[
\Delta^r f(x) = \sup_{x \in \Omega} \left| f(x) - f(x + rh) \right| \quad \text{if } q = \infty.
\]
The norm of the Besov space $B^s_{p,q}(\Omega)$ is defined as
\[
\|f\|_{B^s_{p,q}(\Omega)} := \|f\|_{L^p(\Omega)} + \|\Delta^r f\|_{L^p(\Omega)} \quad \text{if } q < \infty,
\]
\[
\|f\|_{B^s_{p,q}(\Omega)} := \|f\|_{L^p(\Omega)} + \|\Delta^r f\|_{L^p(\Omega)} \quad \text{if } q = \infty.
\]
The Besov space is $B^s_{p,q}(\Omega) = \{f \in L^p(\Omega) : \|f\|_{B^s_{p,q}(\Omega)} < \infty\}$.

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

**Definition 9 ($B^s_{p,q}$ Functions on $M$).** Let $M$ be a compact smooth manifold of dimension $d$. Let $\{U_i, \phi_i\}_{i=1}^{C_M}$ be a finite atlas on $M$ and $\{\rho_i\}_{i=1}^{C_M}$ be a partition of unity on $M$ such that $\supp(\rho_i) \subset U_i$. A function $f : M \to \mathbb{R}$ is in $B^s_{p,q}(M)$ if
\[
\|f\|_{B^s_{p,q}(M)} := \sum_{i=1}^{C_M} \||f| \circ \phi_i^{-1}\|_{B^s_{p,q}(\mathbb{R}^d)} < \infty.
\]

Since $\rho_i$ 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 $B^s_{p,q}(\mathbb{R}^d)$ (Triebel, 1992, Chapter 7).

2.3. Convolution and residual block

In this paper, we consider one-sided stride-one convolution in our network. Let $\mathcal{W} = \{W_{j,k,l}\} \in \mathbb{R}^{C^\times K \times C}$ 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
\[
y = \mathcal{W} * z, \quad y_{i,j} = \sum_{k=1}^{K} \sum_{l=1}^{L} W_{j,k,l} z_{i+k-1,l}, \tag{2}\]
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 \( x \), each residual block computes
\[ x + F(x) \]
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).

\[ F = \{ W_{j,k,l} \}_{j,k,l} \]
\[ \mathcal{W} = \{ W_{j,k,l} \}_{j,k,l} \in \mathbb{R}^{C' \times K \times C} \]
\[ \mathcal{W}' = \{ W'_{j,k,l} \}_{j,k,l} \in \mathbb{R}^{C' \times K \times C} \]

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: \( \text{ReLU}(z) = \max(z, 0) \). The ConvResNet we consider consists of a padding layer and several residual blocks followed by a fully connected feed-forward 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
\[
P(A) = [A \ 0 \ \cdots \ 0] \in \mathbb{R}^{D \times C_2}. \]
Then the matrix \( P \) is a padding layer.

In the \( m \)-th block, let \( \mathcal{W}_m = \{ W^{(1)}_m, \ldots, W^{(L_m)}_m \} \) and \( B_m = \{ B^{(1)}_m, \ldots, B^{(L_m)}_m \} \) 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
\[
\text{Conv}_{\mathcal{W}_m, B_m}(Z) = \text{ReLU}(W^{(L_m)}_m * \ldots \ldots \text{ReLU} \left( W^{(1)}_m * Z + B^{(1)}_m \right) \ldots \ldots B^{(L_m)}_m), \tag{3}\]
with ReLU applied entrywise. Denote
\[
Q(x) = (\text{Conv}_{\mathcal{W}_M, B_M} + \text{id}) \circ \ldots \circ (\text{Conv}_{\mathcal{W}_1, B_1} + \text{id}) \circ P(x). \tag{4}\]

For networks only consisting of residual blocks, we define the network class as
\[
\mathcal{C}^\text{Conv}(M, L, J, K, \kappa) = \{ Q(x) \text{ is the form of } (4) \text{ with } M \text{ residual blocks.} \}
\]

Each block has filter size bounded by \( K \), number of channels bounded by \( J \), max \( L_m \leq L \),
\[
\max_{m,l} \| W^{(l)}_m \|_\infty \lor \| B^{(l)}_m \|_\infty \leq \kappa. \tag{5}\]
Based on the network \( Q \) in (4), a ConvResNet has an additional fully connected layer and can be expressed as
\[
f(x) = WQ(x) + b \tag{6}\]
where \( W \) and \( b \) are the weight matrix and the bias in the fully connected layer. The class of ConvResNets is defined as
\[
\mathcal{C}(M, L, J, K, \kappa_1, \kappa_2, R) = \{ f \mid f(x) = WQ(x) + b \text{ with } Q \in \mathcal{C}^\text{Conv}(M, L, J, K, \kappa_1), \}
\]
\[
\| W \|_\infty \lor \| b \| \leq \kappa_2, \| f \|_L^\infty \leq R. \}
\]
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 \( x \in \mathcal{M}, \| x \|_\infty \leq B \).

Assumption 2. The reach of \( \mathcal{M} \) is \( \tau > 0 \).

Assumption 3. Let \( 0 < p, q, s < \infty, d/p + 1 \leq s < \infty \). 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

\[
\frac{\| f - f^* \|_\infty}{\| f^* \|_{L^\infty}} = O(\varepsilon), \tag{8}\]
The constant hidden in $O(\cdot)$ depend on $d, s, \frac{2d}{sp-d}, p, q, c_0, \tau$ and the surface area of $M$.

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 $\varepsilon$ error, the FNN constructed in Suzuki (2019) requires $O(\log(1/\varepsilon)$ depth, $O(\varepsilon^{-D/s})$ width and $O(\varepsilon^{-D/s} \log(1/\varepsilon))$ non-zero 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 $M$. For any $x \in \mathcal{M}$, denote its label by $y \in \{ -1, 1 \}$. The label $y$ follows the following Bernoulli-type distribution

$$P(y = 1|x) = \eta(x), \quad P(y = -1|x) = 1 - \eta(x)$$

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

We assume the following data model:

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

In binary classification, a classifier $f$ predicts the label of $x$ as $\text{sign}(f(x))$. 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

$$\mathcal{E}_\phi(f) = E(\phi(yf(x))).$$

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

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

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

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

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

$$C(n) = \{ f | f = \tilde{g}_2 \circ \tilde{h} \circ \tilde{g}_1 \circ \tilde{\eta} \text{ where}$$

$$\tilde{\eta} \in \mathcal{C}^{\text{Conv}}(M_1, L_1, J_1, K, \kappa_1), \quad \tilde{g}_1 \in \mathcal{C}^{\text{Conv}}(1, 4, 8, 1, \kappa_2),$$

$$\tilde{h} \in \mathcal{C}^{\text{Conv}}(M_2, L_2, J_2, 1, \kappa_1), \quad g_2 \in \mathcal{C}(1, 3, 8, 1, \kappa_3, 1, R) \}$$ (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:

$$\hat{f}_{\phi,n} = \arg \min_{f \in C(n)} \frac{1}{n} \sum_{i=1}^n \phi(y_i f(x_i)).$$ (15)

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

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

$$M_1 = O\left( n^{\frac{2d}{sp-d}} \right), \quad M_2 = O\left( n^{\frac{2}{(s+\kappa_2)(s+\kappa_3)}} \right),$$

$$L_1 = O(\log(1/\varepsilon) + D + \log D), \quad L_2 = O(\log(1/\varepsilon)),$$

$$J_1 = O(D), \quad J_2 = O(1), \quad \kappa_1 = O(1), \quad \log \kappa_2 = O(\log^2 n), \quad \kappa_3 = O(\log n), \quad R = O(\log n)$$

for $C(n)$. Then

$$\mathbb{E}(\mathcal{E}_\phi(\hat{f}_{\phi,n}, f^*_\phi)) \leq Cn^{-\frac{2d}{sp-d}} \log^4 n$$ (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 $M$. The constant hidden in $O(\cdot)$ depends on $d, s, \frac{2d}{sp-d}, p, q, c_0, \tau$ and the surface area of $M$.

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 $C(n)$ is constructed for the following purpose:

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

---

**Figure 4.** The ConvResNet in Theorem 1 contains a padding layer, $M$ residual blocks, and a fully connected (FC) layer. $f^* \in B^{p,q}_s(M)$, if the weight parameters of this ConvResNet are properly chosen, the network yields a function $\hat{f} \in C(M, L, J, K, \kappa_1, \kappa_2)$ satisfying

$$\|\hat{f} - f^*\|_{L^\infty} \leq \varepsilon.$$  (8)

Such a network architecture has

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

$$J = O(D), \quad \kappa_1 = O(1), \quad \log \kappa_2 = O(\log^2(1/\varepsilon)).$$ (9)

The network has a fixed filter size and a fixed number of channels.
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 \( M \).** Since the manifold \( M \) is compact, we can cover \( M \) by a finite collection of open balls \( B_{\omega}(c_i) \) for \( i = 1, \ldots, C_M \), where \( c_i \) is the center of the ball and \( \omega \) is the radius to be chosen later. Accordingly, the manifold is partitioned as \( M = \bigcup_i U_i \) with \( U_i = B_{\omega}(c_i) \cap M \). 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 \( C_M \leq \frac{\text{SA}(M)}{\omega^d} T_d \), where \( \text{SA}(M) \) is the surface area of \( M \) and \( T_d \) is the average number of \( U_i \)'s that contain a given point on \( M \) (Conway et al., 1987, Chapter 2 Equation (1)).

On each partition, we define a projection-based transformation \( \phi_i \) as

\[
\phi_i(x) = a_i V_i^T (x - c_i) + b_i,
\]

where the scaling factor \( a_i \in \mathbb{R} \) and the shifting vector \( b_i \in \mathbb{R}^d \) 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 \( T_{c_i}(M) \). The atlas on \( M \) is the collection \( (U_i, \phi_i) \) for \( i = 1, \ldots, C_M \). See Figure 5 for a graphical illustration of the atlas.

- **Decompose \( f^* \) according to the atlas.** We decompose \( f^* \) as

\[
f^* = \sum_{i=1}^{C_M} f_i \quad \text{with} \quad f_i = f \rho_i, \tag{17}
\]

where \( \{ \rho_i \}_{i=1}^{C_M} \) is a \( C^\infty \) partition of unity with \( \text{supp}(\rho_i) \subset U_i \). The existence of such a \( \{ \rho_i \}_{i=1}^{C_M} \) 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

\[
f^* = \sum_{i=1}^{C_M} (f_i \circ \phi_i^{-1}) \circ \phi_i \times 1_{U_i} \quad \text{with} \quad f_i = f \rho_i, \tag{18}
\]

where \( 1_{U_i} \) is the indicator function of \( U_i \). Since \( \phi_i \) 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 \( [0,1]^d \setminus \phi_i(U_i) \) by 0. The extended function is in \( B_{p,q}([0,1]^d) \) (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 \( \tilde{f}_i \) as

\[
\tilde{f}_i \circ \phi_i^{-1} \approx \tilde{f}_i = \sum_{j=1}^{N} \tilde{f}_{i,j} \quad \text{with} \quad \tilde{f}_{i,j} = \alpha_{k,j}^i M_{k,j,m}^d, \tag{19}
\]

where \( \alpha_{k,j}^i \in \mathbb{R} \) is a coefficient and \( M_{k,j,m}^d : [0,1]^d \to \mathbb{R} \) denotes a cardinal B-spline with indecies \( k, m \in \mathbb{N}^+ \), \( j \in \mathbb{R}^d \). Here \( k \) is a scaling factor, \( j \) 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 \( s \geq d/p + 1 \) (by Assumption 3), setting \( r = +\infty, m = [s] + 1 \) in Lemma 5 (see Appendix C.3) and applying Lemma 4 gives

\[
\left\| \tilde{f}_i - f_i \circ \phi_i^{-1} \right\|_{L^\infty} \leq C c_0 N^{-s/d} \tag{20}
\]

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

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

\[
\tilde{f}^* = \sum_{i=1}^{C_M} \tilde{f}_i \circ \phi_i \times 1_{U_i} = \sum_{i=1}^{C_M} \sum_{j=1}^{N} \tilde{f}_{i,j} \circ \phi_i \times 1_{U_i}. \tag{21}
\]

Such an approximation has error

\[
\left\| \tilde{f}^* - f^* \right\|_{L^\infty} \leq C C_M c_0 N^{-s/d}. \tag{22}
\]

**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 \( \tilde{f}_{i,j} \circ \phi_i \times 1_{U_i} \) using CNNs. In particular, we show that CNNs can implement the cardinal B-spline \( \tilde{f}_{i,j} \), the linear projection \( \phi_i \), the indicator function \( 1_{U_i} \), and the multiplication operation.

- **Implement \( 1_{U_i} \) by CNNs.** Recall our construction of \( U_i \) in Step 1. For any \( x \in M \), we have \( 1_{U_i}(x) = 1 \) if \( d^2_i(x) = \| x - c_i \|_2^2 \leq \omega^2 \); otherwise \( 1_{U_i}(x) = 0 \).

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

\[
1_{U_i}(x) = 1_{[0,\omega^2]} \circ d^2_i \quad \text{for} \quad x \in M. \tag{22}
\]

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

\[
\|d^2_i - d^2_i\|_{L^\infty} \leq 4B^2 D\theta \tag{23}
\]
We also characterize the network sizes for realizing $\tilde{B}$ and $d^2_1$: The network for $\tilde{B}$ has $O(\log(\omega^2/\Delta))$ layers, 2 channels and all weight parameters bounded by $\max(2, \omega^2 - 4B^2D\delta)$; the network for $d^2_1$ 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 $\tilde{f}_{i,j} \circ \phi_i$ by CNNs. Since $\phi_i$ 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 $\tilde{\phi}^{\text{CNN}}_i$.

For $\tilde{f}_{i,j}$, Proposition 3 (see Appendix C.8) shows that for any $\delta \in (0,1)$ and $2 \leq K \leq d$, there exists a CNN $\tilde{f}_{i,j}^{\text{CNN}} \in \mathcal{F}^{\text{CNN}}(L, J, K, \kappa, \kappa)$ with $L = O \left( \frac{1}{\delta} \right), J = O(1), \kappa = O \left( \delta^{-(\log 2)(\frac{2d}{s} + \frac{1}{d})} \right)$ such that when setting $N = C_1 \delta^{-d/s}$, we have

$$\left\| \sum_{j=1}^{N} \tilde{f}_{i,j}^{\text{CNN}} - \tilde{f}_{i} \circ \phi_i \right\|_{L^\infty(\phi_i(U_i))} \leq \delta,$$

where $C_1$ is a constant depending on $s, p, q$ and $d$. The constant hidden in $O(\cdot)$ depends on $d, s, \frac{2d}{s}, p, q, c_0$. 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 $\hat{\times}$ with $L^\infty$ error:

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

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

**Step 4: Implement $\tilde{f}^\ast$ 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 $\tilde{f}_{i,j} \circ \phi_i \times \Pi_{U_i}$, as

$$\tilde{f}_{i,j} \equiv \hat{\times} \left( \tilde{f}_{i,j}^{\text{CNN}} \circ \phi_i^{\text{CNN}}, \tilde{B} \circ d^2_1 \right). \quad (27)$$

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

$$\tilde{f} = \sum_{i=1}^{C_M} \sum_{j=1}^{N} \tilde{f}_{i,j}. \quad (28)$$

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

$$\left\| \tilde{f} - f^* \right\|_{L^\infty} \leq \sum_{i=1}^{C_M} (A_{i,1} + A_{i,2} + A_{i,3})$$

with

$$A_{i,1} = \sum_{j=1}^{N} \left\| \hat{\times} \left( \tilde{f}_{i,j}^{\text{CNN}} \circ \phi_i^{\text{CNN}}, \tilde{B} \circ d^2_1 \right) - \left( \tilde{f}_{i,j}^{\text{CNN}} \circ \phi_i^{\text{CNN}} \right) \right\|_{L^\infty} \leq N \eta,$$

$$A_{i,2} = \left\| \left( \sum_{j=1}^{N} \left( \tilde{f}_{i,j}^{\text{CNN}} \circ \phi_i^{\text{CNN}} \right) \right) \times (\tilde{B} \circ d^2_1) \right\|_{L^\infty} \leq \delta,$$

$$A_{i,3} = \left\| f_i \times (\tilde{B} \circ d^2_1) - f_i \right\|_{L^\infty} \leq \frac{c(\pi + 1)}{\omega(1 - \omega/\tau)} \Delta,$$

where $\delta, \eta, \Delta$ and $\theta$ are defined in (25), (26), (24) and (23), respectively. For any $\varepsilon \in (0,1)$, with properly chosen $\delta, \eta, \Delta$ and $\theta$ as in (53) in Lemma 12, one has

$$\left\| \tilde{f} - f^* \right\|_{L^\infty} \leq \varepsilon. \quad (29)$$

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

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

$$\tilde{f} = \sum_{i=1}^{C_M} \sum_{j=1}^{N} \tilde{f}_{i,j}^{\text{CNN}}, \quad (30)$$

where $N$ is chosen of $O(\varepsilon^{-d/s})$ (see Proposition 3 and Lemma 12). Lemma 18 (see Appendix C.16) shows that $\tilde{f}^{\text{CNN}}$ can be realized by $\tilde{f} \in \mathcal{C}(M, L, J, K_1, K_2)$ with

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

$$J = O(D), \kappa_1 = O(1), \log \kappa_2 = O(\log^2(1/\varepsilon)).$$

## 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 $M$. 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 $M$. 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


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


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 $\Omega$, we denote $f \lesssim g$ if there is a constant $C$ such that $f(x) \leq Cg(x)$ for all $x \in \Omega$. Similarly, we denote $f \gtrsim g$ if there is a constant $C$ such that $f(x) \geq Cg(x)$ for all $x \in \Omega$. 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 $\{ (f_{1}^{i}, f_{2}^{i}) \}_{i=1}^{N}$ is called a $\delta$-bracketing of a function class $\mathcal{F}$ with respect to the norm $\| \cdot \|$ if for any $i$, $\| f_{1}^{i} - f_{2}^{i} \| \leq \delta$ and for any $f \in \mathcal{F}$, there exists a pair $(f_{1}^{i}, f_{2}^{i})$ such that $f_{1}^{i} \leq f \leq f_{2}^{i}$. The $\delta$-bracketing number is defined as the cardinality of the minimal $\delta$-bracketing set and is denoted by $\mathcal{N}_{B}(\delta, \mathcal{F}, \| \cdot \|)$. The $\delta$-bracketing entropy, 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 $\rho$. A $\delta$-cover of $\mathcal{F}$ is a set $\{ f_{1}^{i}, \ldots, f_{N}^{i} \} \subset \mathcal{F}$ such that for any $f \in \mathcal{F}$, there exists $f_{k}^{i}$ for some $k$ such that $\rho(f, f_{k}^{i}) \leq \delta$. The $\delta$-covering number of $\mathcal{F}$ is defined as

$$\mathcal{N}(\delta, \mathcal{F}, \rho) = \inf \{ N : \text{there exists a $\delta$-cover $\{ f_{1}^{i}, \ldots, f_{N}^{i} \}$ of $\mathcal{F}$} \}. $$

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 $\phi$ 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) $\phi$ 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 $a_{n} = O(n^{-\alpha_{0}})$ for some $\alpha_{0} > 0$, 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 $n \in \mathbb{N}$,

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

for some constant $C_{2} > 0$ only depending on $\phi$ and $\eta$.

(A5) For a positive constant $C_{3} > 0$, there exists a sequence $\{ \delta_{n} \}_{n \in \mathbb{N}}$ such that

$$\mathcal{H}_{B} (\delta_{n}, \mathcal{F}_{n}, \| \cdot \|_{L^{2}}) \leq C_{3}e^{-F_{n}^{2}}n \left( \frac{\delta_{n}}{F_{n}} \right)^{2-\nu}$$

for $\{ \mathcal{F}_{n} \}_{n \in \mathbb{N}}$ in (A2), $\{ F_{n} \}_{n \in \mathbb{N}}$ in (A3) and $\nu$ in (A4).
Let $c^2 \sim \max(a_n, \delta_n)$. Then the empirical $\hat{\phi}$-risk minimizer $\hat{f}_{\phi,n}$ over $\mathcal{F}_n$ satisfies
\[ P\left( \mathcal{E}_n(\hat{f}_{\phi,n}, f_\phi^*) \geq \epsilon_n \right) \leq C_5 \exp\left(-C_4 e^{-F_n \epsilon_n \left( c^2_n / (F_n) \right)^{2-\nu}}\right) \] (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 $\phi$ to be Lipschitz. This condition is satisfied in Theorem 2 since $\phi$ is the logistic loss. (A2) is a condition on the bias of $\hat{f}_{\phi,n}$. Take $n$ as the number of samples. (A2) requires the bias to decrease in the order of $O(n^{-\infty})$ for some $c_0$. (A3) requires all functions in the class $\mathcal{F}_n$ to be bounded. (A4) and (A5) are conditions relate to the variance of $\hat{f}_{\phi,n}$. Condition (A4) for logistic loss can be verified using the following lemma:

**Lemma 1** (Lemma 6.1 in Park (2009)). Let $\phi$ 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
\[ \mathbb{E} \left[ \phi(yf) - \phi(yf_\phi^*) \right]^2 \leq C e^F \mathcal{E}_\phi(f, f_\phi^*) \]
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
\[ \mathcal{F}_n = \mathcal{C}^{(n)}, a_n = n^{-\frac{s-1}{2s+4(\max\{d, p\})}}, \log^2 n, F_n = \frac{s}{2s + 2(s \vee d)} \log n, \delta_n = n^{-\frac{s}{2s+2(s \vee d)n}} \log^4 n \] (32)
where $\mathcal{C}^{(n)}$ is the network architecture in Theorem 2.

We first prove the probability bound of $\mathcal{E}_\phi(\hat{f}_{\phi,n}, f_\phi^*)$ by checking conditions (A1)-(A5) in Proposition 2. Note that $\phi$ 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 $f^* = \log \frac{n}{1-\eta}$. As $\eta$ goes to 0 (resp. 1), $f^*$ goes to $\infty$ (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
\[ f_{\phi,n}^* = \begin{cases} F_n, & \text{if } f_{\phi,n}^* > F_n, \\ f_{\phi,n}^*, & \text{if } -F_n \leq f_{\phi,n}^* \leq F_n, \\ -F_n, & \text{if } f_{\phi,n}^* < -F_n. \end{cases} \] (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 $\epsilon \in (0, 1)$ and any $K \leq D$, there exists a ConvResNet architecture
\[ \mathcal{C}(\bar{f}_\phi) = \{ f \mid \hat{g}_2 \circ \tilde{h} \circ \tilde{g}_1 \circ \tilde{\eta} \text{ where } \tilde{\eta} \in \mathcal{C}^{\text{Conv}}(M_1, L_1, J_1, K, \kappa_1), \tilde{g}_1 \in \mathcal{C}^{\text{Conv}}(1, 4, 8, 1, \kappa_2), \tilde{h} \in \mathcal{C}^{\text{Conv}}(M_2, L_2, J_2, 1, \kappa_1), \bar{g}_2 \in \mathcal{C}(1, 3, 8, 1, \kappa_3, 1, R) \} \]
with
\[ M_1 = O(\epsilon^{-d/s}), \quad M_2 = O\left(e^{-F_n \epsilon^{-1}}\right), \quad L_1 = O(\log(1/\epsilon) + D + D), \quad L_2 = O(\log(1/\epsilon)), \]
\[ J_1 = O(D), \quad J_2 = O(1), \quad \kappa_1 = O(1), \quad \log \kappa_2 = O(\log(1/\epsilon)), \quad \kappa_3 = O(\log(F_n / \epsilon) / F_n), \quad R = F_n, \]
such that for any $\eta \in B_{p,q}(\mathcal{M})$ with $||\eta||_{B_{p,q}(\mathcal{M})} \leq c_0$ for some constant $c_0$, and $f_{\phi,n}^*$ be defined as in (33), there exists $\hat{f}_{\phi,n} \in \mathcal{C}(\bar{f}_\phi)$ with
\[ ||\hat{f}_{\phi,n} - f_{\phi,n}^*||_{L^\infty} \leq 4e^{F_n \epsilon}. \]
Moreover, the covering number of $\mathcal{C}(\bar{f}_\phi)$ is bounded by
\[ \mathcal{N}(\delta, \mathcal{C}(\bar{f}_\phi), ||\cdot||_{L^\infty}) = O\left(D_3\epsilon^{-\left(\frac{d+1}{2}\right)}(\log(1/\epsilon) + \log(1/\epsilon) + \log D + F_n + \log(1/\delta)) \right). \]
The constant hidden in $O(\cdot)$ depends on $d, s, \frac{2d}{s-2}, p, q, c_0, \tau$ and the surface area of $\mathcal{M}$. 


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

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

\[
A_n = \{x \in \mathcal{M} : |f_{\phi}^*| \leq F_n\}, \quad A_n^C = \{x \in \mathcal{M} : |f_{\phi}^*| > F_n\}.
\]

We have

\[
\mathcal{E}_\phi(\tilde{f}_{\phi,n}, f_{\phi}^*) = \int_{A_n} \eta(\phi(\tilde{f}_{\phi,n}) - \phi(f_{\phi}^*)) + (1 - \eta) \left(\phi(-\tilde{f}_{\phi,n}) - \phi(-f_{\phi}^*)\right) \mu(dx) \\
= \int_{A_n} \eta(\phi(\tilde{f}_{\phi,n}) - \phi(f_{\phi}^*)) + (1 - \eta) \left(\phi(-\tilde{f}_{\phi,n}) - \phi(-f_{\phi}^*)\right) \mu(dx) \\
+ \int_{A_n^C} \eta(\phi(\tilde{f}_{\phi,n}) - \phi(f_{\phi}^*)) + (1 - \eta) \left(\phi(-\tilde{f}_{\phi,n}) - \phi(-f_{\phi}^*)\right) \mu(dx),
\]

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

\[
T_1 \leq \int_{A_n} \eta|\phi(\tilde{f}_{\phi,n}) - \phi(f_{\phi,n}^*)| + (1 - \eta)|\phi(-\tilde{f}_{\phi,n}) - \phi(-f_{\phi,n}^*)| \mu(dx) \\
\leq \|\phi(\tilde{f}_{\phi,n}) - \phi(f_{\phi,n}^*)\|_{L^\infty} \leq 4e^{F_n}\varepsilon_1.
\]

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 \(0 < p, q \leq \infty\), \(0 < s < \infty\), \(s \geq d/p + 1\), \(\eta \in B^{\phi,q}_{p,q}(M)\) with \(\|\eta\|_{B^{\phi,q}_{p,q}(M)} \leq c_0\) for some constant \(c_0\). Let \(T_2\) be defined as in (34). If \(4e^{F_n}\varepsilon_1 < 1\), the following bound holds:

\[
T_2 \leq 8F_n e^{-F_n}.
\]

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

\[
\mathcal{E}_\phi(\tilde{f}_{\phi,n}, f_{\phi}^*) \leq T_1 + T_2 \leq 4e^{F_n}\varepsilon_1 + 8F_n e^{-F_n}.
\]

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

\[
\mathcal{E}_\phi(\tilde{f}_{\phi,n}, f_{\phi}^*) \leq C_0 n^{-\frac{2d}{d+2(s\vee d)}} \log^2 n
\]

and \(C(F_n) = C(n)\), where \(C(n)\) is defined in Theorem 2. Here \(C_0\) is a constant depending on \(s\) and \(d\). Thus (A2) and (A3) are satisfied with \(a_n = n^{-\frac{2d}{d+2(s\vee d)}} \log^2 n\), \(F_n = \frac{s}{2s+2(s\vee d)} \log n\).

**Verification of (A5).** For (A5), we only need to check that \(\log \mathcal{N}(\delta_n, C(n), \|\cdot\|_{L^\infty}) \leq C_3 n^{F_n^{-1}} \delta_n\) for some constant \(C_3\).

According to Lemma 2 with our choices of \(\varepsilon_1\) and \(F_n\), we have

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

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

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

for some \(C_3\) depending on \(d, D^3 \log D, s, \frac{d}{2p-q}, p, q, c_0, \tau\) and the surface area of \(M\). Therefore (A5) is satisfied.

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

\[
P\left(\mathcal{E}_\phi(\tilde{f}_{\phi,n}, f_{\phi}^*) \geq \epsilon_n\right) \leq C_5 \exp\left(-\frac{C_4 2s + 2(s \vee d) n^{\frac{2(s\vee d)}{d+2(s\vee d)}} \epsilon_n^2}{s \log n}\right)
\]
with \( \epsilon_n^2 \approx \max(a_n, \delta_n) = C_7n^{-\frac{d + 2(\nu + 2)}{3(\nu + 2)}} \log^4 n \) and \( \delta_n \) being the minimizer of the empirical risk in (11). Here \( C_7 \) is a constant depending on \( d, D, \log D, s, \frac{2d}{sp - d}, p, q, c_0, \tau \) and the surface area of \( M \).

Note that (A5) is also satisfied for any \( \delta_n \geq C_7n^{-\frac{d + 2(\nu + 2)}{3(\nu + 2)}} \log^4 n \). Thus

\[
P\left( E_\phi(f_{\phi,n}, f_\phi^+) \geq t \right) \leq C_5 \exp \left( -C_4 \frac{2s + 2(s \vee d) n^{s + 2(\nu + 2)} t}{s} \log n \right) \tag{39}
\]

for any \( t \geq C_7n^{-\frac{d + 2(\nu + 2)}{3(\nu + 2)}} \log^4 n \). Integrating (39), we estimate the expected excess risk as

\[
E( E_\phi(f_{\phi,n}, f_\phi^+) ) = \int_M E_\phi(f_{\phi,n}, f_\phi^+) \mu(dx)
\]

\[
\leq C_7 \int M E_\phi(f_{\phi,n}, f_\phi^+) \leq C_7n^{-\frac{d + 2(\nu + 2)}{3(\nu + 2)}} \log^4 n
\]

\[
+ C_5 \int_{C_7n^{-\frac{d + 2(\nu + 2)}{3(\nu + 2)}} \log^4 n}^{\infty} \exp \left( -C_4 \frac{2s + 2(s \vee d) n^{s + 2(\nu + 2)} t}{s} \log n \right) dt
\]

\[
\leq C_8n^{-\frac{d + 2(\nu + 2)}{3(\nu + 2)}} \log^4 n \tag{40}
\]

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 \( M \).

B. Convolutional neural networks and multi-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

\[
f(x) = W \cdot \text{Conv}_{W,B}(x) \tag{41}
\]

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

\[
\mathcal{F}_{\text{CNN}}(L, J, K, \kappa_1, \kappa_2) = \{ f | f(x) \text{ in the form (41) with } L \text{ layers.} \}
\]

Each convolutional layer has filter size bounded by \( K \).

The number of channels of each layer is bounded by \( J \).

\[
\max_i \|W^{(i)}\|_\infty \vee \|B^{(i)}\|_\infty \leq \kappa_1, \|W\|_\infty \leq \kappa_2 \}. \tag{42}
\]

For MLP, we consider the following form

\[
f(x) = W_L \cdot \text{ReLU}(W_{L-1} \cdots \text{ReLU}(W_1 x + b_1) \cdots + b_{L-1}) + b_L, \tag{43}
\]

where \( W_1, \ldots, W_L, b_1, \ldots, b_L \) are weight matrices and bias vectors of proper sizes, respectively. The class of MLP is defined as

\[
\mathcal{F}_{\text{MLP}}(L, J, \kappa) = \{ f | f(x) \text{ in the form (43) with } L \text{-layers and width bounded by } J. \}
\]

\[
\|W_i\|_{\infty,\infty} \leq \kappa, \|b_i\|_\infty \leq \kappa \text{ for } i = 1, \ldots, L. \tag{44}
\]

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

\[
\mathcal{F}_{\text{MLP}}(L, J, \kappa, R) = \{ f | f(x) \in \mathcal{F}_{\text{MLP}}(L, J, \kappa) \text{ and } \|f\|_\infty \leq R \}.
\]

In some cases we do not need the constraint on the output, we denote such MLP class as \( \mathcal{F}_{\text{MLP}}(L, J, \kappa) \).

C. Lemmas and proofs in Section 4

C.1. Lemma 4 and its proof

**Lemma 4.** Define \( f_i, \phi_i \) as in (17). We extend \( f_i \circ \phi_i^{-1} \) by 0 on \([0, 1]^d \setminus \phi_i(U_i)\) and denote the extended function by \( f_i \circ \phi_i^{-1} \mid_{[0,1]^d} \). Under Assumption 3, we have \( f_i \circ \phi_i^{-1} \mid_{[0,1]^d} \in B^s_{p,q}([0,1]^d) \) with

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

where \( C \) is a constant depending on \( s, p, q, d, \) and \( a \).
To prove Lemma 4, we first give an equivalent definition of Besov functions:

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

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

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

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

**Proof of Lemma 4.** Since $f \in B_{p,q}^s(\mathcal{M})$, according to Definition 9, $f \circ \phi_i^{-1} \in B_{p,q}^s(\mathbb{R}^d)$ in the sense of extending $f \circ \phi_i^{-1}$ by zero on $\mathbb{R}^d \setminus \phi_i(U_i)$, see Triebel (1983, Section 3.2.3) for details. From Assumption 3, $\| f \circ \phi_i^{-1} \|_{B_{p,q}^s(\mathbb{R}^d)} < c_0$. We next restrict $f \circ \phi_i^{-1}$ 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

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

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

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

\[ \square \]

**C.2. Cardinal B-splines**

We give a brief introduction of cardinal B-splines.

**Definition 13 (Cardinal B-spline).** Let $\psi(x) = 1_{[0,1]}(x)$ be the indicator function of $[0,1]$. The cardinal B-spline of order $m$ is defined by taking $m + 1$-times convolution of $\psi$:

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

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

Note that $\psi_m$ 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 $k = (k_1, \ldots, k_d) \in \mathbb{N}^d$ and $j = (j_1, \ldots, j_d) \in \mathbb{N}^d$, we define the $d$ dimensional cardinal B-spline as $M_{k,j,m}^d(x) = \prod_{i=1}^d \psi_m(2^k x_i - j_i)$. When $k_1 = \ldots = k_d = k \in \mathbb{N}$, we denote $M_{k,j,m}^d(x) = \prod_{i=1}^d \psi_m(2^k x_i - j_i)$.

**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 $k \in \mathbb{N}, j \in J(k)$ be

$$\| \{\alpha_{k,j}\} \|_{p,q} = \left( \sum_{k \in \mathbb{N}} \left( \sum_{j \in J(k)} |\alpha_{k,j}|^p \right)^{1/p} \right)^{1/q}.$$

The following lemma, resulted from DeVore & Popov (1988); Dung (2011), gives an error bound for the approximation of functions in $B_{p,q}^s([0,1]^d)$ by cardinal B-splines.
Specifically, any $\tilde{C}$ for some $k$ according to (45) and Lemma 5, we have

$$f_N(x) = \sum_{k=0}^{H} \sum_{j \in J(k)} \alpha_{k,j} m_{k,j,m}(x) + \sum_{k=K+1}^{H^*} \sum_{i=1}^{n_k} \alpha_{k,i} m_{k,i,m}(x),$$

(46)

where $\{\tilde{j}\}_{i=1}^{n_k} \subseteq J(k), H = [c_1 \log(N)/d], H^* = [\nu - 1 \log(\lambda N)] + H + 1, n_k = \lceil \lambda N^2 - \nu(k-H) \rceil$ for $k = H + 1, \ldots, H^*, u = d(1/p - 1/r_+) + \nu = (s-u)/(2u)$. 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 $\{\alpha_{k,j}\}$ such that

$$\|\{\alpha_{k,j}\}\|_{\mathcal{L}^p} \leq C_1 \|f\|_{B^s_p([0,1]^d)}$$

for some constant $C_1$.

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

$$|\alpha_{k,j}^{(i)}| \leq C_0 N^{(\log 2)(\nu^{-1} + c_1 d^{-1})}(d/p-s)_+$$

(47)

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

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

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

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

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

(48)

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

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

(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 $\tilde{x}(-,-)$, such that

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

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 $F_{MLP}(L, J, \kappa)$ can be realized by a CNN architecture $F_{CNN}(L', J', K', \kappa_1', \kappa_2')$ with

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

Specifically, any $\tilde{f}_{MLP} \in F_{MLP}(L, J, \kappa)$ can be realized by a CNN $\tilde{f}_{CNN} \in F_{CNN}(L', J', K', \kappa_1', \kappa_2')$. Furthermore, the weight matrix in the fully connected layer of $\tilde{f}_{CNN}$ has nonzero entries only in the first row.
C.6. Lemma 9 and its proof

**Lemma 9.** Let \( d_1^2 \) and \( \mathbb{I}_{[0, \omega^2]} \) be defined as in (22). For any \( \theta \in (0, 1) \) and \( \Delta \geq 8B^2D\theta \), there exists a CNN \( \tilde{d}_i^2 \) approximating \( d_i^2 \) such that
\[
\| \tilde{d}_i^2 - d_i^2 \|_{L^\infty} \leq 4B^2D\theta,
\]
and a CNN \( \tilde{I}_\Delta \) approximating \( \mathbb{I}_{[0, \omega^2]} \) with
\[
\tilde{I}_\Delta(x) = \begin{cases} 
1, & \text{if } a \leq (1 - 2^{-k})(\omega^2 - 4B^2D\theta), \\
0, & \text{if } a \geq \omega^2 - 4B^2D\theta, \\
2^k((\omega^2 - 4B^2D\theta)^{-1}a - 1), & \text{otherwise}. 
\end{cases}
\]
for \( x \in \mathcal{M} \). The CNN for \( \tilde{d}_i^2 \) has \( O(\log(1/\theta)) \) layers, 6D channels and all weights parameters are bounded by \( 4B^2 \). The CNN for \( \tilde{I}_\Delta \) 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 \( x \in \mathcal{M} \), \( \tilde{I}_\Delta \circ \tilde{d}_i^2(x) \) gives an approximation of \( \mathbb{I}_{U_i} \) satisfying
\[
\tilde{I}_\Delta \circ \tilde{d}_i^2(x) = \begin{cases} 
1, & \text{if } x \in U_i \text{ and } d_i^2(x) \leq \omega^2 - \Delta; \\
0, & \text{if } x \notin U_i; \\
\text{between 0 and 1}, & \text{otherwise}. 
\end{cases}
\]

**Proof.** We first show the existence of \( \tilde{d}_i^2 \). Here \( \tilde{d}_i^2(x) \) is the sum of \( D \) univariate quadratic functions. Each quadratic function can be approximated by a multi-layer perceptron (MLP, see Appendix B for the definition) according to Lemma 7. Let \( h(x) \) be an MLP approximation of \( x^2 \) for \( x \in [0, 1] \) with error \( \theta \), i.e., \( \| h(x) - x^2 \|_\infty \leq \theta \). We define
\[
d_i^2(x) = 4B^2 \sum_{j=1}^{D} \tilde{h} \left( \left| \frac{x_j - c_{i,j}}{2B} \right| \right)
\]
as an approximation of \( d_i^2(x) \), which gives rise to the approximation error \( \| \tilde{d}_i^2 - d_i^2 \|_\infty \leq 4B^2D\theta \). Such a MLP has \( O(\log 1/\theta) \) layers, and width \( 6D \). All weight parameters are bounded by \( 4B^2 \). According to Lemma 8, \( d_i^2 \) can be realized by a CNN, which is denoted by \( \tilde{d}_i^2 \). Such a CNN has \( O(\log 1/\theta) \) layers, 6D channels. All weight parameters are bounded by \( 4B^2 \).

To show the existence of \( \tilde{I}_\Delta \), we use the following function to approximate \( \mathbb{I}_{[0, \omega^2]} \):
\[
\tilde{I}_\Delta(a) = \begin{cases} 
1, & \text{if } a \leq \omega^2 - 4B^2D\theta, \\
0, & \text{if } a \geq \omega^2 - 4B^2Dv, \\
\frac{-1}{2B^2D\theta}a + \frac{\omega^2 - 4B^2D\theta}{2B^2D\theta} - \frac{\omega^2 - 4B^2D\theta}{2B^2D\theta}, & \text{otherwise}. 
\end{cases}
\]
We implement \( \tilde{I}_\Delta(a) \) based on the basic step function defined as: \( g(a) = 2\text{ReLU}(a - 0.5(\omega^2 - 4B^2D\theta)) - 2\text{ReLU}(a - \omega^2 + 4B^2D\theta) \).

Define
\[
g_k(a) = g \circ \cdots \circ g(a)
\]
\[
= \begin{cases} 
0, & \text{if } a \leq (1 - 2^{-k})(\omega^2 - 4B^2D\theta), \\
\omega^2 - 4B^2D\theta, & \text{if } a \geq \omega^2 - 4B^2D\theta, \\
2^k(\omega^2 - 4B^2D\theta) + \omega^2 - 4B^2D\theta, & \text{otherwise}.
\end{cases}
\]
We set \( \tilde{I}_\Delta = 1 - (\omega^2 - 4B^2D\theta)^{-1}g_k \), 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 \( k = \lceil \log(\omega^2/\Delta) \rceil \).

\[\square\]

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 \( j \) be any element in \( \mathbb{N}^d \). There exists a constant \( C \) depending only on \( d \) and \( m \) such that, for any \( \varepsilon \in (0, 1) \) and \( 2 \leq k \leq d \), there exists a CNN \( \tilde{M}_{k,j,m}^d \in \mathcal{F}^{\text{CNN}}(L, J, K, \kappa, \kappa) \) with 
\[
L = 3 + 2[\log_2\left(\frac{3\varepsilon}{C\varepsilon}\right) + 5][\log_2(d \vee m)] + d, J = 2dm(m + 2) + 8d \quad \text{and} \quad \kappa = 2(m + 1)^m \vee 2^k \quad \text{such that for any} \ k \in \mathbb{N} \ \text{and} \ j \in \mathbb{N}^d, 
\]
\[
\|M_{k,j,m}^d - \tilde{M}_{k,j,m}^d\|_{L^\infty([0,1]^d)} \leq \varepsilon, 
\]
and \( \tilde{M}_{k,j,m}^d(x) = 0 \) for all \( x \notin 2^{-k}[0, m + 1]^d \).

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 \( j \) 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 \( \tilde{M}_{k,j,m}^d \in \mathcal{F}^{\text{MLP}}(L, J', \kappa', 1) \) with \( L' = 3 + 2[\log_2\left(\frac{3\varepsilon}{C\varepsilon}\right) + 5][\log_2(d \vee m)], J' = 6dm(m + 2) + 2d \quad \text{and} \quad \kappa' = 2(m + 1)^m \vee 2^k \) such that for any \( k \in \mathbb{N} \) and \( j \in \mathbb{N}^d, \)
\[
\|M_{k,j,m}^d - \tilde{M}_{k,j,m}^d\|_{L^\infty([0,1]^d)} \leq \varepsilon, 
\]
and \( \tilde{M}_{k,j,m}^d(x) = 0 \) for all \( x \notin 2^{-k}[0, m + 1]^d \).

Proof of Lemma 10. According to Lemma 11, there exists an MLP \( \tilde{M}_{k,j,m}^d \in \mathcal{F}^{\text{MLP}}(L', J', \kappa', 1) \) with \( L' = 3 + 2[\log_2\left(\frac{3\varepsilon}{C\varepsilon}\right) + 5][\log_2(d \vee m)], J' = 6dm(m + 2) + 2d \quad \text{and} \quad \kappa' = 2(m + 1)^m \vee 2^k \) such that
\[
\|M_{k,j,m}^d - \tilde{M}_{k,j,m}^d\|_{L^\infty([0,1]^d)} \leq \varepsilon, 
\]
and \( \tilde{M}_{k,j,m}^d(x) = 0 \) for all \( x \notin 2^{-k}[0, m + 1]^d \).

Lemma 8 shows that such an MLP can be realized by a CNN \( \tilde{M}_{k,j,m}^d \in \mathcal{F}^{\text{CNN}}(L, J, K, \kappa, \kappa). \)

\[\square\]

C.8. Proposition 3 and its proof

Proposition 3 shows that if \( N \) and \( \varepsilon_1 \) are properly chosen, \( \sum_{j=1}^N \tilde{f}_{i,j}^{\text{CNN}} \) 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 \( \tilde{f}_{i,j}^{\text{CNN}} \) such that
\[
\left\| \sum_{j=1}^N \tilde{f}_{i,j}^{\text{CNN}} - f_i \circ \phi_i^{-1} \right\|_{L^\infty} \leq \delta, 
\]
where \( C_1 \) is a constant depending on \( s, p, q \) and \( d \).

\( \tilde{f}_{i,j}^{\text{CNN}} \) is a CNN approximation of \( \tilde{f}_{i,j} \) (defined in (19)) and is in \( \mathcal{F}^{\text{CNN}}(L, J, K, \kappa, \kappa) \) with
\[
L = O\left(\log(1/\delta)\right), J = 24d(s+1)(s+3) + 8d, \quad \kappa = O\left(\delta^{-(\log 2)(\frac{2d}{sp-q}+\varepsilon_1 d^{-1})}\right). 
\]

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

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

Note that \( \tilde{f}_{i,j} = \alpha_{k,j,m}^{(i)} \tilde{M}_{k,j,m}^d \) with some coefficient \( \alpha_{k,j,m}^{(i)} \) and index \( k, j, m \) where \( M_{k,j,m}^d \) is a \( d \)-dimensional cardinal B-spline. Lemma 10 shows that \( M_{k,j,m}^d \) can be approximated by a CNN \( \tilde{M}_{k,j,m}^d \) with arbitrary accuracy. Therefore \( \tilde{f}_{i,j} \) can be approximated by a CNN \( \tilde{f}_{i,j}^{\text{CNN}} \) with arbitrary accuracy. Assume \( \|\tilde{M}_{k,j,m}^d - M_{k,j,m}^d\|_{L^\infty} \leq \varepsilon_1 \) for some \( \varepsilon_1 \in (0, 1) \). Then \( \tilde{f}_{i,j}^{\text{CNN}} \in \mathcal{F}^{\text{CNN}}(L, J, K, \kappa, \kappa) \) with
\[
L = O\left(\log(1/\varepsilon_1)\right), J = 24dm(m + 2) + 8d, \quad \kappa = \max\left(\alpha_{k,j,m}^{(i)}; 2^k\right) = O\left(N^{(\log 2)(\nu^{-1}+\varepsilon_1 d^{-1})(1/\nu(d/p-s)+)}\right), 
\]
where the value of \( \kappa \) 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 \( \varepsilon_1 \), \( \sum_{j=1}^N \tilde{f}_{i,j}^{\text{CNN}} \) can approximate \( f_i \circ \phi_i^{-1} \) with arbitrary accuracy.
We decompose the error as
\[
\left\| \sum_{j=1}^{N} \bar{f}_{i,j}^{\text{CNN}} - f_i \circ \phi_i^{-1} \right\|_{L^\infty} \leq \left\| \bar{f}_i - f_i \circ \phi_i^{-1} \right\|_{L^\infty} + \sum_{j=1}^{N} \left\| \bar{f}_{i,j}^{\text{CNN}} - \bar{f}_i \right\|_{L^\infty},
\]
where \(\bar{f}_i\) is defined in (19). We next derive an error bound for each term.

Let \(m\) be the order of the Cardinal B-spline basis. Set \(m = \lceil s \rceil + 1\). According to (20) and Lemma 5,
\[
\left\| \bar{f}_i - f_i \circ \phi_i^{-1} \right\|_{L^\infty} \leq C_0 N^{-s/d}, \quad \left\| \alpha_{k,j}^{(1)} \right\|_{L^p} \leq C_1 \| f \|_{L_p^p},
\]
for some constant \(C\) depending on \(s, p, q\) and \(d\), and some universal constant \(C_1\) with \(\{k\}_{k=1}^{n_k} \subseteq J(k)\), \(H = \lceil c_1 \log(N)/d \rceil\), \(H^* = \lceil \nu^{-1} \log(\lambda N) \rceil + H + 1\), \(n_k = \lceil \lambda N 2^{-\nu(k-H)} \rceil\) 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 \(N = \left\lceil \left( \frac{\delta}{2C_{c_0}} \right)^{-d/s} \right\rceil\), we have \(\left\| \bar{f}_i - f_i \circ \phi_i^{-1} \right\|_{\infty} \leq \delta/2\).

Next we consider the second term in (51). For any \(x \in [0,1]^d\), we have
\[
\left\| \sum_{j=1}^{N} \bar{f}_{i,j}^{\text{CNN}}(x) - \bar{f}_i(x) \right\| \leq \sum_{(k,j) \in S_N} \left| \alpha_{k,j}^{(1)} \right| \left\| M_{k,j,m}^d(x) - \bar{M}_{k,j,m}^d(x) \right\|
\leq \varepsilon_1 \sum_{(k,j) \in S_N} \left| \alpha_{k,j}^{(1)} \right| \left| M_{k,j,m}^d(x) \right|
\leq \varepsilon_1 (m+1)^d (1 + \log(\lambda N))^{-1} + c_1 \log(N)/d + 3 \left( \varepsilon_1 \right)^{(3 \lambda N)^{-1}} \left( \frac{d}{d+1} \right) \left( \frac{2}{d+1} \right) + \delta_1
\leq C_0 \log(N) \left( \nu^{-1} + c_1 \right)^{-1} \left( \frac{d}{d+1} \right) + \delta_1
\leq C_0 \log(N) \left( \frac{2}{\varepsilon} \right) \left( \frac{\delta}{2} \right) \left( \frac{2}{d+1} \right) + \delta_1
\]
with \(C_2\) being some constant depending on \(m, d, s, p, q, \nu^{-1}\). In the second inequality, \(\left| M_{k,j,m}^d(x) \right| \neq 0\) if \(M_{k,j,m}^d(x) \neq 0\) 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 \(x\) and \(2^{k(s-d/p)} \left| \alpha_{k,j}^{(1)} \right| \leq \left\| \alpha_{k,j}^{(1)} \right\|_{L^p} \leq C_1 \| f \|_{L_p^p}. In the fourth inequality we use
\(H^* = \lceil \log(\lambda N) \rceil + H + 1\) and \(H = \lceil c_1 \log(N)/d \rceil\), and the last inequality follows from \(N = \left\lceil \left( \frac{\delta}{2C_{c_0}} \right)^{-d/s} \right\rceil\). Setting \(\varepsilon_1 = \frac{1}{c_2 \log(2)/\delta} \left( \frac{\delta}{2} \right) \left( \frac{d}{d+1} \right) \left( \frac{2}{d+1} \right) + \delta_1\) proves the error bound.

Under Assumption 3, \(s \geq d/p + 1\). Therefore \(\left( \frac{d}{d+1} \right) > 0\) 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 \(\eta\) be the approximation error of the multiplication operator \(\tilde{x}(\cdot, \cdot)\). \(\delta\) be defined as in Proposition 3, \(\Delta\) and \(b\) be defined as in Lemma 9. Assume \(N\) is chosen according to Proposition 3. For any \(i = 1, \ldots, C_M\), we have \(\|\bar{f} - f^*\|_{L^\infty} \leq \sum_{i=1}^{C_M} (A_{i,1} + A_{i,2} + A_{i,3})\) with
\[
A_{i,1} = \sum_{j=1}^{N} \left\| \tilde{x} (\bar{f}_{i,j}^{\text{CNN}} \circ \phi_i^{\text{CNN}}, \bar{I}_\Delta \circ \bar{d}_j^2) - \bar{f}_{i,j}^{\text{CNN}} \circ \phi_i^{\text{CNN}} \times (\bar{I}_\Delta \circ \bar{d}_j^2) \right\|_{L^\infty} \leq C\delta^{-d/s} \eta,
\]
\[
A_{i,2} = \left\| \sum_{j=1}^{N} (\bar{f}_{i,j}^{\text{CNN}} \circ \phi_i^{\text{CNN}}) \times (\bar{I}_\Delta \circ \bar{d}_j^2) - f_i \times (\bar{I}_\Delta \circ \bar{d}_j^2) \right\|_{L^\infty} \leq \delta,
\]
\[
A_{i,3} = \left\| f_i \times (\bar{I}_\Delta \circ \bar{d}_j^2) - f_i \times 1_{U_i} \right\|_{L^\infty} \leq \frac{c(\pi + 1)}{\omega(1 - \omega/\tau)} \Delta.
\]

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

$$
\delta = \frac{\varepsilon}{3C_M}, \quad \eta = \frac{1}{C} \left( \frac{\varepsilon}{3C_M} \right)^{\frac{1}{\beta + 1}}, \quad \Delta = \frac{\omega(1 - \omega/\tau)}{3\varepsilon(\pi + 1)C_M}, \quad \theta = \frac{\Delta}{16B^2 D}
$$

(53)
gives rise to

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

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 $\tilde{x}$:

$$
A_{i,1} = \sum_{j=1}^{N} \left\| \tilde{x}(f_{i,j}^{\text{CNN}} \circ \phi_i^{\text{CNN}}, \tilde{I}_\Delta \circ \tilde{d}_i^2) - f_{i,j}^{\text{CNN}} \circ \phi_i^{\text{CNN}} \times (I_\Delta \circ d_i^2) \right\|_{L^\infty} \leq N \eta \leq C\delta^{\frac{d}{d/s}}\eta,
$$

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, $A_{i,2} \leq \delta$.

$A_{i,3}$ measures the error from CNN approximation of the chart determination function. The bound of $A_{i,3}$ can be derived using Chen et al. (2019a, Proof of Lemma 4.5) since $f_1 \circ \phi_i^{-1}$ is a Lipschitz function and its domain is in $[0, 1]^d$. □

**C.10. CNN size quantification of $f_{i,j}$**

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

- $d_i^2$ has $O(\log(1/\varepsilon) + D + \log D)$ layers, 6D channels and all weights parameters are bounded by $4B^2$.
- $I_\Delta$ has $O(\log(1/\varepsilon))$ layers with 2 channels. All weights are bounded by $\max(2, \omega^2)$.
- $\tilde{x}$ has $O(\log 1/\varepsilon)$ layers with 6 channels. All weights are of $\max(c_0^2, 1)$.
- $f_{i,j}^{\text{CNN}}$ has $O(\log 1/\varepsilon)$ layers with $[24d(s + 1)(s + 3) + 8d]$ channels. All weights are in the order of $O\left(\varepsilon^{-\left(\frac{1}{2} + c_1d^{-1}\right)}\right)$ where $c_1$ is defined in Lemma 5.
- $\phi_i^{\text{CNN}}$ 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 $M$. 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 $F^{\text{CNN}}(L_1, J_1, K_1, \kappa_1, \kappa_1)$ be a CNN architecture from $\mathbb{R}^D \rightarrow \mathbb{R}$ and $F^{\text{CNN}}(L_2, J_2, K_2, \kappa_2, \kappa_2)$ be a CNN architecture from $\mathbb{R} \rightarrow \mathbb{R}$. Assume the weight matrix in the fully connected layer of $F^{\text{CNN}}(L_1, J_1, K_1, \kappa_1)$ and $F^{\text{CNN}}(L_2, J_2, K_2, \kappa_2, \kappa_2)$ has nonzero entries only in the first row. Then there exists a CNN architecture $F^{\text{CNN}}(L, J, K, \kappa, \kappa)$ from $\mathbb{R}^D \rightarrow \mathbb{R}$ with

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

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

In Lemma 13 and the following lemmas, the subscript of $F^{\text{CNN}}$ 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 convolutional layers to replace and realize this fully connected layer.

Denote $f_1$ and $f_2$ by

$$
f_1(x) = W_1 \cdot \text{Conv}_{W_1, b_1}(x) \quad \text{and} \quad f_2(x) = W_2 \cdot \text{Conv}_{W_2, b_2}(x).
$$
where $\mathcal{W}_1 = \left\{ W_1^{(i)} \right\}_{i=1}^{L_1}, B_1 = \left\{ B_1^{(i)} \right\}_{i=1}^{L_1}, \mathcal{W}_2 = \left\{ W_2^{(i)} \right\}_{i=1}^{L_2}, B_2 = \left\{ B_2^{(i)} \right\}_{i=1}^{L_2}$ are sets of filters and biases and $\text{Conv(W}_1, B_1, \text{Conv(W}_2, B_2)$ are defined in (3). In the rest of this proof, we will choose proper weight parameters in $\mathcal{W}, \mathcal{B}$ and $\mathcal{W}$ such that $f(x) \in \mathcal{F}^{\text{CNN}}(L, J, K, \kappa)$ is in the form of

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

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

For $1 \leq l \leq L_1 - 1$, we set $W_l^{(i)} = W_1^{(i)}, B_l^{(i)} = B_1^{(i)}$.

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

$\mathcal{W}_1^{(L_1)} = (W_1^{(1)})_{1,1}, \mathcal{W}_2^{(L_1)} = -(W_1^{(1)})_{1,1}$

and $B_1^{(L_1)} = 0$. Here $\mathcal{W}_2^{(L_1)} \in \mathbb{R}^{2 \times 1 \times M}$ 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

$$\left[ (f_1(x))_+ \quad (f_1(x))_- \right]$$

where * 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

$\mathcal{W}_1^{(L_1+1)} = \left[ (W_2^{(1)})_{i,1} \quad -(W_2^{(1)})_{i,1} \right]_1, B_1^{L+1} = \left[ B_2^{(1)} \right]_1$

where $i$ varies from 1 to the number of output channels of $W_2^{(1)}$. Here $\mathcal{W}_2^{(L_1+1)}$ is a size-one filter whose number of output channels is the same as that of $W_2^{(1)}$.

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

$\mathcal{W}_1^{(l)} = W_2^{(l-L_1)}, B_1^{l} = \left[ B_2^{(l-L_1)} \right]_1$, for $l = L_1 + 1, \ldots, L_1 + L_1 - 1$.

For $l = L$, we set

$W = \left[ W_2 \right]_1$.

With the above settings, the lemma is proved.

\[\square\]

C.12. Lemma 14 and its proof

Lemma 14 is used to prove Lemma 17.

**Lemma 14.** Let $f_1 \in \mathcal{F}^{\text{CNN}}(L_1, J_1, K_1, \kappa_1, \kappa_1)$ be a CNN from $\mathbb{R}^D \rightarrow \mathbb{R}$ and $f_2 \in \mathcal{F}^{\text{CNN}}(L_2, J_2, K_2, \kappa_2, \kappa_2)$ be a CNN from $\mathbb{R}^D \rightarrow \mathbb{R}$. 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

$\text{Conv(W}, B(x) = \left[ (f_1(x))_+ \quad (f_1(x))_- \quad (f_2(x))_+ \quad (f_2(x))_- \right] \in \mathbb{R}^{D \times 4}$.

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 $L_1 > L_2$.

Denote $f_1$ and $f_2$ by

$f_1(x) = W_1 \cdot \text{Conv(W}_1, B_1(x)$ and $f_2(x) = W_2 \cdot \text{Conv(W}_2, B_2(x)$,

where $\mathcal{W}_1 = \left\{ W_1^{(i)} \right\}_{i=1}^{L_1}, B_1 = \left\{ B_1^{(i)} \right\}_{i=1}^{L_1}, \mathcal{W}_2 = \left\{ W_2^{(i)} \right\}_{i=1}^{L_2}, B_2 = \left\{ B_2^{(i)} \right\}_{i=1}^{L_2}$, are sets of filters and biases and $\text{Conv(W}_1, B_1, \text{Conv(W}_2, B_2)$ are defined in (3). In the rest of this proof, We will choose proper weight parameters in $\mathcal{W}, \mathcal{B}$ such that

$\text{Conv(W}, B(x) = \left[ (f_1(x))_+ \quad (f_1(x))_- \quad (f_2(x))_+ \quad (f_2(x))_- \right] \in \mathbb{R}^{D \times 4}$.
For any CNN architecture $F$, we assume all convolutional layers $f$ where

$$W_{i,:,::}^{(f)} = \begin{bmatrix} (W_{1}^{(f)})_{i,:,::} & 0 \end{bmatrix} \quad \text{for } i = 1, ..., J_1,$$

$$W_{i+1,:,::}^{(f)} = \begin{bmatrix} 0 & (W_{2}^{(f)})_{i-J_1,:,::} \end{bmatrix} \quad \text{for } i = J_1 + 1, ..., J_1 + J_2,$$

$$B^{(f)} = \begin{bmatrix} B_{1}^{(f)} & B_{2}^{(f)} \end{bmatrix}.$$

Each $W^{(f)}$ 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

$$W_{i,:,::}^{(f)} = \begin{bmatrix} (W_{1}^{(f)})_{i,:,::} & 0 \end{bmatrix} \quad \text{for } i = 1, ..., J_1,$$

such that $W^{(f)}$ has size $K_2$ filters.

For the $L_2$-th layer, we set

$$W_{i,:,::}^{(L_2)} = \begin{bmatrix} (W_{1}^{(L_2)})_{i,:,::} & 0 \end{bmatrix} \quad \text{for } i = 1, ..., J_1,$$

$$W_{i+1,:,::}^{(L_2)} = \begin{bmatrix} 0 & (W_{2}^{(L_2)})_{i-J_1,:,::} \end{bmatrix}, \quad W_{i+1+1,:,::}^{(L_2)} = \begin{bmatrix} 0 & -(W_{2}^{(L_2)})_{i-J_1,:,::} \end{bmatrix},$$

$$B^{(L_2)} = \begin{bmatrix} B_{1}^{(L_2)} & 0 \end{bmatrix}.$$

$W^{(L_2)}$ 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

$$W_{i,:,::}^{(f)} = \begin{bmatrix} (W_{1}^{(f)})_{i,:,::} & 0 \end{bmatrix} \quad \text{for } i = 1, ..., J_1,$$

$$W_{i+1,:,::}^{(f)} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \quad W_{i+1+1,:,::}^{(f)} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix},$$

$$B^{(L_2)} = \begin{bmatrix} B_{1}^{(L_2)} & 0 \end{bmatrix}.$$

For the $L_1$-th layer, we set

$$W_{i,:,::}^{(L_1)} = \begin{bmatrix} (W_{1}^{(L_1)})_{i,:,::} & 0 & 0 \end{bmatrix}, \quad W_{2,:,::}^{(L_1)} = \begin{bmatrix} -(W_{1}^{(L_1)})_{i,:,::} & 0 & 0 \end{bmatrix}, \quad W_{3,:,::}^{(L_1)} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \quad W_{4,:,::}^{(L_1)} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix},$$

and $B^{(L_1)} = 0$.

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 $x = [x_1 \cdots x_M]^\top \in \mathbb{R}^M$, define

$$X = \begin{bmatrix} (x_1)_{+} & (x_1)_{-} & \cdots & (x_M)_{+} & (x_M)_{-} \end{bmatrix} \in \mathbb{R}^{N \times (2M)}.$$ For any CNN architecture $F^{\text{CNN}}(L, J, K, \kappa)$ from $\mathbb{R}^M \rightarrow \mathbb{R}$, there exists a CNN architecture $F^{\text{CNN}}(L, M, J, K, \kappa)$ from $\mathbb{R}^{N \times (2M)} \rightarrow \mathbb{R}$ such that for any $f_1 \in F^{\text{CNN}} (L, J, K, \kappa)$, there exists $f \in F^{\text{CNN}} (L, M, J, K, \kappa)$ with $f(X) = f_1(x)$. Furthermore, the fully connected layer of $F^{\text{CNN}} (L, M, J, K, \kappa)$ has nonzero entries only in the first row.

**Proof of Lemma 15.** Denote $f_1$ as

$$f_1 = W_1 \cdot \text{Conv}_{\mathcal{W}_1, B_1}(x)$$

where $\mathcal{W}_1 = \{ W_{1,i}^{(L_1)} \}_{i=1}^{L_1}$, $B_1 = \{ B_{1,i}^{(L_1)} \}_{i=1}^{L_1}$ are sets of filters and biases and $\text{Conv}_{\mathcal{W}_1, B_1}$ 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
\[ f = W \cdot \text{Conv}_{\mathcal{W}, \mathcal{B}}(x) \]
and $f(X) = f_1(x)$.

For the first layer, i.e., $l = 1$, we design $\mathcal{W}^{(1)}$ and $\mathcal{B}^{(1)}$ since $x$ 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
\[
\mathcal{W}^{(1)}_{(i-1)M+j,:} = \begin{bmatrix} 0 & (\mathcal{W}^{(1)}_{1})_{i,1,:} & - (\mathcal{W}^{(1)}_{1})_{i,1,:} & \cdots & (\mathcal{W}^{(1)}_{1})_{i,K,:} & - (\mathcal{W}^{(1)}_{1})_{i,K,:} & 0 \end{bmatrix}
\]
where $0$ is of size $1 \times (j - 1), \bar{0}$ 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
\[
\mathcal{W}^{(1)}_{(i-1)M+j,:} = \begin{bmatrix} 0 & (\mathcal{W}^{(1)}_{1})_{i,1,:} & - (\mathcal{W}^{(1)}_{1})_{i,1,:} & \cdots & (\mathcal{W}^{(1)}_{1})_{i,K-j+1,:} & - (\mathcal{W}^{(1)}_{1})_{i,K-j+1,:} & 0 \end{bmatrix}
\]
where $0$ is of size $1 \times (j - 1)$. The bias is set as
\[
\mathcal{B}^{(1)} = \begin{bmatrix} (\mathcal{B}^{(1)}_{1})_{1,:}^T & \cdots & (\mathcal{B}^{(1)}_{1})_{J,:}^T \end{bmatrix}.
\]

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
\[
\mathcal{W}^{(l)}_{(i-1)M+j,:} = \begin{bmatrix} 0 & (\mathcal{W}^{(l)}_{l})_{i,1,:} & - (\mathcal{W}^{(l)}_{l})_{i,1,:} & \cdots & (\mathcal{W}^{(l)}_{l})_{i,K,:} & - (\mathcal{W}^{(l)}_{l})_{i,K,:} & 0 \end{bmatrix}
\]
where $0$ is of size $1 \times (j - 1), \bar{0}$ 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
\[
\mathcal{W}^{(l)}_{(i-1)M+j,:} = \begin{bmatrix} 0 & (\mathcal{W}^{(l)}_{l})_{i,1,:} & - (\mathcal{W}^{(l)}_{l})_{i,1,:} & \cdots & (\mathcal{W}^{(l)}_{l})_{i,K-j+1,:} & - (\mathcal{W}^{(l)}_{l})_{i,K-j+1,:} & 0 \end{bmatrix}
\]
where $0$ is of size $1 \times (j - 1)$. The bias is set as
\[
\mathcal{B}^{(l)} = \begin{bmatrix} (\mathcal{B}^{(l)}_{1})_{1,:}^T & \cdots & (\mathcal{B}^{(l)}_{1})_{J,:}^T \end{bmatrix}.
\]

For the fully connected layer, we set
\[
W = \begin{bmatrix} (\mathcal{W}^{(1)}_{1})_{1,:}^T & \cdots & (\mathcal{W}^{(1)}_{1})_{J,:}^T \end{bmatrix}.
\]

With these choices, the lemma is proved. \qed

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 $f \in \mathcal{F}^{\text{CNN}}(L, J, K, \kappa_1, \kappa_2)$, there exists $\tilde{f} \in \mathcal{F}^{\text{CNN}}(L, J, K, \alpha^{-1} \kappa_1, \alpha L \kappa_2)$ such that $\tilde{f}(x) = f(x)$.

Proof of Lemma 16. This lemma is proved using the linear property of ReLU and convolution. Let $f$ be any CNN in $\mathcal{F}^{\text{CNN}}(L, J, K, \kappa_1, \kappa_2)$. Denote its architecture as
\[ f(x) = W \cdot \text{Conv}_{\mathcal{W}, \mathcal{B}}(x). \]

Define $\tilde{W} = \alpha^L W$ and $\tilde{\mathcal{W}}, \tilde{\mathcal{B}}$ as
\[ \tilde{\mathcal{W}}^{(l)} = \alpha^{-1} \mathcal{W}^{(l)}, \tilde{\mathcal{B}}^{(l)} = \alpha^{-1} \mathcal{B}^{(l)} \]
for any $l \in (1, L)$. Set
\[ \tilde{f}(x) = \tilde{W} \cdot \text{Conv}_{\tilde{\mathcal{W}}, \tilde{\mathcal{B}}}(x) \]
We have $\tilde{f} \in \mathcal{F}^{\text{CNN}}(L, J, K, \alpha^{-1} \kappa_1, \alpha L \kappa_2)$ and $\tilde{f}(x) = f(x)$ since ReLU($cx$) = $c$ReLU($x$) for any $c > 0$. \qed
C.15. Lemma 17 and its proof

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

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

\[
L = O\left(\log(1/\varepsilon) + D + \log D\right), \quad J = \lfloor 48d(s+1)(s+3) + 28d + 6D \rfloor, \quad \kappa_1 = 1, \quad \log \kappa_2 = O\left(\log^2 \frac{1}{\varepsilon}\right)
\]

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

**Proof of Lemma 17.** According to Lemma 13, there exists a CNN \( g_{i,j} \) realizing \( \tilde{f}_{i,j}^\text{CNN} \circ \phi_{i,j}^\text{CNN} \) and a CNN \( \tilde{g}_i \) realizing \( \mathbb{I}_\Delta \circ \tilde{d}_i^2 \). Using Lemma 14, one can construct a CNN excluding the fully connected layer, denoted by \( \tilde{g}_{i,j} \), such that

\[
\tilde{g}_{i,j}(x) = \begin{bmatrix} (g_{i,j}(x))_+ & (g_{i,j}(x))_- & (\tilde{g}_i(x))_+ & (\tilde{g}_i(x))_- \end{bmatrix} .
\]

Here \( \tilde{g}_{i,j} \) has \( \lfloor 48d(s+1)(s+3) + 28 \rfloor \) channels.

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

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

\[
L = O\left(\log \frac{1}{\varepsilon} + D + \log D\right), \quad J = \lfloor 28d(s+1)(s+3) + 18d \rfloor + 6D, \quad \kappa = O\left(\varepsilon^{-\left(\log 2\right)^2 \left(\frac{2d}{sp-q} + c_0d^{-1}\right)}\right).
\]

and \( K \) can be any integer in \( [2, D] \).

We next rescale all parameters in convolutional layers of \( \tilde{\tilde{f}}_{i,j}^\text{CNN} \) to be no larger than 1. Using Lemma 16, we can realize \( \tilde{\tilde{f}}_{i,j}^\text{CNN} \) by \( \tilde{f}_{i,j}^\text{CNN} \in \mathcal{F}^\text{CNN}(L, J, K, \alpha^{-1}\kappa, \alpha^{-1}\kappa) \) for any \( \alpha > 1 \). Set \( \alpha = C'\varepsilon^{-\left(\log 2\right)^2 \left(\frac{2d}{sp-q} + c_0d^{-1}\right)}(8KD)^{-1}M^{-\frac{1}{2}} \) where \( C' \) \( \tilde{\tilde{f}}_{i,j}^\text{CNN} \) is a constant such that \( \kappa \leq C'\varepsilon^{-\left(\log 2\right)^2 \left(\frac{2d}{sp-q} + c_0d^{-1}\right)}(8KD)^{-1}M^{-\frac{1}{2}} \). With this \( \alpha \), we have \( \tilde{f}_{i,j}^\text{CNN} \in \mathcal{F}^\text{CNN}(L, J, K, \kappa_1, \kappa_2) \) with

\[
L = O\left(\log (1/\varepsilon) + D + \log D\right), \quad J = \lfloor 28d(s+1)(s+3) + 18d \rfloor + 6D, \quad \kappa_1 = (8KD)^{-1}M^{-\frac{1}{2}} = O(1), \quad \log \kappa_2 = O\left(\log^2 \frac{1}{\varepsilon}\right).
\]

\( \square \)

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 \( \mathcal{F}^\text{CNN}(L, J, K, \kappa_1, \kappa_2) \) be any CNN architecture from \( \mathbb{R}^D \) to \( \mathbb{R} \). Assume the weight matrix in the fully connected layer of \( \mathcal{F}^\text{CNN}(L, J, K, \kappa_1, \kappa_2) \) has nonzero entries only in the first row. Let \( M \) be a positive integer. There exists a ConvResNet architecture \( \mathcal{C}(M, L, J, K_1, \kappa_2(1 \vee \kappa_1^{-1})) \) such that for any \( \{f_i(x)\}_{m=1}^M \in \mathcal{F}^\text{CNN}(L, J, K, \kappa_1, \kappa_2) \), there exists \( \bar{f} \in \mathcal{C}(M, L, J, \kappa_1, \kappa_2(1 \vee \kappa_1^{-1})) \) with

\[
\bar{f}(x) = \sum_{m=1}^M \tilde{f}_m(x).
\]

**Proof of Lemma 18.** Denote the architecture of \( f_m \) by

\[
f_m(x) = W_m \cdot \text{Conv}_{\mathcal{W}_m, \mathcal{B}_m}(x)
\]

with \( \mathcal{W}_m = \{ W_m^{(l)} \}_{l=1}^L \), \( \mathcal{B}_m = \{ B_m^{(l)} \}_{l=1}^L \). In \( \tilde{f} \), denote the weight matrix and bias in the fully connected layer by \( \tilde{W}, \tilde{b} \) and the set of filters and biases in the \( m \)-th block by \( \tilde{\mathcal{V}}_m \) and \( \tilde{\mathcal{B}}_m \), respectively. The padding layer \( P \) in \( \tilde{f} \) 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 subnetwork $\text{Conv}_{\bar{W}_m, \bar{B}_m} : \mathbb{R}^{D \times 3} \rightarrow \mathbb{R}^{D \times 3}$ such that for any $Z \in \mathbb{R}^{D \times 3}$ in the form of
\[
Z = \begin{bmatrix} x & * & * \end{bmatrix},
\]
we have
\[
\text{Conv}_{\bar{W}_m, \bar{B}_m}(Z) = \begin{bmatrix} 0 & \frac{\kappa_1}{\kappa_2} (f_m)_+ + \frac{\kappa_2}{\kappa_1} (f_m)_- & * \\
0 & * & * \end{bmatrix}
\]
where $*$ denotes some entries that we do not care.

For any $m$, the first layer of $\bar{f}_m$ takes input in $\mathbb{R}^D$. As a result, the filters in $\mathcal{W}^{(1)}_m$ are in $\mathbb{R}^D$. We pad these filters by zeros to get filters in $\mathbb{R}^{D \times 3}$ and construct $\mathcal{W}^{(1)}_m$ as
\[
(\bar{W}_m^{(1)})_{j,:,:} = \begin{bmatrix} (\mathcal{W}_m^{(1)})_{j,:,:} & 0 & 0 \end{bmatrix}.
\]
For any $Z$ in the form of (55), we have $\bar{W}_m^{(1)} \ast Z = \mathcal{W}_m^{(1)} \ast x$. For the filters in the following layers and all biases, we simply set
\[
\mathcal{W}_m^{(l)} = \mathcal{W}_m^{(1)} \quad \text{for } l = 2, \ldots, L - 1,
\]
\[
\mathcal{B}_m^{(l)} = \mathcal{B}_m^{(1)} \quad \text{for } l = 1, \ldots, L - 1.
\]
In $\text{Conv}_{\bar{W}_m, \bar{B}_m}$, 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 $\bar{B}_m^{(L)} = 0$ and $\bar{W}_m^{(L)}$ as size one filters with three output channels in the form of
\[
(\bar{W}_m^{(L)})_{1,:,:} = 0, \quad (\bar{W}_m^{(L)})_{2,:,:} = \frac{\kappa_1}{\kappa_2} (W_m)_{1,:}, \quad (\bar{W}_m^{(L)})_{3,:,:} = -\frac{\kappa_1}{\kappa_2} (W_m)_{1,:}.
\]
Under such choices, (56) is proved and all parameters in $\mathcal{W}_m, \bar{B}_m$ are bounded by $\kappa_1$.

By composing all residual blocks, one has
\[
(\text{Conv}_{\bar{W}_M, \bar{B}_M} + \text{id}) \circ \cdots \circ (\text{Conv}_{\bar{W}_1, \bar{B}_1} + \text{id}) \circ P(x) = \begin{bmatrix} \frac{\kappa_2}{\kappa_1} \sum_{m=1}^M (f_m)_+ + \frac{\kappa_1}{\kappa_2} \sum_{m=1}^M (f_m)_- & * & * \\
* & * & * \\
\vdots & \vdots & \vdots \end{bmatrix}.
\]

The fully connect layer is set as
\[
\bar{W} = \begin{bmatrix} 0 & \frac{\kappa_2}{\kappa_1} & -\frac{\kappa_1}{\kappa_2} \\
0 & \frac{\kappa_1}{\kappa_2} & 0 \\
0 & 0 & 0 \end{bmatrix}, \quad \bar{b} = 0.
\]
The weights in the fully connected layer are bounded by $\kappa_2(1 + \kappa_1^{-1})$.

Such a ConvResNet gives
\[
\bar{f}(x) = \sum_{m=1}^M (f_m(x))_+ - \left( \sum_{m=1}^M (f_m(x))_- \right) = \sum_{m=1}^M f_m(x).
\]

\[\square\]

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 $\{x_i\}_{i=1}^n$ be i.i.d. samples from some probability distribution. Let $F$ be a class of functions whose magnitude is bounded by $F$. Let $v \geq \sup_{f \in F} \operatorname{Var}(f(x))$, $v > 0$ be a constant. Assume there are constants $M > 0$, $0 < \lambda < 1$ such that

\[\begin{align*}
&B1 \quad \mathcal{H}_B(v^{1/2}, F, \| \cdot \|_{L_2}) \leq \lambda nM^2 / (8(4v + MF/3)), \\
&B2 \quad M \leq \lambda v / (4F), \quad v^{1/2} \leq F.
\end{align*}\]
(B3) if $\lambda M / 8 \leq v^{1/2}$, then

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

Then

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

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

$$E_n(f) = \frac{1}{n} \sum_{i=1}^{n} [\phi(y, f_n(x_i)) - \phi(y, f(x_i))] - \mathbb{E}[\phi(y, f_n(x_i)) - \phi(y, f(x_i))]. \quad (57)$$

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

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

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 $i = 1, ..., n$ 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

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

which implies $\mathcal{F}_{n,i}$ is an empty set for $2^{i-1} \epsilon_n^2 > C_1 F_n$. We set $i_n = \inf\{i \in \mathbb{N} : 2^{i-1} \epsilon_n^2 > C_1 F_n\}$. 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 $\mathcal{E}_\phi(f, f_\phi^*) \leq a_n \leq \epsilon_n^2/2$, we have

$$\inf_{f \in \mathcal{F}_{n,i}} \mathbb{E}[\phi(y, f(x)) - \phi(y, f_n(x))] = \inf_{f \in \mathcal{F}_{n,i}} \left[ \mathcal{E}_\phi(f, f_\phi^*) - \mathcal{E}_\phi(f, f_n) \right] \geq 2^{i-1} \epsilon_n^2.$$ 

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

$$\mathbb{P} \left( \mathcal{E}_\phi(\tilde{f}_{\phi, n}, f_\phi^*) \geq \epsilon_n^2 \right) \leq \sum_{i=1}^{i_n} \mathbb{P} \left( \sup_{f \in \mathcal{F}_{n,i}} \mathbb{E}(f) \geq \mathbb{E}[\phi(y, f(x)) - \phi(y, f_n(x))] \right) \leq \sum_{i=1}^{i_n} \mathbb{P} \left( \sup_{f \in \mathcal{F}_{n,i}} \mathbb{E}(f) \geq M_{n,i} \right),$$

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

$$\sup_{f \in \mathcal{F}_{n,i}} \mathbb{E} \left[ \left( \phi(y, f(x)) - \phi(y, f_n(x)) \right)^2 \right] \leq 2 \sup_{f \in \mathcal{F}_{n,i}} \mathbb{E} \left[ \left( \phi(y, f(x)) - \phi(y, f_\phi^*(x)) \right)^2 + \left( \phi(y, f_n(x)) - \phi(y, f_\phi^*(x)) \right)^2 \right] \leq 2 C_2 e^{F_n} F_n^{2-v} \left( \sup_{f \in \mathcal{F}_{n,i}} \mathcal{E}_\phi(f, f_\phi^*)^\nu + \mathcal{E}_\phi(f_n, f_\phi^*)^\nu \right) \leq 2 C_2 e^{F_n} F_n^{2-v} \left( 2^{i-2} \epsilon_n^2 \right)^\nu \leq 4^{i+1} C_2 e^{F_n} F_n^{2-v} \left( 2^{i-2} \epsilon_n^2 \right)^\nu \leq 4^{i+1} C_2 e^{F_n} F_n^{2-v} \epsilon_n^2.$$

Define $\mathcal{G}_{n,i} = \{ g = \phi(y, f_n(x)) - \phi(y, f(x)) : f \in \mathcal{F}_{n,i} \}$. For any $g \in \mathcal{G}_{n,i}$, $\text{Var}(g) \leq 4^{i+1} C_2 e^{F_n} F_n^{2-v} M_{n,i}^{\nu}$. Since $f_n, f \in \mathcal{F}_n$, we have $\|g\|_{L^\infty} \leq C_1 \|f - f\| \leq 2C_1 F_n$.

To apply Lemma 19 on $\mathcal{G}_{n,i}$, we set $\lambda = 1/2$, $F = D_1 F_n$, $M = M_{n,i}$ and $v = v_{n,i} = D_2 F_n^{2-v} M_{n,i}^\nu$ with

$$D_1 = \frac{1}{8(2C_1)^{1-v}}, \quad D_2 = \max \left\{ 4^{1+\nu} C_2 e^{F_n}, 64(2C_1)^{1-v} \right\}.$$
Since $D_1 \geq 2C_1$ and $D_2 \geq 4^{1+\nu}C_2e^{F_n}$, we have $\sup_{g \in \mathcal{G}_n, i} \|g\|_{L^{\infty}} \leq F$ and $\sup_{g \in \mathcal{G}_n, i} \text{Var}(g) \leq v_{n,i}$. We first check the validation of (B2). When $M_{n,i} \leq 2C_1F_n$, $D_2 \geq 64(2C_1)^{2-\nu}$, \[
 \frac{v}{F^2} = \frac{v_{n,i}}{D_1^2F_n^2} \leq \frac{64(2C_1)^{2-\nu}D_2^2F_n^{2-\nu}(2C_1F_n)^\nu}{D_2^2F_n^2} = \frac{64(2C_1)^{2-\nu}}{D_2} \leq 1,
\]
m_{n,i} = M_{n,i}^{1-\nu}M_{n,i}^\nu \leq (2C_1F_n)^{1-\nu}M_{n,i}^\nu = \frac{8(2C_1)^{1-\nu}D_2^{2-\nu}M_{n,i}^\nu}{8D_2F_n} \leq \frac{v_{n,i}}{8F_n^2},
when $F_n$ is large enough. Thus (B2) is satisfied.

For (B3), note that for $g_1 = \phi(y_f(x)) - \phi(y_f(x))$, $g_2 = \phi(y_f(x)) - \phi(y_f(x))$ where $f_1, f_2 \in \mathcal{F}_{n,i}$, $|g_1 - g_2| \leq C_1|f_1 - f_2|$. We have \[
\mathcal{H}_B(\delta, \mathcal{G}_{n,i}, \| \cdot \|_{L^2}) \leq \mathcal{H}_B(C_1\delta, \mathcal{F}_{n,i}, \| \cdot \|_{L^2}) \leq \mathcal{H}_B(C_1\delta, \mathcal{F}_{n,i}, \| \cdot \|_{L^2}),
\]
where the second inequality comes from $\mathcal{F}_{n,i} \subset \mathcal{F}_n$. Since $M_{n,i}^{-1}\int_{\lambda M_{n,i}/32}^{\nu_{n,i}^{1/2}} (\mathcal{H}_B(\tau, \mathcal{G}_{n,i}, \| \cdot \|_{L^2}))^{1/2} d\tau$ is a non-increasing function of $i$, \[
M_{n,i}^{-1}\int_{\lambda M_{n,i}/32}^{\nu_{n,i}^{1/2}} (\mathcal{H}_B(\tau, \mathcal{G}_{n,i}, \| \cdot \|_{L^2}))^{1/2} d\tau
\leq M_{n,i}^{-1}\int_{M_{n,i}/64}^{\nu_{n,i}^{1/2}} (\mathcal{H}_B(C_1\tau, \mathcal{F}_{n,i}, \| \cdot \|_{L^2}))^{1/2} d\tau
\leq M_{n,i}^{-1}\int_{M_{n,i}/64}^{\nu_{n,i}^{1/2}} (\mathcal{H}_B(C_1^{2}F_n/128, \mathcal{F}_{n,i}, \| \cdot \|_{L^2}))^{1/2}
\leq (D_2F_n^{2-\nu})^{1/2}M_{n,i}^{\nu/2-1}C_3\epsilon^{-F_n}M_{n,i}^{2-\nu}F_n^{1-\nu/2} - F_n^{1-\nu/2}C_2^{2-\nu}F_n^{1-\nu/2}
\leq C_3^{\nu/2}C_1^{\nu/2}M_{n,i}^{\nu/2-1}C_3\epsilon^{-F_n}M_{n,i}^{2-\nu}F_n^{1-\nu/2} - F_n^{1-\nu/2}C_2^{2-\nu}F_n^{1-\nu/2}
= (2^{6\nu-12}e^{-F_n}C_3C_2^{2-\nu}D_2)^{1/2}n^{\nu/2}.
\]
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 $M_{n,i} \leq \lambda v_{n,i}/(4F) \leq \frac{1}{8}v_{n,i}$ which implies $v_{n,i}^{1/2} \geq 8M_{n,i}/16 = \lambda M_{n,i}/8$. Thus the condition in (B3) is satisfied. From (B3), we derive \[
\left(\mathcal{H}_B(\nu_{n,i}^{1/2}, \mathcal{G}_{n,i}, \| \cdot \|_{L^2})\right)^{1/2} \leq \frac{M_{n,i}}{v_{n,i}^{1/2} - M_{n,i}/64} \int_{M_{n,i}/64}^{\nu_{n,i}^{1/2}} (\mathcal{H}_B(\tau, \mathcal{G}_{n,i}, \| \cdot \|_{L^2}))^{1/2} d\tau
\leq \frac{M_{n,i}}{v_{n,i}^{1/2} - M_{n,i}/64} \cdot 2^{-23/2}n^{1/2} \leq \frac{M_{n,i}}{3v_{n,i}^{1/2}} \cdot 2^{-23/2}n^{1/2} = \frac{M_{n,i}^{2\nu}}{2^{19/2}v_{n,i}^{1/2}},
\]
where in the third inequality we have $M_{n,i} \leq 16\sqrt{v_{n,i}}$. Again from (B2), $M_{n,i} \leq \lambda v_{n,i}/(4F) = v_{n,i}/(8F)$. Therefore \[
\lambda M_{n,i}^{2\nu n} \geq 8(4v_{n,i} + M_{n,i}F/3) \geq 16(4v_{n,i} + M_{n,i}F/3) \geq (64 + 2/3)v_{n,i}
\geq \frac{M_{n,i}^{2\nu n}}{9 \cdot 2^{19/2}v_{n,i}^{1/2}} \geq \mathcal{H}_B(\nu_{n,i}^{1/2}, \mathcal{G}_{n,i}, \| \cdot \|_{L^2})
\]
and (B1) is verified.

Apply Lemma 19 to each $\mathcal{H}_{n,i}$, we get \[
P \left(\mathcal{E}_{\delta}(\tilde{f}_{\phi,n}, f_n) \geq \epsilon_n^{2}\right) \leq \sum_{i=1}^{n} \exp \left(\frac{-nM_{n,i}^2}{4(4v_{n,i} + M_{n,i}F/3)}\right)
\leq \sum_{i=1}^{\infty} \exp \left(-C_4nM_{n,i}^2/v_{n,i}\right) \leq \sum_{i=1}^{\infty} \exp \left(-C_4n(2^{1/2})^{2-\nu}e^{-F_n}(\epsilon_n^{2}/F_n)^{2-\nu}\right)
\leq C_3 \exp \left(-C_5e^{-F_n}(\epsilon_n^{2}/F_n)^{2-\nu}\right).
\]
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 $\tilde{f}_{\phi,n}$ such that $\|\tilde{f}_{\phi,n} - f^*_{\phi,n}\|_{L^\infty} \leq 4eF_n\varepsilon$. Then we show that $\tilde{f}_{\phi,n}$ can be realized by a ConvResNet $\tilde{f}_{\phi,n}$.

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

**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

$$\tilde{F}^{(F_n)} = \{g_{F_n} \circ \tilde{h}_n \circ g_n \circ \tilde{\eta}\} \quad (58)$$

where

$$\tilde{\eta} \in \mathcal{C}(M, L, k, \kappa_1, \kappa_2)$$

with

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

$$\tilde{h}_n \in \mathcal{F}^{\text{MLP}}(L_2, J_2, \kappa_2, F_n)$$

with

$$L_2 = O(\log(eF_n/\varepsilon)), \quad J_2 = O(eF_n^{-1}\log(eF_n/\varepsilon)), \quad \kappa_2 = O(eF_n).$$

and

$$g_n(z) = \text{ReLU} \left( -\text{ReLU} \left( -z + \frac{eF_n}{1 + eF_n} \right) + \frac{eF_n - 1}{1 + eF_n} \right) + \frac{1}{1 + eF_n},$$

$$g_{F_n} = \text{ReLU} \left( -\text{ReLU} \left( -z + F_n \right) + 2F_n \right) - F_n.$$

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

$$\|\tilde{f}_{\phi,n} - f^*_{\phi,n}\|_{L^\infty} \leq 4eF_n\varepsilon.$$

**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

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

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

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

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

The function $\max(\min(z, \frac{eF_n}{1 + eF_n}), \frac{1}{1 + eF_n})$ can be realized by $g_n(z) = \text{ReLU} \left( -\text{ReLU} \left( -z + \frac{eF_n}{1 + eF_n} \right) + \frac{eF_n - 1}{1 + eF_n} \right) + \frac{1}{1 + eF_n}$

Then $g_n \circ \tilde{\eta}$ is an approximation of $\eta_n$.

Define

$$h_n = \max \left( \min \left( \log \left( \frac{2eF_n}{1 - z} \right), F_n \right), -F_n \right)$$

for $z \in [0, 1]$ which is a Lipschitz function with Lipschitz constant $(1 + eF_n)^2/eF_n$. According to Chen et al. (2019a, Theorem 4.1), there exists an MLP $\tilde{h} \in \mathcal{F}^{\text{MLP}}(L, J, \kappa)$ such that

$$\|\tilde{h} - h_n \cdot \frac{eF_n}{(1 + eF_n)^2}\|_{L^\infty} \leq \varepsilon_2(1 + eF_n)^{-1}$$

with

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

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

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

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

$$\tilde{f}_{\phi,n} = g_{F_n} \circ \tilde{h}_n \circ g_n \circ \tilde{\eta}$$
as an approximation of \( f_{\phi,n}^* \). Then the error of \( \tilde{f}_{\phi,n} \) can be decomposed as
\[
\| \tilde{f}_{\phi,n} - f_{\phi,n}^* \|_{L^\infty} = \|(g_{F_n} \circ \tilde{h}_n) \circ (g_n \circ \tilde{\eta}) - h_n \circ \eta_n\|_{L^\infty} \\
\leq \|(g_{F_n} \circ \tilde{h}_n) \circ (g_n \circ \tilde{\eta}) - h_n \circ \eta_n\|_{L^\infty} + \|(h_n \circ (g_n \circ \tilde{\eta}) - h_n \circ \eta_n\|_{L^\infty} \\
\leq \varepsilon_2 + \frac{1}{eF_n}^2 \epsilon \|\eta_n - g_n \circ \tilde{\eta}\|_{L^\infty} \leq \frac{1 + 2}{eF_n} \epsilon \varepsilon_1 + \varepsilon_2.
\]
Choosing \( \varepsilon_2 = \frac{1 + 2}{eF_n} \epsilon \varepsilon_1 \) gives rise to \( \| \tilde{f}_{\phi,n} - f_{\phi,n}^* \|_{L^\infty} \leq 4eF_n \epsilon \varepsilon_1 \). With this choice, we have
\[
L_2 = O(\log(1/\varepsilon_1)), \quad J_2 = O(\epsilon_1^{-1} \log(1/\varepsilon_1)), \quad \kappa_2 = O(eF_n).
\]
Setting \( \varepsilon_1 = \varepsilon \) proves the lemma.

To show that \( \tilde{F}(F_n) \) 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 \( C(M, L, J, K, \kappa_1, \kappa_2) \) be the ConvResNet structure defined in Theorem 1. Its covering number is bounded by
\[
\log \mathcal{N}(\delta, C(M, L, J, K, \kappa_1, \kappa_2), \| \cdot \|_{L^\infty}) = O \left( \frac{3\varepsilon^{-d/s} \log(1/\varepsilon)}{\log(3/\varepsilon)} \right).
\]

**Lemma 21** is proved based on the following lemma:

**Lemma 22 (Lemma 4 of Oono & Suzuki (2019)).** Let \( C(M, L, J, K, \kappa_1, \kappa_2) \) be a class of ConvResNet architecture from \( \mathbb{R}^D \) to \( \mathbb{R} \). Let \( \kappa = \kappa_1 \lor \kappa_2 \). For \( \delta > 0 \), we have
\[
\mathcal{N}(\delta, C(M, L, J, K, \kappa_1, \kappa_2), \| \cdot \|_{L^\infty}) \leq (2\kappa A_1/\delta)^{\Lambda_2},
\]
where
\[
\Lambda_1 = (8M + 12)D^2(1 \lor \kappa_2)(1 \lor \kappa_1)\bar{\rho}^+, \quad \Lambda_2 = M(16D^2K + 4D) + 4D^2 + 1
\]
with \( \bar{\rho} = (1 + \rho)^M \leq (1 + M)^M \leq e. \) Moreover, we have \( \rho^+ = 1 + M\rho^+ = (1 + 4DK\kappa_1)^D \). Therefore,
\[
\log \mathcal{N}(\delta, C(M, L, J, K, \kappa_1, \kappa_2), \| \cdot \|_{L^\infty}) = O \left( \frac{3\varepsilon^{-d/s} \log(1/\varepsilon)}{\log(3/\varepsilon)} \right).
\]

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

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

**Lemma 23.** Let \( C(F_n) \) be defined as in Lemma 2. The network composition class \( \tilde{F}(F_n) \) defined in Lemma 20 can be realized by a ConvResNet class \( C(F_n) \). Moreover, the covering number of \( C(F_n) \) is bounded by
\[
\log \mathcal{N}(\delta, C(F_n), \| \cdot \|_{L^\infty}) = O \left( \frac{3\varepsilon^{-d/s} \log(1/\varepsilon)}{\log(3/\varepsilon)} \right).
\]

**Proof of Lemma 23.** In this proof, we show that each part of \( \tilde{f}_{\phi,n} \) in Lemma 20 can be realized by ConvResNet architectures. Specifically, we show that \( \tilde{h}_n, g_n, \tilde{h}_n \) can be realized by residual blocks \( \tilde{h}_n, g_n, \tilde{h}_n \), and \( g_{F_n} \) can be realized by a ConvResNet \( \tilde{g}_{F_n} \). In the following, we show the existence of each ingredient.
Realize \( \tilde{\eta} \) by residual blocks. In Lemma 20, \( \tilde{\eta} \in C(M^{(n)}, L^{(n)}, J^{(n)}, K^{(n)}, \kappa_1^{(n)}, \kappa_2^{(n)}) \) with
\[
M^{(n)} = O\left(\varepsilon^{-d/s}\right),\quad L^{(n)} = O(\log(1/\varepsilon) + D + \log D),\quad J^{(n)} = O(D),\quad \kappa^{(n)} = O(1),\quad \log \kappa_2^{(n)} = O(\log^2(1/\varepsilon)).
\]
By Lemma 21, the covering number of this architecture is bounded by
\[
\log \mathcal{N}\left(\delta, C(M^{(n)}, L^{(n)}, J^{(n)}, K^{(n)}, \kappa_1^{(n)}, \kappa_2^{(n)}), \| \cdot \|_\infty \right)
= O\left(\varepsilon^{-(\frac{2}{\delta})^d} \log(1/\varepsilon) \left(\log^2(1/\varepsilon) + \log D + \log(1/\delta)\right)\right).
\]
Excluding the final fully connected layer, denote all of the residual blocks of \( \tilde{\eta} \) by \( \tilde{\eta}^{(\text{Conv})} \), then \( \tilde{\eta} \in C^{(n)} \) with \( C^{(n)} = C^{\text{Conv}}(M^{(n)}, L^{(n)}, J^{(n)}, K^{(n)}, \kappa_1^{(n)}) \) and
\[
\log \mathcal{N}\left(\delta, C^{(n)}, \| \cdot \|_\infty \right) \leq \log \mathcal{N}\left(\delta, C(M^{(n)}, L^{(n)}, J^{(n)}, K^{(n)}, \kappa_1^{(n)}, \kappa_2^{(n)}), \| \cdot \|_\infty \right)
= O\left(D^3 \varepsilon^{-d/s} \log(1/\varepsilon) \left(\log^2(1/\varepsilon) + \log D + \log(1/\delta)\right)\right).
\]
We denote the \( i \)-th row (the \( i \)-th element of all channels) of the output of the residual-blocks \( \tilde{\eta}^{(\text{Conv})} \) by \( (\tilde{\eta}^{(\text{Conv})})_{i,:} \).
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 \( \tilde{\eta} \) has the form \( (\tilde{\eta}^{(\text{Conv})})_{1,:} = \frac{\kappa^{(n)}}{\kappa_2^{(n)}} \left[\begin{array}{c} \ast \ 	ilde{\eta}_+ \\
 \tilde{\eta}_- \end{array}\right] \). Here \( \ast \) 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 \( \tilde{\eta} \) except we need to pad the filters and biases by \( 0 \) to be compatible with the additional channels. Then the output of \( \tilde{\eta}^{(\text{Conv})} \) is \( \frac{\kappa^{(n)}}{\kappa_2^{(n)}} \left[\begin{array}{cccccc} \ast & \tilde{\eta}_+ & \tilde{\eta}_- & 0 & 0 & 0 \end{array}\right] \).

Realize \( g_n \) by residual blocks. To realize \( g_n \), we add another block with 4 layers with filters and biases \( \left\{W_{g_n}^{(i)}, B_{g_n}^{(i)}\right\}_{i=1}^5 \) where \( W_{g_n}^{(i)} \in \mathbb{R}^{8 \times 1 \times 8} \), \( B_{g_n}^{(i)} \in \mathbb{R}^{8 \times 8} \). We set the parameters in the first layer as
\[
\left(W_{g_n}^{(1)}\right)_{2,1,:} = \begin{bmatrix} 0 & \frac{\kappa_2^{(n)}}{\kappa_1^{(n)}} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},
\left(W_{g_n}^{(1)}\right)_{3,1,:} = \begin{bmatrix} 0 & 0 & \frac{\kappa_2^{(n)}}{\kappa_1^{(n)}} & 0 & 0 & 0 & 0 & 0 \end{bmatrix},
\left(W_{g_n}^{(1)}\right)_{i,4,:} = 0 \quad \text{for } i = 1, 4, 5, ..., 8,
\]
and \( B_{g_n}^{(1)} = 0 \). This layer scales the output of \( \tilde{\eta} \) back to \( \left[ \begin{array}{cccccc} \ast & \tilde{\eta}_+ & \tilde{\eta}_- & 0 & 0 & 0 \end{array}\right] \). Then we use the other 4 layers to realize \( g_n \). The second layer is set as
\[
\left(W_{g_n}^{(2)}\right)_{4,1,:} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},
\left(W_{g_n}^{(2)}\right)_{i,1,:} = 0 \quad \text{for } i = 1, 2, 3, 5, 6, 7, 8,
\left(B_{g_n}^{(2)}\right)_{1,:} = \begin{bmatrix} 0 & 0 & \frac{e_{p_n}}{1+e_{p_n}} & 0 & 0 & 0 & 0 \end{bmatrix},
\left(B_{g_n}^{(2)}\right)_{i,:} = 0 \quad \text{for } i = 2, ..., 8.
\]
The third layer is set as
\[
\left(W_{g_n}^{(3)}\right)_{4,1,:} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},
\left(W_{g_n}^{(3)}\right)_{i,1,:} = 0 \quad \text{for } i = 1, 2, 3, 5, 6, 7, 8,
\left(B_{g_n}^{(3)}\right)_{1,:} = \begin{bmatrix} 0 & 0 & \frac{e_{p_n-1}}{1+e_{p_n}} & 0 & 0 & 0 & 0 \end{bmatrix},
\left(B_{g_n}^{(3)}\right)_{i,:} = 0 \quad \text{for } i = 2, ..., 8.
The forth layer is set as

\[
\begin{align*}
(W^{(4)}_{g_n})_{4,:} &= [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0], \\
(W^{(4)}_{g_n})_{i,:} &= 0 \quad \text{for } i = 1, 2, 3, 5, 6, 7, 8, \\
(B^{(4)}_{g_n})_{1,:} &= [0 \ 0 \ 0 \ \frac{1}{1+e^{\varepsilon_n}} \ 0 \ 0 \ 0 \ 0], \\
(B^{(4)}_{g_n})_{i,:} &= 0 \quad \text{for } i = 2, ..., 8.
\end{align*}
\]

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

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

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

\[
M^{(g_n)} = 1, \quad L^{(g_n)} = 4, \quad J^{(g_n)} = 8, \quad K^{(g_n)} = 1, \quad \kappa^{(g_n)} = O\left(\frac{\kappa^{(g_n)}}{\kappa^{(1)}} \vee \frac{e^{F_n}}{1 + e^{F_n}}\right).
\]

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

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

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

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

\[
C^{(h_n)} = C(M^{(h_n)}, L^{(h_n)}, J^{(h_n)}, K^{(h_n)}, \kappa_1^{(h_n)} \kappa_2^{(h_n)}),
\]

and

\[
M^{(h_n)} = O\left(\varepsilon_2^{-1}\right), \quad L^{(h_n)} = O\left(\log(1/\varepsilon_2)\right), \quad J^{(h_n)} = O(1), \quad K^{(h_n)} = 1, \\
\kappa_1^{(h_n)} = O(1), \quad \log \kappa_2^{(h_n)} = O(\log(L_{h_n}/\varepsilon_2)),
\]

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

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

Note that such \(\tilde{h}\) is from \(R\) to \(R\). 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 \(\varepsilon_2 = \frac{(1 + e^{F_n})^2}{e^{F_n}}\) and \(L_{h_n} = (1 + e^{F_n})^2/e^{F_n}\) yields

\[
M^{(h_n)} = O\left(e^{-F_n} \varepsilon_2^{-1}\right), \quad L^{(h_n)} = O\left(\log(1/\varepsilon)\right), \quad J^{(h_n)} = O(1), \\
\kappa_1^{(h_n)} = O(1), \quad \log \kappa_2^{(h_n)} = O\left(\log(e^{F_n}/\varepsilon)\right)
\]

and

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

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

\[
(\bar{h}^{(Conv)})_{1,:} = \frac{\kappa_1^{(h_n)}}{\kappa_2^{(h_n)}} \begin{bmatrix} * & * & * & (\bar{h}_n)_+ & (\bar{h}_n)_- & 0 & 0 \end{bmatrix}.
\]
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 $\{W_{g_{F_n}}, B_{g_{F_n}}^{(l)}\}_{l=1}^{3}$ and $\{W, b\}$, respectively. Here $W_{g_{F_n}} \in \mathbb{R}^{8 \times 1 \times 8}$, $B_{g_{F_n}}^{(l)} \in \mathbb{R}^{8 \times 8}$, $W \in \mathbb{R}^{8 \times 8}$ and $b \in \mathbb{R}$.

The first layer is set as

\[
\begin{align*}
W_{g_{F_n}}^{(1)} &= \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}, \\
W_{g_{F_n}}^{(1)} &= \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}, \\
B_{g_{F_n}}^{(1)} &= 0 	ext{ for } i = 1, 2, 3, 4, 7, 8,
\end{align*}
\]

This layer scales the output of $h^{(\text{Conv})}$ back to $[\ast \ast \ast \ast (\tilde{h}_n)_+ (\tilde{h}_n)_- 0 0]$. The rest layers are used to realize $g_{F_n}$.

The second layer is set as

\[
\begin{align*}
W_{g_{F_n}}^{(2)} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \\
W_{g_{F_n}}^{(2)} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \\
B_{g_{F_n}}^{(2)} &= 0 	ext{ for } i = 1, \ldots, 6,
\end{align*}
\]

The third layer is set as

\[
\begin{align*}
W_{g_{F_n}}^{(3)} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \\
W_{g_{F_n}}^{(3)} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \\
B_{g_{F_n}}^{(3)} &= 0 	ext{ for } i = 1, \ldots, 6,
\end{align*}
\]

The first row of the output of the third layer is

\[
[\ast \ast \ast \ast \ast \ast \min((\tilde{h}_n)_+, F_n) \min((\tilde{h}_n)_-, F_n)].
\]

Then the fully connected layer is set as

\[
W_{1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}, \\
W_{i} = 0 \text{ for } i = 2, \ldots, 8
\]

and $b = 0$.

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

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

According to Lemma 22, substituting the expressions of $\kappa_1^{(h_n)}, \kappa_2^{(h_n)}$ gives rise to $\log N(\delta, C^{(g_{F_n})}, \| \cdot \|_{L^\infty}) = O \left( D^2 (\log(1/\varepsilon) + \log F_n + \log(1/\delta)) \right)$. 
The resulting network $\tilde{f}_{\phi,n} \equiv \tilde{g}_F \circ \tilde{h}^{(\text{Conv})} \circ \tilde{g}_n \circ \tilde{g}^{(\text{Conv})}$ is a ConvResNet and

$$\tilde{f}_{\phi,n}(x) = \tilde{f}_{\phi,n}(x)$$

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

$$\log N(\delta, C(F_n), \| \cdot \|_{L^{\infty}})$$

$$\leq \log N \left( \delta, C(n), \| \cdot \|_{L^{\infty}} \right) + \log N \left( \delta, C(g_n), \| \cdot \|_{L^{\infty}} \right)$$

$$+ \log N \left( \delta, C(g_{F}), \| \cdot \|_{L^{\infty}} \right) + \log N(\delta, C(F_n), \| \cdot \|_{L^{\infty}})$$

$$= O(D^3 e^{-\left(\frac{2}{3} \sqrt{8} \right)} \log(1/\varepsilon) \left( \log^2(1/\varepsilon) + \log D + F_n + \log(1/\delta) \right)).$$

The constants hidden in $O(\cdot)$ depend on $d, s, \frac{2d}{np-w}, p, q, c_0, r$ and the surface area of $M$. \hfill $\square$

**Proof of Lemma 2.** Lemma 2 is a direct result of Lemma 20 and Lemma 23. \hfill $\square$

**D.3. Proof of Lemma 3**

**Proof of Lemma 3.** We divide $A^C_n$ into two regions: $\{x \in M : f^+_n > F_n\}$ and $\{x \in M : f^-_n < -F_n\}$. A bound of $T_2$ is derived by bounding the integral on both regions.

Let us first consider the region $\{x \in M : f^+_n > F_n\}$. Since $\eta = e^{f^-_n} / (1 + e^{f^-_n})$, we have

$$\eta \phi(f^+_n) + (1 - \eta) \phi(-f^+_n) = \frac{e^{f^-_n}}{1 + e^{f^-_n}} \phi(f^+_n) + \frac{1}{1 + e^{f^-_n}} \phi(-f^+_n)$$

$$\leq \phi(F_n) + \sup_{z \geq F_n} \frac{\log(1 + e^{z})}{1 + e^{z}}$$

$$\leq \log(1 + e^{-F_n}) + \frac{\log(1 + e^{F_n})}{1 + e^{F_n}}$$

$$\leq 2F_n e^{-F_n}. \quad \text{(59)}$$

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

$$\eta \phi(\tilde{f}_{\phi,n}) + (1 - \eta) \phi(-\tilde{f}_{\phi,n}) = \frac{e^{f^-_n}}{1 + e^{f^-_n}} \phi(\tilde{f}_{\phi,n}) + \frac{1}{1 + e^{f^-_n}} \phi(-\tilde{f}_{\phi,n})$$

$$\leq \phi(F_n - 1) + \frac{\log(1 + e^{\tilde{f}_{\phi,n}})}{1 + e^{f^-_n}}$$

$$\leq \log(1 + e^{-(F_n - 1)}) + \frac{\log(1 + e^{F_n})}{1 + e^{F_n}}$$

$$\leq 2F_n e^{-F_n}. \quad \text{(60)}$$

Combining (59) and (60) gives

$$\left[ \left( \eta \phi(f^+_n) + (1 - \eta) \phi(-f^+_n) \right) - \left( \eta \phi(\tilde{f}_{\phi,n}) + (1 - \eta) \phi(-\tilde{f}_{\phi,n}) \right) \right] \leq 4F_n e^{-F_n}. \quad \text{(61)}$$

Now consider the region $\{x \in M : f^-_n < -F_n\}$, we have

$$\eta \phi(f^-_n) + (1 - \eta) \phi(-f^-_n) = \frac{e^{f^-_n}}{1 + e^{f^-_n}} \phi(f^-_n) + \frac{1}{1 + e^{f^-_n}} \phi(-f^-_n)$$

$$= \frac{1}{1 + e^{-f^-_n}} \phi(f^-_n) + \frac{e^{-f^-_n}}{1 + e^{-f^-_n}} \phi(-f^-_n)$$

$$\leq \phi(F_n) + \sup_{z \leq -F_n} \frac{\log(1 + e^{-z})}{1 + e^{-z}}$$

$$\leq \log(1 + e^{-F_n}) + \frac{\log(1 + e^{F_n})}{1 + e^{F_n}}$$

$$\leq 2F_n e^{-F_n}. \quad \text{(62)}$$
On this region, \(-F_n \leq \tilde{f}_{\phi,n} \leq -F_n + 1\). Thus
\[
\eta \phi(\tilde{f}_{\phi,n}) + (1 - \eta) \phi(-\tilde{f}_{\phi,n}) = \frac{1}{1 + e^{-\tilde{f}_{\phi,n}}} \phi(\tilde{f}_{\phi,n}) + \frac{e^{-\tilde{f}_{\phi,n}}}{1 + e^{-\tilde{f}_{\phi,n}}} \phi(-\tilde{f}_{\phi,n}) \\
\leq \phi(F_n - 1) + \frac{\log(1 + e^{-\tilde{f}_{\phi,n}})}{1 + e^{-\tilde{f}_{\phi,n}}} \\
\leq \log(1 + e^{-(F_n - 1)}) + \frac{\log(1 + e^{F_n})}{1 + e^{F_n}} \\
\leq 2F_n e^{-F_n}.
\] (63)

Combining (62) and (63) gives
\[
\left| \left[ \eta \phi(f_{\phi}^*) + (1 - \eta) \phi(-f_{\phi}^*) \right] - \left[ \eta \phi(\tilde{f}_{\phi,n}) + (1 - \eta) \phi(-\tilde{f}_{\phi,n}) \right] \right| \leq 4F_n e^{-F_n}.
\] (64)

Putting (61) and (64) together, we have
\[
T_2 \leq \int_{A_n^c} \left| \left[ \eta \phi(f_{\phi}^*) + (1 - \eta) \phi(-f_{\phi}^*) \right] - \left[ \eta \phi(\tilde{f}_{\phi,n}) + (1 - \eta) \phi(-\tilde{f}_{\phi,n}) \right] \right| \mu(d\mathbf{x}) \leq 8F_n e^{-F_n}.
\]
\[
\Box
\]