

Article

Not peer-reviewed version

Approximating Functions with Multi-Features on Sphere by Deep Convolutional Neural Networks

[Junyan Huang](#) *

Posted Date: 3 December 2025

doi: 10.20944/preprints202512.0342.v1

Keywords: convolutional neural networks; feature extraction; curse of dimensionality; composite functions approximation; Sobolev space



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Approximating Functions with Multi-Features on Sphere by Deep Convolutional Neural Networks

Junyan Huang 

School of Mathematics, Sun Yat-sen University, Guangzhou, China; huangjy563@mail2.sysu.edu.cn

Abstract

In recent years, deep convolutional neural networks (DCNNs) have demonstrated remarkable success in approximating functions with multiple features. However, several challenges remain unresolved, including the approximation of target functions in Sobolev spaces defined on the unit sphere, and the extension of the types for intrinsic functions. To address these issues, we propose a DCNNs architecture with multiple downsampling layers to approximate multi-feature functions in Sobolev spaces on the unit sphere. Our method facilitates automatic feature extraction without requiring prior knowledge of the underlying composite structure and alleviates the curse of dimensionality in function approximation by extracting general smooth and spherical polynomial features. Compared with previous approaches, the proposed DCNNs architecture is more effective in capturing a variety of features.

Keywords: convolutional neural networks; feature extraction; curse of dimensionality; composite functions approximation; Sobolev space

MSC: 68T07, 68Q32, 41A25

1. Introduction

Deep neural networks have demonstrated exceptional computational power and benefited greatly from the availability of large-scale data [1], leading to remarkable performance across a wide range of scientific and engineering applications. Among these architectures, deep convolutional neural networks (DCNNs), which consist of multiple convolutional layers, have proven particularly effective. DCNNs have been successfully applied to speech recognition, image classification, strategic decision-making in board games, and many other practical tasks [2,3,25–27].

In the context of approximation theory, DCNNs have shown strong potential [4–7,12,28], and their approximation capabilities are often regarded as a form of nonlinear approximation [31]. Early foundational work [13,29,30] examined shallow neural networks with sigmoidal-type activation functions for function approximation. More recently, DCNNs with rectified linear unit (ReLU) activation functions have been shown to possess universal approximation properties and continuous functions [8]. Furthermore, [9] demonstrated that DCNNs require only one-eighth of the free parameters needed by fully connected neural networks (FNNs) to approximate smooth functions.

Nevertheless, several challenges persist. For instance, the error rates reported in [10,11,19] suffer from the curse of dimensionality, with the approximation error deteriorating as the input dimension d increases. In an effort to address this, [21] considered the approximation of composite functions of the form $f(Q(x))$ in the Sobolev space $W_{\infty}^{\beta}(\mathbb{B}^d)$ for $(0 < \beta \leq 1)$, where $Q(x)$ is polynomial. This approach achieved an error rate of $O(\epsilon^{\frac{-1}{\beta}})$ for $(0 < \epsilon < 1)$. However, this method assumes a single feature, implying strong dependence among the input variables.

It has been shown that convolutional layers can effectively extract relevant features [32,33]. Neural networks, due to their mesh-free architectures, have also demonstrated the ability to overcome the curse of dimensionality when the target functions exhibit intrinsic structures [17,18,34], possess fast

decaying Fourier coefficients [13,14], or belong to general mixed smooth function spaces [15,16]. To further reduce variable dependence and alleviate the curse of dimensionality, [20] proposed the approximation of functions with multiple features in Sobolev space $W_\infty^\beta([0,1]^d)$ for $(0 < \beta \leq 1)$. This setting is more realistic for complex tasks, where numerous features are typically required to describe objects. In this case, the error rate becomes $O(\epsilon^{-\frac{d^*}{\beta}})$, where d^* denotes the number of features and $d^* \leq d$. This indicates that the curse of dimensionality can be mitigated by extracting features; however, the error rate is still worse than in the case $d^* = 1$, as the input variables exhibit reduced dependence.

Despite these advances, several challenges remain unresolved. This paper focuses on approximating functions with multiple features defined on the unit sphere, i.e., $f^* \in W_\infty^\beta(\mathbb{S}^{d-1})$ for $\beta \in (0,1]$. The target functions are of the form:

$$f^*(x) = f(F_1(x), F_2(x), \dots, F_{d^*}(x)), \quad x \in \mathbb{S}^{d-1}, \quad (1)$$

where each $\{F_\tau(x)\}_{\tau=1}^{d^*}$ is a smooth function in $W_\infty^r(\mathbb{S}^{d-1})$ with $(r > 0)$. In addition, for further alleviating the curse of dimensionality in function approximation, we approximate functions with spherical polynomial features and functions with symmetric polynomial features in Section 3.4 and 3.5 respectively. The DCNN architecture proposed in Section 3.2 is capable of extracting three types of features: general smooth features, polynomial features, and symmetric polynomial features. In contrast, DCNNs constructed in [20] are limited to extracting only polynomial and symmetric polynomial features.

2. Preliminaries

We begin by establishing notation for function classes. Let $\mathbb{S}^{d-1} = \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$ denote the unit sphere in $\mathbb{R}^d$.

2.1. Deep Convolutional Neural Networks with Down Sampling

DCNNs, constructed using convolutional layers, have demonstrated significant effectiveness in tasks, such as image classification [3]. These networks consist of a sequence of convolutional filters denoted by $\mathbf{w} = \{w^{(j)} : \mathbb{Z} \rightarrow \mathbb{R}\}_{j=1}^J$, where each filter $w^{(j)}$ has infinite support over the index set $\{0, \dots, s^{(j)}\}$ for some $s^{(j)} \in \mathbb{N}$, referred to as filter length. In this work, we assume a uniform filter length across layers, that is, $s^{(j)} \equiv s \in \mathbb{N}$, which yields the dimensionality sequence $\{d_j = d + js\}_{j=1}^J$. The activation function employed is the ReLU.

$$\sigma(u) = \begin{cases} u & u \geq 0, \\ 0 & u < 0. \end{cases} \quad (2)$$

Given a filter w with support on the set $\{0, \dots, s\}$, its convolution with a sequence $v = (v_1, \dots, v_{d_{j-1}})$ produces a new sequence denoted by $w * v$ defined as follows:

$$(w * v)_i = \sum_{k \in \mathbb{Z}} w_{i-k} v_k = \sum_{k=1}^{d_{j-1}} w_{i-k} v_k, \quad i \in \mathbb{Z}, \quad (3)$$

which has support on the set $\{1, \dots, d_{j-1} + s\}$. A DCNN comprising J hidden layers, with neuron mappings $\{h^{(j)} : \mathbb{R} \rightarrow \mathbb{R}^{d_j}\}_{j=1}^J$ is defined iteratively starting from the input vector $h^{(0)} = x$, $x = (x_1, \dots, x_d)$ and

$$h^{(j)}(x) = \sigma(T^{(j)}h^{(j-1)}(x) - b^{(j)}). \quad (4)$$

where $T^{(j)}$ represents the Toeplitz type convolutional matrix.

$$T^{(j)} := (w_{i-k}^{(j)})_{i=1, \dots, d_j, k=1, \dots, d_{j-1}}, \quad (5)$$

associated with a filter $w^{(j)}$ of filter length s and $d_j \in N$ given explicitly by

$$T^{(j)} = \begin{bmatrix} w_0^{(j)} & 0 & 0 & \dots & \dots & 0 \\ w_1^{(j)} & w_0^{(j)} & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ w_s^{(j)} & w_{s-1}^{(j)} & \dots & w_0^{(j)} & 0 & \dots & 0 \\ 0 & w_s^{(j)} & \dots & w_1^{(j)} & w_0^{(j)} & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \dots & \dots & 0 & w_s^{(j)} & \dots & \dots & w_0^{(j)} \\ \dots & \dots & \dots & 0 & w_s^{(j)} & \dots & w_1^{(j)} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & w_s^{(j)} & w_{s-1}^{(j)} \\ 0 & \dots & \dots & \dots & \dots & 0 & w_s^{(j)} \end{bmatrix} \in \mathbb{R}^{d_j \times d_{j-1}}. \quad (6)$$

A constraint is imposed on the bias vectors $\{b^{(j)}\}_{j=1}^J$ associated with the convolutional layers,

$$b_{s+1}^{(j)} = \dots = b_{d_j-s}^{(j)}, \quad j = 1, \dots, J. \quad (7)$$

Definition 1 (Deep Convolutional Neural Networks with Downsampling). Let $x = (x_1, x_2, \dots, x_d) \in \mathbb{S}^{d-1}$ represent the input data vector, and let $s \in \mathbb{N}$ denote the filter length. A downsampled DCNNs with mappings $\{h^{(j)} : \mathbb{R}^d \rightarrow \mathbb{R}^{d_j}\}_{j=1}^J$, incorporating downsampling operations at selected layers, is defined recursively. The layer widths $\{d_j\}_{j=1}^J$ are determined as follows: $d_0 = d$, $d_{J_1} = \lfloor \frac{d+J_1s}{D_1} \rfloor$, $d_{J_2} = \lfloor \frac{d_{J_1}+(J_2-J_1)s}{D_2} \rfloor$, $d_{J_k} = \lfloor \frac{d_{J_{k-1}}+(J_k-J_{k-1})s}{D_k} \rfloor$ and more generally,

$$d_j = d_{j-1} + s, \quad j \in \{1, 2, \dots, J\} \setminus \{J_1, J_2, \dots, J_k\}$$

is a sequence of vectors of functions defined iteratively by $h^{(0)} = x$. The downsampling operator $\mathfrak{D}_D : \mathbb{R}^{\hat{D}} \rightarrow \mathbb{R}^{\lfloor \hat{D}/D \rfloor}$, parameterized by scaling factor $D \leq \hat{D}$, is defined by

$$\mathfrak{D}_D(v) = (v_{iD})_{i=1}^{\lfloor \hat{D}/D \rfloor}, \quad v = (v_i)_{i=1}^{\hat{D}} \in \mathbb{R}^{\hat{D}},$$

where $\lfloor u \rfloor$ denotes the integer part of $u > 0$. Denote the activated affine mapping $\mathcal{C}_{T,b}$ induced by the connection matrix T and bias vector b as

$$\mathcal{C}_{T,b}(x) = \sigma(Tx - b),$$

and

$$h^{(j)}(x) = \begin{cases} \mathcal{C}_{T^{(j)},b^{(j)}}(h^{(j-1)}(x)) & \text{if } J_{k-1} < j < J_k, \\ \mathfrak{D}_{D_k} \circ \mathcal{C}_{T^{(j)},b^{(j)}}(h^{(j-1)}(x)) & \text{if } j = J_k, \end{cases}$$

where $J_0 = 0$, and let $\{D_1, D_2, \dots, D_k\}$ denote the downsampling scaling factors corresponding to the layers $\{J_1, J_2, \dots, J_k\}$. The bias vectors satisfy

$$(b^{(j)})_{s+1} = (b^{(j)})_{s+2} = \dots = (b^{(j)})_{d_{j-1}}, \quad j \neq \{J_1, J_2, \dots, J_k\}.$$

In this study, we refer to J_1 as the first group of convolutional layers, while $J_2 - J_1$ constitutes the second group, and in general, $J_k - J_{k-1}$ represents the k -th group of convolutional layers.

2.2. Spherical Harmonics and Sobolev Space on the Sphere

Let $\mathcal{H}_k^d$ represent the set of all spherical harmonics of degree k on $\mathbb{S}^{d-1}$. This space can be characterized as the eigenfunction corresponding to the Laplace-Beltrami operator Δ_0 on $\mathbb{S}^{d-1}$ (see [22], Chapter 1.4):

$$\mathcal{H}_k^d = \{F \in C^2(\mathbb{S}^{d-1}) : \Delta_0 F = -\lambda_k F\}, \quad (8)$$

where $\lambda_k = k(k+d-2)$. The dimension of the linear space $\mathcal{H}_k^d$ is

$$N(k, d) = \binom{k+d-1}{k} - \binom{k+d-3}{k-2} = O(k^{d-2}). \quad (9)$$

Let $\text{Proj}_k : L_2(\mathbb{S}^{d-1}) \rightarrow \mathcal{H}_k^d$ represent the orthogonal projection from $L_2(\mathbb{S}^{d-1})$ to $\mathcal{H}_k^d$. The spaces $\mathcal{H}_k^d$, where $k \in \mathbb{Z}_+$, are mutually orthogonal with respect to the inner product of $L_2(\mathbb{S}^{d-1})$ and possess a spherical harmonic expansion:

$$F = \sum_{k=0}^{\infty} \text{Proj}_k F = \sum_{k=0}^{\infty} \sum_{l=1}^{N(k,d)} \hat{F}_{k,l} Y_{k,l}, \quad (10)$$

where $\{Y_{k,l}\}_{l=1}^{N(k,d)}$ forms an orthonormal basis of $\mathcal{H}_k^d$, and $\hat{F}_{k,l}$ represents the Fourier coefficients of F , which are given by the expression:

$$\hat{F}_{k,l} = \langle F, Y_{k,l} \rangle_{L_2(\mathbb{S}^{d-1})} = \int_{\mathbb{S}^{d-1}} F(x) Y_{k,l}(x) d\mu(x). \quad (11)$$

The normalized spherical measure μ on $\mathbb{S}^{d-1}$ is defined by the surface area $\omega_d = \frac{2\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2})}$, such that $\mu(\mathbb{S}^{d-1}) = 1$.

In addition, $\text{Proj}_k F(x)$ has an integral representation given by:

$$\text{Proj}_k F(x) = \int_{\mathbb{S}^{d-1}} F(y) Z_k(x, y) d\mu(y), \quad x, y \in \mathbb{S}^{d-1}, \quad (12)$$

where

$$Z_k(x, y) = \sum_{l=1}^{N(k,d)} Y_{k,l}(x) Y_{k,l}(y), \quad x, y \in \mathbb{S}^{d-1}. \quad (13)$$

It can be easily demonstrated that $Z_k(x, y)$ is the reproducing kernel of $\mathcal{H}_k^d$ independent of the choice of $\{Y_{k,l}\}_{l=1}^{N(k,d)}$. Furthermore, with $\lambda = \frac{d-2}{2}$,

$$Z_k(x, y) = \frac{k+\lambda}{\lambda} C_k^\lambda(\langle x, y \rangle), \quad x, y \in \mathbb{S}^{d-1}, \quad (14)$$

where $C_k^\lambda(t)$ is the Gegenbauer polynomial of degree k with parameter $\lambda > \frac{-1}{2}$, as discussed, for instance, in [22]. Consequently, for any $x, y \in \mathbb{S}^{d-1}$, $\frac{k+\lambda}{\lambda} C_k^\lambda(\langle x, y \rangle)$ serves as a reproducing kernel of $\mathcal{H}_k^d$ in the sense that

$$\int_{\mathbb{S}^{d-1}} p(y) \frac{k+\lambda}{\lambda} C_k^\lambda(\langle x, y \rangle) d\mu(y) = p(x), \quad \forall p \in \mathcal{H}_k^d. \quad (15)$$

Definition 2 (Sobolev Space on Sphere). Let $3 \leq d \in \mathbb{N} \setminus \{0\}$, and for $r > 0$, the Sobolev space $W_\infty^r(\mathbb{S}^{d-1})$ is defined as

$$W_\infty^r(\mathbb{S}^{d-1}) := \left\{ F \in L_\infty(\mathbb{S}^{d-1}) : \|(-\Delta_0 + I)^{r/2} F\|_{L_\infty(\mathbb{S}^{d-1})} < \infty \right\}, \quad (16)$$

with the norm given by

$$\|F\|_{W_\infty^r(\mathbb{S}^{d-1})} := \|(-\Delta_0 + I)^{r/2} F\|_{L_\infty(\mathbb{S}^{d-1})} = \left\| \sum_{k=0}^{\infty} (1 + \lambda_k)^{\frac{r}{2}} \sum_{l=1}^{N(k,d)} \hat{F}_{k,l} Y_{k,l} \right\|_{L_\infty(\mathbb{S}^{d-1})}. \quad (17)$$

2.3. Near-Optimal Approximation and Cubature Formula

The optimal approximation of a function using polynomial spaces of varying degrees is generally nonlinear. In spherical harmonic analysis, a valuable method for approximation is the application of a linear operator, denoted as L_n .

Let $\eta \in C^\infty([0, \infty))$ with $\eta(t) = 1$ be a smooth function, such that $t \in [0, 1]$, $0 \leq \eta(t) \leq 1$ for $t \in [1, 2]$, and $\eta(t) = 0$ for $t \geq 2$. Define

$$k_n(t) = \sum_{k=0}^{2n} \left(\eta\left(\frac{k}{n}\right) \right)^2 \frac{\lambda + k}{\lambda} C_k^\lambda(t), \quad t \in [-1, 1],$$

and introduce a sequence of linear operators K_n , for $n \in \mathbb{Z}_+$, on the space $L_\infty(\mathbb{S}^{d-1})$ as follows:

$$K_n(F)(x) = \int_{\mathbb{S}^{d-1}} F(y) k_n(\langle x, y \rangle) d\mu(y), \quad x \in \mathbb{S}^{d-1}.$$

This integral operator is bounded and yields effective approximations for functions belonging to Sobolev spaces.

Proposition 1 ([19]). For $n \in \mathbb{N}$, $r > 0$ and $F \in W_\infty^r(\mathbb{S}^{d-1})$, there holds

$$\|F - K_n(F)\|_{L_\infty(\mathbb{S}^{d-1})} \leq C_\eta 2^{d-1} n^{-r} \|F\|_{W_\infty^r(\mathbb{S}^{d-1})},$$

where C_η represents a constant depending only on the function η . Furthermore, with a constant $C_{\eta,d} > 0$ depending on d and η ,

$$\|K_n(F)\|_{L_\infty(\mathbb{S}^{d-1})} \leq C_{\eta,d} \|F\|_{W_\infty^r(\mathbb{S}^{d-1})}.$$

To obtain a discrete representation of $K_n(F)$, a cubature formula is employed for integrating polynomials of degree up to $4n$ on the unit sphere $\mathbb{S}^{d-1}$, with $d \geq 3$ ([23], Theorem 3.1).

(Cubature Formula) There exist a constant $C_d > 0$ that depends only on d , such that for any $m \geq C_d n^{d-1}$, there exist positive constants γ_l and points $y_l \in \mathbb{S}^{d-1}$, for $l = 1, \dots, m$, satisfying the following relation:

$$\int_{\mathbb{S}^{d-1}} F(x) d\mu(x) = \sum_{l=1}^m \gamma_l F(y_l), \quad \forall F \in \Pi_{4n}(\mathbb{S}^{d-1}), \quad (18)$$

where $\Pi_{4n}(\mathbb{S}^{d-1})$ denotes to the space of polynomials of degree up to $4n$ on $\mathbb{S}^{d-1}$. Furthermore, for $F \in \Pi_{4n}(\mathbb{S}^{d-1})$, the following inequality holds:

$$\|F\|_{L_\infty(\mathbb{S}^{d-1})} \asymp \max_{l=1, \dots, m} n^{d-1} \gamma_l |F(y_l)|, \quad (19)$$

where $A \asymp B$ indicates that existence of $C_1 > 0$ and $C_2 > 0$ that are independent of n or m , such that $C_1 A \leq B \leq C_2 A$. In particular, we say that the family $\{(\gamma_l, y_l)\}_{l=1}^m$ follows a cubature rule of degree $4n$.

3. Approximating Functions with Multiple Features on Sphere

In practical deep learning applications, the input dimension d of the target function is often quite large. As d increases, the approximation rates generally worsen exponentially, leading to what is known

as the curse of dimensionality. This poses significant challenges when working with high-dimensional functions.

However, when the input variables are interdependent, meaning there are inherent structures within the target functions, DCNNs can help alleviate the curse of dimensionality by making use of these underlying relationships.

Our first target is to approximate composite function with general smooth features

$$f \in W_{\infty}^{\beta}([-B_F, B_F]^{d^*}), \quad \beta \in (0, 1]$$

with $B_F = \max_{\tau} \{\|F_{\tau}\|_{\infty}\}_{\tau=1}^{d^*}$:

$$f^*(x) = f(F_1(x), F_2(x), \dots, F_{d^*}(x)), \quad x \in \mathbb{S}^{d-1}, \quad (20)$$

where $d^* \leq d$, and $\{F_{\tau}(x)\}_{\tau=1}^{d^*} \in W_{\infty}^r(\mathbb{S}^{d-1})$ ($r > 0$). Since the outer function f is Lipschitz- β continuous, the norm of f is given by

$$\|f\|_{W_{\infty}^{\beta}([-B_F, B_F]^{d^*})} := \sup_{x \in [-B_F, B_F]^{d^*}} |f(x)| + \sup_{x \neq y \in [-B_F, B_F]^{d^*}} \frac{|f(x) - f(y)|}{|x - y|^{\beta}}. \quad (21)$$

Theorem 1. Let $2 \leq s \leq d$, $d \geq 3$, $d^* \leq d$, $N \in \mathbb{N}$, and let $\{F_{\tau}(x)\}_{\tau=1}^{d^*}$ be smooth functions. Define $B_F = \max\{\|F_{\tau}\|_{\infty}\}_{\tau=1}^{d^*}$, and let $f \in W_{\infty}^{\beta}([-B_F, B_F]^{d^*})$ with $\beta \in (0, 1]$. For approximating functions of the form (20), there exists a downsampled DCNNs as defined in Definition 1 (explicitly constructed as in Section 3.1), with downsampling scaling parameters $\{d, 1\}$ at layers $\{J_1, J_2\}$, respectively, and filters $\{w^{(j)}\}_{j=1}^{J_2}$ of uniform length s , bias vectors $\{b^{(j)}\}_{j=1}^{J_2+1}$, connection matrix $F^{(J_2+1)}$, and coefficient vector $c^{(J_2+1)} \in \mathbb{R}^N$, such that

$$\|c^{(J_2+1)} \cdot h^{(J_2+1)} - f^*\|_{L_{\infty}(\mathbb{S}^{d-1})} \leq c_1 \|f\|_{W_{\infty}^{\beta}} N^{-\frac{\beta}{d^*}}$$

where $c_1 = c_1(d^*, d, \beta, F_{\tau}, \eta)$, and $\eta \in C^{\infty}([0, \infty])$ defined in Section 2.3.

3.1. Proof of Theorem 1

Before the proof, we firstly introduce the two Lemmas for approximating inner and outer functions.

Define a kernel:

$$l_n(t) = \sum_{k=0}^{2n} \eta\left(\frac{k}{n}\right) \frac{\lambda + k}{\lambda} C_k^{\lambda}(t), \quad t \in [-1, 1], \quad (22)$$

which is a polynomial of degree $2n$. According to [19], combined with Proposition 1 and the cubature formula, we have:

$$\begin{aligned} \|F_{\tau} - K_n(F_{\tau})\|_{L_{\infty}(\mathbb{S}^{d-1})} &= \left\| F_{\tau} - \sum_{l=1}^m \gamma_l L_n(F_{\tau})(y_l) l_n(\langle y_l, \cdot \rangle) \right\|_{L_{\infty}(\mathbb{S}^{d-1})} \\ &\leq C_{\eta} 2^{d-1} n^{-r} \|F_{\tau}\|_{W_{\infty}^r(\mathbb{S}^{d-1})}, \end{aligned} \quad (23)$$

where

$$L_n(F_{\tau})(y_l) = \int_{\mathbb{S}^{d-1}} F_{\tau}(z) l_n(\langle y_l, z \rangle) d\mu(z),$$

and $\{(\gamma_l, y_l)\}_{l=1}^m$ follows a cubature rule of degree $4n$.

For $N_1 \in \mathbb{N}$, let $\mathbf{t} = \{t_i\}_{i=1}^{2N_1+3}$ be the uniform mesh on $[-1 - \frac{1}{N_1}, 1 + \frac{1}{N_1}]$ with $t_i = -1 + \frac{i-2}{N_1}$. Construct a linear operator I_t on $C[-1, 1]$ by

$$I_t(l_n)(u) = \sum_{i=2}^{2N_1+2} l_n(t_i) \delta_i(u), \quad u \in [-1, 1], \quad l_n \in C[-1, 1], \quad (24)$$

where $\delta_i \in C(\mathbb{R})$, $i = 2, \dots, 2N_1 + 2$, is given by

$$\delta_i(u) = N_1(\sigma(u - t_{i-1}) - 2\sigma(u - t_i) + \sigma(u - t_{i+1})). \quad (25)$$

To facilitate the analysis of the number of free parameters, we define a linear operator $\mathcal{L}_{N_1} : \mathbb{R}^{2N_1+1} \rightarrow \mathbb{R}^{2N_1+3}$, which acts on $\zeta = (\zeta_i)_{i=1}^{2N_1+1} \in \mathbb{R}^{2N_1+1}$ as follows:

$$(\mathcal{L}_{N_1}(\zeta))_i = \begin{cases} \zeta_2, & \text{for } i = 1, \\ \zeta_3 - 2\zeta_2, & \text{for } i = 2, \\ \zeta_{i-1} - 2\zeta_i + \zeta_{i+1}, & \text{for } 3 \leq i \leq 2N_1 + 1, \\ \zeta_{2N_1+1} - 2\zeta_{2N_1+2}, & \text{for } i = 2N_1 + 2, \\ \zeta_{2N_1+2}, & \text{for } i = 2N_1 + 3. \end{cases}$$

This operator $\mathcal{L}_{N_1}$ facilitates the representation of the approximation operator I_t on $C([-1, 1])$ in terms of $\{\sigma(\cdot - t_i)\}_{i=1}^{2N_1+3}$ as

$$I_t(l_n) = N_1 \sum_{i=1}^{2N_1+3} \left(\mathcal{L}_{N_1} \left(\{l_n(t_k)\}_{k=2}^{2N_1+2} \right) \right)_i \sigma(\cdot - t_i). \quad (26)$$

By [19,24], we have

$$\begin{aligned} & \left\| \sum_{l=1}^m \gamma_l L_n(F_\tau)(y_l) l_n(\langle y_l, \cdot \rangle) - \sum_{l=1}^m \gamma_l L_n(F_\tau)(y_l) I_t(l_n)(\langle y_l, \cdot \rangle) \right\|_\infty \\ & \leq \frac{C 3^{d+2} n^{d+3} C_1 C_{\eta,d}}{N_1^2} \|F_\tau\|_{W_\infty^r}, \end{aligned} \quad (27)$$

where $C > 0$ is a constant. Combining this with (23) and (27), we obtain the following Lemma:

Lemma 1. Let $d \geq 3$, $r > 0$, $n \in \mathbb{N}$, $N_1 \in \mathbb{N}$. There exist $\gamma_l \in \mathbb{R}$ and $y_l \in \mathbb{S}^{d-1}$, $l = 1, 2, \dots, m$, such that for any $F_\tau \in W_\infty^r(\mathbb{S}^{d-1})$ ($\tau = \{1, 2, \dots, d^*\}$),

$$\left\| F_\tau - \sum_{l=1}^m \gamma_l L_n(F_\tau)(y_l) I_t(l_n)(\langle y_l, \cdot \rangle) \right\|_\infty \leq \tilde{c} \|F_\tau\|_{W_\infty^r(\mathbb{S}^{d-1})} \max \left\{ n^{-r}, \frac{n^{d+3}}{N_1^2} \right\}, \quad (28)$$

where $\tilde{c} = C_\eta 2^{d-1} + C \cdot 3^{d+2} C_1 C_{\eta,d}$.

For approximating the outer function f of the form (20), we use the result of shallow sigmoidal neural networks approximation with scaling [35].

Lemma 2 ([35]). Let $d^*, N \in \mathbb{N}$, $\beta > 0$. For any $f \in W_\infty^\beta([-B_F, B_F]^{d^*})$, there exists $\{\hat{a}_k\}_{k=1}^N \in \mathbb{R}^{d^*}$, $\{\hat{b}_k\}_{k=1}^N \in \mathbb{R}$, $\{\hat{c}_k\}_{k=1}^N \in \mathbb{R}$, and a constant $C_{d^*,\beta}$ depending on d^* and β such that

$$\sup_{z \in [-B_F, B_F]^{d^*}} \left| f(z) - \sum_{k=1}^N \hat{c}_k \hat{\sigma}(\hat{a}_k \cdot z - \hat{b}_k) \right| \leq B_F^\beta C_{d^*,\beta} \|f\|_{W_\infty^\beta} N^{-\frac{\beta}{d^*}}, \quad (29)$$

where $\hat{\sigma}$ is the activation function of the sigmoidal type.

Proof of Theorem 1. For the approximate target functions of the form (20), we construct DCNNs based on a series of convolutional layers and a final fully connected layer. The convolutional transformations are represented by matrices $\{T^{(j)} := T^{w^{(j)}} \in \mathbb{R}^{(d_{j-1}+s) \times d_{j-1}}\}_{j=1}^{J_2}$, each induced by corresponding filter $\mathbf{w} := \{w^{(j)}\}_{j=1}^{J_2}$ each supported in $\{0, 1, \dots, s\}$. Associated with these layers are bias vectors

$\{b^{(j)} \in \mathbb{R}^{d_{j-1}+s}\}_{j=1}^{J_2}$. The network concludes with a fully connected layer $h^{(J_2+1)} : \mathbb{R}^{d_{J_2}} \rightarrow \mathbb{R}^N$ defined by a connection matrix $F^{(J_2+1)} \in \mathbb{R}^{N \times d_{J_2}}$ and a bias vector $b^{(J_2+1)} \in \mathbb{R}^N$, expressed as

$$h^{(J_2+1)}(x) = \hat{\sigma}(F^{(J_2+1)}h^{(J_2)}(x) - b^{(J_2+1)}),$$

where $\hat{\sigma}$ is the activation function of the sigmoidal type. The activation function $\hat{\sigma}$ that satisfies the assumptions:

$$\hat{\sigma}^{(i)}(u) \neq 0, \quad u \in \mathbb{R}, \quad i \in \mathbb{Z}_+,$$

and for some integer $q \neq 1$, we have $\lim_{u \rightarrow -\infty} \frac{\hat{\sigma}(u)}{|u|^q} = 0$ and $\lim_{u \rightarrow +\infty} \frac{\hat{\sigma}(u)}{|u|^q} = 1$.

The associated hypothesis space $\mathfrak{H}$, used for learning and function approximation, encompasses all output functions defined by the filter sequence $\mathbf{w} = \{w^{(j)}\}_{j=1}^{J_2}$, the connection matrix $F^{(J_2+1)}$, bias sequence $\mathbf{b} = \{b^{(j)}\}_{j=1}^{J_2+1}$, and a coefficient vector $c \in \mathbb{R}^N$, is given by:

$$\mathfrak{H} := \left\{ c^{(J_2+1)} \cdot h^{(J_2+1)}(x) : \mathbf{w}, \mathbf{b}, F^{(J_2+1)}, c^{(J_2+1)} \right\}.$$

We design the initial convolutional layer to extract linear features by employing a convolutional factorization approach [8,10], which facilitates the construction of ridge functions for approximating the smooth components $\{F_\tau(x)\}_{\tau=1}^{d^*}$. Let $s \geq 2$ and consider a sequence $U = (U_k)_{k=-\infty}^\infty$ supported on the index set $\{0, \dots, \mathcal{M}\}$ with $\mathcal{M} \leq 0$. Then, there exists a finite sequence of filters $\{w^{(j)}\}_{j=1}^p$, each supported in $\{0, \dots, s\}$, where $p \leq \lceil \frac{\mathcal{M}}{s-1} \rceil$, such that the following convolutional factorization is satisfied:

$$U = w^{(p)} * w^{(p-1)} * \dots * w^{(2)} * w^{(1)}. \quad (30)$$

For $m \in \mathbb{N}$ and $\mathbf{y} = \{y_1, \dots, y_m\} \subset \mathbb{S}^{d-1}$, we define U as a sequence supported on $\{0, \dots, md-1\}$, given by: $U_{(l-1)d+(d-k)-(y_l)_k}$ for $l \in \{1, \dots, m\}$ and $k \in \{1, \dots, d\}$. Let $\mathcal{M} = md-1$, $p \leq \lceil \frac{\mathcal{M}}{s-1} \rceil$. There exists a sequence of filters $\mathbf{w} = \{w^{(j)}\}_{j=1}^{J_1}$ supported on $\{0, \dots, s\}$ with $J_1 \geq \lceil \frac{\mathcal{M}}{s-1} \rceil$ that satisfies the convolutional factorization:

$$U = w^{(J_1)} * w^{(J_1-1)} * \dots * w^{(2)} * w^{(1)}.$$

Here, for $j = p+1, \dots, J_1$, we take $w^{(j)}$ to be a delta sequence, i.e., $\{1, 0, \dots, 0\}$:

$$T^{(J_1)} \dots T^{(2)} T^{(1)} = T^{(J_1,1)} = (U_{i-k})_{i=1, \dots, d+J_1s, k=1, \dots, d} \in \mathbb{R}^{(d+J_1s) \times d}, \quad (31)$$

where $T^{(j)}$ represents the Toeplitz matrix in (6).

The next step involves constructing the bias vectors in the neural network. We define this as:

$$\|w\|_1 = \sum_{k=-\infty}^{\infty} |w_k|, \quad (32)$$

For the first layer, the bias is set as:

$$b^{(1)} = -\|w\|_1 \mathbf{I}_{d_0}$$

and

$$b^{(j)} = \left(\prod_{p=1}^{j-1} \|w^{(p)}\|_1 \right) T^{(j)} \mathbf{I}_{d_{j-1}} - \left(\prod_{p=1}^j \|w^{(p)}\|_1 \right) \mathbf{I}_{d_j} \quad (33)$$

and for layers $j = 2, \dots, J_1$, the bias vectors are:

$$b_{s+1}^{(j)} = \dots = b_{d_j-s}^{(j)}. \quad (34)$$

Note that for $\|x\|_\infty \leq 1$ for $x \in \mathbb{S}^{d-1}$. Let $\|h\|_\infty = \max\{\|h_j\|_\infty : j = 1, \dots, q\}$ for a vector of functions $h : \mathbb{S}^{d-1} \rightarrow \mathbb{R}^q$. It is known that for $h : \mathbb{S}^{d-1} \rightarrow \mathbb{R}^{d_{j-1}}$, we have the following inequality:

$$\|T^{(j)}h\|_\infty \leq \|w^{(j)}\|_1 \|h\|_\infty, \quad (35)$$

Therefore, the components of $h^{(J_1)}(x)$ satisfy the following:

$$\left(h^{(J_1)}(x)\right)_{ld} = \langle y_l, x \rangle + B^{(J_1)}, \quad l = 1, \dots, m, \quad (36)$$

where $B^{(J_1)} = \Pi_{p=1}^{J_1} \|w^{(p)}\|_1$. We have

$$\mathfrak{D}_d\left(h^{(J_1)}(x)\right) = \begin{bmatrix} \langle y_1, x \rangle \\ \vdots \\ \langle y_m, x \rangle \\ 0 \\ \vdots \\ 0 \end{bmatrix} + B^{(J_1)} \mathbf{I}_{\lfloor (d+J_1s)/d \rfloor}. \quad (37)$$

Since $J_1 \geq \left\lceil \frac{md-1}{s-1} \right\rceil$, we obtain the following:

$$\frac{d+J_1s}{d} \geq 1 + \frac{md-1}{d} \frac{s}{s-1} > 1 + \frac{md-1}{d} \geq m, \quad (38)$$

Therefore, we have $d_{J_1} = \lfloor (d+J_1s)/d \rfloor \geq m$. To satisfy the constraint $J_1 \geq \left\lceil \frac{md}{s-1} \right\rceil$ for the DCNNs, and $m = \lfloor C_d + 1 \rfloor n^{d-1}$ for the cubature formula, we need to select $n = \left\lfloor \left(\frac{(s-1)J_1}{\lfloor C_d + 1 \rfloor d} \right)^{\frac{1}{d-1}} \right\rfloor$.

Considering that $|y_l \cdot x| \leq 1$, we define re exist two groups of convolutional layers with filters $\{w^{(j)}\}_{j=1}^{J_2}$ of uniform length s and biases $\{b^{(j)}\}_{j=1}^{J_2}$ that satisfy the restriction (7) for $j \neq J_2$. These layers are downsampled in the J_1 -th layer by a scale factor d , ensuring that the output of the J_2 -th layer, denoted $h^{(J_2)}(x) \in \mathbb{R}^{d_{J_2}}$ is given by:

$$\left(h^{(J_2)}(x)\right)_{(i-1)d_{J_1}+l} = \begin{cases} \sigma(y_l \cdot x - t_i) & \text{if } 1 \leq l \leq m, 1 \leq i \leq 2N_1 + 3, \\ 0 & \text{otherwise,} \end{cases} \quad (39)$$

where $J_1 \geq \left\lceil \frac{md-1}{s-1} \right\rceil$, $J_2 - J_1 \geq \left\lceil \frac{(2N_1+3)d_{J_1}}{s-1} \right\rceil$. So $d_{J_1} = \lfloor (d+J_1s)/d \rfloor \geq m$ and $d_{J_2} = d_{J_1} + (J_2 - J_1)s > (2N_1 + 3)d_{J_1} \geq (2N_1 + 2)d_{J_1} + m$. Therefore, we choose the bias vector $b^{(J_2)}$ as:

$$\begin{aligned} \left(b^{(J_2)}\right)_k &= B^{(J_1)} \left(\Pi_{p=J_1+1}^{J_2-1} \|w^{(p)}\|_1\right) \left(T^{(J_2)} \mathbf{I}_{d_{J_2}-1}\right)_k \\ &+ \begin{cases} t_i, & \text{if } (i-1)d_{J_1} + 1 \leq k \leq (i-1)d_{J_1} + m, 1 \leq i \leq 2N_1 + 3, \\ B^{(J_1)}, & \text{otherwise.} \end{cases} \end{aligned}$$

Finally, we can express $h^{(J_2)}(x)$ as:

$$h^{(J_2)}(x) = \left[H_1^T H_2^T \cdots H_{2N_1+3}^T 0 \cdots 0 \right]^T \in \mathbb{R}^{d_{J_2}}, \quad (40)$$

where

$$H_i^T = \left[\sigma(y_1 \cdot x - t_i) \sigma(y_2 \cdot x - t_i) \cdots \sigma(y_m \cdot x - t_i) 0 \cdots 0 \right] \in \mathbb{R}^{d_{J_1}}, \quad i = 1, 2, \dots, 2N_1 + 3. \quad (41)$$

We define $v_i = \left(\mathcal{L}_{N_1} \left(\{l_n(t_k)\}_{k=2}^{2N_1+2} \right) \right)_i$ for $i = 1, 2, \dots, 2N_1 + 3$. Additionally, we define the zero matrices: the $m \times (d_{J_1} - m)$ matrix as $\mathbf{O}$, and the $m \times (d_{J_2} - (2N_1 + 2)d_{J_1} - m)$ matrix as $\hat{\mathbf{O}}$. Then, we have:

$$F^{(N_1)} = N_1 \left[v_1 \mathbf{I}_m \mathbf{O} \ v_2 \mathbf{I}_m \mathbf{O} \ \cdots \ v_{2N_1+3} \mathbf{I}_m \hat{\mathbf{O}} \right] \in \mathbb{R}^{m \times d_{J_2}}. \quad (42)$$

Next, to obtain $\sum_{l=1}^m \gamma_l L_n(F_\tau)(y_l) \mathbf{I}_t(l_n)(\langle y_l, \cdot \rangle)$ ($\tau = 1, 2, \dots, d^*$), by (27), we obtain as follows:

$$F^{(\gamma)} = \begin{bmatrix} \gamma_1 L_n(F_1)(y_1) & \gamma_2 L_n(F_1)(y_2) & \cdots & \gamma_m L_n(F_1)(y_m) \\ \gamma_1 L_n(F_2)(y_1) & \gamma_2 L_n(F_2)(y_2) & \cdots & \gamma_m L_n(F_2)(y_m) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_1 L_n(F_{d^*})(y_1) & \gamma_2 L_n(F_{d^*})(y_2) & \cdots & \gamma_m L_n(F_{d^*})(y_m) \end{bmatrix} \in \mathbb{R}^{d^* \times m}. \quad (43)$$

Define the connection matrix $F^{(\hat{\mathbf{a}})} \in \mathbb{R}^{N_2 \times d^*}$ for the fully connected layer as the following:

$$F^{(J_2+1)} = F^{(\hat{\mathbf{a}})} F^{(\gamma)} F^{(N_1)} = \begin{bmatrix} \hat{a}_1 \\ \vdots \\ \hat{a}_{N_2} \end{bmatrix} F^{(\gamma)} F^{(N_1)}, \quad (44)$$

where $\hat{a}_k \in \mathbb{R}^{d^*}$ ($k = 1, 2, \dots, N_2$). Next, select the bias vector $b^{(J_2+1)} \in \mathbb{R}^{N_2}$ as $b^{(J_2+1)} = [\hat{b}_1, \hat{b}_2, \dots, \hat{b}_{N_2}]^T$. Additionally, define the coefficient vector $c^{(J_2+1)} \in \mathbb{R}^{N_2}$ as $c^{(J_2+1)} = [\hat{c}_1, \hat{c}_2, \dots, \hat{c}_{N_2}]^T$.

We refer to N and s as structural parameters because the architecture of the DCNNs is entirely defined by these two parameters. The remaining parameters are termed training parameters, as they can be trained or selected once s and N are specified.

The error rate for smooth features was established in Lemma 1. By setting $n = N_1^{\frac{2}{d+3+r}}$, we obtain the following expression

$$\left\| F_\tau - \sum_{l=1}^m \gamma_l L_n(F_\tau)(y_l) \mathbf{I}_t(l_n)(\langle y_l, \cdot \rangle) \right\|_\infty = \|F_\tau - \hat{F}_\tau\|_\infty \leq \tilde{c} \|F_\tau\|_{W_\infty^r(\mathbb{S}^{d-1})} N_1^{\frac{-2r}{d+3+r}}. \quad (45)$$

Denote $S(x) = (F_1(x), F_2(x), \dots, F_{d^*}(x))$ and $\hat{S}(x) = (\hat{F}_1(x), \hat{F}_2(x), \dots, \hat{F}_{d^*}(x))$. We have

$$\left| f(S(x)) - f(\hat{S}(x)) \right| \leq \|f\|_{W_\infty^\beta} \left(\tilde{c} \sqrt{d^*} \max_{\tau=1}^{d^*} \left\{ \|F_\tau\|_{W_\infty^r(\mathbb{S}^{d-1})} \right\} \right)^\beta N_1^{\frac{-2r\beta}{d+3+r}}, \quad (46)$$

where $\tilde{c} = C_\eta 2^{d-1} + C \cdot 3^{d+2} C_1 C_{\eta,d}$.

Using (46) and Lemma 2, the resulting error rates can be derived as follows:

Taking $N_2 = N$ and $N_1 = \left\lceil N_2^{\frac{d+3+r}{2rd^*}} \right\rceil$, we have

$$\begin{aligned} & \left| c^{(J_2+1)} \cdot h^{(J_2+1)}(x) - f(S(x)) \right| \\ & \leq \left| c^{(J_2+1)} \cdot h^{(J_2+1)}(x) - f(\hat{S}(x)) \right| + \left| f(\hat{S}(x)) - f(S(x)) \right| \\ & \leq B_F^\beta C_{d^*,\beta} \|f\|_{W_\infty^\beta} N_1^{\frac{-\beta}{d^*}} + \|f\|_{W_\infty^\beta} \left(\tilde{c} \sqrt{d^*} \max_{\tau=1}^{d^*} \left\{ \|F_\tau\|_{W_\infty^r(\mathbb{S}^{d-1})} \right\} \right)^\beta N_1^{\frac{-2r\beta}{d+3+r}} \\ & \leq c_1 \|f\|_{W_\infty^\beta} N_1^{\frac{-\beta}{d^*}}, \end{aligned}$$

where $c_1 = C_{d^*,\beta} B_F^\beta + \left(\tilde{c} \sqrt{d^*} \max_{\tau=1}^{d^*} \left\{ \|F_\tau\|_{W_\infty^r(\mathbb{S}^{d-1})} \right\} \right)^\beta$. $\square$

Remark 1. Since $N_1 = \left\lceil N^{\frac{d+3+r}{2d^*}} \right\rceil$, exponent part depend on input dimension d . For approximating functions in $W_\infty^\beta(\mathbb{S}^{d-1})$, the index of inner function r and the number of features d^* should satisfy the following condition:

$$\frac{d+3+r}{2r} \leq d^* < d,$$

such that the curse of dimensionality can be reduced.

We give a comparison with [10] in Table 1. Assume that the index of the outer functions β are equal. Our error rate is much faster when the index of the inner function r and the number of features d^* satisfy $\frac{d+3+r}{2r} \leq d^* < d$ ($d \geq 3$).

Table 1. Approximation rates of extracting features or not.

Regularity	Range	Error rate	features
$f^* \in W_\infty^\beta(\mathbb{S}^{d-1})$ [10]	$0 < \beta \leq 1$ [10]	$O(N^{\frac{-\beta}{d+1}})$ [10]	None [10]
$f^* \in W_\infty^\beta(\mathbb{S}^{d-1})$	$0 < \beta \leq 1$	$O(N^{-\frac{\beta}{d^*}})$	General Smooth

Theorem 1 concerns the approximation of composite functions in Sobolev spaces on the unit sphere $f^* \in W_\infty^\beta(\mathbb{S}^{d-1})$ ($\beta \in (0, 1]$). In contrast, related previous studies—such as the approximation of composite functions in $W_\infty^\beta([0, 1]^d)$ ($\beta \in (0, 1]$)—have been discussed in [20]. When $d^* = d$ and $\beta = 1$, meaning that there are d general smooth features on sphere, corresponding to that of approximating functions in d -dimension cube [11], the following corollary is obtained. It illustrates that the DCNNs we constructed can also approximate or represent functions in a cube.

Corollary 1. Let $2 \leq s \leq d$, $d \geq 3$, $N \in \mathbb{N}$. For approximating functions $f \in W_\infty^1([0, 1]^d)$, there exists a downsampled DCNNs as defined in Definition 1 (explicitly constructed as in Section 3.1), with downsampling scaling parameters $\{d, 1\}$ at layers $\{J_1, J_2\}$, respectively, and filters $\{w^{(j)}\}_{j=1}^{J_2}$ of uniform length s , bias vectors $\{b^{(j)}\}_{j=1}^{J_2+1}$, connection matrix $F^{(J_2+1)}$, and coefficient vector $c^{(J_2+1)} \in \mathbb{R}^N$, such that

$$\left\| c^{(J_2+1)} \cdot h^{(J_2+1)} - f \right\|_{L_\infty([0, 1]^d)} \leq c'_1 \|f\|_{W_\infty^\beta} N^{\frac{-1}{d}}$$

where $c'_1 = c'_1(d, \eta)$, and $\eta \in C^\infty([0, \infty])$ defined in Section 2.3.

3.2. Approximating Functions with Polynomial Features

The approximation of functions using polynomial features is also discussed in [20], where the domain is considered as $[0, 1]^d$. In Theorem 1, we derived the error bounds for functions with smooth features defined on the unit sphere and introduced the reproducing kernel $\frac{k+\lambda}{\lambda} C_k^\lambda(\langle x, y \rangle)$ associated with the space $\mathcal{H}_k^d$ in Section 2.2.

To reduce the curse of dimensionality, we leverage the reproducing kernel to approximate functions with polynomial features on the unit sphere. Specifically, we consider the function:

$$f^*(x) = f(P_{1,q_1}(x), P_{2,q_2}(x), \dots, P_{d^*,q_{d^*}}(x)) = f(P(x)), \quad x \in \mathbb{S}^{d-1} \quad (47)$$

where $\{P_{\tau,q_\tau}(x)\}_{\tau=1}^{d^*}$ denotes spherical polynomials of degree q_τ .

Define $B_P = \max\{\|P_{\tau,q_\tau}\|_\infty\}_{\tau=1}^{d^*}$, assume further that $f \in W_\infty^\beta([-B_P, B_P]^{d^*})$ ($\beta \in (0, 1]$). Recall the polynomial $k_n(t)$ introduced in Section 2.3., suppose $\max\{q_\tau\}_{\tau=1}^{d^*} \leq 2n$, we have the following Theorem:

Theorem 2. Let $2 \leq s \leq d$, $d \geq 3$, $d^* \leq d$, $N \in \mathbb{N}$, $n \in \mathbb{N}$. Let $\{P_{\tau,q_\tau}(x)\}_{\tau=1}^{d^*}$ be spherical polynomials of degree q_τ , satisfying $\max\{q_\tau\}_{\tau=1}^{d^*} \leq 2n$, $B_P = \max\{\|P_{\tau,q_\tau}\|_\infty\}_{\tau=1}^{d^*}$ and suppose $f \in W_\infty^\beta([-B_P, B_P]^{d^*})$ with

$\beta \in (0, 1]$. For approximating functions of the form (47), there exists a downsampled DCNNs, as defined in Definition 1 (explicitly constructed in Section 3.1), with downsampling scaling parameters $\{d, 1\}$ applied at layers $\{J_1, J_2\}$, respectively. The network uses filters $\{w^{(j)}\}_{j=1}^{J_2}$ of uniform length s , bias vectors $\{b^{(j)}\}_{j=1}^{J_2+1}$, a connection matrix $F^{(J_2+1)}$, and a coefficient vector $c^{(J_2+1)} \in \mathbb{R}^N$, such that

$$\left\| c^{(J_2+1)} \cdot h^{(J_2+1)} - f^* \right\|_{L_\infty(\mathbb{S}^{d-1})} \leq c_2 \|f\|_{W_\infty^\beta} \cdot n^{(d+3)\beta} \cdot N^{-\frac{\beta}{d^*}}$$

where $c_2 = c_2(d^*, d, \beta, B_P, n, \eta)$, and $\eta \in C^\infty([0, \infty])$ defined in Section 2.3.

Proof. Let $P_{\tau, q_\tau}(x)$ be a spherical polynomial of degree q_τ , where $q_\tau \leq 2n$. According to (15), we have:

$$P_{\tau, q_\tau}(x) = \int_{\mathbb{S}^{d-1}} P_{\tau, q_\tau}(y) k_n(\langle x, y \rangle) d\mu(y).$$

Applying (18) and (19), and setting $m = \lfloor C_d + 1 \rfloor n^{d-1}$. Since $P_{\tau, q_\tau}(y) k_n(\langle x, y \rangle)$ is a spherical polynomial in y of degree at most $4n$, there exist nodes $y_1, y_2, \dots, y_m \in \mathbb{S}^{d-1}$ and positive weights $\gamma_1, \gamma_2, \dots, \gamma_m > 0$ such that

$$P_{\tau, q_\tau}(x) = \sum_{l=1}^m \lambda_l P_{\tau, q_\tau}(y_l) k_n(\langle x, y_l \rangle), \quad x \in \mathbb{S}^{d-1}.$$

Similar to the proof of Theorem 1, we can derive the following error rates by selecting $N_1 = N_2 = N$:

$$\begin{aligned} & \left| c^{(J_2+1)} \cdot h^{(J_2+1)}(x) - f(P(x)) \right| \\ & \leq \left| c^{(J_2+1)} \cdot h^{(J_2+1)}(x) - f(\hat{P}(x)) \right| + \left| f(\hat{P}(x)) - f(P(x)) \right| \\ & \leq B_P^\beta C_{d^*, \beta} \|f\|_{W_\infty^\beta} N_2^{-\frac{\beta}{d^*}} + \|f\|_{W_\infty^\beta} \left(c' B_P \sqrt{d^*} \right)^\beta n^{(d+3)\beta} N_1^{-2\beta} \\ & \leq c_2 \|f\|_{W_\infty^\beta} n^{(d+3)\beta} N^{-\frac{\beta}{d^*}}, \end{aligned}$$

where $c_2 = B_P^\beta C_{d^*, \beta} + \left(c' B_P \sqrt{d^*} \right)^\beta$, and $c' = C \cdot 3^{d+2} C_1 C_{\eta, d}$.
□

Remark 2. Extracting spherical polynomial features can also derive the rates of approximating functions in $W_\infty^1([0, 1]^d)$ when taking $d^* = d$ and $\beta = 1$.

Remark 3. Compared with [20], our spherical polynomial features are not restricted to the same degree, require only less or equal to $2n$.

We provide a comparison with [10] in Table 2. Clearly, for same index β , our results provide faster error rates since d^* can be much smaller than d ($d \geq 3$). Unlike in the case of extracting general smooth features (Theorem 1), the index of the inner function r and the number of features d^* do not need to be limited to a specific range.

Table 2. Approximation rates of extracting features or not.

Regularity	Range	Error rate	features
$f^* \in W_\infty^\beta(\mathbb{S}^{d-1})$ [10]	$0 < \beta \leq 1$ [10]	$O(N^{-\frac{\beta}{d+1}})$ [10]	None [10]
$f^* \in W_\infty^\beta(\mathbb{S}^{d-1})$	$0 < \beta \leq 1$	$O(N^{-\frac{\beta}{d^*}})$	Spherical Polynomial

3.3. Approximating Functions with Symmetric Polynomial Features

Even though d^* much smaller than d , in practical applications of machine learning d^* is large. In order to further improve the rates of approximation, we consider the polynomial features $\{P_{\tau,q_\tau}(x)\}_{\tau=1}^{d^*}$ in (47), which exhibits symmetric structures with degree n . We denote these features by $\{Q_{\tau,n}(x)\}_{\tau=1}^{d^*}$, where $Q_{\tau,n}(x) = Q_{\tau,n}(x_1, x_2, \dots, x_d)$ and $x \in \mathbb{S}^{d-1}$. For any permutation in the symmetric group $Q_{\tau,n}$:

$$Q_{\tau,n}(x_1, \dots, x_i, \dots, x_j, \dots, x_d) = Q_{\tau,n}(x_1, \dots, x_j, \dots, x_i, \dots, x_d), \quad (48)$$

Therefore, the target function is of the form

$$f^*(x) = f(Q_{1,n}(x), Q_{2,n}(x), \dots, Q_{d^*,n}(x)), \quad x \in \mathbb{S}^{d-1}, \quad (49)$$

and $f \in W_\infty^\beta([-B_Q, B_Q]^{d^*})$ ($\beta \in (0, 1]$) with $B_Q = \max\{\|Q_{\tau,n}\|_\infty\}_{\tau=1}^{d^*}$.

Theorem 3. Let $2 \leq s \leq d$, $d \geq 3$, $n < d^* \leq d$, $N \in \mathbb{N}$, $n \in \mathbb{N}$, $\{Q_{\tau,n}(x)\}_{\tau=1}^{d^*}$ be a symmetric spherical polynomial of degree n , and $f \in W_\infty^\beta([-B_Q, B_Q]^{d^*})$ with $B_Q = \max\{\|Q_{\tau,n}\|_\infty\}_{\tau=1}^{d^*}$ and $\beta \in (0, 1]$. To approximate functions of the form (49), one can construct a downsampled DCNNs as described in Definition 1 (explicitly defined in Section 3.1), employing downsampling scaling parameters $\{d, 1\}$ at layers $\{J_1, J_2\}$, respectively. The network uses filters $\{w^{(j)}\}_{j=1}^{J_2}$ of uniform length s , bias vectors $\{b^{(j)}\}_{j=1}^{J_2+1}$, a connection matrix $F^{(J_2+1)}$, a coefficient vector $c^{(J_2+1)} \in \mathbb{R}^N$, such that

$$\left\| c^{(J_2+1)} \cdot h^{(J_2+1)} - f^* \right\|_{L_\infty(\mathbb{S}^{d-1})} \leq c_3 n^{(d+3)\beta} N^{-\frac{\beta}{n}}$$

where $c_3 = c_3(d^*, d, \beta, f, Q_{\tau,n}, n, \eta)$, and $\eta \in C^\infty([0, \infty])$ defined in Section 2.3.

Proof. Let the elementary symmetric polynomials on $x \in \mathbb{S}^{d-1}$ be denoted by

$$q_k(x) = \sum_{1 \leq i_1 < i_2 < \dots < i_k \leq d} x_{i_1} x_{i_2} \dots x_{i_k}, \quad k = 1, 2, \dots, d. \quad (50)$$

According to the fundamental theorem of symmetric polynomials, any symmetric polynomial can be uniquely expressed as a polynomial in the variables $(q_1(x), q_2(x), \dots, q_d(x))$. Given that the symmetric polynomials $\{Q_{\tau,n}(x)\}_{\tau=1}^{d^*}$ are of degree n , and $n \leq d$, they have a unique expression as a polynomial in $(q_1(x), q_2(x), \dots, q_n(x))$. Consequently, for $Q_{\tau,n}$ with $n \leq d^* \leq d$, there exists a unique polynomial $\tilde{Q}_{\tau,n}$ such that

$$Q_{\tau,n}(x) = \tilde{Q}_{\tau,n}(q_1(x), q_2(x), \dots, q_n(x)), \quad \tau = 1, 2, \dots, d^*. \quad (51)$$

Therefore, the target function $f^*(x)$ can expressed as

$$\begin{aligned} f^*(x) &= f(Q_{1,n}(x), Q_{2,n}(x), \dots, Q_{d^*,n}(x)) \\ &= f(\tilde{Q}_{1,n}(q_1(x), q_2(x), \dots, q_n(x)), \dots, \tilde{Q}_{d^*,n}(q_1(x), q_2(x), \dots, q_n(x))) \\ &= \tilde{f}(q_1(x), q_2(x), \dots, q_n(x)), \end{aligned} \quad (52)$$

for some function $\tilde{f}$ defined on $\mathbb{R}^n$ as

$$\tilde{f}(z) = f(\tilde{Q}_{1,n}(z), \dots, \tilde{Q}_{d^*,n}(z)), \quad z \in \mathbb{R}^n. \quad (53)$$

Since the polynomial $\tilde{Q}_{\tau,n} \in W_\infty^1([-B_q, B_q]^n)$ with semi-norm $|\tilde{Q}_{\tau,n}|_{W_\infty^1}$, and where $B_q = \max_k \{\|q_k\|_\infty\}_{k=1}^n$, it follows that for any $z_1 \neq z_2 \in [-B_q, B_q]^n$, we have:

$$\begin{aligned}
\tilde{f}(z_1) - \tilde{f}(z_2) &= f(\tilde{Q}_{1,n}(z_1), \dots, \tilde{Q}_{d^*,n}(z_1)) - f(\tilde{Q}_{1,n}(z_2), \dots, \tilde{Q}_{d^*,n}(z_2)) \\
&\leq \left(\sqrt{d^*} \max_{\tau} \{ |\tilde{Q}_{\tau,n}(z_1) - \tilde{Q}_{\tau,n}(z_2)| \} \}_{\tau=1}^{d^*} \right)^{\beta} \\
&\leq \left(\sqrt{d^*} \max_{\tau} \{ |\tilde{Q}_{\tau}|_{W_{\infty}^1} \}_{\tau=1}^{d^*} |z_1 - z_2| \right)^{\beta}.
\end{aligned} \tag{54}$$

Since $\{q_k(x)\}_{k=1}^n$ are spherical polynomials of degree up to n , let us denote

$$q(x) = (q_1(x), q_2(x), \dots, q_n(x)) \quad \text{and} \quad \hat{q}(x) = (\hat{q}_1(x), \hat{q}_2(x), \dots, \hat{q}_n(x)).$$

Similar to the proof of Theorem 2, by selecting $N_1 = N_2 = N$, we obtain:

$$\begin{aligned}
&\left| c^{(J_2+1)} \cdot h^{(J_2+1)}(x) - \tilde{f}(q(x)) \right| \\
&\leq \left| c^{(J_2+1)} \cdot h^{(J_2+1)}(x) - \tilde{f}(\hat{q}(x)) \right| + \left| \tilde{f}(\hat{q}(x)) - \tilde{f}(q(x)) \right| \\
&\leq B_q^{\beta} C_{d^*,\beta} \|f\|_{W_{\infty}^{\beta}} N_2^{-\frac{\beta}{n}} + \left(B_q \max_{\tau} \{ |\tilde{Q}_{\tau}|_{W_{\infty}^1} \}_{\tau=1}^{d^*} \sqrt{d^*} \right)^{\beta} (c' n^{d+3})^{\beta} N_1^{-2\beta} \\
&\leq B_q^{\beta} C_{d^*,\beta} \left(\left(B_q \max_{\tau} \{ |\tilde{Q}_{\tau}|_{W_{\infty}^1} \}_{\tau=1}^{d^*} \sqrt{d^*} \right)^{\beta} + \|f\|_{W_{\infty}^{\beta}} \right) N_2^{-\frac{\beta}{n}} \\
&\quad + \left(B_q \max_{\tau} \{ |\tilde{Q}_{\tau}|_{W_{\infty}^1} \}_{\tau=1}^{d^*} \sqrt{d^*} \right)^{\beta} (c' n^{d+3})^{\beta} N_1^{-2\beta}, \\
&\leq c_3 n^{(d+3)\beta} N^{-\frac{\beta}{n}},
\end{aligned} \tag{55}$$

where $c_3 = \left(B_q \max_{\tau} \{ |\tilde{Q}_{\tau}|_{W_{\infty}^1} \}_{\tau=1}^{d^*} \sqrt{d^*} \right)^{\beta} (c'^{\beta} + B_q^{\beta} C_{d^*,\beta}) + B_q^{\beta} C_{d^*,\beta} \|f\|_{W_{\infty}^{\beta}}$, and $c' = C \cdot 3^{d+2} C_1 C_{\eta,d}$.
□

4. Conclusions

This paper proposes the construction of DCNNs with multiple downsampled layers for approximating functions with multi-features—namely, general smooth features, polynomial features and symmetric polynomial features on the unit sphere. The result of Theorem 1 implies that if the index of the inner function r and the number of features d^* satisfy $\frac{d+3+r}{2r} \leq d^* < d$, the curse of dimensionality can be reduced. For further reducing the curse of dimensionality, we also derive error rates for cases where the features are either polynomials or symmetric polynomials on the unit sphere, as shown in Theorem 2 and 3. This implies that the DCNNs developed in Section 3.2 demonstrate the capability to extract three types of features: general smooth features, polynomial features, and symmetric polynomial features. In comparison, the DCNNs in [20] are limited to extracting only polynomial and symmetric polynomial features. In addition, compare to [20], when extracting polynomial features on sphere, our spherical polynomial features are not restricted to the same degree. Although our input vector x is spherical, we show that the DCNNs designed in Section 3.2 can also approximate or represent functions in $W_{\infty}^1([0, 1]^d)$ by extracting general smooth or spherical polynomial features, as shown in Corollary 1. These results underscore the superior learning capacity of the proposed DCNNs framework in capturing complex and high-dimensional feature representations on the unit sphere.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Goodfellow, I. ; Bengio, Y. and Courville, A. *Deep Learning*. MIT Press, **2016**.
2. Hinton, G. E. ; Osindero, S. and Teh, Y. W. A fast learning algorithm for deep belief nets. *Neural Computation*. **2006**, 18, 1527-1554.
3. Krizhevsky, A. ; Sutskever, I. and Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. **2017**, 60, 84-90.
4. DeVore, R. ; Hanin, B. and Petrova, G. Neural network approximation. *ACTA Numerica*. **2021**, 30, 327-444.
5. Elbrächter, D. ; Perekrestenko, D. ; Grohs, P. and Böelcskei, H. Deep Neural Network Approximation Theory. *IEEE Transactions on Information Theory*. **2021**, 67, 2581-2623.
6. Bartolucci, F. ; De Vito, E. ; Rosasco, L. and Vigogna, S. Understanding neural networks with reproducing kernel Banach spaces. *Applied and Computational Harmonic Analysis*. **2023**, 62, 194-236.
7. Song, L. H. ; Liu, Y. ; Fan, J. and Zhou, D. X. Approximation of smooth functionals using deep ReLU networks. *Neural Networks*. **2023**, 166, 424-436.
8. Zhou, D. X. Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*. **2020**, 48, 787-794.
9. Zhou, D. X. Theory of deep convolutional neural networks: Downsampling. *Neural Networks*. **2020**, 124, 319-327.
10. Fang, Z. Y. ; Feng, H. ; Huang, S. and Zhou, D. X. Theory of deep convolutional neural networks II: Spherical analysis. *Neural Networks*. **2020**, 131, 154-162.
11. Yarotsky, D. Error bounds for approximations with deep ReLU networks. *Neural Networks*. **2017**, 94, 103-114.
12. Zhou, D. X. Deep distributed convolutional neural networks: Universality. *Analysis and Applications*. **2018**, 16, 895-919.
13. Barron, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*. **1993**, 39, 930-945.
14. Klusowski, J. and Barron, A. R. Approximation by Combinations of ReLU and Squared ReLU Ridge Functions With ℓ^1 and ℓ^0 Controls. *IEEE Transactions on Information Theory*. **2018**, 64, 7649-7656.
15. Suzuki, T. Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality. *International Conference on Learning Representations*. New Orleans, United States, 2019.
16. Montanelli, H. and Du, Q. New Error Bounds for Deep ReLU Networks Using Sparse Grids. *SIAM Journal on Mathematics of Data Science*. **2019**, 1, 78-92.
17. Bach, F. Breaking the Curse of Dimensionality with Convex Neural Networks. *Journal of Machine Learning Research*. **2017**, 18, 19.
18. Bauer, B. and Kohler, M. On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *Annals of Statistics*. **2019**, 47, 2261-2285.
19. Feng, H. ; Huang, S. and Zhou, D. X. Generalization Analysis of CNNs for Classification on Spheres. *IEEE Transactions on Neural Networks and Learning Systems*. **2023**, 34, 6200-6213.
20. Mao, T. ; Shi, Z. J. and Zhou, D. X. Approximating functions with multi-features by deep convolutional neural networks. *Analysis and Applications*. **2023**, 21, 93-125.
21. Mao, T. ; Shi, Z. J. and Zhou, D. X. Theory of deep convolutional neural networks III: Approximating radial functions. *Neural Networks*. **2021**, 144, 778-790.
22. Dai, F. and Xu, Y. *Approximation Theory and Harmonic Analysis on Spheres and Balls*. Springer, **2013**.
23. Brown, G. and Dai, F. Approximation of smooth functions on compact two-point homogeneous spaces. *Journal of Functional Analysis*. **2005**, 220, 401-423.
24. De Boor, C. and Fix, G. J. Spline approximation by quasiinterpolants. *Journal of Approximation Theory*. **1973**, 8, 19-45.
25. Feng, H. ; Hou, S. Z. ; Wei, L. Y. and Zhou, D. X. CNN models for readability of chinese texts. *Mathematical Foundations of Computing*. **2022**, 5, 351-362.
26. Ahmed, R. ; Fahim, A. I. ; Islam, M. ; Islam, S. and Shatabda, S. Dolg-next: Convolutional neural network with deep orthogonal fusion of local and global features for biomedical image segmentation. *Neurocomputing*. **2023**, 546, 126362.
27. Silver, D. ; Schrittwieser, J. ; Simonyan, K. ; et al. Mastering the game of Go without human knowledge. *Nature*. **2017**, 550, 354-359.
28. Herrmann, L. ; Opschoor, J. and Schwab, C. Constructive Deep ReLU Neural Network Approximation. *Journal of Scientific Computing*. **2022**, 90, 75.

29. Cybenko,G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*. **1989**, 2, 303-314.
30. Hornik,K. ; Stinchcombe,M. and White,H. Multilayer feedforward networks are universal approximators. *Neural Networks*. **1989**, 2, 359-366.
31. DeVore, R. ; Howard, R. and Micchelli, C. Optimal nonlinear approximation. *Manuscripta Mathematica*. **1989**, 63, 469-478.
32. Mallat, S. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. **2016**, 374, 20150203.
33. Wiatowski, T.and Bölcskei, H. A Mathematical Theory of Deep Convolutional Neural Networks for Feature Extraction. *IEEE Transactions on Information Theory*. **2018**, 64, 1845-1866.
34. Mhaskar, H. and Poggio, T. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*. **2016**, 14, 829–848.
35. Mhaskar, H. N. and Micchelli, C. A. Approximation by superposition of sigmoidal and radial basis functions. *Advances in Applied Mathematics*. **1992**, 13, 350-373.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.