Stability of Spherical Convolutional Neural Networks to Rotation Diffeomorphisms

Zhan Gao†, Fernando Gama‡, and Alejandro Ribeiro†
†Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, USA
‡Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, USA
E-mails: {gaozhan, ariberio}@seas.upenn.edu, fgama@berkeley.edu

***Abstract*—Spherical convolutional neural networks (Spherical CNNs) learn nonlinear representations from spherical signals. These are mathematical models for data arising in 3-D objects and have found applications in computer vision, light detection and ranging (LIDAR), and planning among others. The Spherical CNN comprises a cascade of layers, each with a series of spherical convolutions (spherical filters) followed by a pointwise nonlinearity. This paper investigates the impact that structure perturbations in spherical signals have on Spherical CNN outputs. We consider general perturbations as rotation diffeomorphisms in the spherical surface, and show that Spherical CNNs with Lipschitz filters are stable to such perturbations. In particular, we establish that the output difference of Spherical CNN induced by the diffeomorphism perturbation is bounded by the perturbation size. This result also shows the role of the nonlinearity and the architecture width and depth, and indicates how Spherical CNNs exploit the rotational structure of spherical signals to gain superior performance. We corroborate theoretical findings in 3-D object classification, and observe stable performance to rotation diffeomorphisms.**

***Index Terms*—Spherical convolutional neural networks, spherical filters, stability analysis, rotation diffeomorphisms**

I. Introduction

Data arising in 3-D applications such as object classification in computer vision [1], panoramic video processing in self-driving cars [2] and 3-D surface reconstruction in medical imaging [3] can be described as belonging to a spherical surface. We can thus model this data as spherical signals, essentially assigning a scalar (or vector) value to each point on the sphere, providing a mathematical representation for 3-D data [4], [5]. Spherical signals are supported on the spherical surface and processing such signals requires architectures capable of exploiting this structure. In particular, we are concerned about exploiting the rotational structure of spherical signals, which can be captured by means of the rotation group [6], [7]. The latter is a mathematical group defined in the space of spherical surface equipped with the operation of rotation [8], [9]. Since this is a group, it admits a definition of spherical convolutional filter, which is a linear operator that computes weighted, rotated combinations of the values of a spherical signal [10], [11]. Spherical convolutional filters effectively exploit the inherent structure in 3-D data to extract higher-level features and are key operators defining spherical convolutional neural networks.

Spherical convolutional neural networks (Spherical CNNs) are information processing architectures that capture nonlinear relationships between the 3-D data and the relevant information. Spherical CNNs consist of a cascade of layers, each of which applies a series of spherical convolutional filters followed by a pointwise nonlinearity [12]. The inclusion of nonlinearities and multiple layers dons Spherical CNNs with an enhanced representation power, which is capable of capturing nonlinear relationships. Inherited from spherical convolutions, Spherical CNNs are adapted to the rotational structure embedded in spherical signals and exhibit superior performance [12]–[15].

Considering the evident success of Spherical CNNs in processing 3-D data, in this paper, we focus on analyzing the properties that Spherical CNNs exhibit as they pertain to the rotational structure present in spherical signals. To be more precise, we characterize how spherical convolutional filters and Spherical CNNs react to structure perturbations in input signals, offering an insight into the reasons behind their observed success. Our rationale is that if two images are close to each other, the processing architectures shall yield similar outputs since they likely capture the same information in their data structures. We thus define a notion of rotation diffeomorphism (Def. 1) describing general perturbations in the spherical surface [16], which is a local rotation of the signal (i.e. a perturbation of the spherical space where each point may be rotated by a different angle). We prove that spherical convolutions with Lipschitz filters are stable to diffeomorphism perturbations (Thm. 1), and that this property is inherited by Spherical CNNs (Thm. 2). Stability to such perturbations is of paramount importance in 3-D applications such as object identification [17], where small changes in spherical signals may be introduced by a different viewing angle or distance.

We start this paper by introducing spherical convolutions built for spherical signals on the rotation group and defining spherical convolutional neural networks (Section II). We model general perturbations in spherical signals as rotation diffeomorphisms, and show that spherical convolutions with Lipschitz filters are stable to such perturbations, i.e., the filter output difference is bounded linearly by the perturbation size, where the proportionality constant depends on the Lipschitz condition (Section III). We then carry over the stability analysis from spherical convolutions to Spherical CNNs (Section IV). These results indicate that these architectures yield similar outputs for signals that are perturbed by small rotation diffeomorphisms. We corroborate theory in the problem of 3-D object classification in Section V and draw the conclusion in Section VI.

II. Spherical Convolutional Neural Networks

Consider the spherical surface $\mathbb{S}_2 \subset \mathbb{R}^3$ contained in $\mathbb{R}^3$. Given a Euclidean coordinate system $\mathcal{S}(x, y, z)$, a point $u = (x_u, y_u, z_u) \in \mathbb{S}_2$ can be characterized by three variables: the polar angle $\theta_u = \arctan(\sqrt{x_u^2 + y_u^2}/z_u) \in [0, \pi]$ measured from the z-direction, the azimuth angle $\phi_u = \arctan(x_u/y_u) \in [0, 2\pi)$ along the xy-plane and the radial distance $d = \sqrt{x_u^2 + y_u^2 + z_u^2}$

from the origin. Without loss of generality, we set $d=1$ on the unit sphere. We can represent a point $u \in \mathbb{S}_2$ by the vector

$$u=(\theta_u,\phi_u)=\Big[\sin(\theta_u)\cos(\phi_u),\sin(\theta_u)\sin(\phi_u),\cos(\theta_u)\Big]. \quad (1)$$

For points with polar angles $\theta_u = 0$ or $\theta_u = \pi$, the azimuth angles are assumed to be zero. Spherical signal is defined as the map $x : \mathbb{S}_2 \to \mathbb{R}$ that assigns a signal value $x(u) \in \mathbb{R}$ to each point $u \in \mathbb{S}_2$ on the sphere. Equivalently, it can be represented as the map from the angular variables (θ_u, ϕ_u) to the real line $\mathbb{R}$, i.e. $x(u) = x(\theta_u, \phi_u)$. These signals typically describe 3-D data, e.g., LIDAR images, X-ray models, etc. In 3-D object classification, for instance, the spherical signal may be the distance between the object center and the farthest intersection point along the ray direction (ϕ_u, θ_u), i.e., $x(\phi_u, \theta_u) = d(\phi_u, \theta_u)$.

A. Spherical Convolution

Rotation operation. An elementary operation for spherical signals is the rotation r, which is characterized by a rotation point $N^r \in \mathbb{S}_2$ and a rotation angle $\beta^r \in [0, 2\pi)$. It displaces points on the sphere by β^r degrees along the axis that passes through the origin and the point N^r. To describe the rotation with angular variables [cf. (1)], let $r^\theta : \mathbb{S}_2 \to \mathbb{R}$ be the polar angle displacement and $r^\phi : \mathbb{S}_2 \to \mathbb{R}$ the azimuth angle displacement induced by r. We can parametrize the rotation $r : \mathbb{S}_2 \to \mathbb{S}_2$ as

$$\begin{aligned} r\circ u = r\circ(\theta_u,\phi_u) = [&\sin(\theta_u + r^\theta(\theta_u,\phi_u))\cos(\phi_u + r^\phi(\theta_u,\phi_u)),\\ &\sin(\theta_u + r^\theta(\theta_u,\phi_u))\sin(\phi_u + r^\phi(\theta_u,\phi_u)),\\ &\cos(\theta_u + r^\theta(\theta_u,\phi_u))] \end{aligned} \quad (2)$$

where r^θ and r^ϕ are given by the Rodrigues' rotation formula [18]. Taking the azimuthal rotation as an example, the rotation axis coincides with the z-axis of $\mathcal{S}(x, y, z)$, the polar angle remains unchanged $r^\theta(\theta_u, \phi_u) = 0$, and the azimuth angle is displaced by a constant $r^\phi(\theta_u, \phi_u) = \beta^r$ for all $u = (\theta_u, \phi_u) \in [0, \pi] \times [0, 2\pi)$. The set of rotation operations about the origin (2) then defines a mathematical group under the operation of composition, referred to as the 3-*D rotation group* $\mathsf{SO}(3)$ [19].

Spherical convolution. The spherical surface $\mathbb{S}_2$ equipped with the rotation group $\mathsf{SO}(3)$ allows defining a convolution operation for spherical signals. Given two signals $x, h : \mathbb{S}_2 \to \mathbb{R}$, the spherical convolution $*_{\mathsf{SO}(3)}$ between them is

$$y(u) = (h *_{\mathsf{SO}(3)} x)(u) = \int_{\mathsf{SO}(3)} h(r^{-1} \circ u)x(r)\ dr \quad (3)$$

where $y(u)$ is the output spherical signal, and $x(r)$ is the concise notation for $x(r \circ u_0)$ with $u_0 = (0, 0)$ the point given by $\theta_u = \phi_u = 0$. Spherical convolutions exploit the rotational structure present in spherical signals to generate higher-level features.

To describe the spherical convolution with angular variables [cf. (1)], we first represent the rotation operation $r \in \mathsf{SO}(3)$ in terms of the ZYZ Euler parametrization [20]

$$r = r_{\phi_r\theta_r\rho_r} = r^z_{\phi_r} r^y_{\theta_r} r^z_{\rho_r} \quad (4)$$

where $r^z_{\phi_r}$, $r^y_{\theta_r}$ and $r^z_{\rho_r}$ are the rotations along z-, y- and z-axis, each one with a rotation angle of $\phi_r \in [0, 2\pi)$, $\theta_r \in [0, \pi]$ and $\rho_r \in [0, 2\pi)$. The Euler parametrization allows us to decompose any rotation r as three consecutive rotations. Recall the normalized Haar measure on the rotation group [21]

$$dr = \frac{d\phi_r}{2\pi}\frac{\sin(\theta_r)d\theta_r}{2}\frac{d\rho_r}{2\pi}, \quad (5)$$

and we can rewrite (3) as

$$\begin{aligned} y(u) &= (h *_{\mathsf{SO}(3)} x)(u) \quad (6)\\ &= \frac{1}{8\pi^2}\int \Big(h\big(r^{-1}_{\phi_r\theta_r\rho_r}\circ u\big)x\big(r_{\phi_r\theta_r\rho_r}\big)\Big) sin(\theta_r)\ d\theta_r d\phi_r d\rho_r \end{aligned}$$

where $r_{\phi_r\theta_r\rho_r}$ is the rotation parametrized by the Euler angles [cf. (4)] and $r^{-1}_{\phi_r\theta_r\rho_r}$ is its inverse. By noting that $r_{\phi_r\theta_r\rho_r} \circ u_0 = (\theta_r, \phi_r)$ and using the Rodrigues' formula (2), we further get

$$\begin{aligned} y(\theta_u,\phi_u) &= (h *_{\mathsf{SO}(3)} x)(\theta_u,\phi_u)\\ &= \frac{1}{8\pi^2}\int \Big(h\big(\theta_u + r^{\theta\ -1}_{\phi_r\theta_r\rho_r}(\theta_u,\phi_u), \phi_u + r^{\phi\ -1}_{\phi_r\theta_r\rho_r}(\theta_u,\phi_u)\big)\cdot\\ &\qquad x(\theta_r,\phi_r)\Big)\cdot\ \sin(\theta_r)\ d\theta_r d\phi_r d\rho_r. \end{aligned} \quad (7)$$

where $r^{\theta\ -1}_{\phi_r\theta_r\rho_r}$ and $r^{\phi\ -1}_{\phi_r\theta_r\rho_r}$ are polar and azimuth angle displacements induced by $r^{-1}_{\phi_r\theta_r\rho_r}$. We see that (7) is equivalent to (3) while describing it with specific angular variables, based on which we could evaluate the output of spherical convolution mathematically. Put simply, we parametrize the spherical point with polar and azimuth angles and the rotation operation with Euler angles. This allows us to carry out the integral in (7) explicitly, serving for the stability analysis in next sections.

B. Spherical convolutional neural network

Spherical convolutional neural network consists of a cascade of L layers, each of which applies a series of spherical convolutions followed by a pointwise nonlinearity. At layer $\ell = 1, \ldots, L$, there is a collection of F input features (spherical signals) $\{x^f_{\ell-1}\}^F_{f=1}$. These features are processed by a series of spherical convolutions to obtain intermediate features

$$y^{fg}_\ell = h^{fg}_\ell *_{\mathsf{SO}(3)} x^f_{\ell-1},\ \forall\ f = 1, \ldots, F,\ g = 1, \ldots, G, \quad (8)$$

where $x^f_{\ell-1}$ is convolved with the spherical filter h^{fg}_ℓ to output y^{fg}_ℓ. The latter are aggregated over index f and passed through a pointwise nonlinearity $\sigma(\cdot)$ to generate the gth output feature

$$x^g_\ell = \sigma\Big(\sum_{f=1}^F y^{fg}_\ell\Big),\ \forall\ g = 1, \ldots, G. \quad (9)$$

Operations (8)-(9) define the recursive operation at each layer, which is equivalent to filtering F input features through a bank of FG spherical filters, aggregating the output of each gth filter across all F input features, and applying a pointwise nonlinearity $\sigma(\cdot)$. The number of layers L and features F as well as the specific form of the nonlinearity $\sigma(\cdot)$ are typically design choices, while the collection of spherical filters $\mathcal{H} = \{h^{fg}_\ell\}_{\ell fg}$ are *learned* by minimizing some objective function over a training set. Without loss of generality, we assume a single input $x_0 = x$ and a single output $\Phi(x; \mathcal{H}) = x_L$ where $\Phi : \mathbb{S}_2 \to \mathbb{S}_2$ represents the nonlinear mapping given by the Spherical CNN (9). To simplify notation, we denote the spherical convolution as a filtering operation, i.e. $y = h *_{\mathsf{SO}(3)} x := H(x)$.

Inherited from spherical convolutions, the Spherical CNN (9) is designed to leverage the rotational structure embedded in 3-D data. Thus, we expect it to exhibit strong performance on 3-D learning tasks (in fact, this is observed in practice [12]). In next sections, we demonstrate the perturbation stability of spherical convolutions and further Spherical CNNs. This property illustrates exactly how Spherical CNNs leverage the data structure, and help understand the observed superior performance.

(a) Unperturbed space (b) Perturbed space

Figure 1. Rotation diffeomorphism on the sphere [cf. Def 1]. (a) Original sphere. (b) Perturbed sphere after rotation diffeomorphism.

III. Stability of Spherical Convolutions

Adopting the spherical convolution as the main operation to process spherical signals exploits the rotational structure of 3-D data and exhibits improved performance. In this section, we explore the effect that structure perturbations in spherical signals have on the output of spherical convolution, to explain theoretically its superior performance.

A. Diffeomorphism perturbation

In general, we are interested in how the processing architecture based on spherical convolutions fares when acting on two signals that are similar, but not quite the same. In a sense, we want the architecture to yield a similar output if the input is perturbed by small changes probably due to noise or other unimportant causes [22], [23].

Denote by $r_u \in \mathsf{SO}(3)$ the rotation operation that maps the point $u_0 = (0,0) \in \mathbb{S}_2$ to the point $u \in \mathbb{S}_2$ along the shortest arc, i.e. $u = r_u \circ u_0$. We are interested in perturbations in the spherical surface $\tau : \mathbb{S}_2 \to \mathbb{S}_2$ that can be described as local rotations, i.e. rotations whose characterization changes depending on the specific point u to which it is applied. Let $\tau \circ u$ be the resulting point after the perturbation τ is applied. We can similarly obtain the point $\tau \circ u = r_{\tau \circ u} \circ u_0$ by applying the rotation $r_{\tau \circ u}$ to u_0. Combining these notions of r_u and $r_{\tau \circ u}$ we formally define a diffeomorphism perturbation as follows.

Definition 1 (Diffeomorphism perturbation)**.** Let $\tau : \mathbb{S}_2 \to \mathbb{S}_2$ be a diffeomorphism, i.e. a bijective differentiable function whose inverse $\tau^{-1} : \mathbb{S}_2 \to \mathbb{S}_2$ is differentiable as well. This diffeomorphism can be used to define a local rotation $\tau_u \in \mathsf{SO}(3)$ as

$$\tau_u = r_{\tau \circ u} \circ r_u^{-1}, \ \forall \ u \in \mathbb{S}_2. \tag{10}$$

The diffeomorphism perturbation is the set of local rotations $\{\tau_u\}_{u \in \mathbb{S}_2}$ [cf. (10)] that satisfy $\|\tau\| < \infty$ and $\|\nabla \tau\| < \infty$ for

$$\|\tau\| := \max_{u \in \mathbb{S}_2} \left\{ \beta^{\tau_u} \right\}, \tag{11}$$

where β^{τ_u} is the rotation angle of the local rotation τ_u [cf. (10)]; and for

$$\|\nabla \tau\| = \max_{(\theta_u, \phi_u) \in [0,\pi] \times [0, 2\pi)} \left\{ \left| \frac{\partial \tau^\theta}{\partial \theta_u} \right|, \left| \frac{\partial \tau^\phi}{\partial \phi_u} \right| \right\}. \tag{12}$$

where τ^θ and τ^ϕ are polar and azimuthal angle displacements induced by τ such that $\tau^\theta(\theta_u, \phi_u) = \tau_u^\theta(\theta_u, \phi_u)$ and $\tau^\phi(\theta_u, \phi_u) = \tau_u^\phi(\theta_u, \phi_u)$ at each point $u \in \mathbb{S}_2$. The signal resulting from a diffeomorphism perturbation to the spherical structure of a given signal x is denoted as x_τ such that

$$x_\tau(u) = x(\tau \circ u) \ , \ \forall \ u \in \mathbb{S}_2. \tag{13}$$

The diffeomorphism τ applied to the spherical signal is called a rotation diffeomorphism.

The rotation diffeomorphism is, essentially, a set of local rotations where rotation axes and rotation angles depend on specific points on the sphere. We remark that since rotations can displace a point $u \in \mathbb{S}_2$ to any other point on the sphere, this representation can actually model any structure perturbation in the spherical space. We measure the *perturbation size* by $\|\tau\|$ and $\|\nabla \tau\|$ [cf. (11), (12)], and an example of a rotation diffeomorphism is illustrated in Fig. 1 [16].

B. Stability to diffeomorphism perturbations

We have now defined diffeomorphism perturbations of the signal, as those that modify the underlying space $\mathbb{S}_2$ by using local rotations, i.e. rotations whose size depends on the specific point in space. Before claiming our main result, we first define the normalized spherical norm to quantify spherical signals.

Definition 2 (Normalized spherical norm)**.** For a spherical signal $x : \mathbb{S}_2 \to \mathbb{R}$, the normalized spherical norm is defined as

$$\|x\| = \Big(\frac{1}{2\pi^2} \int x(\theta_u, \phi_u)^2 d\theta_u d\phi_u \Big)^{\frac{1}{2}}. \tag{14}$$

where $\theta_u \in [0, \pi]$ and $\phi_u \in [0, 2\pi)$ describe the signal support in the spherical coordinate system [cf. (1)]. For a collection of signals $\mathbf{x} = \{x^f\}_{f=1}^F$, we define $\|\mathbf{x}\| = \sum_{f=1}^F \|x^f\|$.

This is a proper norm in the sense that it is (i) absolutely scalable, (ii) positive definite, and (iii) satisfies the triangular inequality. It allows us to define the normed space $\mathbb{L}^2(\mathbb{S}_2) = \left\{ \mathbf{x} : \mathbb{S}_2 \to \mathbb{R} \ : \ \|\mathbf{x}\| < \infty \right\}$ that collects finite-energy spherical signals. Without loss of generality, any valid norm can be applied here for the stability analysis. We adopt the above norm corresponding to our mathematical descriptions for spherical signals [cf. (1)]. Next, we restrict our attention to the family of *Lipschitz filters*.

Definition 3 (Lipschitz filter)**.** A filter $h : \mathbb{S}_2 \to \mathbb{R}$ is a *Lipschitz filter*, if there exists a constant C_h such that

$$|h(u)| \leq C_h \ , \ \frac{|h(u_1) - h(u_2)|}{\|u_1 - u_2\|_2} \leq C_h \tag{15}$$

for all $u, u_1, u_2 \in \mathbb{S}_2$, where $\| \cdot \|_2$ is the Euclidean norm in $\mathbb{R}^3$.

Now we can formally characterize the stability of the spherical convolution to diffeomorphism perturbations [cf. Def. 1] in the input spherical signal.

Theorem 1 (Stability of spherical convolutional filters)**.** Let $x \in \mathbb{L}^2(\mathbb{S}_2)$ be a spherical signal, and H be a Lipschitz spherical convolutional filter w.r.t. C_h [cf. Def 3]. Consider a rotation diffeomorphism τ [cf. Def 1] that satisfies

$$\|\tau\| \leq \epsilon \ , \ \|\nabla \tau\| \leq \epsilon \leq \frac{1}{2}. \tag{16}$$

Then, for the perturbed signal x_τ [cf. (13)] it holds that

$$\|H(x) - H(x_\tau)\| \leq 8 C_h \epsilon \|x\| + \mathcal{O}(\epsilon^2) \tag{17}$$

where $\| \cdot \|$ is the normalized spherical norm [Def. 2].

Proof. See the extended version [24]. □

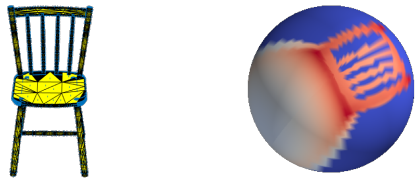

Figure 2. 3-D model (left) and spherical signal (right) of a chair.

Theorem 1 establishes that spherical convolutional filters are stable to diffeomorphism perturbations in the domain. More specifically, the difference in the output between filtering x or a perturbed version x_τ of it, is bounded. In fact, it is bounded linearly by the size of the perturbation ϵ, and the proportionality constant $8C_h$ depends on the spherical filter characteristics. A careful design of the filter can reduce the constant C_h and thus lead to more stable filters. The constant term 8 depends on the domain, and can be improved by further restricting the class of filters or the size of perturbations. Furthermore, the proportionality constant is independent of the input signal x such that the bound holds uniformly for all spherical signals.

IV. Stability of Spherical Convolutional Neural Networks

Spherical CNNs $\Phi(x;\mathcal{H})$ [cf. (9)] are information processing architectures to learn nonlinear representations from spherical signals. The stability of the Spherical CNN to diffeomorphism perturbations (Def. 1) is inherited from the stability of spherical convolutions. The inclusion of pointwise nonlinearities in the processing pipeline affects the stability constant. In particular, we consider nonlinearities that are Lipschitz, as defined next.

Definition 4 (Lipschitz nonlinearity)**.** A nonlinearity $\sigma : \mathbb{R} \to \mathbb{R}$ satisfying $\sigma(0) = 0$ is Lipschitz if there exists a constant $C_\sigma > 0$ such that

$$|\sigma(a) - \sigma(b)| \leq C_\sigma |a - b|, \ \forall \ a, b \in \mathbb{R}. \tag{18}$$

When Spherical CNNs are built from Lipschitz filters (Def. 3) and use Lipschitz nonlinearities (Def. 4), they pertain the stability to diffeomorphism perturbations (Def. 1).

Theorem 2. Let $x \in \mathbb{L}^2(\mathbb{S}_2)$ be a spherical signal and $\Phi(\cdot;\mathcal{H})$ be a Spherical CNN [cf. (9)] consisting of L layers, with F features per layer, built with Lipschitz filters w.r.t. C_h [cf. Def 3] and Lipschitz nonlinearities w.r.t. C_σ [cf. Def. 4]. Consider a rotation diffeomorphism τ [cf. Def 1] that satisfies

$$\|\tau\| \leq \epsilon \ , \ \|\nabla\tau\| \leq \epsilon \leq \frac{1}{2}. \tag{19}$$

Then, for the perturbed signal x_τ [cf. (13)] it holds that

$$\|\Phi(x;\mathcal{H}) - \Phi(x_\tau;\mathcal{H})\| \leq 8(C_\sigma C_h)^L F^{L-1} \epsilon \|x\| + \mathcal{O}(\epsilon^2) \tag{20}$$

where $\|\cdot\|$ is the normalized spherical norm [Def. 2].

Proof. See the extended version [24]. □

Theorem 2 determines that Spherical CNNs are stable to diffeomorphism perturbations. More concretely, the difference between the output of a Spherical CNN applied to an input signal and to its perturbed version is bounded proportionally to the size of the perturbation ϵ. This means that if the signals are close to each other, the outputs of the Spherical CNN will be close as well. The proportionality constant depends on the learned filters through C_h and depends on the neural network design choices through F, L and C_σ. We see that the deeper the Spherical CNN is, the looser the bound is, mainly due to the propagation of the difference through the architecture. In any case, the output difference of the Spherical CNN is linear in the size of the perturbation, guaranteeing stability of the architecture. The latter explains how Spherical CNNs exploit the rotational structure present in 3-Data and how Spherical CNNs maintain performance under admissible perturbations.

V. Numerical Experiments

We have proved that Spherical CNNs are stable under diffeomorphism perturbations. This property illustrates how Spherical CNNs adequately exploit the data structure to improve their learning performance. In fact, they have already been shown successful in classification tasks [12]. Thus, in what follows, we focus solely on corroborating theoretical findings by numerical experiments with admissible perturbations; for comparative performance with other methods, please refer to [12].

Dataset and problem setting. The shape classification problem of 3-D object is considered on the ModelNet40 dataset [25], i.e., given a spherical signal, the goal is to find out which class its represented object belongs to. There are 40 classes in the dataset, where we use 9683 samples for training and 29595 samples for testing. We parametrize the spherical signal in a 64×64 resolution; see Fig. 2 for an example of spherical signal.

Architecture and training. We consider the Spherical CNN of 8 layers, each containing 16, 16, 32, 32, 64, 64, 128 and 128 features, with the ReLU nonlinearity. For a readout layer, we apply a global weighted average pooling for a descriptor vector and the latter is projected into the number of object classes. We train the architecture for 50 epochs with the ADAM optimizer and a batch size of 16 samples. The learning rate is $1 \cdot 10^{-3}$ and is divided by 5 on epochs 30 and 40, respectively.

Diffeomorphism types. We test the output changes of a Spherical CNN when input signals are subject to four different types of rotation diffeomoprhisms. Namely, we carry out local rotations along the latitude, indicating different perturbation severity. For type 1, we rotate every other sampled point at each latitude with a random degree drawn from $[-3, 3]$, where we assume the clockwise direction as the positive direction. Type 2 rotates every other sampled point with a random degree drawn from $[-6, 6]$. Type 3 considers rotations of each sampled point with a random degree drawn from $[-3, 3]$. Finally, type 4 perturbs blocks of 3 samples at each latitude separately, rotating the second point to the third point and interpolating the values of the remaining sampled points, where the maximal degree change is approximately 6 degrees. Fig. 3 shows perturbed spherical signals stemming from these four rotation diffeomorphisms.

Stability to perturbations. Table I shows the classification accuracy of the trained Spherical CNN when assuming all signals in the test set are perturbed by each of the aforementioned diffeomorphism types. The unperturbed classification accuracy is 0.864 for reference. In general, the Spherical CNN exhibits strong robustness to the rotation diffeomorphism in all cases, as expected from Theorem 2. We observe that type 1 has little effect on the classification accuracy, while types 2 and

(a) (b) (c) (d)

Figure 3. Rotation diffeomorphism on the spherical signal. (a) Type 1. (b) Tyep 2. (c) Type 3. (d) Type 4.

Table I: Test classification accuracy of the Spherical CNN for Diffeomorphism 1-4. Relative root mean square error (RMSE) of the Spherical CNN output features for Diffeomorphism 1-4.

Diffeomorphism	Classification accuracy	Relative RMSE
Type 1	0.863	0.0614
Type 2	0.856	0.0801
Type 3	0.858	0.0715
Type 4	0.828	0.1152

3 slightly decrease the accuracy, likely due to the increase of the maximal degree change and the number of perturbed sampled points. Type 4 is most severe as observed from Fig. 3d, however, the Spherical CNN only suffers from 0.036 accuracy loss indicating the stability of the Spherical CNN to rotation diffeomorphisms. Similarly, Table I shows the relative root mean square error (RMSE) of the output features of the final spherical convolution layer under above rotation diffeomorphisms. The relative RMSEs maintain low values in all cases, which also implies the diffeomorphism stability. A higher classification accuracy typically corresponds to a lower relative RMSE.

VI. Conclusion

This paper discussed the stability of spherical convolutional filters and Spherical CNNs to structure perturbations in spherical signals. We modeled general perturbations as rotation diffeomorphisms in the spherical space, and proved spherical filters are stable to such perturbations by analyzing in the spherical coordinate system with Euler parametrization. We then showed that Spherical CNNs inherit the stability to diffeomorphism perturbations (provided that the involved filters are Lipschitz), and analyzed the explicit role played by the perturbation size, filter property, nonlinearity and architecture width and depth on the stability of Spherical CNNs. These results establish that as long as two signals are similar, their outputs of spherical filters and Spherical CNNs will also be similar. It illustrates the ways in which Spherical CNNs are capable of exploiting the rotational structure in 3-D data and thus, generalize well to unseen or perturbed samples. These theoretical findings were corroborated through numerical experiments of 3-D object classification.

References

[1] M. Yavartanoo, E. Y. Kim, and K. M. Lee, "Spnet: Deep 3d object classification and retrieval using stereographic projection," in *14th Asian Conference on Computer Vision*, Perth, Australia, 2-6 Dec. 2018, Springer, pp. 691–706.

[2] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[3] L. Maier-Hein, P. Mountney, A. Bartoli, et al., "Optical techniques for 3d surface reconstruction in computer-assisted laparoscopic surgery," *Medical image analysis*, vol. 17, no. 8, pp. 974–996, Dec. 2013.

[4] R. S. Llonch, E. Kokiopoulou, I. Tošić, and P. Frossard, "3d face recognition with sparse spherical representations," *Pattern Recognition*, vol. 43, no. 3, pp. 824–834, 2010.

[5] F. J. Simons and A. Plattner, "Scalar and vector slepian functions, spherical signal estimation and spectral analysis," *arXiv preprint arXiv:1306.3184 [physics.data-an]*, 13 June 2013.

[6] A. Makadia, L. Sorgi, and K. Daniilidis, "Rotation estimation from spherical images," in *17th International Conference on Pattern Recognition*, Cambridge, UK, 26 Aug. 2004, IEEE, pp. 590–593.

[7] D. E Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," in *30th Conference on Computer Vision and Pattern Recognition*, Hawaii, HO, 22-25July 2017, IEEE, pp. 5028–5037.

[8] G. S. Chirikjian, A. B. Kyatkin, and A. C. Buckingham, "Engineering applications of noncommutative harmonic analysis: with emphasis on rotation and motion groups," *Appl. Mech. Rev.*, vol. 54, no. 6, pp. B97–B98, 2001.

[9] I. M. Gelfand, R. A. Minlos, and Z. Y. Shapiro, *Representations of the Rotation and Lorentz Groups and their Applications*, Courier Dover Publications, 2018.

[10] A. Derighetti, *Convolution operators on groups*, Springer, 2011.

[11] Y. Su and K. Grauman, "Learning spherical convolution for fast features from 360 imagery," in *31st Conference on Neural Information Processing Systems*, Long Beach, CA, 4-9 Dec. 2017, pp. 529–539, Neural Inform. Process. Foundation.

[12] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis, "Learning so(3) equivariant representations with spherical cnns," in *European Conference on Computer Vision*, Munich, Germany, 8-14 Sep. 2018, pp. 52–68.

[13] T. S. Cohen, M. Geiger, J. Khler, and M. Welling, "Spherical cnns," in *6th International Conference on Learning Representations*, Vancouver, BC, 30 APR-3 MAY 2018.

[14] N. Perraudin, M. Defferrard, T. Kacprzak, and R. Sgier, "Deepsphere: Efficient spherical convolutional neural network with healpix sampling for cosmological applications," *Astronomy and Computing*, vol. 27, pp. 130–146, 2019.

[15] C. Jiang, J. Huang, K. Kashinath, P. Marcus, and M. Niessner, "Spherical cnns on unstructured grids," *arXiv preprint arXiv:1901.02039 [cs.CV]*, 7 Jan. 2019.

[16] M. Rosenthal, W. Wu, E. Klassen, and A. Srivastava, "Nonparametric spherical regression using diffeomorphic mappings," *arXiv:1702.00823v1 [stat.OT]*, 2 Feb. 2017.

[17] Z. Cao, Q. Huang, and R. Karthik, "3d object classification via spherical projections," in *2017 International Conference on 3D Vision*, Qingdao, China, 10-12Oct. 2017, IEEE, pp. 566–574.

[18] J. E. Mebius, "Derivation of the euler-rodrigues formula for three-dimensional rotations from the general formula for four-dimensional rotations," *arXiv preprint arXiv:math/0701759 [math.GM]*, 26 Jan. 2007.

[19] P. J. Kostelec and D. N. Rockmore, "Ffts on the rotation group," *Journal of Fourier analysis and applications*, vol. 14, no. 2, pp. 145–179, 2008.

[20] J. Stuelpnagel, "On the parametrization of the three-dimensional rotation group," *SIAM review*, vol. 6, no. 4, pp. 422–430, 1964.

[21] J. R. Driscoll and D. M. Healy, "Computing fourier transforms and convolutions on the 2-sphere," *Advances in applied mathematics*, vol. 15, no. 2, pp. 202–250, 1994.

[22] K. Chen and L. Liu, "Geometric data perturbation for privacy preserving outsourced data mining," *Knowledge and information systems*, vol. 29, no. 3, pp. 657–695, 2011.

[23] Mohammad A. Nazari S., S. Gholtashi, A. R. Kahoo, W. Chen, and Y. Chen, "Data-driven multitask sparse dictionary learning for noise attenuation of 3d seismic data," *Geophysics*, vol. 82, no. 6, pp. V385–V396, 2017.

[24] Z. Gao, F. Gama, and A. Ribeiro, "Spherical convolutional neural networks: Stability to perturbations in so (3)," *arXiv preprint arXiv:2010.05865 [cs.LG]*, 12 Oct. 2020.

[25] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *28th Conference on Computer Vision and Pattern Recognition*, Boston, MA, 7-12 June 2015, IEEE, pp. 1912–1920.