Concentric Spherical Neural Network for 3D Representation Learning

James Fox
Georgia Institute of Technology
jfox43@gatech.edu

Bo Zhao
University of California, San Diego
bozhao@ucsd.edu

Beatriz Gonzalez del Rio
University of Valladolid
beatriz@metodos.fam.cie.uva.es

Sivasankaran Rajamanickam
Sandia National Laboratories
srajama@sandia.gov

Rampi Ramprasad
Georgia Institute of Technology
rramprasad3@gatech.edu

Le Song
Mohamed bin Zayed University of Artificial Intelligence
le.song@mbzuai.ac.ae

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9892358

***Abstract*—Learning 3D representations of point clouds that generalize well to arbitrary orientations is a challenge of practical importance in domains ranging from computer vision to molecular modeling. The proposed approach uses a concentric spherical spatial representation, formed by nesting spheres discretized the icosahedral grid, as the basis for structured learning over point clouds. We propose rotationally equivariant convolutions for learning over the concentric spherical grid, which are incorporated into a novel architecture for representation learning that is robust to general rotations in 3D. We demonstrate the effectiveness and extensibility of our approach to problems in different domains, such as 3D shape recognition and predicting fundamental properties of molecular systems.**

I. Introduction

Real-world 3D point cloud data today come from a variety of sources, with examples such as LiDAR sensors, RGBD cameras, and even snapshots from molecular simulations. Learning suitable representations of point clouds for data-driven modeling is well-motivated by applications spanning different domains, such as autonomous vehicles, scene understanding, and molecular modeling. However, it remains challenging and yet important to be able to generalize well to any 3D orientation of point clouds. Many different methods have been proposed over the years for structured learning over point cloud data [1]–[6], but many of these methods likewise suffer from catastrophic loss of performance when encountering arbitrarily oriented data at test time. One path to addressing this problem is to introduce rotations in training. While augmenting training in this manner is helpful, it has disadvantages of reducing training and model parameter efficiency, and furthermore there a significant performance gap remains.

Instead of relying on augmentation, an alternative strategy is to extract initial features that are rotationally *invariant*, from the input. This can be achieved using hand-crafted descriptors [7]–[12] or local reference frames [13]. Since the inputs to the model are invariant to rotation, then so is the output. While this strategy effectively eliminates the performance gap for non-rotated vs. rotated data, their absolute performance is also considerably lower compared to what was previously achievable (without invariant features). However, extracting rotationally invariant features from the input in a hand-crafted manner may significantly restrict what the model is able to learn, as important spatial relationships may be irrecoverably lost.

Another strategy is to incorporate rotations into the design of the model itself, which is the principle adopted by this work. Specifically, we focus on designing model layers which are *equivariant* to rotations. Broadly speaking, this property is satisfied if a rotation on the input feature space of a layer results in a corresponding rotation on its output feature space. Equivariance allows preservation of the input geometry throughout layers of the model, an advantage in hierarchical and descriptive feature learning. This principle already underpins the success of many models in the 2D domain, such as the equivariance of convolutional neural networks (CNNs) to translations [14] and other symmetry groups [15].

We propose a novel model, the Concentric Spherical Neural Network (CSNN) that is equivariant to 3D rotations and suitable for point cloud analysis. The design of CSNN centers around two main components: (1) a new 3D spatial grid structure and (2) equivariant convolutional layers. The proposed spatial structure is formed by nesting multiple spheres, each discretized by the icosahedral grid. The icosahedral grid is desirable as it results in a highly regular spatial sampling of the sphere. To learn over the concentric icosahedral grid, we propose distinct intra-sphere and inter-sphere convolutions which are combined to learn features within and between spheres. The resulting convolutions are rotationally equivariant and also scalable, scaling linearly with respect to the grid resolution.

We experimentally evaluate CSNN on two different types of problems: (1) 3D shape recognition and (2) molecular modeling. For the former problem, the goal is to correctly classify benchmark 3D that can appear in any orientation. For the latter problem, the objective is to predict the electronic density of states, a fundamental property of atomistic systems that is also invariant to rotation.

This work was funded through the National Science Foundation under Award Number 1900017, and Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

978-1-7281-8671-9/22/$31.00 ©2022 IEEE

To summarize our contributions: (1) We propose a new model to address the problem of generalizing to rotations in 3D representation learning through design of a concentric spherical structure and equivariant convolutions, (2) experimentally demonstrate the effectiveness of our approach by improving state-of-the-art for problems of 3D shape recognition and resolving electronic structure of atomistic systems, and (3) make our implementation and datasets publicly available at https://github.com/foxjas/CSNN for reproducibility.

II. Background and Related Work

This section provides background on the principle of rotational equivariance and further discussion of related work. To begin, we define a feature map $h : B_3 \to \mathbb{R}$ as a function which maps positions on the unit ball (spherical volume) to scalar features (without loss of generality). We define a rotation R on the feature map by the following operator:

$$[L_R h](\boldsymbol{x}) = h(R^{-1}\boldsymbol{x}) \tag{1}$$

In other words, each position $\boldsymbol{x} \in B_3$ is related by rotation to a corresponding position in the original feature map.

The space rotations is characterized by its group $\mathcal{R}$; for example, the space of all 3D rotations, which is continuous, belongs to the $SO(3)$ group. A layer Φ is *equivariant* to the rotation group $\mathcal{R}$ if it commutes with all its rotations:

$$\Phi[L_R h](\boldsymbol{x}) = [L_R(\Phi h)](\boldsymbol{x}), \quad \forall R \in \mathcal{R} \tag{2}$$

In other words, feeding a rotated input through the layer it the same as feeding the original input to the layer and rotating its output. Equivariance of layers ensures that their composition is also equivariant, making the entire model equivariant. To make the entire model invariant to rotation, it suffices to apply an appropriate global pooling operator after the equivariant layers, prior to output.

Equivariant models have been proposed for different structures in the 3D context. Most related to our work is Spherical CNNs [16]–[21], designed for convolutional feature learning over spherical images. However, they are not suitable for direct application to point clouds, with a fundamental limitation being the loss of information in constraining spatial representation from 3D domain to a 2D manifold. Some efforts have also been made to extend Spherical CNNs to point clouds [22], [23]. Rao et. al. [22] first learns a projection of points to the sphere, before spherical convolutions. You et. al. [23] proposes a spherical coordinate voxel grid as the basis for applying spherical convolutions, which has limitation in non-uniform spherical resolution and performance. Compared to prior work, our approach combines the use of a highly uniform 3D spatial grid with design of convolutions for directly and efficiently learning features over concentric spheres, and achieves state-of-the-art performance in practice.

III. Architecture Design

The primary goal of our proposed approach is to learn complete representations of 3D point clouds in a rotationally equivariant and scalable manner. To achieve this goal, we first propose a spatial structure of concentric spheres at different radii, each discretized by the icosahedral grid. The proposed construction organizes 3D space volumetrically by spherical and radial components. The icosahedral grid provides a highly regular sampling of the sphere, which provides efficient use of spatial resolution and also permits design of scalable and rotationally-equivariant convolutions. We propose using two separate convolutions together to learn volumetric features over concentric spheres: (1) graph-based convolution to learn features within spheres, and (2) co-radial convolutions to learn features between spheres. The proposed convolutions can be extended to different spatial scales via pooling and downsampling (following the regular properties of the icosahedral grid), resulting in the hierarchical convolutional architecture of Fig. 1.

A. Concentric Spherical Discretization

In this section we explain in detail our method of discretization by concentric spheres. We further present our approach to converting arbitrary point cloud data to initial feature channels over this spatial structure.

Concentric Icosahedral Grid. The initial icosahedron has 12 vertices forming 20 equilateral triangular faces. To increase grid resolution, each face can be sub-divided, with resolution scaling as $|V| = 10 * 4^l + 2$ (l is target discretization level). We implement concentric spheres by stacking R identical icosahedral grids to form the radial dimension (see Fig. 3). Assuming normalization to unit radius, concentric spheres are uniformly distributed over radii $[\frac{1}{R}, \frac{2}{R}, ..., 1]$. Assuming single-channel feature map, the resulting grid is the matrix $\boldsymbol{H} \in \mathbb{R}^{R \times |V|}$, where each vertex is indexed by the sphere it belongs to, and its position on the sphere. The icosahedral discretization results in a highly regular spatial sampling on the sphere, which provides very uniform spherical resolution. While resolution is not uniform between spheres (sampling density is higher for spheres closer to the center), this difference is largely explained by a scaling factor, and does not inhibit the design of rotationally equivariant convolutions.

Point Cloud to Concentric Spheres. We now consider the problem of converting a point cloud $\boldsymbol{P} \in \mathbb{R}^{N \times 3}$ to a concentric spherical feature input $\mathbf{H} \in \mathbb{R}^{R \times |V| \times C}$, where C is number of channels. While the concentric grid representation is defined discretely, the space point positions are continuous. To summarize the contribution of points in a continuous fashion we use a Gaussian radial basis function (RBF)

$$f(\boldsymbol{x}) = \sum_{j=1}^{N} \phi(||\boldsymbol{x} - \boldsymbol{P}_j||_2^2) \tag{3}$$

where N is the number of data point and $\phi = \exp(-\gamma r^2)$. In practice we limit computation to a local neighborhood of points (instead of considering all points), and choose γ such that contributions of points on the border of the neighborhood decay to a small value. Instead of computing the summation in Eq. 1 with respect to all points, for each data point we update the features of vertices in a local neighborhood. Restricting

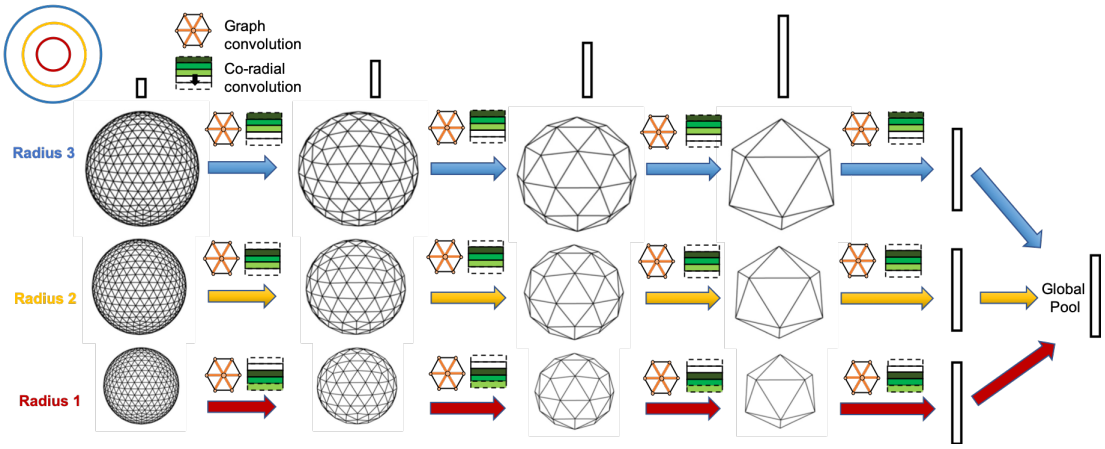


Fig. 1: Example architecture with three concentric spheres. Graph convolutions are followed by co-radial convolutions, at each level of spherical resolution. Co-radial convolution (in this example) has spatial window of three co-radial vertices, with padding applied to maintain radial dimensions across convolutions. Each arrow indicates vertex neighborhood pooling and downsampling, after which convolutions proceed with new filters at coarser spatial resolution. Global pooling is applied to obtain final feature representation.

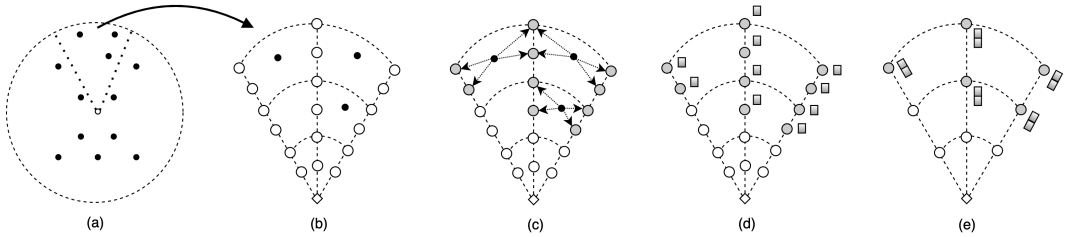


Fig. 2: Illustrative example of converting point cloud to concentric icosahedral grid (2D cross-section view). (a) An example point cloud (black points) is contained by a maximal radius sphere. (b) The spherical volume is further partitioned into 6 concentric spheres, co-radially. (c) Each point has a contribution to vertices in a local neighborhood (gray circles), resulting in (d) single channel feature per vertex (gray square). (e) Co-radial vertices are further grouped, resulting in smaller subset of concentric spheres with multi-channel inputs. In this example, grouping results in 3 concentric spheres where vertices have 2 input channels each.

computation to fixed size local neighborhoods instead of computing the summation in Eq. 1 with respect to all points means the overall point cloud to concentric spherical grid conversion is $O(N)$. We refer to Fig. 2(a)-(d) for illustrative example of the conversion.

Instead of restricting to 1-to-1 assignment of vertices to their corresponding sphere, we can group further group vertices into smaller number of spheres by concatenating features of co-radial vertices as input channels, as shown in Fig. 2(e). Letting R' be the initial number of spheres, grouping the features from co-radial vertices across R groups results in feature tensor $\mathbf{H} \in \mathbb{R}^{R \times |V| \times \frac{R'}{R}}$, where R is the number of spheres represented spatially, and $\frac{R'}{R}$ is the number of spheres represented via input channels. The proposed grouping mechanism gives ability to represent radial resolution either spatially or input-channel wise. This flexibility is beneficial for tackling computational and memory limitations, as the complexity of convolutions scales with the spatial dimensions of the grid rather than initial input channel dimensions.

B. Concentric Spherical Convolutions

In this section we present our implementation of rotationally equivariant intra-sphere and inter-sphere convolutions for feature learning. Proof of equivariance is further provided in Sec. III-D.

Intra-sphere convolutions. The objective for intra-sphere convolutions is to learn localized features within each sphere, in a rotationally equivariant fashion. We use localized graph convolutional filters for this objective. We construct the undirected graph $G^{(l)} = (V^{(l)}, E^{(l)})$ corresponding to level l icosahedron $I^{(l)}$. The vertex set $V^{(l)}$ corresponds one-to-one with the vertex set of $I^{(l)}$, but projected to unit sphere. To form the edge set $E^{(l)}$, we connect vertices of $V^{(l)}$ corresponding to face edges of the icosahedron. The resulting graph is highly regular, as all vertices are degree six beyond $I^{(0)}$, and all edges within each sphere are also approximately equidistant. We adopt the graph convolutional operator from [24] (but omitting degree-based normalization), and introduce additional notation to define graph convolution in our context. Let $\mathbf{H} \in \mathbb{R}^{R \times |V| \times C}$ denote a C channel tensor of features, and $\boldsymbol{Z} \in \mathbb{R}^{C \times F}$ be

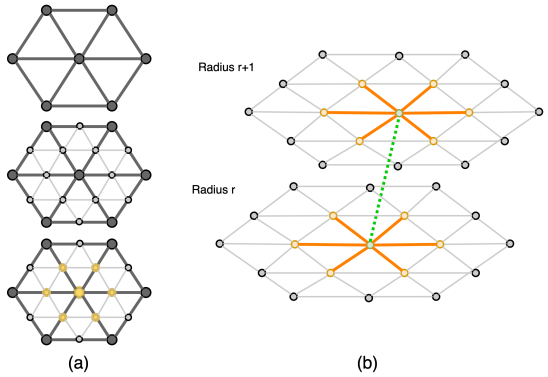


Fig. 3: (a) The icosahedral grid is formed by vertices of equilateral triangles (top), which can be recursively subdivided to form a higher resolution grid (middle). This also defines a natural vertex neighborhood and hierarchy for pooling and downsampling, where yellow highlighted vertices (bottom) are involved in pooling. (b) Two spherical grids are stacked, corresponding to consecutive concentric spheres. Graph convolution involves vertices within the same sphere (convolution neighborhood highlighted orange). Conversely, co-radial convolution involves co-radial vertices between the two spheres (green dotted line).

learnable weights, where C and F are input and output channel sizes. We use to $N(u)$ denote neighbors of vertex u in graph G. We also introduce subscript t to indicate convolutional layer number, $i \in [0, R-1]$ to index the radial dimension, and $u \in [0, |V|-1]$ to index the vertices. The layer $t+1$ intra-sphere convolution output for vertex u of sphere i is then given by Eq. 4, where σ is a nonlinear activation function:

$$\mathbf{H}_{i,u}^{(t+1)} = \sigma\Big(\sum_{v \in N(u)} \mathbf{H}_{i,v}^{(t)} \boldsymbol{Z}^{(t)}\Big) \tag{4}$$

Inter-sphere convolutions. We introduce *co-radial convolutions* for learning features between spheres. To achieve this, we view co-radial vertices as an ordered sequence and use 1D convolution to learn localized features between spheres (see Fig. 3(b) for illustration). We introduce some additional notation to describe co-radial convolutions: let K be the size of the 1D convolution kernel window. We pad inputs in the radial dimension such that the number of spheres R is maintained spatially across convolutions. Let $\boldsymbol{W} \in \mathbb{R}^{K \times C \times F}$ be a tensor of shared parameters, where C and F are input and output channel sizes. The layer $t+1$ co-radial convolution output for vertex u of sphere i is:

$$\mathbf{H}_{i,u}^{(t+1)} = \sigma\Big(\sum_{k=-\lfloor \frac{K}{2} \rfloor}^{\lfloor \frac{K}{2} \rfloor} \mathbf{H}_{i+k,u}^{(t)} \mathbf{W}_{k+\lfloor \frac{K}{2} \rfloor}^{(t)}\Big) \tag{5}$$

C. Complexity Analysis

The neighborhood size of both graph and co-radial convolution filters are constant, as are their filter parameters. Therefore the overall complexity of both intra-sphere and inter-sphere convolution is $O(R \times |V|)$, where $|V|$ is the resolution of the icosahedral grid, while the factor R corresponds from stacking multiple such grids. In other words, the convolutions scale linearly with respect to total grid size.

D. Equivariance of Convolutions

Since our work is based on the icosahedral grid, we focus equivariance analysis with respect to the icosahedral rotation group $\mathcal{I}$, a subgroup of $SO(3)$ containing 60 discrete symmetries. We start with the definition of intra-sphere convolution for single-channel feature map (without loss of generality):

$$\Phi h(\boldsymbol{x}_{i,j}) = \sigma\Big(\theta \sum_{\boldsymbol{x}_{i,k} \in N(\boldsymbol{x}_{i,j})} h(\boldsymbol{x}_{i,k})\Big) \tag{6}$$

where $\boldsymbol{x}_{i,j}$ is the position corresponding to a vertex in the concentric spherical grid, indexed by radial and spherical dimension. Additionally, θ is trainable parameter, $N(\boldsymbol{x}_{i,j})$ denotes positions of vertices in the neighborhood of the vertex at $\boldsymbol{x}_{i,j}$, and σ is nonlinearity function. Equivariance of the proposed layer is shown as follows:

$$\Phi[L_R h](\boldsymbol{x}_{i,j}) = \Phi h(R^{-1}\boldsymbol{x}_{i,j}) \tag{7}$$

$$= \sigma\Big(\theta \sum_{\tilde{\boldsymbol{x}}_{i,k} \in N(\tilde{\boldsymbol{x}}_{i,j})} h(\tilde{\boldsymbol{x}}_{i,k}), \quad \tilde{\boldsymbol{x}}_{i,k} = R^{-1}\boldsymbol{x}_{i,k}\Big) \tag{8}$$

$$= [L_R(\Phi h)](\boldsymbol{x}_{i,j}) \tag{9}$$

The second equality follows from rotation $R \in \mathcal{I}$ being an isometric transformation that maps the icosahedral sphere onto itself. This means that each vertex position $\boldsymbol{x}_{i,j}$ of the rotated feature map corresponds to an vertex unique position $R^{-1}\boldsymbol{x}_{i,j}$ in the original feature map, and that vertex neighborhoods are also preserved. The final equality follows from applying Eq. 1 and Eq. 6.

Next, we show that intra-sphere convolution is also rotationally equivariant:

$$\phi h(\boldsymbol{x}_{i,j}) = \sigma\Big(\sum_{k=-\lfloor \frac{K}{2} \rfloor}^{\lfloor \frac{K}{2} \rfloor} h(\boldsymbol{x}_{i+k,j}) \beta_{k+\lfloor \frac{K}{2} \rfloor}\Big) \tag{10}$$

where K is the size of the co-radial kernel and β denotes trainable parameter. We show that the convolution ϕ commutes with rotation:

$$\phi[L_R h](\boldsymbol{x}_{i,j}) = \phi h(R^{-1}\boldsymbol{x}_{i,j}) \tag{11}$$

$$= \sigma\Big(\sum_{k=-\lfloor \frac{K}{2} \rfloor}^{\lfloor \frac{K}{2} \rfloor} h(\tilde{\boldsymbol{x}}_{i+k,j}) \beta_{k+\lfloor \frac{K}{2} \rfloor}\Big) \tag{12}$$

$$= [L_R(\phi h)](\boldsymbol{x}_{i,j}) \tag{13}$$

where $\tilde{\boldsymbol{x}}_{i+k,j} = R^{-1}\boldsymbol{x}_{i+k,j}$. The second equality follows trivially from the fact that co-radial vertices remain co-radial after shared rotation, thereby preserving neighborhood for convolution.

IV. Experiments

We demonstrate the effectiveness of our approach for 3D object classification in Sec. IV-A, and applied to resolving fundamental properties of electronic structure in materials in Sec. IV-B. We further study important components to the performance of our approach in Sec. IV-C. Each experiment was run on a single NVIDIA V100 GPU.

A. Point Cloud Classification

We consider the ModelNet40 3D object recognition task, where CSNN uses the centroid of each object as the rotational reference for evaluation and the center for global feature extraction. We use the pre-processed point clouds from [2], with 1024 points each. In total there are 12308 shapes from 40 categories, with 9840 shapes used for training and 2468 for testing.

Architecture and Hyperparameters. See Fig. 4 for overview of model architecture and layers. Batch normalization and ReLU activation is applied after each convolution and hidden layer. The point cloud is converted to initial concentric spherical features following the procedure in III-A. We trained separate models for the two types of rotations evaluated: one for z-axis aligned rotations, and one for general $SO3$ rotations. For sake brevity, we only detail the hyperparameters for model trained on $SO3$ rotations, and refer to our codebase for details on the other model. In our experiment, CSNN uses $L = 4$, $R = 20$, $C = 8$, corresponding to level 4 icosahedral resolution (2562 vertices/sphere), 20 spatial spheres, and 8 input channels per spatial sphere. The model is trained using Adam optimizer for 60 epochs, batch size of 32, initial learning rate 3.9e-4 with decay factor 0.1, and early termination when learning rate falls below 1e-5. For regularization, CSNN uses dropout of 0.14 and weight decay of 2.7e-7. We follow prior work in augmenting training with random translation, re-scaling, and positional jitter of input point clouds. Specifically, we apply random uniform translation in the range of $[-0.1, 0.1]$, random Gaussian noise for positional jitter with standard deviation of 0.01, and random uniform re-scaling by a factor of $[0.8, 1.2]$ applied independently to each axis.

Results. For experimental evaluation, we consider two types of rotations in training and/or testing, following convention from earlier work: z-axis aligned rotations and arbitrary rotations ($SO3$). The latter is the most general and challenging. We compare with related work organized into three categories, based on their strategy for handling rotations. Results from running each baseline are presented in Table I. Our method achieves state of the art performance in z/z and SO3/SO3 settings, when training and testing draws from the same space of rotation. Our approach is not best, but remains competitive even when restricted to z-axis rotations while testing on general rotations (z/SO3 setting). This difference is likely due to artifacts of discretization from the initial mapping of points to the concentric icosahedral grid, which is largely mitigated by $SO3$ rotations in training. The weakness of methods which rely solely on training augmentation [4]–[6] is highlighted when testing on arbitrary rotations. Methods like RI-GCN [13] rely on designing invariant features at input to achieve rotational invariance, and so performance is consistent regardless of rotations seen in training. However, the overall performance of invariant methods is lower across settings compared to equivariant methods such as ours and [22], suggesting this form of invariance comes at a cost.

Method	Strategy	Params	z/z	z/SO3	SO3/SO3
PointNet [2]	Augmentation	3.5M	87.5	22.9	84.9
DGCNN [4]	Augmentation	1.8M	90.7	35.5	89.0
ShellNet [6]	Augmentation	470K	89.2	22.9	84.8
KPConv [5]	Augmentation	6.1M	90.0	27.5	85.0
SPHNet [12]	Invariance	2.9M	86.5	85.6	87.0
RIConv [11]	Invariance	0.7M	87.0	87.0	87.2
RI-GCN [13]	Invariance	4.4M	89.2	**89.3**	89.1
SFCNN [22]	Equivariance	9.2M	90.8	84.2	89.6
PRIN [23]	Equivariance	1.7M	76.5	81.9	81.0
CSNN	Equivariance	4.0M	**91.0**	88.3	**90.1**

TABLE I: ModelNet40 object classification overall accuracy, considering two types of rotations: z-axis aligned, and more general $SO3$ rotations. For example, $SO3/SO3$ indicates training and testing with arbitrary rotations of input data. Strategy refers to how rotations are handled. Our approach (CSNN) achieves state-of-the-art performance in two out of three settings.

B. Resolving Electronic Structure of Materials

Accurate molecular dynamics simulation from quantum-mechanical principles is critical to many applications, such as the design of advanced materials or the study of materials' properties under extreme conditions. However, accurately scaling simulations to systems beyond hundreds of atoms is a problem of primary concern. The main bottleneck appears when solving the quantum mechanical questions, which provides fundamental properties describing the electronic structure. Recent ML efforts have tried to overcome this issue by effectively predicting the electronic structure (output) from the atomic structure (input) [25]–[29]. Here we aim to effectively and accurately predict the electronic density of states (DOS), a rotation-invariant quantity describing the energy distribution of the electrons within an atomic snapshot. From the DOS, the band energy, an essential component of the total energy of the system, can be calculated. Due to the atomic nature of the problem, we propose learning atom-centered descriptors of local environments end-to-end, enabling data-driven and more flexible representations compared to hand-crafted descriptors of prior work of [27], [30]. We further introduce and make publicly available a dataset consisting of geometrically diverse structures of graphene, and show that the use of CSNN lowers overall error for calculating band energy, and increases the number of structures resolved to chemical accuracy.

Dataset. The datasets consists of eight different types of graphene allotropes: graphene sheet, graphite, three different fullerenes, and three single-walled nanotubes–see Fig. 5a). There are 200 atomic snapshots per structure, generated from snapshots of DFT molecular dynamics simulations run using VASP [31], [32]. The number of atoms per snapshot range from

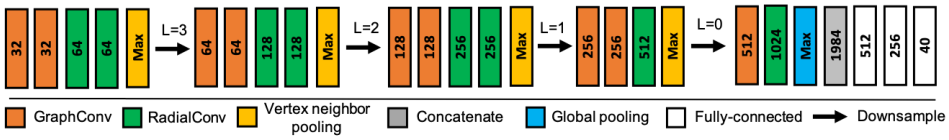


Fig. 4: Architecture for ModelNet40 classification. Number of output channels are shown for each layer, where applicable. $L = 4$ is initial level of discretization of icosahedral spheres. Radial convolution uses kernel size of 3 for spatial size (co-radial vertices).

20 up to 152. After alignment with respect to the vacuum energy used as global reference, the resulting DOS curve is binned into 310 windows of 0.1 eV each, from -30 to 1 eV. We refer to [27] for more details on the data generation and preparation process. In total there are 1280 training, 320 validation, and 160 test snapshots.

Problem Formulation. Each input is a snapshot of positions of carbon atoms, represented by 3D coordinates. Additionally, the atoms are located inside a unit cell with periodic boundary conditions. The prediction target, the DOS, is a fixed dimension vector $\boldsymbol{y} \in \mathbb{R}^{310}$. We evaluate the predicted DOS by using it to compute an important downstream property of interest, the band energy:

$$E_{band} = \int_{-\infty}^{E_F} DOS(\epsilon)\epsilon d\epsilon \tag{14}$$

where ϵ is the energy and E_F is the Fermi level. The integration has an upper bound of E_F, a physical limit representing the highest energy of the bound electrons. E_F is calculated as the energy at which the cumulative integral of the DOS curve equals the total number of electrons in the system. Since this limit is a function of the DOS integral and electron number, we introduce an additional prediction target in terms of the cumulative DOS (FDOS). Including the FDOS during training enables better resolution of E_F and lowers band energy error. The resulting objective function to minimize is:

$$L = \alpha * L_{DOS}(\boldsymbol{y}, \hat{\boldsymbol{y}}) + (1 - \alpha) * L_{FDOS}(\boldsymbol{y}, \hat{\boldsymbol{y}}) \tag{15}$$

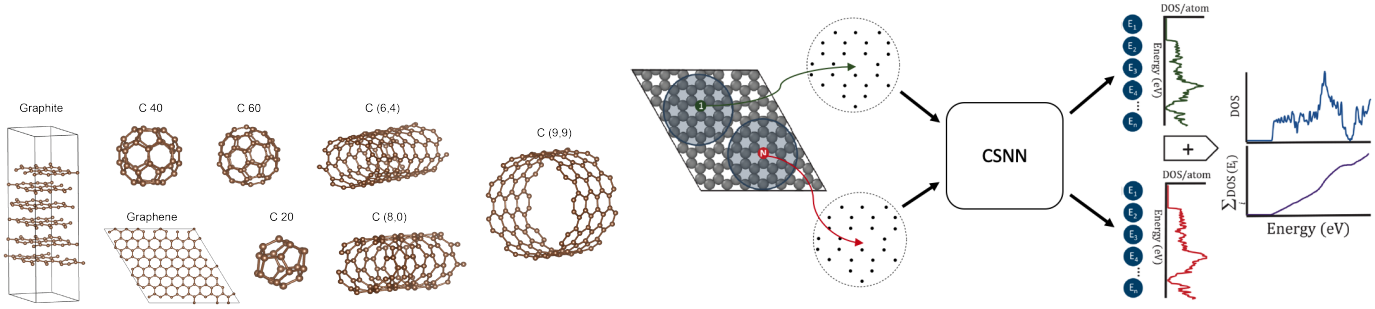
where $\hat{\boldsymbol{y}}$ is the predicted DOS and α controls the relative weighting between DOS and FDOS mean squared error losses.

Architecture and Hyperparameters. To model the total DOS of a system, we predict the contribution of each atom to the overall DOS, where each atom's contribution is a function of its local atomic atomic environment. The closer the neighboring atoms are to the target, the stronger the effect they have on the target's properties. To account for this effect, a fixed cutoff radius of 7 angstroms is used in experiments, eliminating the effect of neighbors that are further away. Our approach is then applied to learn a suitable descriptor of each local environment for mapping to atom-wise DOS contributions, which are then summed to obtain the overall DOS. This workflow is illustrated in Fig. 5b. The input to CSNN is then locally-centered point clouds, corresponding to atom-centered environments. To convert a point cloud to concentric spherical feature map, each point is assigned a contribution to its nearest vertex. The contribution is determined as the inverse of the point's distance from center, based on the aforementioned neighbor effect. We refer to Fig. 6 for details of model layers. In our experiment, CSNN uses $L = 3$, $R = 1$, $C = 32$, corresponding to level 4 icosahedral resolution (642 vertices/sphere), 1 spatial spheres, and 32 input channels per spatial sphere. The main consideration for using $R = 1$ was computational and memory efficiency, as pushing co-radial information to input channels does not require adding additional spatial dimensions. We also tested a channel-wise single-sphere ($C = 1$) version of the model which performed considerably worse at 0.036 eV/atom mean absolute error, highlighting the need for concentric spherical information. The model is trained using Adam optimizer for 1500 epochs, batch size of 32 snapshots, initial learning rate 5e-4 with decay factor 0.1 and early termination when learning rate falls below 1e-5. A weight decay of 1e-7 is applied for regularization. Using $\alpha = 0.1$ provided best performance for weighting DOS and FDOS losses. A single 1D convolution with kernel size of 3 is applied to smooth each atom's predicted DOS contribution prior to summing.

Results. We present our results in Table II. AGNI [33], [34] is a hand-crafted descriptor method for extracting rotationally invariant features of atomic environments applied to this problem and dataset in [27]. SchNet [9] is a rotation-invariant neural message passing model for learning atom-centered features, and is considered a strong baseline for many atomistic ML problems. Our rotationally equivariant approach achieves the lowest mean error in resolving band energy on the test set, reducing overall error by 24% relative to previous best. Our approach also demonstrates the ability of learned descriptors to improve over the performance of hand-crafted descriptors for this problem. Since the dataset is composed of different types of geometries, we further group test error by each type. Our approach achieves the lowest mean error on six out of eight structures. In absolute terms, it is also important for methods to achieve achieve chemical accuracy (0.043 eV/atom) to be of practical use, a widely adopted threshold in computational chemistry. Our approach is also able to resolve more structure (seven of eight) to chemical accuracy compared to prior approaches.

C. Performance Analysis

In this section we analyze the number of concentric spheres, a key component to the performance of our proposed architecture. We evaluate on the ModelNet40 dataset for SO3/SO3 test accuracy. To rule out potential impact of other factors, the number of trainable model parameters and all other training hyperparameters were kept identical. Recall that the concentric



(a) (b)

Fig. 5: (a) Illustrations of each type of carbon structure present in the dataset. Only atom position information is available for experiments; bonds shown are illustrative only. (b) From left to right: each atom's local atomic environment is the input to our proposed approach for learning DOS. CSNN (shared across inputs) learns an atom-centered descriptor from each environment, which is used to predict an atom-wise DOS contribution. All atoms' contributions are summed to obtain total DOS of the atomistic configuration.

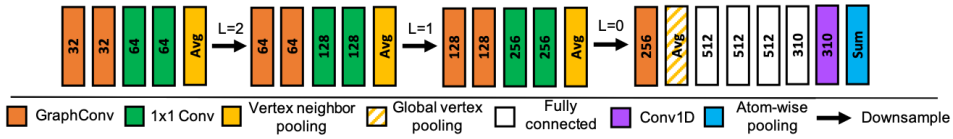


Fig. 6: Architecture for DOS prediction. Number of output channels are shown for each layer, where applicable. $L = 3$ is initial icosahedral spherical resolution. 1x1 convolution is applied within channels, without spatial component. 1D convolution is applied to smooth the predicted DOS curve, represented by output energy bins.

Model	Type	Params	Graphene	Graphite	C20	C40	C60	C(6,4)	C(9,9)	C(8,0)	Overall
AGNI [33]	Hand-crafted	392K	0.021	0.053	0.052	**0.030**	**0.010**	0.046	0.027	0.026	0.033
SchNet [35]	Learned	976K	0.042	0.045	0.065	**0.030**	0.022	0.022	0.033	0.019	0.035
CSNN	Learned	1.06M	**0.013**	**0.039**	**0.051**	0.033	0.017	**0.020**	**0.014**	**0.015**	**0.025**

TABLE II: Comparison of different models applied to predicting density of states (DOS). We report the mean error (eV/atom) in resolving band energy using predicted DOS aggregated over all snapshots, as well as by structure type. CSNN achieves lowest overall error, as well as lowest error in six out of eight structures.

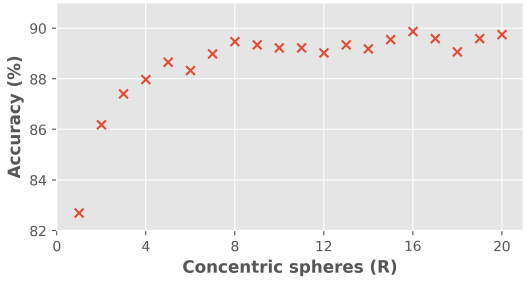


Fig. 7: Number of concentric spheres used (spatial) vs. classification accuracy. Evaluation is performed on ModelNet40 dataset for arbitrary rotations of samples.

spherical mapping can be represented in two different ways: (1) input channel-wise or (2) spatially. We focus on isolating the impact of the number of spatial spheres, as this directly impacts the cost of convolutions and would need justification. We use variable R to denote the number of spheres represented spatially, while the number of spheres represented via input channels is fixed at $C = 1$. We present results in Fig. 7,

Spheres	1	2	4	8	16	32
Time/epoch (s)	50	66	86	83	104	205

TABLE III: Time per training epoch in seconds vs. number of concentric spatial spheres.

which show that increasing concentric resolution significantly improves classification accuracy, with relative improvement of 8.7% from $R = 1$ to $R = 16$. We additionally compare the cost in training time (per epoch) from additional spatial spheres in Table III. Compared to single sphere, training time is increased by a factor of 2 for $R = 16$, which is where accuracy peaks. From this analysis we observe significant accuracy benefits from concentric spheres, without incurring prohibitive runtime costs.

V. Conclusion

In this work we present a novel approach, the Concentric Spherical Neural network, to address the problem of generalizing to rotations in representation learning of 3D point cloud data. We achieve this by proposing a new convolutional

approach based on the structure of concentric spheres and principles of equivariant design. We experimentally demonstrate the effectiveness our approach applied to different problems in computer vision and quantum chemistry domains, respectively. In the former, CSNN improves state-of-the-art on a standard 3D classification benchmark, ModelNet40, in handling arbitrary orientations of common objects. In the latter, CSNN is used to more accurately resolve the band energy of carbon-based materials by up to 24% compared to prior approaches.

REFERENCES

[1] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Sep 2015, p. 922–928. [Online]. Available: http://ieeexplore.ieee.org/document/7353481/

[2] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[3] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108.

[4] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, Oct. 2019. [Online]. Available: https://doi.org/10.1145/3326362

[5] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6411–6420.

[6] Z. Zhang, B.-S. Hua, and S.-K. Yeung, "Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct 2019, p. 1607–1616. [Online]. Available: https://ieeexplore.ieee.org/document/9010996/

[7] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, "Quantum-chemical insights from deep tensor neural networks," *Nature Communications*, vol. 8, no. 1, p. 13890, Apr 2017.

[8] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 1263–1272.

[9] K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K.-R. Müller, "Schnet: A continuous-filter convolutional neural network for modeling quantum interactions," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 992–1002.

[10] C. Chen, G. Li, R. Xu, T. Chen, M. Wang, and L. Lin, "Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4994–5002.

[11] Z. Zhang, B.-S. Hua, D. W. Rosen, and S.-K. Yeung, "Rotation invariant convolutions for 3d point clouds deep learning," in *2019 International Conference on 3D Vision (3DV)*. IEEE, 2019, pp. 204–213.

[12] A. Poulenard, M.-J. Rakotosaona, Y. Ponty, and M. Ovsjanikov, "Effective rotation-invariant point cnn with spherical harmonics kernels," in *2019 International Conference on 3D Vision (3DV)*. IEEE, 2019, pp. 47–56.

[13] S. Kim, J. Park, and B. Han, "Rotation-invariant local-to-global representation learning for 3d point cloud," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 8174–8185.

[14] T. Cohen and M. Welling, "Group equivariant convolutional networks," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 2990–2999. [Online]. Available: http://proceedings.mlr.press/v48/cohenc16.html

[15] M. Weiler and G. Cesa, "General e (2)-equivariant steerable cnns," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[16] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling, "Spherical CNNs," *International Conference on Learning Representations*, 2018.

[17] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis, "Learning SO(3) equivariant representations with spherical CNNs," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 52–68.

[18] C. M. Jiang, J. Huang, K. Kashinath, Prabhat, P. Marcus, and M. Niessner, "Spherical CNNs on unstructured grids," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkl-43C9FQ

[19] T. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling, "Gauge equivariant convolutional networks and the icosahedral CNN," in *International Conference on Machine Learning*, 2019, pp. 1321–1330.

[20] M. Defferrard, M. Milani, F. Gusset, and N. Perraudin, "Deepsphere: a graph-based spherical cnn," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=B1e3OlStPB

[21] Q. Yang, C. Li, W. Dai, J. Zou, G.-J. Qi, and H. Xiong, "Rotation equivariant graph convolutional network for spherical image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4303–4312.

[22] Y. Rao, J. Lu, and J. Zhou, "Spherical fractal convolutional neural networks for point cloud recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 452–460.

[23] Y. You, Y. Lou, Q. Liu, Y.-W. Tai, W. Wang, L. Ma, and C. Lu, "Pointwise rotation-invariant network with adaptive sampling and 3D spherical voxel convolution," *The AAAI Conference on Artificial Intelligence*, 2020.

[24] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *International Conference on Learning Representations*, 2017.

[25] A. Chandrasekaran, D. Kamal, R. Batra, C. Kim, L. Chen, and R. Ramprasad, "Solving the electronic structure problem with machine learning," *npj Computational Materials*, vol. 5, no. 1, pp. 1–7, 2019.

[26] A. Fabrizio, A. Grisafi, B. Meyer, M. Ceriotti, and C. Corminboeuf, "Electron Density Learning of Non-Covalent Systems," *Chem. Sci.*, vol. 10, no. 41, pp. 9424–9432, 2019.

[27] B. G. del Rio, C. Kuenneth, H. D. Tran, and R. Ramprasad, "An efficient deep learning scheme to predict the electronic structure of materials and molecules: The example of graphene-derived allotropes," *The Journal of Physical Chemistry A*, vol. 124, no. 45, pp. 9496–9502, 2020, pMID: 33138367. [Online]. Available: https://doi.org/10.1021/acs.jpca.0c07458

[28] D. Kamal, A. Chandrasekaran, R. Batra, and R. Ramprasad, "A charge density prediction model for hydrocarbons using deep neural networks," *Machine Learning: Science and Technology*, vol. 1, no. 2, p. 025003, mar 2020.

[29] J. A. Ellis, L. Fiedler, G. A. Popoola, N. A. Modine, J. A. Stephens, A. P. Thompson, A. Cangi, and S. Rajamanickam, "Accelerating finite-temperature kohn-sham density functional theory with deep neural networks," *Phys. Rev. B*, vol. 104, p. 035120, Jul 2021. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.104.035120

[30] C. Ben Mahmoud, A. Anelli, G. Csányi, and M. Ceriotti, "Learning the electronic density of states in condensed matter," *Physical Review B*, vol. 102, no. 23, p. 235130, Dec 2020.

[31] G. Kresse and J. Furthmüller, "Efficient Iterative Schemes for Ab Initio Total-Energy Calculations using a Plane-Wave Basis Set. ," *Phys. Rev. B*, vol. 54, no. 54, pp. 11 169–86, 1996.

[32] ——, "Efficiency of Ab-Initio Total Energy Calculations for Metals and Semiconductors using a Plane-Wave Basis Set." *J. Comput. Mater. Sci.*, vol. 6, no. 6, pp. 15–50, 1996.

[33] V. Botu and R. Ramprasad, "Learning scheme to predict atomic forces and accelerate materials simulations," *Phys. Rev. B*, vol. 92, p. 094306, Sep 2015. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.92.094306

[34] V. Botu, R. Batra, J. Chapman, and R. Ramprasad, "Machine learning force fields: Construction, validation, and outlook," *Journal of Physical Chemistry C*, vol. 121, pp. 511–522, 2016.

[35] K. Schütt, P.-J. Kindermans, H. E. S. Felix, S. Chmiela, A. Tkatchenko, and K.-R. Müller, "Schnet: A continuous-filter convolutional neural network for modeling quantum interactions," in *Advances in neural information processing systems*, 2017, pp. 991–1001.