

# On the Approximation and Complexity of Deep Neural Networks to Invariant Functions

Gao Zhang<sup>1</sup>, Jin-Hui Wu<sup>2</sup>, Shao-Qun Zhang<sup>2\*</sup>

<sup>1</sup> *Institute of Mathematics, Nanjing Normal University, Nanjing, 210046, China*

<sup>2</sup> *National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China*

---

## Abstract

Recent years have witnessed a hot wave of deep neural networks in various domains; however, it is not yet well understood theoretically. A theoretical characterization of deep neural networks should point out their approximation ability and complexity, i.e., showing which architecture and size are sufficient to handle the concerned tasks. This work takes one step on this direction by theoretically studying the approximation and complexity of deep neural networks to invariant functions. We first prove that the invariant functions can be universally approximated by deep neural networks. Then we show that a broad range of invariant functions can be asymptotically approximated by various types of neural network models that includes the complex-valued neural networks, convolutional neural networks, and Bayesian neural networks using a polynomial number of parameters or optimization iterations. We also provide a feasible application that connects the parameter estimation and forecasting of high-resolution signals with our theoretical conclusions. The empirical results obtained on simulation experiments demonstrate the effectiveness of our method.

*Key words:* Deep Neural Networks, Theoretical Characterization, Invariant Functions, Approximation, Polynomial Complexity

---

## 1. Introduction

During the past decades, deep neural network (DNN) has become a mainstream model of machine learning in many real-world applications, such as computer vision [15], speech recognition [10, 26], and machine

---

\*Shao-Qun Zhang is the corresponding author. Email: zhangsq@lamda.nju.edu.cn

translation [2]. DNNs profit from the great approximation ability to exploit the structure of the concerned data, and different network architectures lead to different approximation abilities of the underlying data structure. For instance, in modeling high-resolution signals, the input data often comes in the form of a list of adjoint descriptions like the complex-valued formation, rotation invariance, and translation invariance. The goal of such tasks is to forecast some macroscopic quantities depending on these input specifications. In these cases, there are some invariant transformations with respect to permutation, rotation, and translation on the list of signals. These transformations can be regarded as specific groups, or equally sub-groups of the symmetric groups on the apposite coordinates of the feature vectors, and such transformations leave the underlying structure of the concerned signals or data invariant.

Some researchers have focused on real-world data with invariant structures, especially when using DNNs for such tasks. From the practical perspective, some studies attempted to reduce the complexity for approximating some rotation-invariant transformations using DNNs by exploiting data augmentation approaches [6, 21]. From the theoretical perspective, Yarotsky [31] proved the universal approximations of invariant maps by introducing specific architectures and symmetry groups. Similar results were also applied to some permutation invariant/equivariant functions by Sannai et al. [24]. Li et al. [18] provided general sufficient conditions for DNNs with various architectures to achieve universal approximation through dynamical systems. Despite the practical and theoretical progresses, in-depth analyses of approximation ability, especially complexity of DNNs, are far from clear.

In this paper, we theoretically investigate the approximation and complexity of DNNs to invariant functions. In comparison with previous studies, this work provides in-depth analysis of the approximation and optimization complexity of DNNs, especially answering which architecture and how many parameters are sufficient to approximate the concerned tasks. Our main contributions are summarized as follows:

- We show the universal approximation property of deep neural networks to invariant functions.
- We prove that a kind of rotation-invariant functions can be asymptotically approximated by the complex-valued neural networks within a polynomial number of parameters.
- We prove that a kind of translation-invariant functions can be asymptotically approximated by the convolutional neural networks within a polynomial number of parameters.
- We prove that a broad range of invariant functions can be universally approximated by Bayesian neural networks within polynomial optimization iterations.

- We also provide a feasible application that connects the parameter estimation and forecasting of high-resolution signals with our theoretical conclusions and demonstrate our theoretical results using simulation experiments.

The rest of this paper is organized as follows. Section 2 introduces some useful terminologies and related studies. Section 3 presents the universal approximation and useful properties of deep neural networks to invariant functions. Section 4 provides some theoretical results on complexity of various types of neural networks. Section 5 explores a feasible application, which shows that exploiting the complexity advantages of neural networks can strengthen the estimation and forecasting effectiveness of high-resolution signals. Section 6 concludes this work with discussions and prospects.

## 2. Preliminaries

This section first provides some useful notations, necessary terminologies, and seminal progresses related to our topic.

### 2.1. Notations and Terminologies

Let  $[N] = \{1, 2, \dots, N\}$  be an integer set for  $N \in \mathbb{N}^+$ , and  $|\cdot|_{\#}$  denotes the number of elements in a collection, e.g.,  $|[N]|_{\#} = N$ . Given two functions  $g, h: \mathbb{N}^+ \rightarrow \mathbb{R}$ , we denote by  $h = \Theta(g)$  if there exist positive constants  $c_1, c_2$ , and  $n_0$  such that  $c_1g(n) \leq h(n) \leq c_2g(n)$  for every  $n \geq n_0$ ;  $h = \mathcal{O}(g)$  if there exist positive constants  $c$  and  $n_0$  such that  $h(n) \leq cg(n)$  for every  $n \geq n_0$ ;  $h = \Omega(g)$  if there exist positive constants  $c$  and  $n_0$  such that  $h(n) \geq cg(n)$  for every  $n \geq n_0$ ;  $h = o(g)$  if for any  $c > 0$ , there exists a positive constant  $n_0$  such that  $h(n) < cg(n)$  for every  $n \geq n_0$ .

The general linear group over field  $\mathbb{F}$ , denoted by  $\mathbf{GL}(n, \mathbb{F})$ , is the set of  $n \times n$  invertible matrices with entries in  $\mathbb{F}$ . Especially, we define that a special linear group  $\mathbf{SL}(n, \mathbb{F})$  is the subgroup of  $\mathbf{GL}(n, \mathbb{F})$ , which consists of matrices with determinant 1. For any field  $\mathbb{F}$ , the  $n \times n$  orthogonal matrices form the following subgroup

$$\mathbf{O}(n, \mathbb{F}) = \{\mathbf{P} \in \mathbf{GL}(n, \mathbb{F}) \mid \mathbf{P}^{\top} \mathbf{P} = \mathbf{P} \mathbf{P}^{\top} = \mathbf{E}_n\}$$

of the general linear group  $\mathbf{GL}(n, \mathbb{F})$ . Similarly, we have the special orthogonal group, denoted as  $\mathbf{SO}(n, \mathbb{F})$ , which consists of all orthogonal matrices of determinant 1 and is a normal subgroup of  $\mathbf{O}(n, \mathbb{F})$ . Therefore, this group is also called the rotation group.

Let  $\mathcal{C}(K, \mathbb{R})$  be the set of all scalar functions  $f : K \rightarrow \mathbb{R}$ , which are continuous on  $K \subset \mathbb{R}^n$ . Given  $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)^\top \in \mathbb{N}^n$ , we define

$$D^\alpha f(\mathbf{x}) = \frac{\partial^{\alpha_1}}{\partial x^{\alpha_1}} \frac{\partial^{\alpha_2}}{\partial x^{\alpha_2}} \cdots \frac{\partial^{\alpha_n}}{\partial x^{\alpha_n}} f(\mathbf{x}),$$

where  $\mathbf{x} = (x_1, x_2, \dots, x_n) \in K$ . Further, we define

$$\mathcal{C}^l(K, \mathbb{R}) = \{f \in \mathcal{C}(K, \mathbb{R}) \mid D^\alpha f \in \mathcal{C}(K, \mathbb{R})\}$$

for  $\boldsymbol{\alpha} \in \mathbb{N}^n$  with  $|\boldsymbol{\alpha}| \triangleq \sum_{i \in [n]} \alpha_i \leq l$ . For  $1 \leq p$ , we define

$$\mathcal{L}^p(K, \mathbb{R}) = \left\{ f \in \mathcal{C}(K, \mathbb{R}) \mid \|f\|_{p,K} < \infty \right\},$$

where

$$\|f\|_{p,K} \triangleq \left( \int_K |f(\mathbf{x})|^p d\mathbf{x} \right)^{1/p}.$$

This work considers the Sobolev space  $\mathcal{W}^{l,p}(K, \mathbb{R})$ , defined as the collection of all functions  $f \in \mathcal{C}^l(K, \mathbb{R})$  and  $D^\alpha f \in \mathcal{L}^p(K, \mathbb{R})$  for all  $|\boldsymbol{\alpha}| \in [l]$ , that is,

$$\|D^\alpha f\|_{p,K} = \left( \int_K |D^\alpha f(\mathbf{x})|^p d\mathbf{x} \right)^{1/p} < \infty.$$

Notice that throughout this paper, we employ  $p = 2$  as the default of the above norms and abbreviate  $\|\cdot\|_2$  as  $\|\cdot\|$ .

## 2.2. Approximation and Complexity of Deep Neural Networks

The approximation is one of the key characteristics for investigating the theoretical understanding of deep neural networks despite over-parameterization. In approximation theory, both shallow and deep network models with non-polynomial activations are known to be capable of approximating any continuous function at an exponential cost [7, 9, 12, 17, 4], which provides a legitimate guarantee for neural networks.

Further, Lu et al. [19] and Sun et al. [25] showed the effects of width and depth of deep neural networks on approximation ability, respectively. Kidger and Lyons [14] provided a qualitative difference between deep narrow networks and shallow wide networks. The universal approximation property also holds for other types of neural network models, such as the complex-valued networks [29], convolutional neural networks [36], spiking neural networks [35], etc.

Beyond the universal approximation, the community also focuses on the approximation or computational complexity of neural networks, i.e., showing which architecture and size are sufficient to handle the concerned tasks. There are great efforts on this issue. Eldan and Shamir [8] proved the approximation complexity difference between two-layer and three-layer neural networks. Zhang and Zhou [33] showed the complexity advantages of the complex-reaction networks over the real-valued ones, and Wu et al. [30] extended these conclusions to proving that the FTNet possesses the complexity advantages over the real-valued feed-forward and recurrent neural networks. Zhang and Zhou [35] provided the theoretical support for polynomial approximation and computation complexity of spiking neural networks.

### 3. Approximation to Invariant Functions

This work considers the approximation of neural networks to invariant functions. Subsection 3.1 formally introduces the invariant function and its examples. Subsection 3.2 shows the universal approximation and two useful properties of deep neural networks to invariant functions.

#### 3.1. Invariant Functions

Firstly, we formally define invariant functions as follows.

**Definition 1 (Invariant Functions)** *Let  $\mathcal{G}$  be a concerned group on a set  $K \subset \mathbb{R}^n$ , in which  $\tau : \mathbb{R}^n \rightarrow \mathbb{R}^n$  for any  $\tau \in \mathcal{G}$ . The function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is said to be **invariant to group  $\mathcal{G}$** , or equally a  **$\mathcal{G}$ -invariant function**, if for any  $\tau \in \mathcal{G}$  and  $\mathbf{x} \in K$ , the following holds*

$$f(\tau(\mathbf{x})) = (f \circ \tau)(\mathbf{x}) = f(\mathbf{x}).$$

Next, we introduce some examples of invariant functions, which motivate us to build approximation analyses and applications of deep neural networks in the following sections.

**Example 1: Permutation.** In mathematics, a permutation  $\tau$  of a set  $K$  is a rearrangement of the elements in set  $K$ . Taking an example of  $K = \{1, 2, 3\}$ , there are six permutations (with all the possible orderings) of set  $K$ , namely  $(1, 2, 3)$ ,  $(1, 3, 2)$ ,  $(2, 1, 3)$ ,  $(2, 3, 1)$ ,  $(3, 1, 2)$ , and  $(3, 2, 1)$ . Obviously, the permutation operation  $\tau$  is a bijective function on set  $K$ . Furthermore, we denote by  $\mathcal{G}$  a permutation group whose elements are permutations of a given set  $K$ . The following group axioms hold for the permutation group  $\mathcal{G}$ :

- **Closure.** If  $\tau, \sigma \in \mathcal{G}$ , then  $\tau \circ \sigma \in \mathcal{G}$ , where  $\circ$  denotes the composition operation on  $\mathcal{G}$ .
- **Identity.** There is an identity permutation, denoted as  $id$  with  $id(x) = x$  for  $x \in K$ . For any  $\tau \in \mathcal{G}$ , one has  $id \circ \tau = \tau \circ id = \tau$ .
- **Associativity.** If  $\tau, \sigma, \pi \in \mathcal{G}$ , then  $\tau \circ \sigma \circ \pi = \tau \circ (\sigma \circ \pi)$ .
- **Invertibility.** For any  $\tau \in \mathcal{G}$ , there exists an inverse permutation  $\tau^{-1}$  such that

$$\tau^{-1} \circ \tau = \tau \circ \tau^{-1} = id .$$

Based on these axioms above, it is obvious that the permutation group  $\mathcal{G}$  on a set  $K$  is the symmetric group, and the elements in  $\mathcal{G}$  can be thought of as bijective functions from the set  $K$  to itself. Precisely, the function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is said to be invariant to permutation group  $\mathcal{G}$ , or equally a  $\mathcal{G}$ -permutation-invariant function, if for any  $\tau \in \mathcal{G}$  and  $\mathbf{x} \in K$ , the following holds  $f(\tau(\mathbf{x})) = (f \circ \tau)(\mathbf{x}) = f(\mathbf{x})$ .

**Example 2: Radial Function.** We say that  $f$  is a radial function if  $f(\mathbf{x}') = f(\mathbf{x})$  for any  $\|\mathbf{x}'\| = \|\mathbf{x}\|$ . Here, we regard  $\tau$  as a norm-preserving operation such that  $\tau(\mathbf{x}) = \mathbf{x}'$  with  $\|\mathbf{x}'\| = \|\mathbf{x}\|$ , and thus, both  $\tau$  and  $f$  obey Definition 1, i.e., radial functions lead to the rotation-invariant transformations. The radial function usually is implemented by data augmentation approaches for reducing the approximation complexity of some designed neural network models [6, 21].

**Example 3: Translation.** Let  $\Delta\mathbf{x} \in \mathbb{R}^n$  be a translation vector that corresponds to the concerned vector  $\mathbf{x} \in \mathbb{R}^n$ . We loosely define  $\tau(\mathbf{x}) = \mathbf{x} + \Delta\mathbf{x}$  to be a  $\Delta\mathbf{x}$ -translation of vector  $\mathbf{x}$ . For convenience, we usually limit the translation vector  $\Delta\mathbf{x}$  inside a unit ball  $\mathfrak{B}_1^n \triangleq \{\mathbf{y} \mid \|\mathbf{y}\| = 1\}$  and employ a scalar  $\eta$  to indicate its norm. Thus, we have

$$\tau(\mathbf{x}) = \mathbf{x} + \eta\Delta\mathbf{x}, \quad \text{for } \Delta\mathbf{x} \in \mathfrak{B}_1^n . \tag{1}$$

Similarly, we have the translation-invariance function  $f$  if

$$f(\tau(\mathbf{x})) = f(\mathbf{x} + \eta\Delta\mathbf{x}) = f(\mathbf{x}) .$$

### 3.2. Universal Approximation for Invariant Functions

This subsection presents our first theorem about the universal approximation to invariant functions as follows.

**Definition 2** *The function  $f(x) : \mathbb{R} \rightarrow \mathbb{R}$  is said to be  **$l$ -finite** if  $f$  is an  $l$ -times differentiable scalar function that satisfies*

$$0 < \left| \int_{\mathbb{R}} D^l f(x) dx \right| < \infty \quad \text{for } l \in \mathbb{N}^+ .$$

**Theorem 1** *Let  $K \subset \mathbb{R}^n$  be a compact set and  $\mathcal{G}$  denotes a concerned group. If the activation function  $\sigma_a$  is  $l$ -finite for some  $l \in \mathbb{N}^+$  and element-wise, then there exists a collection of parameters  $\{\mathbf{w} \in \mathbb{R}^{m \times 1}, \mathbf{v} \in \mathbb{R}^{n \times m}, b_w \in \mathbb{R}, \mathbf{b}_v \in \mathbb{R}^{m \times 1}\}$ , such that the set of functions of form*

$$f(\mathbf{x}) = \mathbf{w}^\top \sigma_a(\mathbf{v}^\top \mathbf{x} + \mathbf{b}_v) + b_w$$

*is dense in the set of the  $\mathcal{G}$ -invariant functions.*

Theorem 1 shows that the one-hidden-layer neural networks with apposite activation functions are universal approximators to the invariant functions that belong to  $\mathcal{C}(K, \mathbb{R})$ . The proof idea of this theorem follows the typical line of thought of universal approximation properties [12, 9, 14] of neural networks, and thus, we here omit the detailed proof of Theorem 1.

Further, we have a useful lemma as follows.

**Lemma 1** *Let  $\mathcal{G}$  be a concerned group on a compact set  $K \subset \mathbb{R}^n$ , in which  $\tau : \mathbb{R}^n \rightarrow \mathbb{R}^n$  for any  $\tau \in \mathcal{G}$ . For the concerned function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , if for arbitrary  $\epsilon > 0$  there exists a  $\mathcal{G}$ -invariant function  $g_\epsilon : \mathbb{R}^n \rightarrow \mathbb{R}$  such that  $\|f(\mathbf{x}) - g_\epsilon(\mathbf{x})\|_{p,K} \leq \epsilon$ , then  $f$  is also a  $\mathcal{G}$ -invariant function.*

Lemma 1 shows the transitivity of invariant functions, which can be used for identifying certain invariant functions and the corresponding operations. Notice that the concerned function  $g_\epsilon$  is related to the

approximation error  $\epsilon$ , which implies that it is possible to measure the functional structure (e.g., approximation complexity investigated in this work) of  $g_\epsilon$  using approximation errors provided  $f$ . Eldan et al. [8] and Zhang et al. [33] have studied the approximation complexity of the common-used fully-connected network (FCN).

**Lemma 2** [8, Theorem 1];[33, Theorem 2] *For the (Lebesgue) measurable activation function  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  with  $|\sigma(x)| \leq C(1 + |x|^\alpha)$  for all  $x \in \mathbb{R}$  and some constants  $C, \alpha > 0$ , there exist a probability measure  $\mu$  and a radial function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  such that there exists a constant  $\delta > 0$  such that*

$$\mathbb{E}_{\mathbf{x} \sim \mu} \left[ (f_R(\mathbf{x}) - f(\mathbf{x}))^2 \right] \geq \delta$$

for any one-hidden-layer FCN  $f_R : \mathbb{R}^d \rightarrow \mathbb{R}$  with exponential parameter number  $\mathcal{O}(de^d/\epsilon)$ .

Lemma 2 shows that a kind of radial functions cannot be approximated by FCNs with exponential ( $\mathcal{O}(C_1(d+1)e^{C_1(d)})$ ) parameters. In the next section, we present the in-depth analysis of the complexity of various types of neural networks.

## 4. Complexity

This section provides the theoretical complexity of various types of neural networks.

### 4.1. Complex-valued Neural Networks

We start this subsection by introducing the Complex-Valued Neural Network (CVNN). Let  $z = z_1 + z_2 i \in \mathbb{C}$  be a complex number where  $i = \sqrt{-1}$ ,  $z_1, z_2 \in \mathbb{R}$ ,  $[z]_{\mathbb{R}} = z_1$ . We denote by  $\bar{z} = z_1 - z_2 i$  and  $|z|^2 = z_1^2 + z_2^2$ . The basic building block of a CVNN can be formalized as

$$\sigma : \mathbb{C}^d \rightarrow \mathbb{C}, \quad \mathbf{z} \mapsto \sigma(\mathbf{w}^\top \mathbf{z}),$$

where  $\mathbf{w} \in \mathbb{C}^d$  and  $\sigma$  are complex-valued connection weights and the activation function, respectively. This work employs the functions with the following complex-homogeneity property as activations of CVNNs

$$\sigma(z) = \frac{\partial \sigma(z)}{\partial z} z \quad \text{and} \quad \sigma(\alpha z) = \alpha \sigma(z), \quad \alpha \in \mathbb{R},$$



such as the zReLU function [34, 28] defined as follows

$$\text{zReLU}(z) = \begin{cases} z, & \text{if } \theta_z \in [0, \pi/2] \cup [\pi, 3\pi/2], \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where  $\theta_z$  denotes the phase (i.e., angle) of the complex-valued scalar  $z$ . CVNNs also employ a pure linear connection as the final layer and extract the real part as the outputs. Therefore, we have established the CVNN, of which the expressive function is denoted by  $f_{\text{CVNN}} : \mathbb{C}^d \rightarrow \mathbb{R}$ . Next, we present the approximation complexity theorem for CVNNs as follows.

**Theorem 2** *For the zReLU activation function, there exist a probability measure  $\mu$ , a radial function  $g : \mathbb{C}^d \rightarrow \mathbb{R}$ , and some  $C_2 < d$  such that for any  $\epsilon > 0$ , there is a one-hidden-layer CVNN  $f_{\text{CVNN}} : \mathbb{C}^d \rightarrow \mathbb{R}$  with  $\mathcal{O}(d^{19/4}/\epsilon)$  parameters such that the following holds*

$$\mathbb{E}_{\mathbf{x} \sim \mu} [(f_{\text{CVNN}}(\mathbf{x}) - g(\mathbf{x}))^2] \leq \frac{\sqrt{3}}{C_2 d^{1/4}} + \epsilon.$$

Theorem 2 shows the polynomial asymptotic approximation complexity of one-hidden-layer CVNNs for approximating the radial functions. Combined with Lemma 2, we can conclude that a kind of radial functions can be asymptotically approximated by the one-hidden-layer CVNN of polynomial parameters, whereas such functions cannot be approximated by real-valued FCNs with exponential parameters.

This paper provides two ways of proving this theorem. Here, we only introduce the proof idea of the more interesting one (full proof is shown in Appendix B), and the other one can be accessed in Appendix C. The proof idea can be summarized as follows. The radial functions are invariant to rotations and dependent on the input norm, corresponding to the phase and norm (i.e., radial) of inputs, respectively. It is observed that the zReLU function in Eq. (2) comprises the radius and phase. Thus, there exist some linear connections (including rotation transformations) such that the combination of some hidden neurons is invariant to rotations, as shown in Lemma 3. Thus, there is a family of radial functions that can be well approximated by a one-hidden-layer CVNN.

We begin our proof of Theorem 2 with some useful lemmas.

**Lemma 3** *Let  $g : [r, R] \rightarrow \mathbb{R}$  be an  $L$ -Lipschitz function for  $0 \leq r \leq R$ . For any  $\epsilon > 0$ , there exist  $C_s \geq 1$ ,  $m \leq C_s R^2 L d / (\sqrt{r} \epsilon)$ , and an expressive function  $f_{\text{CVNN}}(\mathbf{x})$  led by a one-hidden-layer CVNN*

of  $m$  hidden neurons such that

$$\sup_{\mathbf{x} \in \mathbb{C}^d} |g(\|\mathbf{x}\|) - f_{\text{CVNN}}(\mathbf{x})| \leq \epsilon.$$

**Lemma 4** For  $d > C_2 > 0$ ,  $g : \mathbb{C}^d \rightarrow \mathbb{R}$  is an  $L$ -Lipschitz radial function supported on the set

$$\mathcal{S}_\Delta = \{\mathbf{x} \mid C_2\sqrt{d} \leq \|\mathbf{x}\| \leq 2C_2\sqrt{d}\}.$$

For any  $\epsilon > 0$ , there exists an expressive function  $f_{\text{CVNN}}(\mathbf{x})$  led by a one-hidden-layer CVNN with width at most  $C_s(C_2)^{3/2}Ld^{7/4}/\epsilon$  such that

$$\sup_{\mathbf{x} \in \mathbb{C}^d} |g(\mathbf{x}) - f_{\text{CVNN}}(\mathbf{x})| \leq \epsilon.$$

Lemma 4 shows that the  $L$ -Lipschitz radial functions can be approximated by the expressive function  $f_{\text{CVNN}}(\mathbf{x})$  led by a one-hidden-layer CVNN with polynomial parameters.

**Lemma 5** Define

$$g(\mathbf{x}) = \sum_{i=1}^N \beta_i g_i(\mathbf{x}) \quad \text{with} \quad g_i(\mathbf{x}) = \mathbb{I}\{\|\mathbf{x}\| \in \Omega_i\}, \quad (3)$$

where  $\mathbb{I}$  is an indicative function,  $\beta_i \in \{-1, +1\}$ ,  $N$  is a polynomial function of  $d$ , and  $\Omega_i$ 's are disjoint intervals of width  $\mathcal{O}(1/N)$  on values in the range  $\Theta(\sqrt{d})$ . For any  $\beta_i \in \{-1, +1\}$  ( $i \in [N]$ ), there exists a Lipschitz function  $h : \mathcal{S}_\Delta \rightarrow [-1, +1]$  such that

$$\int_{\mathbb{C}^d} (g(\mathbf{x}) - h(\mathbf{x}))^2 \phi^2(\mathbf{x}) \, d\mathbf{x} \leq \frac{3}{(C_2)^2 \sqrt{d}}.$$

Lemma 5 shows that any non-Lipschitz function  $g(\mathbf{x})$  can be approximated by a Lipschitz function with density  $\phi^2$ .

*Proof of Theorem 2.* Let  $g(\mathbf{x}) = \sum_{i=1}^N \beta_i g_i(\mathbf{x})$  be defined by Eq. (3) and  $N \geq 4C_2^{5/2}d^2$ . According to Lemma 5, there exists a Lipschitz function  $h$  with range  $[-1, +1]$  such that

$$\|h(\mathbf{x}) - g(\mathbf{x})\|_{L_2(\mu)} \leq \frac{\sqrt{3}}{C_2 d^{1/4}}.$$

Based on Lemmas 3 and 4, any Lipschitz radial function supported on  $\mathcal{S}_\Delta$  can be approximated by an expressive function  $f_{\text{CVNN}}(\mathbf{x})$  led by a one-hidden-layer CVNN with width at most  $C_3 C_s d^{15/4}/\epsilon$ , where

$C_3$  is a constant relative to  $C_2$  and  $\epsilon$ . Hence, one has

$$\sup_{\mathbf{x} \in \mathbb{C}^d} |h(\mathbf{x}) - f_{\text{CVNN}}(\mathbf{x})| \leq \epsilon,$$

and thus,

$$\|h(\mathbf{x}) - f_{\text{CVNN}}(\mathbf{x})\|_{L_2(\mu)} \leq \epsilon.$$

Hence, the range of  $f(\mathbf{x})$  is in  $[-1 - \epsilon, +1 + \epsilon] \subseteq [-2, +2]$  for  $\epsilon \leq 1$ . Provided the radial function, defined by Eq. (3), we have

$$\|g(\mathbf{x}) - f_{\text{CVNN}}(\mathbf{x})\|_{L_2(\mu)} \leq \|g(\mathbf{x}) - h(\mathbf{x})\|_{L_2(\mu)} + \|h(\mathbf{x}) - f_{\text{CVNN}}(\mathbf{x})\|_{L_2(\mu)} \leq \frac{\sqrt{3}}{C_2 d^{1/4}} + \epsilon.$$

This implies that given constants  $d > C_2 > 0$  and  $C_3 > 0$ , for any  $\epsilon > 0$  and  $\epsilon_i \in \{-1, +1\}$  ( $i \in [N]$ ), target radial function  $g$  can be approximated by an expressive function  $f_{\text{CVNN}}(\mathbf{x})$  led by a one-hidden-layer CVNN with range in  $[-2, +2]$  and width at most  $C_3 C_s d^{15/4} / \epsilon$ , that is,

$$\|g(\mathbf{x}) - f_{\text{CVNN}}(\mathbf{x})\|_{L_2(\mu)} \leq \frac{\sqrt{3}}{C_2 d^{1/4}} + \epsilon.$$

This completes the proof. □

## 4.2. Convolutional Neural Networks

This subsection investigates the approximation complexity of the convolutional neural network (CNN). Now, we present the theoretical conclusion.

**Theorem 3** *Let  $g : \mathbb{R}^d \rightarrow \mathbb{R}$  be the translation-invariant function. Provided a probability measure  $\mu$ , a translation scalar  $\eta > 0$  in Eq. (1), and  $C_2 < d$ , there exists some CNN, expressed by  $f_{\text{CNN}} : \mathbb{R}^d \rightarrow \mathbb{R}$ , with width at most  $\mathcal{O}(d^{17/4} / \epsilon)$  such that*

$$\mathbb{E}_{\mathbf{x} \sim \mu} [(f_{\text{CNN}}(\mathbf{x}) - g(\mathbf{x}))^2] \leq \frac{\sqrt{3}}{C_2 d^{1/4}} + \epsilon.$$

Theorem 3 shows the polynomial asymptotic approximation complexity (i.e.,  $\mathcal{O}(d^{17/4} / \epsilon)$ ) of CNNs with approximation to a kind of translation-invariant functions.

The proof idea of Theorem 3 is similar with that of Theorem 2. We start our proof with a re-formulation of the translation invariance. For the one-dimensional case,  $\tau(\mathbf{x}, \eta) = (\tau(x_1), \dots, \tau(x_d))^\top$ . Let  $\mathbf{w} = (w_1, \dots, w_l)^\top$  be the weighted vector where  $l < d$ , the convolution operation [18] in one dimension can be formulated as

$$\tau(\mathbf{x}; \mathbf{w}) = \left( \sum_{i=1}^l w_i x_i, \sum_{i=1}^l w_i x_{i+1}, \dots, \sum_{i=1}^l w_i x_{i+d-l} \right). \quad (4)$$

Therefore, one can convert the problem about translation-invariant approximation into a new learning problem that finds a collection of apposite weights to enable translation invariance. It suffices to show that there exists a kind of translation-invariant functions that can be approximated by the CNN expression within a polynomial function of  $l$ . We claim that the concerned polynomial function obeys the product of  $l \in \mathcal{O}(d^{17/4})$ . Our claim can be supported by the following lemmas.

**Lemma 6** *Let  $g : [r, R] \rightarrow \mathbb{R}$  be an  $L$ -Lipschitz function for  $0 \leq r \leq R$ . For any  $\epsilon > 0$ , there exist  $C_s \geq 1$ ,  $l \leq C_s R^2 L d / (\sqrt{r} \epsilon)$ , and an expressive function  $f_{\text{CNN}}(\mathbf{x})$  led by a one-hidden-layer CNN of  $l$  hidden neurons such that*

$$\sup_{\mathbf{x} \in \mathbb{C}^d} |g[\tau(\mathbf{x}, \eta)] - f_{\text{CNN}}(\mathbf{x})| \leq \epsilon.$$

**Lemma 7** *For  $d > C_2 > 0$ ,  $g : \mathbb{R}^d \rightarrow \mathbb{R}$  is an  $L$ -Lipschitz radial function supported on the set*

$$\mathcal{S}_\Delta = \left\{ \mathbf{x} : C_2 \sqrt{d} \leq \|\tau(\mathbf{x}, \eta) - \mathbf{x}\| \leq 2C_2 \sqrt{d} \right\}.$$

*For any  $\epsilon > 0$ , there exists an expressive function  $f_{\text{CNN}}(\mathbf{x})$  led by a one-hidden-layer CNN with width at most  $C_s (C_2)^{3/2} L d^{9/4} / \epsilon$  such that*

$$\sup_{\mathbf{x} \in \mathbb{R}^d} |g[\tau(\mathbf{x}, \eta)] - f_{\text{CNN}}(\mathbf{x})| \leq \epsilon.$$

**Lemma 8** *Define*

$$g(\mathbf{x}) = \sum_{i=1}^N \epsilon_i g_i(\mathbf{x}) \quad \text{with} \quad g_i(\mathbf{x}) = \mathbb{I}\{|\mathbf{x}| \in \Omega_i\}, \quad (5)$$

*where  $\epsilon_i \in \{-1, +1\}$ ,  $N$  is a polynomial function of  $d$ , and  $\Omega_i$ 's are disjoint intervals of width  $\mathcal{O}(1/N)$  on values in the range  $\Theta(\sqrt{d})$ . For any  $\epsilon_i \in \{-1, +1\}$  ( $i \in [N]$ ), there exists a Lipschitz function  $h : \mathcal{S}_\Delta \rightarrow [-1, +1]$  such that*

$$\int_{\mathbb{C}^d} (g(\mathbf{x}) - h(\mathbf{x}))^2 \phi^2(\mathbf{x}) \, d\mathbf{x} \leq \frac{3}{(C_2)^2 \sqrt{d}}.$$

*Proof of Theorem 3.* Similar to the proof procedure of Theorem 2, we can finish the proof of Theorem 3 by a straight combination of Lemmas 6, 7, and 8. Provided the setting that  $C_2 < d$  in Lemma 7, we can re-write the polynomial bound as

$$1 \leq C_s(C_2)^{3/2} L d^{9/4} / \epsilon \leq C_3 C_s d^{17/4} .$$

This completes the proof. □

### 4.3. Bayesian Neural Networks

This subsection studies the translation invariance of Bayesian neural networks (BNNs). For  $K \in \mathbb{N}^+$ , we first introduce an  $L$ -layer BNN as follows

$$\begin{cases} z^0 = \mathbf{x} \in \mathcal{D} , \\ z^l = \sigma(\mathbf{W}^l z^{l-1}), \quad \text{for } l \in [L] , \end{cases}$$

where the weight parameters  $\mathbf{w} \triangleq (\mathbf{W}^i)_{i \in [L]}$  are treated as random variables with a prior distribution  $p(\mathbf{w})$ . The BNN can be optimized by estimating the posterior  $p(\mathbf{w} \mid \mathcal{D})$  via the likelihood function  $h(\mathbf{w}; \mathcal{D})$ , or equally  $p(\mathcal{D} \mid \mathbf{w})$ . Conventionally, this optimization can be formulated as follows:

$$q^*(\mathbf{w}) = \arg \min_q \mathbb{E} [\ln q(\mathbf{w}) - \ln p(\mathbf{w} \mid \mathcal{D})] , \tag{6}$$

where the optimization function can be abbreviated as  $\text{KL}[q \parallel p(\cdot \mid \mathcal{D})]$ . Thus, one can solve Eq. (6) by maximizing a lower bound to the model evidence, that is,

$$\ln p(\mathcal{D}) - \text{KL}[q \parallel p(\cdot \mid \mathcal{D})] = \mathbb{E} \left[ \ln \frac{h(\mathbf{w}; \mathcal{D}) p(\mathbf{w})}{q(\mathbf{w})} \right] .$$

We can establish the BNN and have the following conclusion.

**Theorem 4** *The function induced by BNNs is translation-invariant, and the optimization complexity is less than  $\mathcal{O}(\alpha^{-T})$ , where  $\alpha > 1$  and  $T$  denotes the iteration step.*

Theorem 4 shows that the BNN has the translation-invariant property and maintain the lower optimization complexity than deep neural networks.

*Proof.* We start our proof with an over-parameterized linear probabilistic model, i.e.,  $y = \frac{1}{d}\mathbf{w}^\top \mathbf{x} + \epsilon$ , where both  $\mathbf{w}$  and  $\mathbf{x}$  are  $d$ -dimensional vectors,  $\epsilon \in \mathcal{N}(0, \sigma_y^2)$ . Theorem 4 considers the translation invariance, which motivates to ensure that using  $\mathbf{w} + \eta\Delta\mathbf{w}$  (just like Eq.(1)) has the same likelihood with using  $\mathbf{w}$ . It is obvious that there exists a vector  $\Delta\mathbf{w} \in \mathfrak{B}_1^{d-1}$ , such that the following equations hold

$$\begin{cases} \mathbf{w}^\top \mathbf{x} = (\mathbf{w} + \mathbf{B}\Delta\mathbf{w})^\top \mathbf{x} , \\ \mathbf{B} = \begin{bmatrix} \mathbf{I}_{(d-1)\times(d-1)} \\ -(x_1, \dots, x_{d-1})/x_d \end{bmatrix} , \end{cases}$$

which means that

$$\Delta\mathbf{w}^\top \mathbf{B}^\top \mathbf{x} = (x_1, \dots, x_{d-1}, x_d) \begin{bmatrix} \Delta\mathbf{w} \\ -\mathbf{x}_{d-1}^* \Delta\mathbf{w} \end{bmatrix} = 0 ,$$

where  $\mathbf{x}_{d-1}^* = (x_1, \dots, x_{d-1})/x_d$ . Notice that  $\eta$  is a singular value of the matrix  $\mathbf{B}$ . Provided a Gaussian prior  $\mathbf{w} \in \mathcal{N}(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$ , we can approximate the likelihood from

$$p(y | \mathbf{w}, \mathbf{x}) \approx q(\mathbf{w}) \propto \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x) . \quad (7)$$

Thus, it suffices to show the marginal  $q(\mathbf{w} + \mathbf{B}\Delta\mathbf{w})$  over  $\Delta\mathbf{w} \in \mathcal{N}(\mathbf{0}, \beta^2 \mathbf{I}_{(d-1)\times(d-1)})$ . For  $\beta \rightarrow \infty$ , we have

$$\begin{aligned} \int q(\mathbf{w} + \mathbf{B}\Delta\mathbf{w}) p(\Delta\mathbf{w}) d(\Delta\mathbf{w}) &= \int \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x) \cdot \mathcal{N}(\mathbf{0}, \beta^2 \mathbf{I}_{(d-1)\times(d-1)}) d(\Delta\mathbf{w}) \\ &= \mathcal{N}(\boldsymbol{\mu}_x + \mathbf{B} \mathbf{0}_{(d-1) \times 1}, \boldsymbol{\Sigma}_x + \beta^2 \mathbf{B}\mathbf{B}^\top) , \end{aligned}$$

where the second equality holds from [16, Theorem 1]. Thus, the inverse of the covariance matrix can be re-written as

$$\begin{aligned} (\boldsymbol{\Sigma}_x + \beta^2 \mathbf{B}\mathbf{B}^\top)^{-1} &= \boldsymbol{\Sigma}_x^{-1} - \beta^{-2} \boldsymbol{\Sigma}_x^{-1} \mathbf{B} (\mathbf{I}_{d-1} + \beta^2 \mathbf{B}^\top \boldsymbol{\Sigma}_x^{-1} \mathbf{B})^{-1} \mathbf{B}^\top \boldsymbol{\Sigma}_x^{-1} \\ &= \boldsymbol{\Sigma}_x^{-1} - \beta^{-2} \boldsymbol{\Sigma}_x^{-1} \mathbf{B} [\beta^2 (\beta^{-2} \mathbf{I}_{d-1} + \mathbf{B}^\top \boldsymbol{\Sigma}_x^{-1} \mathbf{B})]^{-1} \mathbf{B}^\top \boldsymbol{\Sigma}_x^{-1} \\ &= \boldsymbol{\Sigma}_x^{-1} - \boldsymbol{\Sigma}_x^{-1} \mathbf{B} (\beta^{-2} \mathbf{I}_{d-1} + \mathbf{B}^\top \boldsymbol{\Sigma}_x^{-1} \mathbf{B})^{-1} \mathbf{B}^\top \boldsymbol{\Sigma}_x^{-1} , \end{aligned}$$

where  $\mathbf{I}_{(d-1)\times(d-1)}$  is abbreviated as  $\mathbf{I}_{d-1}$ .

By building the likelihood model in Eq. (7), we can estimate the posterior as follows

$$p(\mathbf{w} | \mathcal{D}) \approx p(\mathbf{w} | \mathbf{x}) \propto q(\mathbf{w}) p(\mathbf{w}) = \mathcal{N}(\boldsymbol{\nu}, \boldsymbol{\Sigma}) , \quad (8)$$

where

$$\begin{cases} \boldsymbol{\nu} = \boldsymbol{\Sigma}_w(\boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_w)^{-1}\boldsymbol{\mu}_x + \boldsymbol{\Sigma}_x(\boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_w)^{-1}\boldsymbol{\mu}_w, \\ \boldsymbol{\Sigma} = \boldsymbol{\Sigma}_x(\boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_w)^{-1}\boldsymbol{\Sigma}_w, \end{cases}$$

which holds according to

$$(\boldsymbol{\Sigma}_w^{-1} + \boldsymbol{\Sigma}_x^{-1})^{-1} = \boldsymbol{\Sigma}_x(\boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_w)^{-1}\boldsymbol{\Sigma}_w = \boldsymbol{\Sigma}_w(\boldsymbol{\Sigma}_w + \boldsymbol{\Sigma}_x)^{-1}\boldsymbol{\Sigma}_x.$$

Combing Eq. (8) with Eq. (7), we have

$$\begin{aligned} p(\mathbf{w} | \mathbf{x}) &\propto q(\mathbf{w})p(\mathbf{w}) \\ &\propto \mathcal{N}(\boldsymbol{\Sigma}_x^{-1}\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x^{-1}) \cdot \mathcal{N}(\boldsymbol{\Sigma}_w^{-1}\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w^{-1}) \\ &= \mathcal{N}(\boldsymbol{\Sigma}_x^{-1}\boldsymbol{\mu}_x + \boldsymbol{\Sigma}_w^{-1}\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_x^{-1} + \boldsymbol{\Sigma}_w^{-1}) \\ &= \mathcal{N}(\boldsymbol{\nu}^*, \boldsymbol{\Sigma}^*), \end{aligned}$$

where

$$\begin{cases} \boldsymbol{\nu}^* = (\boldsymbol{\Sigma}_x^{-1} + \boldsymbol{\Sigma}_w^{-1})^{-1}(\boldsymbol{\Sigma}_x^{-1}\boldsymbol{\mu}_x + \boldsymbol{\Sigma}_w^{-1}\boldsymbol{\mu}_w), \\ \boldsymbol{\Sigma}^* = (\boldsymbol{\Sigma}_x^{-1} + \boldsymbol{\Sigma}_w^{-1})^{-1}. \end{cases}$$

Since  $\beta^2\mathbf{I}_{d-1}$  goes to zero as  $\beta \rightarrow \infty$ , thus one has

$$p_\infty(\mathbf{w} | \mathbf{x}) = \mathcal{N}(\boldsymbol{\nu}^{**}, \boldsymbol{\Sigma}^{**}),$$

where

$$\begin{cases} \boldsymbol{\nu}^{**} = (\boldsymbol{\Sigma}_\infty^{-1} + \boldsymbol{\Sigma}_w^{-1})^{-1}(\boldsymbol{\Sigma}_\infty^{-1}\boldsymbol{\mu}_x + \boldsymbol{\Sigma}_w^{-1}\boldsymbol{\mu}_w), \\ \boldsymbol{\Sigma}^{**} = (\boldsymbol{\Sigma}_\infty^{-1} + \boldsymbol{\Sigma}_w^{-1})^{-1}, \end{cases}$$

and

$$\boldsymbol{\Sigma}_\infty^{-1} = \boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\mathbf{B}(\mathbf{B}^\top\boldsymbol{\Sigma}_x^{-1}\mathbf{B})^{-1}\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}.$$

The above over-parameterized linear probabilistic model can be easily extended to the concerned BNNs.

Along this line, the problem that BNNs allowing the translation invariance coincides with solving the

following equations that model all translation invariance of BNNs at some data point  $\mathbf{x}$ :

$$\begin{cases} \mathbf{z}^0 = \mathbf{x} \in \mathcal{D}, \\ \mathbf{z}^l = \sigma((\mathbf{W}^l + \mathbf{B}^l \Delta \mathbf{W}^l) \mathbf{z}^{l-1}), \quad \text{for } l \in [L], \end{cases}$$

where  $d_l$  denotes the dimension of  $\mathbf{z}^l$  and

$$\mathbf{B}^l = \begin{bmatrix} \mathbf{I}_{(d_l-1) \times (d_l-1)} \\ -(z^1, \dots, z_{d_l-1}^l) / z_{d_l}^l \end{bmatrix}.$$

Similarly, the approximation likelihood can be calculated

$$\begin{cases} q(\mathbf{W}^l) = \mathbb{E}_{\mathbf{z}^{l-1}} [\mathcal{N}(\boldsymbol{\nu}_l^*, \boldsymbol{\Sigma}_l^*)] \\ p(\mathbf{z}^l) = \mathbb{E}_{\mathbf{z}^{l-1}} \left[ \mathbb{E}_{\mathbf{W}^l \in q(\mathbf{W}^l)} \left[ \sigma(\mathbf{W}^l \mathbf{z}^{l-1}) \right] \right] \end{cases} \quad (9)$$

for  $l \in [L]$ , where

$$\begin{cases} \boldsymbol{\nu}_l^* = \mathbf{u}_w + \frac{(\mathbf{z}^l)^\top (\boldsymbol{\mu}_x - \boldsymbol{\mu}_w) \boldsymbol{\Sigma}_w \mathbf{z}^l}{\boldsymbol{\Sigma}^\top (\boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_w) \mathbf{z}^l}, \\ \boldsymbol{\Sigma}_l^* = \boldsymbol{\Sigma}_w - \frac{\boldsymbol{\Sigma} (\mathbf{z}^l)^\top (\mathbf{z}^l)^\top \boldsymbol{\Sigma}^\top}{\boldsymbol{\Sigma}^\top (\boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_w) \mathbf{z}^l}. \end{cases}$$

It is observed that all the first and second moments (i.e.,  $\boldsymbol{\mu}_w, \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_w, \boldsymbol{\Sigma}_x$ ) are constrained to the parts of the overall weight vector that corresponds to  $\mathbf{W}^l$  and the expectation over  $\mathbf{z}^l$  is taken w.r.t.  $\mathbf{z}^l \in p(\mathbf{z}^{l-1})$  for  $l \in [L]$ .

Therefore, we have proved that the function induced by BNNs is translation-invariant. Besides, the above invariant approximation can be optimized using iterative algorithms layer by layer, as shown in Appendix D. Thus, it is obvious that the optimization complexity of BNNs are less than that of DNNs, that is,  $\mathcal{O}(\alpha^{-T})$  for  $\alpha > 1$  and the iteration step indicator  $T \in \mathbb{N}^+$ . This completes the proof.  $\square$

## 5. Applications

The previous section showed the polynomial (asymptotic approximation and optimization) complexity of various neural networks to the invariant function and its variants. Inspired by this recognition, we here present a feasible application, which shows that exploiting the complexity advantages of neural networks can strengthen the effectiveness of the parameter estimation and forecasting of high-resolution signals.



In general, we have some mild assumptions as follows:

**Assumption 1** *We pre-process the signal data to ensure the following settings:*

- i) There are  $n$  sources  $\{s_i\}_{i \in [n]}$  centered at frequency  $\omega_0$ ;*
- ii) Each source is a stationary zero-mean random process or deterministic signal;*
- iii) Additive noise  $\epsilon$  that belongs to a stationary zero-mean random process with covariance matrix  $\Sigma_{\text{noise}}$  is present at all  $2m$  sensors, where  $n \leq m$ .*

Based on the above assumptions, the signals received at the  $i$ -th doublet can be formulated as

$$\begin{cases} p_i(t) = \sum_{l=1}^n s_l(t) a_i(\theta_l) + \epsilon_{p,i}(t) , \\ q_i(t) = \sum_{l=1}^n s_l(t) e^{j\omega_0 \sin(\theta_l)\delta/c} + \epsilon_{q,i}(t) , \end{cases} \quad (10)$$

where  $\theta_l$  is now the direction-of-arrival frequency of the  $l$ -th source relative to the direction of the translation displacement  $\delta$ . Correspondingly, Eq. (10) has a vector-matrix formation

$$\begin{cases} \mathbf{p}(t) = \mathbf{A}\mathbf{s}(t) + \boldsymbol{\epsilon}_{p,\cdot}(t) , \\ \mathbf{q}(t) = \mathbf{A}\boldsymbol{\Phi}\mathbf{s}(t) + \boldsymbol{\epsilon}_{q,\cdot}(t) , \end{cases}$$

where  $\mathbf{p} \in \mathbb{R}^m$ ,  $\mathbf{s} \in \mathbb{R}^n$ ,  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , and

$$\boldsymbol{\Phi} = \text{diag}\{e^{j\omega_0 \sin(\theta_1)\delta/c}, \dots, e^{j\omega_0 \sin(\theta_n)\delta/c}\} .$$

Provided the observed signals, we estimate the parameters  $\mathbf{A}$  and  $\boldsymbol{\Phi}$ . Previously, there have been great efforts on this issue. The representative approaches contain the maximum likelihood (ML) method and maximum entropy (ME) method, which are often widely used but have certain limitations about the bias and sensitivity in parameter estimation [23]. Some researchers employed the statistical time series forecasting models, such as the auto-regressive moving average (ARMA) and auto-regressive integrated moving average (ARIMA) models [5], to handle this issue. Besides, Schmidt and Bienvenu [3] independently exploited the measurement methods in the case of sensor arrays of arbitrary form, of which the more famous one is the multiple signal classification (MUSIC) model. Roy and Kailath [22] presented the ESPRIT by

Table 1: Parameter complexity and performance (Little is better) of comparative models for the task of forecasting high-resolution signals with  $n = 10$  and  $m = 20$ .

Models	Settings	Paras	MSE
MUSIC	–	–	25.37
ESPRIT	–	–	23.14
VARIMA	$(d, p, \Delta, q) = (20, 10, 1, 5)$	$7.0 \times 10^2$	25.70
FCN	size(10,150,1)	$1.6 \times 10^3$	21.09
RNN	size(10,150,1)	$2.4 \times 10^4$	19.32
LSTM	size(10,150,1)	$8.2 \times 10^4$	15.87
Our Work	size(10,150,1)	$3.0 \times 10^4$	<b>13.33</b>
Our Work	size(10,50,1)	$7.1 \times 10^3$	16.62

introducing rotational invariance to improve the procedure of covariance estimation, which reduces the computation and storage consumption in comparison with conventional studies including MUSIC.

Our method here is to improve the effectiveness of the parameter estimation and forecasting of high-resolution signals. The basic idea behind our method is to exploit the rotation invariance of the underlying signal subspace induced by the translation invariance of the sensor array, and then, to employ the FTNet — a special implementation beyond the complex-valued neural network, presented by Zhang and Zhou [34] — to pursue the concerned signals.

To verify the effectiveness of our method, we conduct a simulation experiment with  $n = 10$  and  $m = 20$ . Table 1 lists the comparative results between the classical estimation approaches and our method. It is observed that our method performs better with fewer parameters, which supports our proposed Theorem 2 and recent advance [30, Theorem 3].

Finally, we also provide a discussion about the computational complexity. The primary computational advantage of ESPRIT is that it eliminates the search procedure inherent in all previous methods (ML, ME, MUSIC). ESPRIT produces signal parameter estimates directly in terms of (generalized) eigenvalues. As noted previously, this involves  $\Theta(n^3)$  computations. On the other hand, other high-resolution techniques require a search over the parameter space, in which the exhaustive search is computationally expensive. The significant computational advantage of ESPRIT becomes even more pronounced in multidimensional parameter estimation where the computational load of ESPRIT grows linearly with dimension, while that of MUSIC grows exponentially. Let  $r_i$  denote the number of vectors, the computation required to search over  $m$  dimensions for  $n$  parameter vectors is proportional to  $n \propto \prod_{i=1}^m r_i$ . The results in Table 1 coincide with the above analyses.

## 6. Conclusions, Discussions, and Prospects

In this paper, we present the theoretical understandings of deep neural networks to invariant functions from the perspectives of approximation and complexity. We first show the universal approximation of neural networks to the generalized invariant functions, and then conclude that a broad range of invariant functions can be asymptotically approximated by various types of neural network models using a polynomial number of parameters or optimization iterations. We also provide a feasible application that connects the parameter estimation and forecasting of high-resolution signals with our theoretical results, which demonstrates the effectiveness of our method using simulation experiments.

Theorems 2 and 3 show the advantages of the neural networks equipped with complex-valued or convolution operations in comparison with real-valued ones (from [33, Theorem 2]) via approximation complexity, where CVNNs and CNNs profit from unitary transformation and anti-symmetric multiplication, perhaps two most important characteristics of the complex-valued and convolutional neural networks. In Theorem 2, we adopt a considerably direct way, i.e., regard the relevance of the radius and phase to the target function as a prior (e.g., radial function) that imposes more constraints on a complex-valued neural network than a real-valued one, and thus, the unitary transformation that allows radius scaling and phase rotation (see Lemma 3) yields an advantageous reduction of the approximation complexity. This means that merely doubling the number of real-valued parameters (or neurons) in each layer does not give the equivalent effect observed in a complex-valued neural network, which is consistent with the conclusions in [11, 20]. Alternatively, a complex number  $z = z_1 + z_2 i$  can be written into a matrix form

$$\begin{pmatrix} z_1 & -z_2 \\ z_2 & z_1 \end{pmatrix} = z_1 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + z_2 \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

Combined with the convolution operations, one has

$$\begin{pmatrix} z_1 & -z_2 \\ z_2 & z_1 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} y_1 & -y_2 \\ y_2 & y_1 \end{pmatrix},$$

which yields an anti-symmetric multiplication [1] between the partial derivatives and independent variables, where  $a, b, c, d \in \mathbb{R}$  and  $y = y_1 + y_2 i$ . Once half of entries are known, the other half are fixed [13]. Such an adjoint relation also applies to reduce the complexity for approximating translation operations using convolutional neural networks (see Theorem 3).

In light of the preceding merits, we conjecture that the neural networks equipped with complex-valued or convolution operations have the potential and power of representing the invariant functions with rotational and translation operations. Hence, our work may provide solid support for designating the theoretical legitimacy [27] and characterization [30] of such network models. Besides, it would be interesting to theoretically study feature space transformation [37] and width-depth representation [32] which might be keys to understanding mysteries behind the success of deep neural networks.

This appendix provides the supplementary materials for our work “On the Approximation and Complexity of Deep Neural Networks to Invariant Functions”, constructed according to the corresponding sections therein.

### A. Full Proof of Lemma 1

Let  $\mathbf{x} \in K \subset \mathbb{R}^n$ ,  $\tau \in \mathcal{G}$ , and for arbitrary  $\epsilon > 0$   $g_\epsilon : \mathbb{R}^n \rightarrow \mathbb{R}$  is a  $\mathcal{G}$ -invariant function, satisfying that  $\|f(\mathbf{x}) - g_\epsilon(\mathbf{x})\|_{p,K} \leq \epsilon$ . Then we have

$$\begin{aligned} \|f(\mathbf{x}) - f(\tau(\mathbf{x}))\|_{p,K} &\leq \|f(\mathbf{x}) - g_\epsilon(\mathbf{x})\|_{p,K} + \|g_\epsilon(\mathbf{x}) - g_\epsilon(\tau(\mathbf{x}))\|_{p,K} + \|g_\epsilon(\tau(\mathbf{x})) - f(\tau(\mathbf{x}))\|_{p,K} \\ &\leq \epsilon + 0 + \epsilon = 2\epsilon . \end{aligned}$$

This completes the proof. □

### B. Full Proof of Theorem 2

There are two proof methods for Theorem 2, one based on the rotation group action and one based on the Fourier transformation, which are detailed in this and the following sections, respectively.

#### B.1. Proof of Lemma 3

Let  $f' : \mathbb{R}^n \rightarrow \mathbb{R}$  be a radial function with  $f'(\mathbf{x}) = \|\mathbf{x}\|$ . According to the universal approximation theorems of CVNNs [29], such as [33, Theorem 1], for any  $\epsilon > 0$ ,  $i \in [n]$ , and  $d \geq 1$ , one has

$$\sup_{\mathbf{x} \in \mathbb{C}^d} \left| f'(\mathbf{x}) - [\sigma(\mathbf{v}_i^\top \mathbf{x} + b_i)]_{\mathbb{R}} \right| \leq \epsilon/2 , \quad (11)$$

where  $\mathbf{v}_i \in \mathbb{C}^n$  and  $b_i \in \mathbb{C}$ . Further, we define a new function  $g' : \mathbb{R} \rightarrow \mathbb{R}$  as follows

$$g'(s) = \left[ \sum_{i=1}^{n'} \alpha'_i \sigma(s) + a' \right]_{\mathbb{R}},$$

where  $\alpha'_i, a' \in \mathbb{R}$ . For Lipschitz continuous function  $\sqrt[k]{\cdot}$  and from [33, Lemma 1], we have

$$\sup_{s \in [r^k, R^k]} |g(\sqrt[k]{s}) - g'(s)| \leq \epsilon/2, \quad (12)$$

where  $n' \leq C' L(R^k - r^k)/(\sqrt[k]{r}\epsilon)$  for some constant  $C' > 0$  and integer  $k \geq 2$ . Further, we have

$$|g'(s) - f_{\text{CVNN}}(\mathbf{x})| \leq |g'(s) - f'(\mathbf{x})| + |f'(\mathbf{x}) - f_{\text{CVNN}}(\mathbf{x})|, \quad (13)$$

where  $s = [\mathbf{v}_i^\top \mathbf{x} + b_i]_{\mathbb{R}}$  and

$$f'(\mathbf{x}) = \left[ \sum_{i=1}^{n'} w'_i \sigma(\mathbf{v}_i^\top \mathbf{x} + b_i) + a \right]_{\mathbb{R}},$$

in which  $\{w'_i, a\}$  denotes another collection of weighted parameters that corresponds to

$$f_{\text{CVNN}}(\mathbf{x}) = \left[ \sum_{i \in [n]} w_i \sigma(\mathbf{v}_i^\top \mathbf{x} + b_i) + a \right]_{\mathbb{R}},$$

The first term of Eq. (13) can be bounded by  $\epsilon/4$  from [33, Lemma 1] for any  $s \in [r^k, R^k]$ . The second term is at most  $\epsilon/4$  when  $m \geq m'$  from Eq. (11). This follows that

$$|g'(s) - f_{\text{CVNN}}(\mathbf{x})| \leq \epsilon/2. \quad (14)$$

Combining with Eqs. (12) and (14), we have

$$|g(\|\mathbf{x}\|) - f_{\text{CVNN}}(\mathbf{x})| \leq |g(\sqrt[k]{s}) - g'(s)| + |g'(s) - f_{\text{CVNN}}(\mathbf{x})| \leq \epsilon,$$

where  $\mathbf{x} \in \mathbb{C}^d$  and  $s \in [r^k, R^k]$ . We finally obtain

$$m \leq C_s (R^k - r^k) L d / (\sqrt[k]{r}\epsilon),$$

provided  $m \leq dm'$  and  $C' \leq C_s$ . Finally, we can complete the proof by setting  $k = 2$  in the above upper bound.  $\square$

### B.2. Proof of Lemma 4

Let  $r = C_2\sqrt{d}$ ,  $R = 2C_2\sqrt{d}$ , and  $d \geq 1$ , then we have  $r \geq 1$ , which satisfies the condition of [33, Lemma 1]. Invoke Lemma 3 to construct the concerned CVNN and define  $\epsilon' \leq \epsilon/d$ . Then for any  $L$ -Lipschitz radial function  $g : \mathbb{C}^d \rightarrow \mathbb{R}$  supported on  $\mathcal{S}_\Delta$ , we have

$$\sup_{\mathbf{x} \in \mathbb{C}^d} |g(\mathbf{x}) - f_{\text{CVNN}}(\mathbf{x})| \leq \epsilon' ,$$

where the width of the hidden layer is bounded by

$$m \leq \frac{C_s(C_2)^{3/2}dL}{\epsilon}d^{3/4} \leq \frac{C_s(C_2)^{3/2}L}{\epsilon}d^{7/4} .$$

This completes the proof.  $\square$

### A.3. Proof of Lemma 5

Define a branch function

$$h_i(\mathbf{x}) = \begin{cases} \max\{\mathbb{I}\{\|\mathbf{x}\| \in \Omega_i\}, ND_i\}, & \text{if } B_i = 1, \\ 0 & , \text{ if } B_i = 0, \end{cases}$$

where  $B_i$  denotes a binary indicator and

$$D_i = \min \left\{ \left| \|\mathbf{x}\| - \left(1 + \frac{i-1}{N}\right) C_2\sqrt{d} \right|, \left| \|\mathbf{x}\| - \left(1 + \frac{i}{N}\right) C_2\sqrt{d} \right| \right\} .$$

Let  $h(\mathbf{x}) = \sum_{i=1}^N \beta_i h_i(\mathbf{x})$ , where  $\beta_i \in \{-1, +1\}$ ,  $B_i = 1$ ,  $\Omega_i$ 's are disjoint intervals,  $h_i(\mathbf{x})$  is an  $N$ -Lipschitz function. Thus,  $h$  is also an  $N$ -Lipschitz function. So we have

$$\begin{aligned} \int_{\mathbb{C}^d} \left( h(\mathbf{x}) - \sum_{i=1}^N \epsilon_i g_i(\mathbf{x}) \right)^2 \phi^2(\mathbf{x}) \, d\mathbf{x} &= \int_{\mathbb{C}^d} \sum_{i=1}^N \epsilon_i^2 (h_i(\mathbf{x}) - g_i(\mathbf{x}))^2 \phi^2(\mathbf{x}) \, d\mathbf{x} \\ &= \sum_{i=1}^N \int_{\mathbb{C}^d} (h_i(\mathbf{x}) - g_i(\mathbf{x}))^2 \phi^2(\mathbf{x}) \, d\mathbf{x} \\ &\leq \frac{3}{(C_2)^2 \sqrt{d}}, \end{aligned}$$

where the last inequality holds from [8, Lemma 22]. This completes the proof.  $\square$

Finally, we can complete the proof of Theorem 2.

*Finishing the Proof of Theorem 2.* Let  $g(\mathbf{x}) = \sum_{i=1}^N \epsilon_i g_i(\mathbf{x})$  be defined by Eq. (3) and  $N \geq 4C_2^{5/2} d^2$ . According to Lemma 5, there exists a Lipschitz function  $h$  with range  $[-1, +1]$  such that

$$\|h(\mathbf{x}) - g(\mathbf{x})\|_{L_2(\mu)} \leq \frac{\sqrt{3}}{C_2 d^{1/4}}.$$

Based on Lemmas 3 and 4, any Lipschitz radial function supported on  $S_\Delta$  can be approximated by an expressive function  $f_{\text{CVNN}}(\mathbf{x})$  led by a one-hidden-layer CVNN with width at most  $C_3 C_s d^{15/4} / \epsilon$ , where  $C_3$  is a constant relative to  $C_2$  and  $\epsilon$ . This means that

$$\sup_{\mathbf{x} \in \mathbb{C}^d} |h(\mathbf{x}) - f_{\text{CVNN}}(\mathbf{x})| \leq \epsilon.$$

Thus, we have

$$\|h(\mathbf{x}) - f_{\text{CVNN}}(\mathbf{x})\|_{L_2(\mu)} \leq \epsilon.$$

Hence, the range of  $f_{\text{CVNN}}(\mathbf{x})$  is in  $[-1 - \epsilon, +1 + \epsilon] \subseteq [-2, +2]$  given  $\epsilon \leq 1$ . For the radial function in Eq. (3), we have

$$\|g(\mathbf{x}) - f_{\text{CVNN}}(\mathbf{x})\|_{L_2(\mu)} \leq \|g(\mathbf{x}) - h(\mathbf{x})\|_{L_2(\mu)} + \|h(\mathbf{x}) - f_{\text{CVNN}}(\mathbf{x})\|_{L_2(\mu)} \leq \frac{\sqrt{3}}{C_2 d^{1/4}} + \epsilon.$$

This implies that given constants  $m > C_2 > 0$  and  $C_3 > 0$ , for any  $\epsilon > 0$  and  $\beta_i \in \{-1, +1\}$  ( $i \in [N]$ ), the target radial function  $g$  can be approximated by an expressive function  $f_{\text{CVNN}}(\mathbf{x})$  led by a one-hidden-

layer CVNN with range in  $[-2, +2]$  and width at most  $C_3 C_s d^{15/4} / \epsilon$ , that is,

$$\|g(\mathbf{x}) - f_{\text{CVNN}}(\mathbf{x})\|_{L_2(\mu)} \leq \frac{\sqrt{3}}{C_2 d^{1/4}} + \epsilon.$$

This completes the proof.  $\square$

### C. Another Proof for Theorem 2

The proof idea can be summarized as follows. Given any permutation operation on the input channels, there are some invariant properties on the complex-valued operations. In other words, the expressive function led by a one-hidden-layer CVNN admits the rotation transformations. Therefore, it suffices to show that before final weighted aggregation, the component expressive functions of hidden neurons can approximate any  $L$ -Lipschitz radial function. Then we can find a special radial function that can be well approximated by these hidden expressive functions within the polynomial parameter complexity.

**Lemma 9** *Let  $\sigma$  and  $f_{\text{CVNN}}(\mathbf{x})$  be an  $l$ -finite function and the expressive function led by a one-hidden-layer CVNN, respectively. For any  $\epsilon > 0$  and  $\mathbf{A} \in \mathbf{SO}(n, \mathbb{R})$ , there exists some time such that*

$$|f_{\text{CVNN}}(\mathbf{A}\mathbf{x}) - \|\mathbf{x}\| \leq \epsilon.$$

**Lemma 10** *Let  $\sigma$  be an  $l$ -finite function,  $g : \mathbb{R} \rightarrow \mathbb{R}$  is a scalar function, and  $f_{\text{CVNN}}(\mathbf{x})$  is the expressive function led by a one-hidden-layer CVNN, where  $\mathbf{x} \in \mathcal{S}(r)$  and  $\mathcal{S}(r)$  is a sphere supported with density  $\phi^2$  for  $0 < r < \infty$ . For any  $\epsilon > 0$ , it holds*

$$|f_{\text{CVNN}}(\mathbf{x}) - g(\|\mathbf{x}\|)| \leq \epsilon$$

*if and only if*

$$|f_{\text{CVNN}}(\mathbf{A}\mathbf{x}) - g(\|\mathbf{x}\|)| \leq \epsilon,$$

*for any  $\mathbf{A} \in \mathbf{SO}(n, \mathbb{R})$ .*

**Lemma 11** *Define a radial function  $g' : \mathbb{R} \rightarrow \mathbb{R}$  with the form of*

$$g'(\|\mathbf{x}\|) \triangleq \sum_{i=1}^N \beta_i \mathbb{I}\{\|\mathbf{x}\| \in \Omega_i\},$$



where  $N$  is a polynomial function of  $d$ ,  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_N) \in \{-1, +1\}^N$ , and  $\Omega_i$ 's are disjoint intervals of width  $\mathcal{O}(1/N)$  on values in the range  $\Theta(\sqrt{d})$ . For  $C_2, C_3 > 0$  with  $d > C_2$ , any  $\epsilon > 0$ , and any choice of  $\epsilon_i \in \{-1, +1\}$  ( $i \in [N]$ ), there exists some matrix  $\mathbf{A} \in \mathbf{SO}(n, \mathbb{R})$  such that

$$|g'(\|\mathbf{x}\|) - f_{\text{CVNN}}(\mathbf{A}\mathbf{x})| \leq \frac{\sqrt{3}}{C_2 d^{1/4}} + \epsilon,$$

where  $f_{\text{CVNN}}(\mathbf{x})$  indicates the expressive function led by a one-hidden-layer CVNN with range in  $[-2, +2]$  and at most  $C_3 C_s d^{15/4} / \epsilon$  hidden neurons.

*Finishing the Proof of Theorem 2* The rest proof is a straightforward combination of Lemmas 9, 10 and 11. Define a radial function  $g' : \mathbb{R} \rightarrow \mathbb{R}$  with the form of

$$g'(\|\mathbf{x}\|) \triangleq \sum_{i=1}^N \beta_i \mathbb{I}\{\|\mathbf{x}\| \in \Omega_i\},$$

where  $N$  is a polynomial function of  $m$ ,  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_N) \in \{-1, +1\}^N$ , and  $\Omega_i$ 's are disjoint intervals of width  $\mathcal{O}(1/N)$  on values in the range  $\Theta(\sqrt{d})$ . Obviously,  $g'$  is equivalent to another radial function defined in Eq. (3) as follows

$$g(\mathbf{x}) = \sum_{i=1}^N \beta_i g_i(\mathbf{x}) \quad \text{with} \quad g_i(\mathbf{x}) = \mathbb{I}\{\|\mathbf{x}\| \in \Omega_i\},$$

provided  $\mathbf{x} \in \mathcal{S}(r)$ . From Lemma 11, it is observed that provided the concerned radial function  $g(\mathbf{x})$  with a collection of choices  $\boldsymbol{\beta}$ , there exists  $m \in \mathcal{O}(C_3 C_s d^{15/4} / \epsilon)$  such that  $g(\mathbf{x})$  can be approximated by an expressive function  $f_{\text{CVNN}}(\mathbf{x})$  led by a one-hidden-layer CVNN with range in  $[-2, +2]$  and at most  $m$  hidden neurons, such that

$$|g'(\|\mathbf{x}\|) - f_{\text{CVNN}}(\mathbf{A}\mathbf{x})| \leq \frac{\sqrt{3}}{C_2 d^{1/4}} + \epsilon.$$

Let  $\mathbf{A} \in \mathbf{SO}(n, \mathbb{R})$ . If  $\|\mathbf{x}\| \in \Omega_i \subseteq \mathcal{S}(r)$  for  $0 < r < \infty$ , then it holds  $\|\mathbf{A}\mathbf{x}\| \in \Omega_i \subseteq \mathcal{S}(r)$  and  $g'(\|\mathbf{x}\|) = g'(\|\mathbf{A}\mathbf{x}\|)$ . According to Lemma 10, we have

$$\begin{aligned} \|g(\mathbf{x}) - f_{\text{CVNN}}(\mathbf{x}, t)\|_{L_2(\mu)} &= |g'(\|\mathbf{x}\|) - f_{\text{CVNN}}(\mathbf{x})| \\ &= |g'(\|\mathbf{x}\|) - f_{\text{CVNN}}(\mathbf{A}\mathbf{x})| \\ &\leq \frac{\sqrt{3}}{C_2 d^{1/4}} + \epsilon, \end{aligned}$$

without any incremental change in the parameter complexity. This completes the proof. □

#### D. Iterative Algorithms for Bayesian Neural Networks

Here, we show the iterative algorithm for optimizing the invariant approximation using BNNs layer by layer in Subsection 4.3. The procedure is listed as follows:

- Provided data  $\mathbf{x}$ , using Eq. (9) and the point estimation for the weights in the  $l$ -th layer to optimize the parameters (i.e., mean vector and covariance matrix) of the weights in the  $(l - 1)$ -th layer for  $2 \leq l \leq L$ ;
- One generates distribution  $p$  of activation variables  $\mathbf{z}^l$  composed of the  $l$ -th layer by exploiting Eq. (9) under the optimal  $q(\mathbf{W}^l)$ ;
- The above steps can be repeated until the last layer, (that is,  $l = L$ ), and then  $q(\mathbf{W}^l)$  converges over iteration.

It is observed that this iterative procedure is no larger than optimizing the weights of DNNs, since all previous layers are full-characterized by the distribution of the latent activation variables  $\mathbf{z}^l$  of the hidden layers.

#### References

- [1] R. Abraham and J. Marsden. *Foundations of Mechanics*. American Mathematical Soc., 1978.
- [2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, 2014.
- [3] A. J. Barabell, J. Capon, DF. Delong, JR. Johnson, and KD. Senne. Performance comparison of superresolution array processing algorithms. *NASA*, (19980211580), 1998.
- [4] A. Barron. Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14(1):115–133, 1994.

- [5] G. EP. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.
- [6] S. Chen, E. Dobriban, and J. H. Lee. A group-theoretic framework for data augmentation. *The Journal of Machine Learning Research*, 21(1):9885–9955, 2020.
- [7] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.
- [8] R. Eldan and O. Shamir. The power of depth for feedforward neural networks. In *Proceedings of the 29th Annual Conference on Learning Theory*, pages 907–940, 2016.
- [9] K.-I. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):183–192, 1989.
- [10] A. Graves, A.-R. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, 2013.
- [11] A. Hirose. *Complex-valued Neural Networks: Theories and Applications*, volume 5. World Scientific, 2003.
- [12] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.
- [13] B. Joshua, L. Qian, and X. Li. A survey of complex-valued neural networks. *arXiv:2101.12249*, 2021.
- [14] P. Kidger and T. Lyons. Universal approximation with deep narrow networks. In *Proceedings of the 33rd Annual Conference on Learning Theory*, pages 2306–2327, 2020.
- [15] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105, 2012.
- [16] R. Kurle, R. Herbrich, T. Januschowski, Y. Wang, and J. Gasthaus. On the detrimental effect of invariances in the likelihood for variational inference. *arXiv:2209.07157*, 2022.
- [17] M. Leshno, V. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.

- [18] Q. Li, T. Lin, and Z. Shen. Deep neural network approximation of invariant functions through dynamical systems. *arXiv:2208.08707*, 2022.
- [19] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. In *Advances in Neural Information Processing Systems 30*, pages 6232–6240, 2017.
- [20] N. Mönning and S. Manandhar. Evaluation of complex-valued neural networks on real-valued classification tasks. *arXiv:1811.12351*, 2018.
- [21] F. Quiroga, F. Ronchetti, L. Lanzarini, and A. F. Bariviera. Revisiting data augmentation for rotational invariance in convolutional neural networks. In *Proceedings of 8th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, pages 127–141, 2018.
- [22] R. Roy and T. Kailath. ESPRIT-estimation of signal parameters via rotational invariance techniques. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(7):984–995, 1989.
- [23] R. Roy, A. Paulraj, and T. Kailath. Estimation of signal parameters via rotational invariance techniques-ESPRIT. In *MILCOM 1986-IEEE Military Communications Conference: Communications-Computers: Teamed for the 90's*, volume 3, pages 41–6, 1986.
- [24] A. Sannai, Y. Takai, and M. Cordonnier. Universal approximations of permutation invariant/equivariant functions by deep neural networks. *arXiv:1903.01939*, 2019.
- [25] S. Sun, W. Chen, L. Wang, X. Liu, and T.-Y. Liu. On the depth of deep neural networks: A theoretical view. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 2066–2072, 2016.
- [26] I. Sutskever, O. Vinyals, and Q. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112, 2014.
- [27] C. Trabelsi. *Stabilizing and Enhancing Learning for Deep Complex and Real Neural Networks*. PhD thesis, Ecole Polytechnique, Montreal (Canada), 2019.
- [28] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. Pal. Deep complex networks. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.
- [29] F. Voigtlaender. The universal approximation theorem for complex-valued neural networks. *arXiv:2012.03351*, 2020.

- [30] J.-H. Wu, S.-Q. Zhang, Y. Jiang, and Z.-H. Zhou. Theoretical exploration of flexible transmitter model. *arXiv:2111.06027*, 2021.
- [31] D. Yarotsky. Universal approximations of invariant maps by neural networks. *Constructive Approximation*, 55(1):407–474, 2022.
- [32] S.-Q. Zhang and F.-L. Fan. Neural network gaussian processes by increasing depth. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [33] S.-Q. Zhang, W. Gao, and Z.-H. Zhou. Towards understanding theoretical advantages of complex-reaction networks. *Neural Networks*, 151:80–93, 2022.
- [34] S.-Q. Zhang and Z.-H. Zhou. Flexible transmitter network. *Neural Computation*, 33(11):2951–2970, 2021.
- [35] Shao-Qun Zhang and Zhi-Hua Zhou. Theoretically provable spiking neural networks. *Advances in Neural Information Processing Systems 35*, 2022.
- [36] D.-X. Zhou. Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, 48(2):787–794, 2020.
- [37] Z.-H. Zhou. Why over-parameterization of deep neural networks does not overfit? *Science China Information Sciences*, 64(1):1–3, 2021.