Universality of group convolutional neural networks based on ridgelet analysis on groups

Anonymous Author(s)
Affiliation
Address
email

Abstract

We investigate the approximation property of group convolutional neural networks (GCNNs) based on the ridgelet theory. We regard a group convolution as a matrix element of a group representation, and formulate a versatile GCNN as a nonlinear mapping between group representations, which covers typical GCNN literatures such as a cyclic convolution on a multi-channel image, permutation-invariant datasets (Deep Sets), and $\mathrm{E}(n)$-equivariant convolutions. The ridgelet transform is an analysis operator of a depth-2 network, namely, it maps an arbitrary given target function f to the weight γ of a network $S[\gamma]$ so that the network represents the function as $S[\gamma] = f$. It has been known only for fully-connected networks, and this study is the first to present the *ridgelet transform for (G)CNNs*. Since the ridgelet transform is given as a closed-form integral operator, it provides a constructive proof of the *cc-universality of GCNNs*. Unlike previous universality arguments on CNNs, we do not need to convert/modify the networks into other universal approximators such as invariant polynomials and fully-connected networks.

1 Introduction

Since the historical success of AlexNet [1], convolutional neural networks (CNNs) have been applied for a variety of machine learning tasks [2]. In the research field of *geometric deep learning* [3], *group equivariant CNNs* have been developed to capture the inductive bias behind a variety of datasets such as sets and point clouds [4, 5], graphs [6, 7], manifolds, groups and homogeneous spaces [8–10, 6].

In the theoretical aspect, however, the approximation property of CNNs is less investigated than that of fully-connected networks. To this date, several authors have shown that some CNNs are *universal*, that is, they can approximate some class of continuous maps with any precision [11–16]. These studies are still limited because (1) the network components such as the activation function and convolution are limited, and/or (2) the proofs are indirect or existential since they rely on the universality of other universal approximators such as fully-connected network and invariant polynomials.

On the other hand, the approximation property of fully-connected networks has been investigated in the 1990s, with gradually increasing the resolution from abstract to concrete, starting from existential proofs based on the Hahn-Banach theorem [17] and the Stone-Weierstrass theorem [18], indirect proofs based on the Fourier transform [19, 20], the Radon transform [21, 22], B-splines [23, 24], to constructive proofs based on the integral representation [25], ridge functions [26], and the *ridgelet transform* [27–29]. For deep fully-connected ReLU networks, more detailed approximation properties have been revealed [30–32] in the 2010s.

In this study, we investigate the approximation property of CNNs based on the ridgelet transform—an analysis operator that maps a given function f to the weight parameter γ in a single hidden layer of a neural network. To overview the main contributions, we describe the formulation of GCNNs.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

A Typical Convolution Layer for Images. Given an $m_1 \times m_2$-dimensional n_{in}-channel input image $\boldsymbol{x} \in \mathbb{R}^{m_1 \times m_2 \times n_{in}}$, a typical *convolution layer* with $w_1 \times w_2$-dimensional $n_{in} \times n_{out}$-channels filter $\boldsymbol{a} \in \mathbb{R}^{w_1 \times w_2 \times n_{in} \times n_{out}}$ and n_{out}-channel bias $\boldsymbol{b} \in \mathbb{R}^d$ followed by the elementwise activation function $\sigma : \mathbb{R} \to \mathbb{R}$ and the aggregation with output coefficients $\boldsymbol{c} \in \mathbb{R}^d$ is given by

$$S[\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}](\boldsymbol{x})(i,j) = \sum_{\ell=1}^{n_{out}} c^\ell \sigma \left(\sum_{k=1}^{n_{in}} \sum_{p=1}^{w_1} \sum_{q=1}^{w_2} a_{pq}^{k\ell} x_{i+p,j+q}^k - b^\ell \right) \tag{1}$$

for each pixel at $(i,j) \in \mathbb{R}^{(m_1 - w_1 + 1) \times (m_2 - w_2 + 1)}$.

Here, in this study, the output channels (indexed by n_{out}) are aggregated soon after the activation function, which may be slightly different from an ordinary formulation of CNNs. We assume this aggregation for technical reasons, but we may understand this as a part of the subsequent layer.

In the standard formulation of GCNNs, the convolution in the pixel directions (i,j) is reformulated as a group convolution with respect to the product group $G = \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2}$ of cyclic groups $\mathbb{Z}_{m_i}$ $(i = 1, 2)$. So, the input image is understood as a function on a group G (or a homogeneous space G/H on which G acts). In addition, the inner product in the channel direction k is also understood as a part of the convolution with respect to the trivial action of G on the 'fiber' $\mathbb{R}^c$. In this sense, the multi-channel image is understood as a vector-valued function (or a section of a vector bundle).

The Integral Representation $S[\gamma]$ of Group Convolution Layer. Let G be an *arbitrary* group, and $\mathcal{H}$ be an *arbitrary* vector space of feature vector x and filter a. In this study, we formulate a group convolution layer in an integral form, called the *integral representation*, as

$$S[\gamma](x)(g) := \int_{\mathcal{H} \times \mathbb{R}} \gamma(a,b) \sigma\big((a * x)(g) - b\big) \mathrm{d}a\mathrm{d}b, \quad x \in \mathcal{H},\ g \in G. \tag{2}$$

This is an infinite-dimensional linear representation of a depth-2 group convolutional neural network. (It is linear in the coefficient function γ.) Here, each function $x \mapsto \sigma((a * x)(g) - b)$ represents a single convolutional neuron, or a feature map of input x parametrized by (a,b). The integration over (a,b) implies that all the possible neurons are assigned in advance, and thus $S[\gamma]$ can be understood as a *continuous neural network*. We note, however, that if we take γ to be a finite sum of Dirac's measures such as $\gamma_n := \sum_{\ell=1}^n c^\ell \delta_{(a^\ell, b^\ell)}$, then the integral representation can also exactly reproduce a finite model: $S[\gamma_n](x)(g) = \sum_{\ell=1}^n c^\ell \sigma((a^\ell * x)(g) - b^\ell)$ for $x \in \mathcal{H}, g \in G$. In summary, $S[\gamma]$ is a mathematical model of shallow neural networks with *any* width ranging from finite to infinite. In particular, the locality of distribution γ reflects the sparsity/low-dimensionality of filters a^ℓ.

The Ridgelet Transform $R[f;\rho](a,b)$ is a right inverse (or pseudo-inverse) operator of the integral representation operator S. As displayed in the main section, the ridgelet transform is given as a closed-form integral operator $R[f;\rho]$. Here, $f : \mathcal{H} \to C(G)$ is a target function to be approximated, and $\rho : \mathbb{R} \to \mathbb{C}$ is an auxiliary function, called the *ridgelet function*. Under mild conditions, it satisfies the *reconstruction formula*

$$S[R[f;\rho]] = (\!(\sigma, \rho)\!) f, \tag{3}$$

where $(\!(\cdot,\cdot)\!)$ denote a scalar product of σ and ρ. Therefore, as long as the product $(\!(\sigma,\rho)\!)$ is neither 0 nor ∞, we can normalize ρ to satisfy $(\!(\sigma,\rho)\!) = 1$ so that $S[R[f;\rho]] = f$.

In other words, R and S are analysis and synthesis operators, and thus play the same roles as the Fourier (F) and inverse Fourier (F^{-1}) transforms respectively, in the sense that the reconstruction formula $S[R[f;\rho]] = (\!(\sigma,\rho)\!) f$ corresponds to the Fourier inversion formula $F^{-1}[F[f]] = f$. Despite the common belief that neural network parameters are a blackbox, the closed-form expression of ridgelet transform clearly describes how the network parameters are organized, which is a clear advantage of the integral representation theory. Furthermore, the integral representation theory can deal with a wide range of activation functions without any modification, *not only ReLU* but all the tempered distribution $\mathcal{S}'(\mathbb{R})$ [see, e.g., 33].

The original ridgelet transform is discovered in the 1990s independently by Murata [27], Candès [28] and Rubin [29]. In the context of deep learning theory, Savarese et al. [34] and their followers [35–38] investigate the ridgelet transform for the specific case of fully-connected ReLU layers to establish representer theorems. Furthermore, as applications of ridgelet analysis for fully-connected

networks, it has been shown that finite models trained by regularized empirical risk minimization (RERM) converges to a certain unique ridgelet transform [39], and any parameter distribution γ satisfying $S[\gamma] = f$ can always be represented as (not always single but) a linear combination of ridgelet transforms [40].

Main Contributions. To this date, the closed-form expression of the ridgelet transform is known only for fully-connected layers [41], and this study is the first to derive the ridgelet transform for a general class of GCNNs. As an application, we show the cc-universality of GCNNs for a general class of group equivalent continuous vector-valued functions in a constructive and unified manner.

2 Preliminaries

2.1 Notation

For any topological space X, $C(X), C_0(X)$ and $C_c(X)$ denote the collections of all continuous functions on X, continuous functions on X that vanish at infinity, and continuous functions on X with compact support, respectively. We note that when X is compact, then these three spaces are equivalent, i.e., $C(X) = C_0(X) = C_c(X)$.

For any integer $d > 0$, $\mathcal{S}(\mathbb{R}^d)$ and $\mathcal{S}'(\mathbb{R}^d)$ denote the classes of Schwartz test functions (or rapidly decreasing functions) and tempered distributions on $\mathbb{R}^d$, respectively. Namely, $\mathcal{S}'$ is the topological dual of $\mathcal{S}$. We note that $\mathcal{S}'(\mathbb{R})$ includes truncated power functions $\sigma(b) = b_+^k = \max\{b, 0\}^k$ such as step function for $k = 0$ and ReLU for $k = 1$.

Fourier Transform The Fourier transform on the Euclidean space $\mathbb{R}^m$ and its inversion formula has been defined (at least) three different function classes: $L^1(\mathbb{R}^m), L^2(\mathbb{R}^m)$ and $\mathcal{S}'(\mathbb{R}^m)$. When $f \in L^1(\mathbb{R}^m)$ and $\widehat{f} \in L^1(\mathbb{R}^m)$, the inversion formula holds at every continuous point $\boldsymbol{x}$ of f, which is a pointwise equality. When $f \in L^2(\mathbb{R}^m)$, the inversion formula holds in L^2, which is not a pointwise equality because the equality $f = g$ in L^2 is defined as $f(\boldsymbol{x}) = g(\boldsymbol{x})$ for a.e. $\boldsymbol{x}$. Similarly, when $f \in \mathcal{S}'(\mathbb{R}^m)$, the inversion formula holds in $\mathcal{S}'$. We use the third definition for computing the Fourier transform of activation functions $\sigma \in \mathcal{S}'(\mathbb{R})$ such as ReLU and $\tanh$. We refer to Grafakos [42] for more details on the Fourier transform for distributions.

2.2 Group Representation

Let G be a group, let $\mathcal{H}$ be a vector space over a field $\mathbb{K}$, and let $GL(\mathcal{H})$ be the general linear group on $\mathcal{H}$. A *group representation* T of the group G on the vector space $\mathcal{H}$ is a group homomorphism from G to $GL(\mathcal{H})$, that is, a map $T : G \to GL(\mathcal{H}); g \mapsto T_g$ satisfying $T_{gh} = T_g T_h$ for all $g, h \in G$. Here, $\mathcal{H}$ is called the *representation space*, or sometimes the *G-module*.

Regular Representation. Let $\mathcal{H}$ be a vector space of functions on G. The *(left) regular representation* L is a group representation defined on $\mathcal{H}$ as

$$L_g[x](h) := x(g^{-1}h), \quad g, h \in G,\ x \in \mathcal{H}. \tag{4}$$

In particular, when G is a locally compact Haussdorf (LCH) group, then it has a (left) invariant measure μ, and we can define the collection $L^2(G)$ of all square integrable functions on G with respect to the canonical inner product $\langle x, y\rangle_{L^2(G)} := \int_G x(g)\overline{y(g)}\mathrm{d}\mu(g)$ for any measurable functions $x, y : G \to \mathbb{C}$. It is known that the regular representation on $\mathcal{H} = L^2(G)$ is a unitary representation.

Dual Representation. For any group representation $T : G \to GL(\mathcal{H})$, the *dual representation* $T^* : G \to GL(\mathcal{H}^*)$ is a group representation defined on the dual vector space $\mathcal{H}^*$ as the transpose of $T_{g^{-1}}$, that is, $T_g^* = T_{g^{-1}}^\top$. When $\mathcal{H}$ is a Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, then it satisfies the following relation:

$$\langle T_g[x], y\rangle_{\mathcal{H}} = \langle x, T_{g^{-1}}^*[y]\rangle_{\mathcal{H}}, \quad g \in G,\ x, y \in \mathcal{H}. \tag{5}$$

Matrix Element. For any group representation $T : G \to GL(\mathcal{H})$, the *matrix element* (or the *matrix coefficient*) of T is a bilinear functional $f_{a,x}$ on G defined by

$$f_{a,x}(g) := a[T_g[x]], \quad g \in G,\ x \in \mathcal{H},\ a \in \mathcal{H}' \tag{6}$$

where x is a vector in $\mathcal{H}$ and $a \in \mathcal{H}'$ is a continuous linear functional on $\mathcal{H}$. When $\mathcal{H}$ is a Hilbert space, then it can be written as

$$f_{a,x}(g) = \langle T_g[a], x\rangle_{\mathcal{H}}, \quad g \in G,\ a, x \in \mathcal{H}. \tag{7}$$

In the next section, we use this quantity as the generalized form of the *group convolution*.

2.3 Universality

The notion of *universality* in machine learning can be rephrased as the *density* in mathematics, and thus it has several definitions [43, 44]. In this study, we establish the so-called *cc-univesality*, one of the standard universalities in the neural network theory.

***cc*-Universality.** Let X be a topological space, and let NN be a collection of functions (e.g., neural networks) on X. The *cc*-univesality of NN is defined as the density of NN in $C(X)$ endowed with the topology of *compact convergence*, that is, for any compact subset $K \subset X$, continuous function $f \in C(K)$, and all $\varepsilon > 0$, there exists a function $g \in \mathrm{NN}$ such that

$$\left\| f - g|_K \right\|_{C(K)} := \sup_{x \in K} |f(x) - g(x)| < \varepsilon, \tag{8}$$

where $g|_K$ denotes the restriction of g to K.

3 Formulations

We introduce the notions of (G, T)-convolution and (G, T)-equivariance, which are natural extensions of group convolution and group equivariance, respectively. Then, we formulate the integral representation of (G, T)-convolutional neural network.

3.1 (G, T)-Convolution

Definition 1. Let G be a group, let $\mathcal{H}$ be a Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ over a field $\mathbb{K}$, and let $T : G \to GL(\mathcal{H})$ be a representation of G on $\mathcal{H}$. For any $x, y \in \mathcal{H}$ and $g \in G$, we define the (G, T)-*convolution* as

$$(x *_T y)(g) := \langle T_g[x], y\rangle_{\mathcal{H}}. \tag{9}$$

First of all, this is simply a paraphrase of the *matrix element of group representation* (see the previous section). Nevertheless, we call it a *convolution* simply because it covers a wide range of previously proposed 'group convolutions' in the GCNN literature.

For example, an orthodox group convolution is reproduced when T is the regular representation (of a LCH group G) on $\mathcal{H} = L^2(G)$, that is,

$$T_g[x](h) := x(g^{-1}h), \quad \text{and} \quad \langle T_g[a], x\rangle_{L^2(G)} = \int_G a(g^{-1}h)\overline{x(h)}\mathrm{d}\mu(h) = (a *_G \overline{x})(g). \tag{10}$$

More specifically, the cyclic convolution for an n-channel image $\boldsymbol{x} = (x^k_{ij}) \in \mathbb{R}^{m_1 \times m_2 \times n}$ is understood as the case when $G = \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2}$, $\mathcal{H} = \mathbb{R}^{m_1 \times m_2 \times n}$ and

$$T_{(p,q)}[\boldsymbol{x}](i, j, k) = x^k_{i-p,j-q}, \quad \text{so that} \quad \langle T_{(p,q)}[\boldsymbol{a}], \boldsymbol{x}\rangle_{\mathbb{R}^{m_1 \times m_2 \times n}} = \sum_{i,j,k} a^k_{i-p,j-q} x^k_{ij}. \tag{11}$$

the inner product in the channel direction is understood as the

On the other hand, the 'input feature space' $\mathcal{H}$ can be an arbitrary abstract vector space, which is more abstract than the conventional group convolution where the feature space is instantiated as a function space on G, such as $L^2(G)$ and $L^2(G/H)$. We consider this as an advantage for geometric understanding of CNNs, since the theory becomes free from the specific instantiation of the feature x.

3.2 (G,T)-Equivariance

Associated with the extended convolution, we introduce the notion of equivariance as below.

Definition 2. We say a function $f : \mathcal{H} \to \mathbb{C}^G$ is (G,T)-*equivariant* when

$$f(T_g^*[x])(h) = L_g[f[x]](h) = f(x, g^{-1}h), \quad x \in \mathcal{H},\ g, h \in G. \tag{12}$$

That is, we restrict the term 'equivariance' for the special case of the regular representation on $C(G)$, which is simply because our main results assume this case. Besides, we state the condition for the dual representation T^*, which is simply because our GCNN satisfies this condition. We remark that this does not restrict the choice of representations, since T itself is the dual of T^*, i.e., $T = (T^*)^*$.

3.3 (G,T)-Equivariant Continuous Vector-Valued Function $f : \mathcal{H} \to C(G)$

Let $\mathcal{X}$ and $\mathcal{Y}$ denote any Banach spaces, and let $\|\cdot\|_{\mathcal{X}'}$ denote any norm for (nonlinear) functions $f' : \mathcal{X} \to \mathbb{R}$. For a vector-valued function $f : \mathcal{X} \to \mathcal{Y}$, a vector-valued norm is induced as $\|f\|_{\mathcal{X}\to\mathcal{Y}} := \big\|\|f\|_{\mathcal{Y}}\big\|_{\mathcal{X}'}$. Specifically, in the main results, we deal with the following uniform norm:

$$\|f\|_{C(\mathcal{H};C(G))} := \|f\|_{C(\mathcal{H})\to C(G)} := \sup_{x\in\mathcal{H}} \Big| \sup_{g\in G} |f(x)(g)| \Big|. \tag{13}$$

By $C_{equi}(\mathcal{H}; C(G))$, we denote the Banach space of all (G,T)-*equivariant* $C(G)$-*valued continuous* functions on $\mathcal{H}$.

Remark for avoiding potential confusions: A vector-valued function $f : \mathcal{X} \to \mathcal{Y}$ is *not* always a linear operator, that is, we cannot expect the linearity: $f(\alpha x + \beta y) \neq \alpha f(x) + \beta f(y)$ in general. Since the linearity is a special form of nonlinearity, the Banach space V of vector-valued functions $f : \mathcal{X} \to \mathcal{Y}$ endowed with the vector-valued norm $\|\cdot\|_V$ is 'intuitively larger' than the Banach space L of bounded linear operators $g : \mathcal{X} \to \mathcal{Y}$ endowed with the operator norm $\|\cdot\|_L$.

3.4 Fourier Transform on The Abstract Feature Space $\mathcal{H}$

For the sake of simplicity, we assume that the real dimension $m := \dim_{\mathbb{R}} \mathcal{H}$ of $\mathcal{H}$ is *finite*.

Lebesgue Measure on $\mathcal{H}$. We define the Lebesgue measure λ on $\mathcal{H}$ by pushing forward the Lebesgue measure $\mathrm{d}\boldsymbol{x}$ on $\mathbb{R}^m$ via an isometric linear embedding $\mathbb{R}^m \to \mathcal{H}$. For example, let $\{\mathsf{e}_i\}_{i\in[m]}$ be an orthonormal basis of $\mathcal{H}$, and put a linear embedding $\phi(\boldsymbol{x}) := \sum_{i\in[m]} x_i \mathsf{e}_i$, then (it preserves the length and) we can induce the Lebesgue measure λ on $\mathcal{H}$ as the push forward measure $\lambda = \phi_\sharp \mathrm{d}\boldsymbol{x}$ so that the volume of a hypercube $E = \{\sum_{i\in[m]} c_i \mathsf{e}_i \mid c_i \in [a_i, b_i]\}$ in $\mathcal{H}$ is calculated as in $\mathbb{R}^m$: $\lambda(E) = \prod_{i\in[m]} |b_i - a_i|$, and the integration is calculated as

$$\int_E f(x)\mathrm{d}\lambda(x) = \int_{\mathbb{R}^m} 1_X \left(\sum_{i=1}^m x_i \mathsf{e}_i\right) f\left(\sum_{i=1}^m x_i \mathsf{e}_i\right) \prod_{i=1}^m \mathrm{d}x_i = \int_{\phi^{-1}(X)} f \circ \phi(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \tag{14}$$

for any measurable subset $E \subset \mathcal{H}$ and measurable function $f : \mathcal{H} \to \mathbb{C}$. As far as there is no risk of confusion, we denote $\mathrm{d}x$ instead of $\mathrm{d}\lambda(x)$.

Fourier Transform on $\mathcal{H}$. Based on λ, we induce the Fourier transform on $\mathbb{R}^m$ as below: For any function $f : \mathcal{H} \to \mathbb{C}$,

$$\widehat{f}(y) := \int_{\mathcal{H}} f(x) e^{-i\langle x,y\rangle_{\mathcal{H}}} \mathrm{d}\lambda(x), \quad f(x) \overset{\star}{=} \frac{1}{(2\pi)^m} \int_{\mathcal{H}} \widehat{f}(y) e^{i\langle x,y\rangle_{\mathcal{H}}} \mathrm{d}\lambda(y). \tag{15}$$

Here, the equality $\overset{\star}{=}$ in the inversion formula holds in a variety of different senses (see the previous section).

Again, $\mathcal{H}$ remains abstract. Depending on the embedding ϕ, each instantiation of $\mathcal{H}$ looks different. Nevertheless, these share the equivalent structures induced from $\mathbb{R}^m$. We consider this as also an advantage for geometric understanding of CNNs.

4 Main Results

Based on the formulations in the previous sections, we define the (G,T)-convolutional neural networks and the corresponding ridgelet transform.

Throughout this section, $T : G \to GL(\mathcal{H})$ is a representation of a group G on a finite-dimensional Hilbert space $\mathcal{H}$ over a field $\mathbb{K}$ endowed with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and a Lebesgue measure $\mathrm{d}x$. Let $m := \dim_{\mathbb{R}} \mathcal{H}$ be the real dimension of $\mathcal{H}$, and $k := \dim_{\mathbb{R}} \mathbb{K}$ be the real dimension of $\mathbb{K}$, that is, $k = 1$ for $\mathbb{K} = \mathbb{R}$ and $k = 2$ for $\mathbb{K} = \mathbb{C}$.

4.1 Integral Representation of (G,T)-Convolutional Neural Network

Definition 3. For any functions $\gamma : \mathcal{H} \times \mathbb{K} \to \mathbb{C}$ and $\sigma : \mathbb{K} \to \mathbb{C}$, we define the *integral representation of (G,T)-convolutional neural network* as a vector-valued function $\mathcal{H} \to \mathbb{C}^G$,

$$S[\gamma](x)(g) := \int_{\mathcal{H}\times\mathbb{K}} \gamma(a,b)\sigma((a *_T x)(g) - b)\mathrm{d}\lambda(a)\mathrm{d}b, \quad x \in \mathcal{H},\ g \in G. \tag{16}$$

Here, we call γ a coefficient function, and σ the activation function. If there is no risk of confusion, we abbreviate $\mathrm{d}\lambda(a)$ as $\mathrm{d}a$.

It is easy to see that a (G,T)-CNN is (G,T)-equivariant. In fact, for every $g, h \in G$,

$$\begin{aligned} S[\gamma](T_g[x])(h) &= \int_{\mathcal{H}\times\mathbb{K}} \gamma(a,b)\sigma(\langle T_h[a], T_g^*[x]\rangle_{\mathcal{H}} - b)\mathrm{d}a\mathrm{d}b \\ &= \int_{\mathcal{H}\times\mathbb{K}} \gamma(a,b)\sigma(\langle T_{g^{-1}}T_h[a], x\rangle_{\mathcal{H}} - b)\mathrm{d}a\mathrm{d}b = S[\gamma](x)(g^{-1}h). \end{aligned} \tag{17}$$

4.2 Ridgelet Transform and Bilinear Form of Activation Function

Definition 4. For any functions $f : \mathcal{H} \to \mathbb{C}^G$ and $\rho : \mathbb{K} \to \mathbb{C}$, we define the *ridgelet transform* as

$$R[f;\rho](a,b) := \int_{\mathcal{H}} f(x)(e)\overline{\rho(\langle a, x\rangle_{\mathcal{H}} - b)}\mathrm{d}x, \quad (a,b) \in \mathcal{H} \times \mathbb{K}. \tag{18}$$

Here e denotes the identity element of G.

Definition 5. For any tempered distribution $\sigma \in \mathcal{S}'(\mathbb{R})$ and function $\rho \in \mathcal{S}(\mathbb{R})$, put the bilinear product

$$(\!(\sigma, \rho)\!) := (2\pi)^{m-k} \int_{\mathbb{K}} \sigma^\sharp(\omega)\overline{\rho^\sharp(\omega)}|\omega|^{-m}\mathrm{d}\omega. \tag{19}$$

Here, $\cdot^\sharp$ denotes the Fourier transform on $\mathbb{K}$, which is identified with on $\mathbb{R}^k$ with $k = \dim_{\mathbb{R}} \mathbb{K}$. We note that $\sigma^\sharp$ is defined in the sense of tempered distributions.

The derivations of the ridgelet transform and bilinear product are clarified in the proof of the reconstruction formula. Some readers who are familiar with the classical ridgelet transform may notice that the ridgelet transform for GCNN is formally the same as the one for fully-connected NNs. This may look asymmetric, since $\rho((a *_T x)(g) - b)$ looks intuitively more symmetric than $\rho(\langle a, x\rangle_{\mathcal{H}} - b)$ for GCNNs. In fact, this symmetry is not broken, because at the identity element $g = e$, the equality $\langle a, x\rangle_{\mathcal{H}} = (a *_T x)(e)$ holds, thus we can understand the 'ridgelet transform with $\rho((a *_T x)(g) - b)$' is applied at the single point $g = e$, which works because (in the main theorem) the target function is assumed to be (G,T)-equivariant.

4.3 Reconstruction Formula

Theorem 1 (Main Theorem 1/2)**.** *Assume (A1) that $f : \mathcal{H} \to \mathbb{C}^G$ is (T,G)-equivariant, i.e.,*

$$f(T_g^*[x])(h) = f(x)(g^{-1}h), \quad \textit{for every } x \in \mathcal{H} \textit{ and } g, h \in G; \tag{20}$$

and (A2) that f satisfies at least one of the following conditions: Write $f_e := f(\bullet, e)$, and (A2a) both f_e and $\widehat{f_e}$ are absolute-integrable, i.e., $f_e, \widehat{f_e} \in L^1(\mathcal{H})$, (A2b) f_e is square-integrable, i.e., $f_e \in$

$L^2(\mathcal{H})$, or (A2c) f_e is a tempered distribution, i.e., $f_e \in \mathcal{S}'(\mathcal{H})$. Then, the following reconstruction formula holds:

$$S[R[f;\rho]](x)(g) = \int_{\mathcal{H}\times\mathbb{K}} R[f;\rho](a,b)\sigma((a *_T x)(g) - b)\mathrm{d}a\mathrm{d}b \overset{\star}{=} ((\sigma,\rho))f(x)(g), \tag{21}$$

where the equality $\overset{\star}{=}$ holds at every continuous point x_c of f for (A2a), in L^2 for (A2b), and in $\mathcal{S}'$ for (A2c), respectively.

The proof is given in Appendix A.1.

4.4 cc-Universality

Finally, we state the second half of our main results. Since the reconstruction formula holds for an arbitrary function f, by discretizing the reconstruction formula $S[\gamma] = f$ with $\gamma = R[f;\rho]$ into a finite sum

$$f_n(x)(g) := \sum_{i=1}^{n} c_i\sigma((a_i *_T x)(g) - b_i), \tag{22}$$

in a 'nice' manner so that the discretized coefficient function $\gamma_n := \sum_{i=1}^{n} c_i\delta_{(a_i,b_i)}$ reproduces $\gamma = R[f;\rho]$ in a limit of $n \to \infty$, we can construct a sequence $\{f_n\}_{n\in\mathbb{N}}$ of finite (G,T)-CNNs that converges to an arbitrary target function f,

$$f_n \to f, \quad n \to \infty. \tag{23}$$

This is the primitive idea behind the constructive proof of the following cc-universality of finite (G,T)-CNNs based on ridgelet analysis.

To state an assumption on the activation function σ, we introduce the forward difference operator Δ_θ^n with difference $\theta > 0$, defined as

$$\Delta_\theta^1[\sigma](t) := \sigma(t+\theta) - \sigma(t), \quad \Delta_\theta^{n+1}[\sigma](t) := \Delta_\theta^1 \circ \Delta_\theta^n[\sigma](t). \tag{24}$$

Theorem 2 (Main Theorem 2/2)**.** *For an activation function $\sigma \in \mathcal{S}'(\mathbb{K})$, assume (A3) that there exist $n \in \mathbb{N}$ and $\theta > 0$ such that $\Delta_\theta^n[\sigma]$ is bounded and Lipschitz continuous. Let NN be the collection of finite (G,T)-CNNs, that is, $\mathrm{NN} := \bigcup_{n\in\mathbb{N}}\{f_n(x)(g) = \sum_{i=1}^{n} c_i\sigma((a_i *_T x)(g) - b_i) \mid (a_i,b_i,c_i) \in \mathcal{H}\times\mathbb{K}\times\mathbb{C}, i \in [n]\}$. Then, NN is cc-universal, that is, for any (G,T)-equivariant $C(G)$-valued continuous function $f \in C_{equi}(\mathcal{H};C(G))$, and for any compact set $K \subset \mathcal{H}$, there exists a sequence $\{f_n\}_{n\in\mathbb{N}} \subset \mathrm{NN}$ of finite GCNNs such that*

$$\|f - f_n\|_{C(K;C(G))} = \sup_{x\in K}\sup_{g\in G}|f(x)(g) - f_n(x)(g)| \to 0, \quad n \to \infty. \tag{25}$$

The proof is given in Appendix A.2.

4.5 Remarks

The topological group G covers a quite large class of groups. For example, it contains all the finite groups like $\mathbb{Z}_n$ and S_n, compact groups like $SO(n)$ and $U(n)$, and non-compact groups like $SE(n)$ and $\mathbb{R}^n$ as well. We assumed this simply for considering the class $C(G)$ of "continuous" functions. We note that when G is a finite group, then $C(G)$ is simply identified with a $|G|$-dimensional complex-vector space $\mathbb{C}^{|G|}$, because $C(G) := \{x : G \to \mathbb{C},\ \text{continuous}\} = \mathbb{C}^G$.

The G-module $\mathcal{H}$ is an abstract vector space and thus needs *not* be a space of functions on G such as $C(G)$ and $L^2(G)$, which is one of the major advantages; On the other hand, $\mathcal{H}$ needs to be finite-dimensional, since the proof is based on the Fourier inversion formula on the finite-dimensional Euclidean space $\mathbb{R}^m$. We conjecture that the extension to an infinite-dimensional setting would be a routine for some specialists in functional analysis. As we will see in the examples, however, ordinary input vectors are always essentially finite-dimensional due to the limitations in computational resources, and thus we consider finite-dimensional setting is reasonable.

The representation T needs *not* to be unitary, irreducible, nor square-integrable, since the proof is based only on the existence of the dual representation T^*.

5 Examples

5.1 Finite Periodic Convolution Layer

Example 1 (For 1-dimensional periodic signals)**.** The periodic convolution corresponds to the case when $\mathbb{K} = \mathbb{R}, G = \mathbb{Z}_m \cong [m] = \{0, 1, \ldots, m-1\}$, $\mathcal{H} = L^2(G) \cong \mathbb{R}^m$ equipped with the inner product $\langle x, y \rangle := \frac{1}{m}\sum_{i\in[m]} x_i y_i$, and $T_i[x](j) := x_{j-i}$ thus $(a *_T x)(i) = \frac{1}{m}\sum_{j\in[m]} a_j x_{i+j}$. Therefore, the ridgelet transform and the reconstruction formula are given by

$$R[f;\rho](a,b) = \int_{\mathbb{R}^m} f(x,0)\overline{\rho\left(\frac{1}{m}\sum_{i\in[m]} a_i x_i - b\right)}\mathrm{d}x,$$

$$S[R[f;\rho]](x,i) = \int_{\mathbb{R}^m\times\mathbb{R}} R[f;\rho](a,b)\sigma\left(\frac{1}{m}\sum_{j\in[m]} a_j x_{i+j} - b\right)\mathrm{d}a\mathrm{d}b = (\!(\sigma,\rho)\!) f(x,i).$$

Example 2 (For multi-channel two-dimensional periodic images)**.** A c-channel 2-dimensional image is identified with a vector-valued function $x : \mathbb{Z}_m \times \mathbb{Z}_n \to \mathbb{R}^c$, thus $\mathcal{H} \cong \mathbb{R}^{m\times n\times c}$. Let x_{ij}^k denote the (i,j)-th component in the k-th channel of $x \in \mathcal{H}$. Let $G = \mathbb{Z}_m \times \mathbb{Z}_n$, and put $T_{(p,q)}[x]_{ij}^k := x_{i-p,j-q}^k$. Therefore, the ridgelet transform and the reconstruction formula are given by

$$R[f;\rho](a,b) = \int_{\mathbb{R}^{mnc}} f(x)_{(0,0)}\overline{\rho\left(\frac{1}{mnc}\sum_{k\in[c]}\sum_{i,j\in[m]} a_{ij}^k x_{ij}^k - b\right)}\mathrm{d}x.$$

$$S[R[f;\rho]](x)_{ij} = \int_{\mathbb{R}^{mnc}\times\mathbb{R}} R[f;\rho](a,b)\sigma\left(\frac{1}{mnc}\sum_{k\in[c]}\sum_{p,q\in[m]} a_{pq}^k x_{p+i,q+j}^k - b\right)\mathrm{d}a\mathrm{d}b = (\!(\sigma,\rho)\!) f(x)_{ij}.$$

5.2 (Deep Sets) Permutation Equivariant Maps on A Finite Set

Example 3. Let $\mathcal{H} = \mathbb{R}^m, G \leqslant S_m$ and $T_g[x] = (x_{g^{-1}(1)}, x_{g^{-1}(2)}, \ldots, x_{g^{-1}(m)})$. Thus $\langle a, x\rangle_{\mathcal{H}} = \frac{1}{m}\sum_{i\in[m]} a_i x_i$, and $(a *_T x)(g) = \frac{1}{m}\sum_{p\in[m]} a_p x_{g(p)}$. So,

$$R[f;\rho](a,b) = \int_{\mathbb{R}^m} f(x,e)\overline{\rho\left(\frac{1}{m}\sum_{i\in[m]} a_i x_i - b\right)}\mathrm{d}x,$$

$$S[R[f;\rho]](x,g) = \int_{\mathbb{R}^m\times\mathbb{R}} R[f;\rho](a,b)\sigma\left(\frac{1}{m}\sum_{p\in[m]} a_p x_{g(p)} - b\right)\mathrm{d}a\mathrm{d}b = (\!(\sigma,\rho)\!) f(x,g).$$

5.3 Continuous Periodic Convolution Layer

Example 4. Let $G = \mathbb{T} := \mathbb{R}/2\pi\mathbb{Z} \cong \{e^{i\theta} \mid \theta \in [-\pi,\pi]\}$ be the 1-dimensional torus group, which is one of the most basic continuous group. As a consequence of the Fourier series expansion, $L^2(\mathbb{T})$ is spanned by $\{e^{in\theta} \mid n \in \mathbb{N}\}$. Hence, we can take $\mathcal{H}$ to be an m-dimensional subspace $\mathcal{H} := \{\sum_{|n|<m} x_n e^{in\theta} \mid x_{-n} = x_n \in \mathbb{R}\}$ equipped with an inner product $\langle x, y\rangle_{\mathcal{H}} := \int_{\mathbb{T}} x(\theta)\overline{y(\theta)}\mathrm{d}\theta$. We note that the constraint $x_n = x_{-n}$ implies $\sum_{|n|<m} x_n e^{in\theta} = \sum_{|n|<m} x_n \cos(n\theta)$ and thus any signal $x \in \mathcal{H}$ is a bandlimited real-valued continuous signal with each coefficient x_n being the n-th frequency spectrum. Put $T_\alpha[x](\theta) := x(\theta - \alpha)$, then $(a * x)(\alpha) = \int_{\mathbb{T}} a(\theta)x(\alpha-\theta)\mathrm{d}\theta = \sum_{|n|<m} a_n x_n e^{in\alpha} = \sum_{|n|<m} a_n x_n \cos(n\alpha)$ (by the convolution theorem and the constraint). Therefore,

$$R[f;\rho](a,b) := \int_{\mathbb{R}^m} f(x,0)\overline{\rho\left(\sum_{|n|<m} a_n x_n - b\right)}\mathrm{d}x,$$

$$S[R[f;\rho]](x,\theta) = \int_{\mathbb{R}^m\times\mathbb{R}} R[f;\rho](a,b)\sigma\left(\sum_{|n|<m} a_n x_n \cos(n\theta) - b\right)\mathrm{d}a\mathrm{d}b = (\!(\sigma,\rho)\!) f(x,\theta).$$

Example 5 (Differential operator (with convergence factor))**.** $f_t(x,\theta) := \frac{\mathrm{d}}{\mathrm{d}\theta}x(\theta)\phi_t(x) = \sum_{|n|<m} n x_n \phi_t(x) e^{in\theta}$. Here, $(\phi_t)_{t>0} \subset \mathcal{S}(\mathcal{H})$ is a family of convergence factors (for $R[f_t;\rho]$ to exist) that satisfies (1) the first moment $\int_{\mathcal{H}} |x|_{\mathcal{H}}|\phi_t(x)|\mathrm{d}x$ exists at every t, (2) $\phi_t \to 1$ in the weak sense as $t \to \infty$, and (3) (continuous and) (G,T)-equivariant. For example, we can take a Gaussian $\phi_t(x) = \exp(-|x|_{\mathcal{H}}^2/4t)$. Hence,

$$R[f_t;\rho](a,b) = \int_{\mathbb{R}^m} \sum_{|n|<m} n x_n \phi_t(x)\overline{\rho\left(\sum_{|n|<m} a_n x_n - b\right)}\mathrm{d}x.$$

5.4 Euclidean group $\mathrm{E}(n)$ equivariant map

Example 6. The Euclidean group $\mathrm{E}(n)$ is a semidirect product $\mathbb{R}^n \rtimes \mathrm{O}(n)$ of the translational group $\mathbb{R}^n$ and the orthogonal group $\mathrm{O}(n)$, which acts on $\mathbb{R}^n$ as $(U, s) \cdot t := Ut + s$ for any $t \in \mathbb{R}^n$ and $(U, s) \in O(n) \times \mathbb{R}^n$. So, put $\mathcal{H} \subset L^2(\mathbb{R}^m)$ and $T_{(U,s)}[x](t) := x(U^{-1}(t - s))$. Then,

$$R[f;\rho](a,b) := \int_{\mathcal{H}} f(x,(I,0))\overline{\rho\left(\int_{\mathbb{R}^n} a(t)\overline{x(t)}\mathrm{d}t - b\right)}\mathrm{d}x$$

$$S[R[f;\rho]](x,(U,t)) = \int_{\mathcal{H}\times\mathbb{R}} R[f;\rho](a,b)\sigma\left(\int_{\mathbb{R}^n} a(U^{-1}(t-s))\overline{x(s)}\mathrm{d}s - b\right)\mathrm{d}a\mathrm{d}b = f(x,(U,t))$$

We note that a more efficient representation for $L^2(\mathbb{R}^n)$ and/or a more general representation such as $L^2(\mathrm{SO}(2))$ and $L^2(\mathrm{E}(3))$, have been developed in the context of *steerable* CNNs [45, 46].

6 Related Work

The universality of (G)CNNs have been shown by several authors. Zhou [11, 12] have shown the universality of (non-group) deep ReLU CNNs. Interestingly, the networks increase the expressive power as the depth increases. Yarotsky [13] and Maron et al. [14] are the earliests to show the universality of GCNNs. In particular, the proof by Yarotsky [13] covers finite groups, compact groups, and a few noncompact groups ($\mathbb{R}^m$ and $SE(2)$). However, these universality arguments rely on the universality of the invariant polynomials. Similarly, Kumagai and Sannai [16] and Petersen and Voigtlaender [15] rely on the universality of fully-connected networks. On the other hand, our proof of the cc-universality is given a constructive and unified manner, covering a wide range of group G and feature space $\mathcal{H}$.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105, Lake Tahoe, 2012. Curran Associates, Inc.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[3] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *arXiv preprint: 2104.13478*, 2021.

[4] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep Sets. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[6] Risi Kondor and Shubhendu Trivedi. On the Generalization of Equivariance and Convolution in Neural Networks to the Action of Compact Groups. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2747–2755. PMLR, 2018.

[7] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and Equivariant Graph Networks. In *International Conference on Learning Representations*, 2019.

[8] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018.

[9] Taco S Cohen, Mario Geiger, and Maurice Weiler. A General Theory of Equivariant CNNs on Homogeneous Spaces. In H Wallach, H Larochelle, A Beygelzimer, F d'Alché-Buc, E Fox, and R Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[10] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch-Gordan Nets: a Fully Fourier Space Spherical Convolutional Neural Network. In S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[11] Ding-Xuan Zhou. Deep distributed convolutional neural networks: Universality. *Analysis and Applications*, 16(06):895–919, 2018.

[12] Ding-Xuan Zhou. Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, 48(2):787–794, 2020.

[13] Dmitry Yarotsky. Universal Approximations of Invariant Maps by Neural Networks. *Constructive Approximation*, 2021.

[14] Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman. On the Universality of Invariant Networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4363–4371. PMLR, 2019.

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

[16] Wataru Kumagai and Akiyoshi Sannai. Universal Approximation Theorem for Equivariant Maps by Group CNNs. *arXiv preprint: 2012.13882*, 2020.

[17] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, 1989.

[18] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[19] Bunpei Irie and Sei Miyake. Capabilities of three-layered perceptrons. In *IEEE International Conference on Neural Networks*, pages 641–648. IEEE, 1988.

[20] Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):183–192, 1989.

[21] S. M. Carroll and B. W. Dickinson. Construction of neural nets using the Radon transform. In *International Joint Conference on Neural Networks 1989*, volume 1, pages 607–611. IEEE, 1989.

[22] Yoshifusa Ito. Representation of functions by superpositions of a step or sigmoid function and their applications to neural network theory. *Neural Networks*, 4(3):385–394, 1991.

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

[24] H. N. Mhaskar. Neural Networks for Optimal Approximation of Smooth and Analytic Functions. *Neural Computation*, 8:164–177, 1996.

[25] Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.

[26] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.

[27] Noboru Murata. An integral representation of functions using three-layered betworks and their approximation bounds. *Neural Networks*, 9(6):947–956, 1996.

[28] Emmanuel Jean Candès. *Ridgelets: theory and applications*. PhD thesis, Standford University, 1998.

[29] Boris Rubin. The Calderón reproducing formula, windowed X-ray transforms, and Radon transforms in L^p-spaces. *Journal of Fourier Analysis and Applications*, 4(2):175–197, 1998.

[30] Matus Telgarsky. Benefits of depth in neural networks. In *29th Annual Conference on Learning Theory*, pages 1–23, 2016.

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

[32] Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks*, 108:296–330, 2018.

[33] Sho Sonoda and Noboru Murata. Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, 43(2):233–268, 2017.

[34] Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? In *Proceedings of the 32nd Conference on Learning Theory*, volume 99, pages 2667–2690, Phoenix, USA, 2019. PMLR.

[35] Greg Ongie, Rebecca Willett, Daniel Soudry, and Nathan Srebro. A Function Space View of Bounded Norm Infinite Width ReLU Nets: The Multivariate Case. In *International Conference on Learning Representations*, 2020.

[36] Rahul Parhi and Robert D. Nowak. Neural Networks, Ridge Splines, and TV Regularization in the Radon Domain. *arXiv preprint: 2006.05626*, 2020.

[37] Rahul Parhi and Robert D Nowak. Banach Space Representer Theorems for Neural Networks and Ridge Splines. *Journal of Machine Learning Research*, 22(43):1–40, 2021.

[38] Michael Unser. A Representer Theorem for Deep Neural Networks. *Journal of Machine Learning Research*, 20(110):1–30, 2019.

[39] Sho Sonoda, Isao Ishikawa, and Masahiro Ikeda. Ridge Regression with Over-Parametrized Two-Layer Networks Converge to Ridgelet Spectrum. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021*, volume 130, pages 2674–2682. PMLR, 2021.

[40] Sho Sonoda, Isao Ishikawa, and Masahiro Ikeda. Ghosts in Neural Networks: Existence, Structure and Role of Infinite-Dimensional Null Space. *arXiv preprint: 2106.04770*, 2021.

[41] Sho Sonoda, Isao Ishikawa, and Masahiro Ikeda. Fully-Connected Network on Noncompact Symmetric Space and Ridgelet Transform based on Helgason-Fourier Analysis. *arXiv preprint: 2203.01631*, 2022.

[42] Loukas Grafakos. *Classical Fourier Analysis*. Graduate Texts in Mathematics. Springer New York, 2 edition, 2008.

[43] Bharath K. Sriperumbudur, Kenji Fukumizu, and Gert R. G. Lanckriet. Universality, Characteristic Kernels and RKHS Embedding of Measures. *Journal of Machine Learning Research*, 12 (Jul):2389–2410, 2010.

[44] Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8: 143–195, 1999.

[45] Taco S Cohen and Max Welling. Steerable CNNs. In *International Conference on Learning Representations 2017*, pages 1–14, 2017.

[46] Maurice Weiler and Gabriele Cesa. General E(2)-Equivariant Steerable CNNs. In H Wallach, H Larochelle, A Beygelzimer, F d'Alché-Buc, E Fox, and R Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]