PDO-eS^{2}CNNs: Partial Differential Operator Based Equivariant Spherical CNNs

Abstract

Spherical signals exist in many applications, e.g., planetary data, LiDAR scans and digitalization of 3D objects, calling for models that can process spherical data effectively. It does not perform well when simply projecting spherical data into the 2D plane and then using planar convolution neural networks (CNNs), because the distortion from projection will make translation equivariance ineffective.

Actually, a good principle of designing spherical CNNs is to convert the shift equivariance property in planar CNNs to rotation equivariance in the spherical domain. In this work, we use orientable partial differential operators (PDOs) to design a spherical equivariant CNN, PDO-eS^{2}CNN, which is exactly rotation equivariant in the continuous domain. We then discretize PDO-eS^{2}CNNs, and analyze the equivariance error resulted from discretization. This is the first time that the equivariance error is theoretically analyzed in the spherical domain. In experiments, PDO-eS^{2}CNNs show greater parameter efficiency and achieve new state-of-the-art results on several benchmark datasets.

1 Introduction

Nowadays, many machine learning problems in computer vision require to process spherical data found in various applications; for instance, omnidirectional RGB-D images such as Matterport (Chang et al. 2017), 3D LiDAR scans from self-driving cars (Dewan et al. 2016), molecular modelling (Boomsma and Frellsen 2017), and planetary signals in earth science (Racah et al. 2017). Unfortunately, naively mapping spherical signals to $\mathbb{R}^2$ and then using planar convolution neural networks (CNNs) is destined to fail, because the space-varying distortions introduced by projection will make shift equivariance ineffective.

Actually, the success of planar CNNs is mainly attributed to their shift equivariance (Cohen and Welling 2016): shifting an image and then feeding it through multiple layers is the same as feeding the original image and then shifting the resulted feature maps. Since there do not exist translation symmetries in the spherical domain, a good principle of modifying planar CNNs to spherical CNNs is to convert the shift equivariance property to 3D rotation equivariance in the spherical domain. Motivated by this, Cohen et al. (2018) and Esteves et al. (2018) propose spherical CNNs that are equivariant over the $SO(3)$ group. However, these methods represent the sphere using the spherical coordinates, which over-sample near the poles and cause significant distortion.

To avoid the impact of distortion, many recent works process spherical data using much more uniform representations. Among these methods, Cohen et al. (2019) and Zhang et al. (2019) approximate the sphere using the icosahedron and propose Icosahedral CNN and orientation-aware CNN, respectively. Specifically, Icosahedral CNN (Cohen et al. 2019) is rotation equivariant while orientation-aware CNN (Zhang et al. 2019) is beneficial for some orientation-aware tasks, such as semantic segmentation with preferred orientation. However, these methods need project spherical data to the icosahedron, resulting in inaccurate representations.

Actually, there exist some discretizations of the sphere that are both uniform and accurate, like the icosahedral spherical mesh (Baumgardner and Frederickson 1985) and the HealPIX (Gorski et al. 2005). However, these representations are non-Euclidean structured grids (Bronstein et al. 2017), which have no uniform locality, thus conventional convolutions defined in the Euclidean case (e.g., square lattices) cannot work on them. Accordingly, Jiang et al. (2019) propose MeshConvs, which use orientable parameterized partial differential operators (PDOs) to process spherical signals represented by non-Euclidean structured grids. However, MeshConvs are not rotation equivariant.

In order to address the above issues in the existing methods, we combine the advantages of (Cohen et al. 2019) and (Jiang et al. 2019) together, and propose PDO-eS^{2}CNN, which is an orientable spherical CNN equivariant over $SO(3)$ based on PDOs. The distinction from (Cohen et al. 2019) is that our model is orientation-aware and can work on much more accurate non-Euclidean structured representations instead of icosahedron, and the difference from (Jiang et al. 2019) is that ours is rotation equivariant.

Our contributions are as follows:

- We use PDOs to design an orientable spherical CNN that is exactly equivariant over the $SO(3)$ group in the continuous domain.
- The equivariance of the PDO-eS^{2}CNN becomes approximate after the discretization, and it is the first time that the theoretical equivariance error analysis is provided when

Table 1: Summary of notations in this paper.

$\mathcal{S}^2$	The sphere	$SO(2), SO(3)$	Rotation groups
α, β, γ	The ZYZ-Euler angles	$Z(\alpha), Y(\beta)$	The rotations around z and y axes
R	$R \in SO(3)$ and $R = Z(\alpha_R)Y(\beta_R)Z(\gamma_R)$	$n = (0,0,1)^T$	The north pole
P	$P \in \mathcal{S}^2$, and $P(\alpha,\beta) = Z(\alpha)Y(\beta)n$	$\bar{P}$	The coset representative associated with P
$\bar{P} \cdot SO(2)$	$\{\bar{P}Z(\gamma) \mid \gamma \in [0, 2\pi)\}$, the left coset of $SO(2)$	$E \simeq F$	E is homeomorphic to F
A_R	The 2D rotation matrix simplifying $Z(\gamma_R)$	$C^\infty(\mathcal{S}^2)$	The space of smooth function on $\mathcal{S}^2$
$C^\infty(SO(3))$	The space of smooth function on $SO(3)$	s, so	$s \in C^\infty(\mathcal{S}^2)$ and $so \in C^\infty(SO(3))$
$\pi^S_{\widetilde{R}}[s], \pi^{SO}_{\widetilde{R}}[so]$	The group actions of $\widetilde{R}$ on s and so	U_P	An open set of $\mathcal{S}^2$ containing P
$\widetilde{U}_p$	$\widetilde{U}_p = \varphi_P(U_P) \subset \mathbb{R}^2$	φ_P	The homeomorphism from U_P to $\widetilde{U}_P$
$\bar{s}$	The smooth function on $\mathbb{R}^3$ extended by s	$H(\cdot, \cdot; \boldsymbol{w})$	The polynomail parameterized by $\boldsymbol{w}$
$\partial/\partial x_i^{(A)}$	The PDOs rotated by $A \in SO(2)$	$\nabla_x[f], \nabla_x^2[f]$	The gradient and the Hessian matrix of f
$\nabla_x^{(A)}, (\nabla_x^{(A)})^2$	The operators defined in (6) and (7)	$\chi^{(A)}$	The differential operators defined in (4)
Ψ, Φ	The mappings defined in (9) and (10)	$\boldsymbol{I}, \boldsymbol{F}$	Discrete inputs and intermediate feature maps
f_P	$f_P = \bar{s} \cdot \varphi_P^{-1}$	$O(\cdot)$	The infinitesimal of the same order
$D_P, \hat{D}_P$	Partial derivatives matrix and its estimation	$\hat{\nabla}_x[f], \hat{\nabla}_x^2[f]$	The estimations of $\nabla_x[f]$ and $\nabla_x^2[f]$
$\widetilde{\chi}^{(A)}, \widetilde{\Psi}, \widetilde{\Phi}$	The discretizations of $\chi^{(A)}$, Ψ and Φ	C_N	The N-ary cyclic group

the equivariance is approximate in the spherical domain.

- PDO-eS^{2}CNNs show greater parameter efficiency and achieve new state-of-the-art (SOTA) results on spherical MNIST classification, 2D-3D-S image segmentation and QM7 atomization energy prediction tasks.

The paper is organized as follows. In Section 2, we review some works related to spherical CNNs. In Section 3, we introduce some prior knowledge to make our work easy to understand. In Section 4, we use orientable parameterized PDOs to design PDO-eS^{2}CNN, which is exactly equivariant over $SO(3)$ in the continuous domain. In Section 5, we use Taylor's expansion to estimate PDOs accurately, implement PDO-eS^{2}CNN in the discrete domain, and provide the equivariance error analysis. In Section 6, we evaluate our method on multiple tasks.

2 Related Work

The most straightforward method to process spherical signals is mapping them into the planar domain via the equirectangular projection (Su and Grauman 2017), and then using 2D CNNs. However, this projection will result in severe distortion. Coors, Condurache, and Geiger (2018) and Zhao et al. (2018) implement CNNs on the tangent plane of the spherical image to reduce distortions. Even though, such methods are not equivariant in the spherical domain.

Actually, many works (Cohen and Welling 2016; Cesa and Weiler 2019; Shen et al. 2020; Sosnovik, Szmaja, and Smeulders 2019; Weiler et al. 2018; Ravanbakhsh, Schneider, and Póczos 2017) focus on incorporating equivariance into network architectures. As for spherical data, some works (Bruna et al. 2014; Frossard and Khasanova 2017; Perraudin et al. 2019; Defferrard et al. 2020) represent the sampled sphere as a graph connecting pixels according to distance between them and utilize graph-based methods to process it. Particularly, Perraudin et al. (2019) propose DeepSphere using isotropic filters, and achieve rotation equivariance. Defferrard et al. (2020) improve DeepSphere and achieve a controllable tradeoff between cost and equivariance. However, the isotropic filters they use significantly restrict the capacity of models.

Also, there exist some works (Cohen et al. 2018; Esteves et al. 2018; Kondor, Lin, and Trivedi 2018) using anisotropic filters to achieve rotation equivariance. Specifically, Cohen et al. (2018) extend the group equivariance theory into the spherical domain and use a generalized Fourier transform for implementation. However, these methods only work on nonuniform grids which over-sample near the poles, and as a result, these methods suffer from significant distortion. Cohen et al. (2019) further extend group equivariance to gauge equivariance, which is automatically $SO(3)$ equivariant in the spherical domain. However, their theory cannot show how the feature maps transform w.r.t. rotation transformations explicitly whereas ours can, which makes our theory more transparent and explainable. Cohen et al. (2019) implement gauge equivariant CNNs on the surface of the icosahedron. The icosahedron is not an accurate discretization of the sphere, so their equivariance is weak. By contrast, our method can be applied on accurate discretizations of the sphere, achieving much better equivariance consequently.

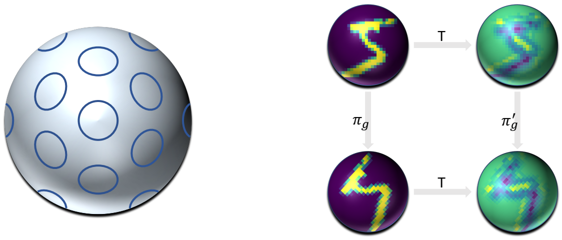
Particularly, empirical results (Jiang et al. 2019; Zhang et al. 2019) show that orientation-aware CNNs can be beneficial for some tasks with orientation information. Zhang et al. (2019) use north-aligned filters to achieve orientation-awareness, while Jiang et al. (2019) use orientable PDOs. In addition, Jiang et al. (2019) can process spherical signals on non-Euclidean structured grids easily using PDOs. However, their models are not equivariant. Our PDO-eS^{2}CNN furthermore incorporates equivariance into the model, and introduces a new weight sharing scheme across filters, which brings greater parameter efficiency.

3 Prior Knowledge

Parameterization of $\mathcal{S}^2$ and $SO(3)$

We use $\mathcal{S}^2$ and $SO(3)$ to denote a sphere and a group of 3D rotations, respectively. Formally,

$$\mathcal{S}^2 = \{(x_1, x_2, x_3) \mid \|x\|_2 = 1\},$$
$$SO(3) = \{R \in \mathbb{R}^3 \mid R^T R = I, \det(R) = 1\}.$$



(a) $S^2 \simeq SO(3)/SO(2)$ (b) Group equivariance on SO(3)

Figure 1: (a) $\mathcal{S}^2 \simeq SO(3)/SO(2)$. $SO(3)$ can be viewed as a bundle of circles over the sphere; (b) Group equivariance on $SO(3)$. Transforming an input by a transformation $g \in SO(3)$ and then passing it through the mapping T is equivalent to first mapping it through T and then transforming the representation.

We use the ZYZ Euler parameterization for $SO(3)$. An element $R \in SO(3)$ can be written as

$$R = Z(\alpha_R)Y(\beta_R)Z(\gamma_R),$$

where ZYZ-Euler angles $\alpha_R \in [0, 2\pi), \beta_R \in [0, \pi]$ and $\gamma_R \in [0, 2\pi)$, and $Z(\alpha)$ and $Y(\beta)$ are rotations around z and y axes, respectively. To be specific,

$$Z(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, Y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}.$$

Accordingly, we have a related parameterization for the sphere. An element $P \in \mathcal{S}^2$ can be written as $P(\alpha, \beta) = Z(\alpha)Y(\beta)n$, where n is the north pole, i.e., $n = (0, 0, 1)^T$. Conversely, we can also calculate α and β if $P = (x_1, x_2, x_3)^T$ is given. To be specific, if $P = (0, 0, 1)^T$, we take $\alpha = \beta = 0$; if $P = (0, 0, -1)^T$, we take $\alpha = 0$ and $\beta = \pi$; otherwise, we have

$$\alpha = \begin{cases} \arccos\left(\frac{x_1}{\sqrt{x_1^2 + x_2^2}}\right) & x_2 \geq 0 \\ 2\pi - \arccos\left(\frac{x_1}{\sqrt{x_1^2 + x_2^2}}\right) & x_2 < 0 \end{cases}, \quad \beta = \arccos(x_3).$$

This parameterization makes explicit the fact that the sphere is a quotient $\mathcal{S}^2 \simeq SO(3)/SO(2)$[1], where $SO(2)$ is the subgroup of $SO(3)$ and contains the rotations around the z axis. Elements of the subgroup $SO(2)$ leave the north pole invariant, and have the form $Z(\gamma)$. The point $P(\alpha, \beta) \in \mathcal{S}^2$ is associated with the coset representative $\bar{P} = Z(\alpha)Y(\beta) \in SO(3)$. This element represents the left coset $\bar{P} \cdot SO(2) = \{\bar{P}Z(\gamma) | \gamma \in [0, 2\pi)\}$. Intuitively, $SO(3)$ can be viewed as a bundle of circles ($SO(2)$) over the sphere, as we show in Figure 1(a). In this way, $\forall R \in SO(3)$, $R \in \bar{P}_R SO(2)$, where $\bar{P}_R = Z(\alpha_R)Y(\beta_R)$. As a result, we can parameterize R as (P_R, A_R), where $P_R = \bar{P}_R n \in \mathcal{S}^2$ and $A_R \in SO(2)$. Specifically, A_R is a 2D rotation matrix, which is a simplification of $Z(\gamma_R)$, i.e.,

$$A_R = \begin{bmatrix} \cos\gamma_R & -\sin\gamma_R \\ \sin\gamma_R & \cos\gamma_R \end{bmatrix}.$$

[1]Given a group $\mathcal{G}$ and its subgroup $\mathcal{H}$, the left cosets $g\mathcal{H}$ of $\mathcal{H}$ partition $\mathcal{G}$, where $g \in \mathcal{G}$. We denote the set of left cosets as $\mathcal{G}/\mathcal{H}$. $E \simeq F$ denotes that E is homeomorphic to F.

Group Actions on Spherical Functions

Inputs and feature maps can be naturally modeled as functions in the continuous domain. Specifically, we model the input s as a smooth function on $\mathcal{S}^2$ and the intermediate feature map so as a smooth function on $SO(3)$. Particularly, the smoothness of so means that if we use the parameterization of $SO(3)$ mentioned above, the feature map $so(P, A)$ is smooth w.r.t. P when A is fixed. So so can also be viewed as a smooth spherical function with infinite channels indexed by $A \in SO(2)$. We use $C^\infty(\mathcal{S}^2)$ and $C^\infty(SO(3))$ to denote the function spaces of s and so, respectively .

In this way, rotation transformations acting on inputs and feature maps can be mathematically formulated as follows.

Actions on Inputs Suppose that $s \in C^\infty(\mathcal{S}^2)$ and $\widetilde{R} \in SO(3)$, then $\widetilde{R}$ acts on s in the following way:

$$\forall P \in \mathcal{S}^2, \quad \pi_{\widetilde{R}}^S[s](P) = s\left(\widetilde{R}^{-1}P\right).$$

Actions on Feature Maps Suppose that $so \in C^\infty(SO(3))$ and $\widetilde{R} \in SO(3)$, then $\widetilde{R}$ acts on so in the following way:

$$\forall R \in SO(3), \quad \pi_{\widetilde{R}}^{SO}[so](R) = so\left(\widetilde{R}^{-1}R\right). \quad (1)$$

If we use the parameterization of $SO(3)$, (1) is of the following more intuitive form:

$$\begin{aligned}\pi_{\widetilde{R}}^{SO}[so](P_R, A_R) &= so\left(P_{\widetilde{R}^{-1}R}, A_{\widetilde{R}^{-1}R}\right) \\ &= so\left(\widetilde{R}^{-1}P_R, A_{\widetilde{R}^{-1}R}\right),\end{aligned}$$

where (P_R, A_R) is the representation of R and $P_{\widetilde{R}^{-1}R} = \widetilde{R}^{-1}Rn = \widetilde{R}^{-1}P_R$.

Group Equivariance

Equivariance measures how the outputs of a mapping transform in a predictable way with the transformation of the inputs. To be specific, let T be a mapping, which could be represented by a deep neural network from the input feature space to the output feature space, and $\mathcal{G}$ is a transformation group. T is called group equivariant if it satisfies

$$\forall g \in \mathcal{G}, \quad T[\pi_g[f]] = \pi'_g[T[f]],$$

where f can be any input feature map in the input feature space, and π_g and π'_g denote how the transformation g acts on input features and output features, respectively.

In our theory, we take the group $\mathcal{G}$ as $SO(3)$, and then focus on utilizing PDOs to design a neural network equivariant to $SO(3)$, as shown in Figure 1(b).

4 PDO-eS^{2}CNNs

Chart-based PDOs

We define an atlas to help define PDOs acting on the spherical functions uniformly. To be specific, an atlas for $\mathcal{S}^2$ is a collection of charts whose domains cover $\mathcal{S}^2$. We denote the atlas as $\{(U_P, \varphi_P) | P \in \mathcal{S}^2\}$, where U_P is an open subset of $\mathcal{S}^2$ containing P and $\varphi_P : U_P \to \widetilde{U}_P$ is a homeomorphism

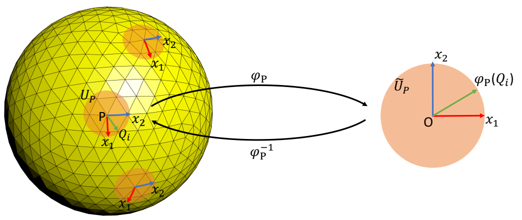


Figure 2: For any $P \in \mathcal{S}^2$, a homeomorphism φ_P maps the chart $U_P \subset \mathcal{S}^2$ to an open subset $\widetilde{U}_P \subset \mathbb{R}^2$. The sphere is presented by a level-3 icosahedral mesh.

from the chart U_P to an open subset $\widetilde{U}_P = \varphi_P(U_P) \subset \mathbb{R}^2$ and $\varphi_P(P) = 0$. The form of φ_P is given by

$$\varphi_P^{-1}(x_1, x_2) = \bar{P}\left(x_1, x_2, \sqrt{1-|x|^2}\right)^T. \tag{2}$$

In this way, as shown in Figure 2, for any point $P \in \mathcal{S}^2$ (except poles), x_1 resp. x_2 point to the north-south and east-west directions in the chart U_P, and the homeomorphism φ_P's are uniformly defined over the sphere, which relate to orientable and uniform PDOs over the sphere.

In order to use PDOs, we suppose that the spherical function s is smooth and denote it as $s \in C^\infty(\mathcal{S}^2)$. s can always be extended to a smooth function $\bar{s}$ defined on $\mathbb{R}^3$, and we denote it as $\bar{s} \in C^\infty(\mathbb{R}^3)$. We emphasize that we need not obtain $\bar{s}$ explicitly from the given s, whereas we only use this notation for ease of derivation. Then the PDOs $\partial/\partial x_i$ and $\partial^2/\partial x_i \partial x_j (i,j=1,2)$[2] act on the spherical function s in the way that these PDOs act on the composite function $\bar{s} \cdot \varphi_P^{-1} \in C^\infty(\mathbb{R}^2)$[3]. Formally, $\forall P \in \mathcal{S}^2$,

$$\frac{\partial}{\partial x_i}[s](P) = \frac{\partial}{\partial x_i}\left[\bar{s} \cdot \varphi_P^{-1}\right](0),$$
$$\frac{\partial^2}{\partial x_i \partial x_j}[s](P) = \frac{\partial^2}{\partial x_i \partial x_j}\left[\bar{s} \cdot \varphi_P^{-1}\right](0).$$

By contrast, Jiang et al. (2019) define PDOs based on the spherical coordinates, which have high resolution near the pole and low resolution near the equator. So the scales of their PDOs are dependent on the latitudes. By contrast, the scales of our chart-based PDOs are independent of locations, resulting in much more uniform feature extration. Our definition of PDOs is also different from that in conventional manifold calculus in that we can deal with second-order PDOs without defining a smooth vector field. Actually, it is impossible to define a non-trivial smooth vector field over the sphere due to the hairy ball theorem (Milnor 1978).

Rotated Parameterized Differential Operators

Following (Jiang et al. 2019; Ruthotto and Haber 2018; Shen et al. 2020), we parameterize convolution kernels using a linear combination of PDOs. Specifically, we refer to H as a parameterized second-order polynomial of 2 variables, i.e.,

$$H(u, v; \boldsymbol{w}) = w_1 + w_2 u + w_3 v + w_4 u^2 + w_5 uv + w_6 v^2, \tag{3}$$

[2]We only consider the PDOs up to the second order in this work.
[3]We use $[\cdot]$ to denote that an operator acts on a function.

where $\boldsymbol{w}$ are learnable parameters. If we take $u = \partial/\partial x_1$ and $v = \partial/\partial x_2$, then $H(\partial/\partial x_1, \partial/\partial x_2; \boldsymbol{w})$ becomes a linear combination of PDOs. For example, if $H(u, v; \boldsymbol{w}) = u^2 + uv$, then $H(\partial/\partial x_1, \partial/\partial x_2; \boldsymbol{w}) = \partial^2/\partial x_1^2 + \partial^2/\partial x_1 \partial x_2$.

We rotate these PDOs with a 2×2 rotation matrix $A \in SO(2)$, and obtain the following rotated parameterized differential operators:

$$\chi^{(A)} = H\left(\frac{\partial}{\partial x_1^{(A)}}, \frac{\partial}{\partial x_2^{(A)}}; \boldsymbol{w}\right), \tag{4}$$

where

$$\left(\frac{\partial}{\partial x_1^{(A)}}, \frac{\partial}{\partial x_2^{(A)}}\right)^T = A^{-1}\left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}\right)^T. \tag{5}$$

As a compact form, we can also rewrite (5) as

$$\nabla_x^{(A)} = A^{-1} \nabla_x, \tag{6}$$

where $\nabla_x = (\partial/\partial x_1, \partial/\partial x_2)^T$ is the gradient operator. In fact, (5) is equivalent to first rotating the coordinate system by Z, and then calculating gradients. In addition, it is easy to get that

$$\begin{aligned}\left(\nabla_x^{(A)}\right)^2 &:= \begin{bmatrix} \frac{\partial}{\partial x_1^{(A)}}\frac{\partial}{\partial x_1^{(A)}} & \frac{\partial}{\partial x_1^{(A)}}\frac{\partial}{\partial x_2^{(A)}} \\ \frac{\partial}{\partial x_1^{(A)}}\frac{\partial}{\partial x_2^{(A)}} & \frac{\partial}{\partial x_2^{(A)}}\frac{\partial}{\partial x_2^{(A)}} \end{bmatrix} \\ &= A^{-1}\begin{bmatrix} \frac{\partial^2}{\partial x_1^2} & \frac{\partial^2}{\partial x_1 \partial x_2} \\ \frac{\partial^2}{\partial x_1 \partial x_2} & \frac{\partial^2}{\partial x_2^2} \end{bmatrix} A = A^{-1} \nabla_x^2 A.\end{aligned} \tag{7}$$

To make it more explicit, we emphasize that by the definition in (4), $\chi^{(A)}$'s are identical polynomials w.r.t. $\partial/\partial x_1^{(A)}$'s and $\partial/\partial x_2^{(A)}$'s, but different polynomials w.r.t. $\partial/\partial x_1$ and $\partial/\partial x_2$. To be specific,

$$\begin{aligned}\chi^{(A)} =& w_1 + (w_2, w_3)\nabla_x^{(A)} + \left\langle \begin{bmatrix} w_4 & \frac{w_5}{2} \\ \frac{w_5}{2} & w_6 \end{bmatrix}, \left(\nabla_x^{(A)}\right)^2 \right\rangle \\ =& w_1 + (w_2, w_3)A^{-1}\nabla_x + \left\langle \begin{bmatrix} w_4 & \frac{w_5}{2} \\ \frac{w_5}{2} & w_6 \end{bmatrix}, A^{-1}\nabla_x^2 A \right\rangle \\ =& w_1 + (w_2, w_3)A^{-1}\nabla_x + \left\langle A \begin{bmatrix} w_4 & \frac{w_5}{2} \\ \frac{w_5}{2} & w_6 \end{bmatrix} A^{-1}, \nabla_x^2 \right\rangle,\end{aligned} \tag{8}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. Particularly, these differential operators $\chi^{(A)}$'s share parameters $\boldsymbol{w}$, indicating great parameter efficiency.

From another point of view, the rotation of differential operators can also be viewed as changing the coefficients of PDOs (see (8)), without changing the orientations of PDOs. Consequently, the rotated parameterized differential operators, $\chi^{(A)}$'s, and the subsequent PDO-eS^{2}CNN are still orientable. By contrast, some rotation equivariant spherical CNNs, such as Icosahedral CNNs (Cohen et al. 2019), assume no preferred orientation, so they are not orientable.

Equivariant Differential Operators

We define two mappings, Ψ and Φ, using the above-mentioned differential operators, $\chi^{(A)}$'s. To be specific, we use Ψ to deal with inputs, which maps an input s to a feature map defined on $SO(3)$: $\forall R \in SO(3)$,

$$\Psi[s](R) = \Psi[s](P_R, A_R) = \chi^{(A_R)}[s](P_R). \quad (9)$$

Then, we use Φ to deal with the resulting feature maps, which maps one feature map defined on $SO(3)$ to another feature map defined on $SO(3)$: $\forall R \in SO(3)$,

$$\begin{aligned}\Phi[so](R) &= \Phi[so](P_R, A_R) \\ &= \int_{SO(2)} \chi_A^{(A_R)}\ [so](P_R, A_R A) d\nu(A), \quad (10)\end{aligned}$$

where ν is a measure on $SO(2)$. As for $\chi_A^{(A_R)}$, we use the subscript A to distinguish the differential operators parameterized by different $\boldsymbol{w}_A$'s. The so on the right hand side should be viewed as a spherical function indexed by $A_R A$ when the operator $\chi_A^{(A_R)}$ acts on it.

Finally, we prove that the above two mappings, Ψ and Φ, are equivariant under arbitrary rotation transformation $\widetilde{R} \in SO(3)$ and show how the outputs transform w.r.t. the transformation of inputs. The proofs of theorems can be found in the Supplementary Material.

Theorem 1 *If $s \in C^\infty(\mathcal{S}^2)$ and $so \in C^\infty(SO(3))$, $\forall \widetilde{R} \in SO(3)$, we have*

$$\Psi\left[\pi_{\widetilde{R}}^S[s]\right] = \pi_{\widetilde{R}}^{SO}\left[\Psi[s]\right], \quad (11)$$

$$\Phi\left[\pi_{\widetilde{R}}^{SO}[so]\right] = \pi_{\widetilde{R}}^{SO}\left[\Phi[so]\right]. \quad (12)$$

Equivariant Network Architectures

It is easy to use the above-mentioned two equivariant mappings, Ψ and Φ, to design an equivariant network. To be specific, according to the working spaces, we set a Ψ as the first layer, followed by multiple Φ's, inserted by pointwise nonlinearities $\sigma(\cdot)$, e.g., ReLUs, which do not disturb the equivariance. Finally, we can get an equivariant network architecture $T[s] = \Phi^{(L)}\left[\cdots\sigma\left(\Phi^{(1)}\left[\sigma(\Psi[s])\right]\right)\right]$.

Theorem 2 *If $s \in C^\infty(\mathcal{S}^2)$, $\forall \widetilde{R} \in SO(3)$, we have*

$$T\left[\pi_{\widetilde{R}}^S[s]\right] = \pi_{\widetilde{R}}^{SO}\left[T[s]\right].$$

That is, transforming an input s by a transformation $\widetilde{R}$ (forming $\pi_{\widetilde{R}}^S$) and then passing it through the network T gives the same result as first mapping s through T and then transforming the representation.

As discussed above, we only consider the case where inputs, s, and intermediate feature maps over $SO(3)$, so, only consist of single channel. In fact, our theory can be easily extended to a more general case where inputs and feature maps consist of multiple channels, and we only need to use multiple Ψ's and Φ's to process inputs and generate outputs.

Besides, in conventional CNNs, we always use 1×1 convolutions to change the numbers of channels without introducing too many parameters. In PDO-eS^{2}CNN, this can be easily achieved by taking $\boldsymbol{w}$ as a one-hot vector. The details are given in the Supplementary Material. We can also incorporate equivariance into other architectures, e.g., ResNets, because shortcut connections do not disturb equivariance.

5 Implementation

Icosahedral Spherical Mesh

In practice, spherical data are always given on discrete domain, instead of continuous domain. The icosahedral spherical mesh (Baumgardner and Frederickson 1985) is among the most uniform and accurate discretization of the sphere. Specifically, a spherical mesh can be obtained by progressively subdividing each face of the unit icosahedron into four triangles and reprojecting each node to unit distance from the origin. We start with the unit icosahedron as the level-0 mesh, and each progressive mesh resolution is one level above the previous. The level-3 icosahedral mesh is shown in Figure 2. The subdivision scheme for triangles also provides a natural coarsening and refinement scheme for the grid, which allows for easy implementations of downsampling and upsampling routines associated with CNN architectures. We emphasize that our method is not limited to the icosahedral spherical mesh, but can also use other discrete representations of the sphere easily, like the HealPIX (Gorski et al. 2005). In this work, we use the icosahedral spherical mesh for ease of implementation.

Estimation of Partial Derivatives

We view the input spherical data $\boldsymbol{I}$ as a discrete function sampled from a smooth spherical function s on the icosahedral spherical mesh vertices $\Omega \subset \mathcal{S}^2$, where $\boldsymbol{I}(P) = s(P), \forall P \in \Omega$, and use a numerical method to estimate partial derivatives at $P \in \Omega$ in the discrete domain. Firstly, we use φ_P to map P and $Q_i (i = 1, 2, \cdots, m)$ into an open set $\widetilde{U}_P \subset \mathbb{R}^2$, where $Q_i \in \Omega$ are the neighbor nodes of P (see Figure 2)[4]. As a result, we get $\varphi_P(P) = 0$, and $\varphi_P(Q_i) = (x_{i1}, x_{i2})$, where $\forall i = 1, 2, \cdots, m$,

$$\left(x_{i1}, x_{i2}, \sqrt{1 - x_{i1}^2 - x_{i2}^2}\right)^T = \bar{P}^{-1} Q_i.$$

We denote $f_P = \bar{s} \cdot \varphi_P^{-1}$, so $f_P(0) = s(P) = \boldsymbol{I}(P)$ and $f_P(x_{i1}, x_{i2}) = s(Q_i) = \boldsymbol{I}(Q_i)$. We use Taylor's expansion to expand f_P at the original point, then we have that $\forall i = 1, 2, \cdots, m$,

$$\begin{aligned}f_P(x_{i1}, x_{i2}) =& f_P(0,0) + x_{i1}\frac{\partial f_P}{\partial x_1} + x_{i2}\frac{\partial f_P}{\partial x_2} + \frac{1}{2}x_{i1}^2\frac{\partial^2 f_P}{\partial x_1^2} \\ &+ x_{i1}x_{i2}\frac{\partial^2 f_P}{\partial x_1 \partial x_2} + \frac{1}{2}x_{i2}^2\frac{\partial^2 f_P}{\partial x_2^2} + O(\rho_i^3) \quad (13)\end{aligned}$$

[4]We only consider the neighbor nodes of P, in analogy with the commonly-used 3×3 convolutions in planar CNNs.

where all above partial derivatives are evaluated at $(0,0)$, and $\rho_i = \sqrt{x_{i1}^2 + x_{i2}^2}$. Thus we have

$$\begin{bmatrix} \vdots \\ f_P(x_{i1}, x_{i2}) - f_P(0) \\ \vdots \end{bmatrix} \approx \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{i1} & x_{i2} & \frac{x_{i1}^2}{2} & x_{i1}x_{i2} & \frac{x_{i2}^2}{2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} D_P,$$

where D_P is a partial derivatives matrix:

$$D_P = \left(\frac{\partial f_P}{\partial x_1}, \frac{\partial f_P}{\partial x_2}, \frac{\partial^2 f_P}{\partial x_1^2}, \frac{\partial^2 f_P}{\partial x_1 x_2}, \frac{\partial^2 f_p}{\partial x_2^2} \right)^T \bigg|_{x_1 = x_2 = 0}.$$

We denote the above approximate equations as $F_P \approx V_P D_P$, and use the least square method to estimate D_P:

$$\hat{D}_P = \arg\min_D \|V_P D - F_P\|_2 = (V_P^T V_P)^{-1} V_P^T F_P.$$

Actually, we can easily estimate any partial derivatives using the similar method so long as we employ the appropriate Taylor's expansions. By contrast, Jiang et al. (2019) can only deal with limited PDOs, including $\partial/\partial x_1, \partial/\partial x_2$, and the Laplacian operator.

Discretization of $SO(2)$

As it is impossible to go through all the $A \in SO(2)$ in (9) and (10), we need to discretize $SO(2)$. To be specific, we discretize the continuous group $SO(2)$ as the N-ary cyclic group C_N, where $C_N = \{e = A_0, A_1, \cdots, A_{N-1}\}$, and

$$A_i = \begin{bmatrix} \cos\frac{2\pi i}{N} & -\sin\frac{2\pi i}{N} \\ \sin\frac{2\pi i}{N} & \cos\frac{2\pi i}{N} \end{bmatrix}.$$

Correspondingly, (9) should be discretized as: $\forall P \in \Omega$ and $i = 0, 1, \cdots, N-1$,

$$\begin{aligned} &\widetilde{\Psi}[\boldsymbol{I}](P,i) = \widetilde{\chi}^{(A_i)}[\boldsymbol{I}](P) \\ =&\left(w_1 + (w_2, w_3) A_i^{-1} \hat{\nabla}_x + \left\langle A_i \begin{bmatrix} w_4 & \frac{w_5}{2} \\ \frac{w_5}{2} & w_6 \end{bmatrix} A_i^{-1}, \hat{\nabla}_x^2 \right\rangle \right) [f_P](0) \\ =& w_1 f_P(0) + (w_2, w_3) A_i^{-1} \hat{\nabla}_x [f_P](0) \\ &+ \left\langle A_i \begin{bmatrix} w_4 & \frac{w_5}{2} \\ \frac{w_5}{2} & w_6 \end{bmatrix} A_i^{-1}, \hat{\nabla}_x^2 [f_P](0) \right\rangle, \end{aligned}$$

where the partial derivatives are estimated using $\boldsymbol{I}$. In this way, when viewed as a spherical function, the output $\widetilde{\Psi}[\boldsymbol{I}]$ consists of N channels, instead of infinite channels indexed by $A \in SO(2)$. Similarly, (10) is discretized as: $\forall P \in \Omega$ and $i = 0, 1, \cdots, N-1$,

$$\widetilde{\Phi}[\boldsymbol{F}](P,i) = \frac{\nu(SO(2))}{N} \sum_{j=0}^{N-1} \widetilde{\chi}_{Z_j}^{(Z_i)} \; [\boldsymbol{F}](P, i \oplus j),$$

where the intermediate feature map $\boldsymbol{F}$ is an N-channel discrete function sampled from the smooth function $so \in C^\infty(SO(3))$, i.e., $\boldsymbol{F}(P,i) = so(P, A_i)$, and $\oplus$ denotes the module-N addition. As a result, $\widetilde{\Psi}$ and $\widetilde{\Phi}$ become discretized PDO-eS^{2}Convs. Particularly, batch normalization (Ioffe and Szegedy 2015) should be implemented with a single scale and a single bias per PDO-eS^{2}Conv feature map in order to preserve equivariance.

Equivariance Error Analysis

As shown in Theorem 1, the equivariance of PDO-eS^{2}Convs Ψ and Φ is exact in the continuous domain, and it becomes approximate because of discretization in implementation. In (13), it is easy to verify that $O(\rho_1) = O(\rho_2) = \cdots = O(\rho_m)$ from the definition of icosahedral spherical mesh, and we write $O(\rho_i) = O(\rho)$ for simplicity. Then, we have the following equivariance error analysis.

Theorem 3 $\forall \widetilde{R} \in SO(3)$,

$$\widetilde{\Psi}\left[\pi_{\widetilde{R}}^S [\boldsymbol{I}]\right] = \pi_{\widetilde{R}}^{SO}\left[\widetilde{\Psi}[\boldsymbol{I}]\right] + O(\rho), \tag{14}$$

$$\widetilde{\Phi}\left[\pi_{\widetilde{R}}^{SO} [\boldsymbol{F}]\right] = \pi_{\widetilde{R}}^{SO}\left[\widetilde{\Phi}[\boldsymbol{F}]\right] + O(\rho) + O\left(\frac{1}{N^2}\right), \tag{15}$$

where transformations acting on discrete inputs and feature maps are defined as $\pi_{\widetilde{R}}^S[\boldsymbol{I}](P) = \pi_{\widetilde{R}}^S[s](P)$ *and* $\pi_{\widetilde{R}}^{SO}[\boldsymbol{F}](P,i) = \pi_{\widetilde{R}}^{SO}[so](P, A_i)$*, respectively.*

Particularly, we note that (Shen et al. 2020) use PDOs to design an equivariant CNN over the Euclidean group, and achieve a quadratic order equivariance approximation for 2D images in the discrete domain. However, they can only deal with the data in the Euclidean space. Virtually, we extend their theory to the non-Euclidean geometry, i.e., the sphere. By contrast, we can only achieve a first order equivariance approximation w.r.t. the grid size ρ, as the representation of the sphere we use is non-Euclidean structured.

6 Experiments

We evaluate our PDO-eS^{2}CNNs on three datasets. The data preprocessing, model architectures and training details for each task are provided in the Supplementary Material.

Spherical MNIST Classification

We follow (Cohen et al. 2018) in the preparation of the spherical MNIST, and prepare non-rotated training and testing (N/N), non-rotated training and rotated testing (N/R) and rotated training and testing (R/R) tasks. The training set and the test set include 60,000 and 10,000 images, respectively. We randomly select 6,000 training images as a validation set, and choose the model with the lowest validation error during training. Inputs are on a level-4 icosahedral spherical mesh. For fair comparison with existing methods, we evaluate our method using a small and a large model, respectively.

As shown in Table 2, when using the small model (73k), our method achieves 99.44% test accuracy on the N/N task. The result decreases to 90.14% on the N/R task, mainly because of the equivariance error after discretization. HexRUNet-C achieves comparable results using slightly more parameters, but it performs significantly worse on N/R and R/R tasks for lack of rotation equivariance. S2CNN performs better on the N/R task because it is nearly exactly equivariant. However, it cannot perform well on two more important tasks, N/N and R/R, because of the distortion from nonuniform sampling. We argue that these two tasks are more important because the training and the test sets of most tasks are of identical distributions.

Table 2: Results on the spherical MNIST dataset with non-rotated (N) and rotated (R) training and test data. The second column marks whether these models are rotation-equivariant in the spherical domain.

Model	Rotation equivariance	N/N	N/R	R/R	#Parms
S2CNN (Cohen et al. 2018)	✓	96	94	95	58k
UGSCNN (Jiang et al. 2019)	✗	99.23	35.60	94.92	62k
HexRUNet-C (Zhang et al. 2019)	✗	99.45	29.84	97.05	75k
PDO-eS^{2}CNN (ours)	✓	99.44 ± 0.06	90.14 ± 0.58	98.93 ± 0.08	73k
SphereNet (Coors, Condurache, and Geiger 2018)	✗	94.4	-	-	196k
FFS2CNN (Kondor, Lin, and Trivedi 2018)	✓	96.4	**96**	96.6	286k
Icosahedral CNN (Cohen et al. 2019)	✓	99.43	69.99	99.31	182k
PDO-eS^{2}CNN (ours)	✓	$\mathbf{99.60 \pm 0.04}$	94.25 ± 0.29	$\mathbf{99.45 \pm 0.05}$	180k

When using the large model (180k), our method results in new SOTA results on the N/N and R/R tasks (99.60% and 99.45%), respectively, which improve the previous SOTA results (99.45% and 99.31%) significantly. Note that the previous SOTA results have been very competitive even for planar MNIST, and the error rates are further reduced by more than 20% using our method. Also, we obtain a more competitive result (94.25%) on the N/R task. By contrast, Icosahedral CNN only achieves 69.99% test accuracy because it is only equivariant over the alternating group $A5$, which merely contains 60 rotational symmetries. FFS2CNN performs the best on this task because it is also nearly exactly equivariant and use much more parameters, but it performs significantly worse on other tasks (N/N and R/R) because of the distortion in representation from nonuniform sampling.

Omnidirectional Image Segmentation

Omnidirectional semantic segmentation is an orientation-aware task since the natural scene images are always upright due to gravity. We evaluate our method on the Stanford 2D-3D-S dataset (Armeni et al. 2017), which contains 1,413 equirectangular images with RGB+depth channels, and semantic labels across 13 different classes. We use the official 3-fold cross validation to train and evaluate our model, and report the mean intersection over union (mIoU) and pixel accuracy (mAcc).

Table 3: mAcc and mIoU comparison on 2D-3D-S.

Model	mAcc	mIoU	#Params
UNet	50.8	35.9	-
Icosahedral CNN (Cohen et al. 2019)	55.9	39.4	-
UGSCNN (Jiang et al. 2019)	54.7	38.3	5.18M
HexRUNet (Zhang et al. 2019)	58.6	43.3	1.59M
PDO-eS^{2}CNN (ours)	$\mathbf{60.4 \pm 1.0}$	$\mathbf{44.6 \pm 0.4}$	0.86M

We achieve new SOTA result on this task, and report our main result in Table 3. As pointed out in (Zhang et al. 2019), the 2D-3D-S dataset is acquired with preferred orientation, thus an orientation-aware system can be beneficial. Our model performs significantly better than icosahedral CNN, mainly because that our model is orientation-aware, while the latter assumes no preferred orientation. Compared with HexRUNet, an orientation-aware model, our method still performs better and achieve a new SOTA result, because we can process spherical data inherently, whereas HexRUNet can only process icosahedron data, which makes big difference. In addition, we use far fewer parameters (0.86M vs. 1.59M), showing great parameter efficiency from weight sharing across rotated filters. The detailed statistics for this task is shown in the Supplementary Material.

Atomization Energy Prediction

Finally, we apply our method to the QM7 dataset (Blum and Reymond 2009; Rupp et al. 2012), where the goal is to regress over atomization energies of molecules given atomic positions p_i, and charges z_i. This dataset contains 7,165 molecules, and each molecule contains up to 23 atoms of 5 types (H, C, N, O, S). We use the official 5-fold cross validation to train and evaluate our model, and report the root mean square error (RMSE).

Table 4: Experimental results on the QM7 task.

Model	RMSE	#Params
MLP/Random CM (Montavon et al. 2012)	5.96 ± 0.48	-
S2CNN (Cohen et al. 2018)	8.47	1.4M
FFS2CNN (Kondor, Lin, and Trivedi 2018)	7.97	1.1M
PDO-eS^{2}CNN (ours)	$\mathbf{3.78 \pm 0.07}$	0.4M

As shown in Table 4, compared with other spherical CNNs, including S2CNN and FFS2CNN, our model halves the RMSE using far fewer parameters (0.4M vs. 1M+), showing greater performance and parameter efficiency. Our method also significantly outperforms the previous SOTA model, MLP trained on randomly permuted Coulomb matrices (CM). In addition, this MLP method is unlikely to scale to large molecules, as it needs a large sample of random permutations, which grows exponentially with the numbers of molecules, as pointed out in (Cohen et al. 2018).

7 Conclusions

In this work, we define chart-based PDOs and then use them to design rotation-equivariant spherical CNNs, PDO-eS^{2}CNNs. PDO-eS^{2}CNNs are easy to implement on non-Euclidean structured representations, and we analyze the equivariance error from discretization. Extensive experiments verify the effectiveness of our method.

One drawback of our work is that the equivariance cannot be preserved as well as S2CNN and FFS2CNN do in the discrete domain. In future work, we will explore more representations of the sphere and numerical calculation methods, in order to improve the equivariance in the discrete domain.

References

Armeni, I.; Sax, A.; Zamir, A. R.; and Savarese, S. 2017. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *ArXiv e-prints* .

Baumgardner, J. R.; and Frederickson, P. O. 1985. Icosahedral Discretization of the Two-Sphere. *SIAM Journal on Numerical Analysis* 22(6): 1107–1115.

Blum, L. C.; and Reymond, J.-L. 2009. 970 Million Druglike Small Molecules for Virtual Screening in the Chemical Universe Database GDB-13. *Journal of the American Chemical Society* 131(25): 8732–8733.

Boomsma, W.; and Frellsen, J. 2017. Spherical Convolutions and Their Application in Molecular Modelling. In *NeurIPS*, 3433–3443.

Bronstein, M. M.; Bruna, J.; Lecun, Y.; Szlam, A.; and Vandergheynst, P. 2017. Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine* 34(4): 18–42.

Bruna, J.; Zaremba, W.; Szlam, A.; and Lecun, Y. 2014. Spectral Networks and Locally Connected Networks on Graphs. In *ICLR*.

Cesa, G.; and Weiler, M. 2019. General E(2)-Equivariant Steerable CNNs. In *NeurIPS*, 14334–14345.

Chang, A. X.; Dai, A.; Funkhouser, T.; Halber, M.; Niebner, M.; Savva, M.; Song, S.; Zeng, A.; and Zhang, Y. 2017. Matterport3D: Learning from RGB-D Data in Indoor Environments. *International Conference on 3D Vision (3DV)* 667–676.

Cohen, T. S.; Geiger, M.; Kohler, J.; and Welling, M. 2018. Spherical CNNs. In *ICLR*.

Cohen, T. S.; Weiler, M.; Kicanaoglu, B.; and Welling, M. 2019. Gauge Equivariant Convolutional Networks and the Icosahedral CNN. In *ICML*, 1321–1330.

Cohen, T. S.; and Welling, M. 2016. Group Equivariant Convolutional Networks. In *ICML*, volume 48, 2990–2999.

Coors, B.; Condurache, A. P.; and Geiger, A. 2018. SphereNet: Learning Spherical Representations for Detection and Classification in Omnidirectional Images. In *ECCV*, 525–541.

Defferrard, M.; Milani, M.; Gusset, F.; and Perraudin, N. 2020. DeepSphere: a Graph-based Spherical CNN. In *ICLR*.

Dewan, A.; Caselitz, T.; Tipaldi, G. D.; and Burgard, W. 2016. Motion-based Detection and Tracking in 3D LiDAR Scans. In *ICRA*, 4508–4513.

Esteves, C.; Allenblanchette, C.; Makadia, A.; and Daniilidis, K. 2018. Learning SO(3) Equivariant Representations with Spherical CNNs. In *ECCV*, 54–70.

Frossard, P.; and Khasanova, R. 2017. Graph-Based Classification of Omnidirectional Images. In *IJCAI*, 860–869.

Gorski, K. M.; Hivon, E.; Banday, A. J.; Wandelt, B. D.; Hansen, F. K.; Reinecke, M.; and Bartelmann, M. 2005. HEALPix: A Framework for High-resolution Discretization and Fast Analysis of Data Distributed on the Sphere. *The Astrophysical Journal* 622(2): 759.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, 448–456.

Jiang, C.; Huang, J.; Kashinath, K.; Prabhat; Marcus, P.; and Niessner, M. 2019. Spherical CNNs on Unstructured Grids. In *ICLR*.

Kondor, R.; Lin, Z.; and Trivedi, S. 2018. ClebschGordan Nets: a Fully Fourier Space Spherical Convolutional Neural Network. In *NeurIPS*, 10117–10126.

Milnor, J. 1978. Analytic Proofs of the Hairy Ball Theorem and the Brouwer Fixed Point Theorem. *The American Mathematical Monthly* 85(7): 521–524.

Montavon, G.; Hansen, K.; Fazli, S.; Rupp, M.; Biegler, F.; Ziehe, A.; Tkatchenko, A.; Lilienfeld, A. V.; and Müller, K.-R. 2012. Learning Invariant Representations of Molecules for Atomization Energy Prediction. In *NeurIPS*, 440–448.

Perraudin, N.; Defferrard, M.; Kacprzak, T.; and Sgier, R. 2019. DeepSphere: Efficient Spherical Convolutional Neural Network with HEALPix Sampling for Cosmological Applications. *Astronomy and Computing* 27: 130–146.

Racah, E.; Beckham, C.; Maharaj, T.; Kahou, S. E.; Prabhat; and Pal, C. 2017. ExtremeWeather: A Large-scale Climate Dataset for Semi-supervised Detection, Localization, and Understanding of Extreme Weather Events. In *NeurIPS*, 3402–3413.

Ravanbakhsh, S.; Schneider, J.; and Póczos, B. 2017. Equivariance through Parameter-sharing. In *ICML*, 2892–2901.

Rupp, M.; Tkatchenko, A.; Müller, K.-R.; and Von Lilienfeld, O. A. 2012. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Physical review letters* 108(5): 058301.

Ruthotto, L.; and Haber, E. 2018. Deep Neural Networks Motivated by Partial Differential Equations. *Journal of Mathematical Imaging and Vision* 1–13.

Shen, Z.; He, L.; Lin, Z.; and Ma, J. 2020. PDO-eConvs: Partial Differential Operator Based Equivariant Convolutions. In *ICML*.

Sosnovik, I.; Szmaja, M.; and Smeulders, A. 2019. Scale-equivariant Steerable Networks. In *ICLR*.

Su, Y.; and Grauman, K. 2017. Learning Spherical Convolution for Fast Features from 360° Imagery. In *NeurIPS*, 529–539.

Weiler, M.; Geiger, M.; Welling, M.; Boomsma, W.; and Cohen, T. S. 2018. 3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data. In *NeurIPS*, 10381–10392.

Zhang, C.; Liwicki, S.; Smith, W. A. P.; and Cipolla, R. 2019. Orientation-Aware Semantic Segmentation on Icosahedron Spheres. In *ICCV*, 3533–3541.

Zhao, Q.; Zhu, C.; Dai, F.; Ma, Y.; Jin, G.; and Zhang, Y. 2018. Distortion-aware CNNs for Spherical Images. In *IJCAI*, 1198–1204.