Continuous SO(3) Equivariant Convolution for 3D Point Cloud Analysis

Jaein Kim[1,4], Hee Bin Yoo[2,4], Dong-Sig Han[3,4], Yeon-Ji Song[1,4], and Byoung-Tak Zhang[1,2,3,4]

[1] Interdisciplinary Program in Neuroscience, Seoul National University
[2] Interdisciplinary Program in Artificial Intelligence, Seoul National University
[3] Department of Computer Science and Engineering, Seoul National University
[4] Artificial Intelligence Institute, Seoul National University
{jykim,hbyoo,dshan,yjsong,btzhang}@bi.snu.ac.kr

Abstract. The inherent richness of geometric information in point cloud underscores the necessity of leveraging group equivariance, as preserving the topological structure of the point cloud up to the feature space provides an intuitive inductive bias for solving problems in 3D space. Since manifesting the symmetry by means of model architecture has an advantage over the dependence on the augmentation, it has been a crucial research topic in the point cloud field. However, existing methods have limitations in the non-continuity of groups or the complex architecture causing computational inefficiency. In this paper, we propose CSEConv: a novel point convolution layer equivariant under continuous SO(3) actions. Its structure is founded on the framework of group theory, realizing the convolution module defined on a sphere. Implementing its filters to be explicit, continuous, and rigorously equivariant functions defined upon the double coset space is the distinctive factor which makes our method more scalable than previous approaches. From the classification experiments on synthetic and real-world point cloud datasets, our method achieves the best accuracy, to the best of our knowledge, amidst point-based models equivariant against continuous rotation group.

Keywords: Geometric deep learning · 3D feature learning · 3D point clouds

1 Introduction

Advancements in 3D data processing technologies have been propelled by the increasing demand for rich geometric data across various application domains, such as robotics, autonomous driving, and augmented reality [18, 19]. This recent trend has given rise to the emergence of large-scale 3D datasets and deep learning techniques exploiting a point cloud. Nonetheless, a point cloud poses computational challenges that neural networks should concern its nature - an unordered set containing a large number of coordinate samples [18].

The equivariance toward symmetry facilitates the mapping of visual input to its representation while maintaining structural information [6]. With respect

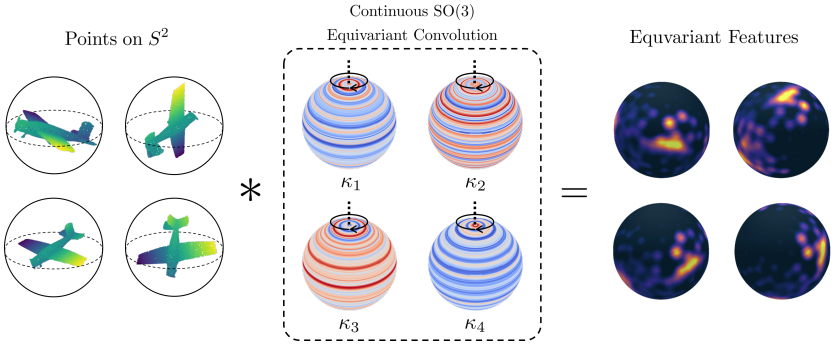

Fig. 1: Continuous SO(3) Equivariant Convolution (CSEConv) defines convolution operation on S^2. It maintains equivariance to the 3D rotation group SO(3) by ensuring the filter functions $(\kappa_1, \cdots, \kappa_4)$ are invariant to SO(2). What makes our method differ from existing convolution methods is that the filters are defined on continuous S^2, equivariant against random 3D rotations.

to point clouds, the learned representation is expected to be equivariant against 3D rotation; thus the model is able to project a complex collection of coordinates onto the symmetric representation space variable to the global topology. Though augmenting training data with random rotations also guides the approximated symmetric representation, it is known both analytically and pragmatically that reliance on augmentation has a limitation on the performance compared to structurally equivariant methods [15, 25]. Therefore, many previous works have proposed their own approaches to manifest innate equivariance, especially employing *group convolution* in 3D space [2, 14, 34, 39, 48].

One of the typical approaches to realize group convolution is to discretize the group domain into finite elements [2, 48]. This makes the explicit integration feasible over the group space, yet it forgoes the rigorous symmetry towards the continuous group actions. On the other hand, some methods preserve strict symmetry by defining filter function or replacing convolution based on harmonic analysis [7, 14, 34]. However, they are not applicable to a point cloud or underachieve in practice due to their computational complexity. Furthermore, both approaches mostly have defined filter functions using regular grids. Since points in point cloud are irregularly sampled, each value at the point between grid cells can only be approximated by gating [2, 48] or weight functions [14, 34]. Hence, previous works on group convolution have exhibited their limitations regarding the efficiency, the rigor of equivariance, and the smoothness of the filter.

This paper proposes a novel rotation-equivariant point convolution network named Continuous SO(3) Equivariant Convolution (CSEConv), leveraging concepts from group theory and diverse techniques from previous convolution research. The proposed method achieves SO(3) equivariance without the mentioned limitations through the following means: we reformulate a convolution operation on SO(3) into an equivalent operation on S^2, where the symmetry against rotations is preserved. Inspired by previous works regarding continu-

ous filter of equivariant convolution on 2D space [26, 27], we realize the filter of CSEConv as explicit and continuous functions as in Figure 1, implemented with simple neural networks instead of tensor-shaped parameters. They are designed to ensure SO(3) equivariance on S^2 with a constraint on their input coordinates, founded from the derivation of our method's formulation. We verify the robust equivariance and outstanding efficiency of the proposed method through comparative analyses with existing rotation-equivariant methods.

CSEConv is evaluated at point cloud datasets with different complexity: ModelNet40 [43] and ScanObjectNN [36]. Utilizing the suggested convolution layer with other techniques for point cloud learning, we implement the model that maps point cloud to feature vectors in a rotation-invariant manner. Notably, our model attains the best performance among methods equivariant to continuous groups in classification tasks, comparable to the non-equivariant methods augmented during training.
Our contributions are summarized as follows:

- We propose CSEConv that maintains a simple convolution structure and is equivariant to continuous SO(3) actions. The symmetry can be held due to the constraint on the filter function based on group theory.
- We implement the filter as the coordinate value-based neural network that shares its parameter along every input point. This enables the cost efficiency of our method while preserving its capacity.
- Performance analyses verify the robust equivariance and efficiency of our method compared to baselines. Benchmark experiments also demonstrate its scalability, as ours outperforms every baseline in classifying randomly rotated ModelNet40 when a model is trained without augmentation.

2 Related Works

2.1 Deep learning of point cloud

In spite of the abundant geometric information inherent in point clouds, its utilization is hindered by high-dimensionality and disorder. As a result, it remains common to transform point clouds into more summarized formats [3, 20, 21] in application domains. The preprocessing techniques enable the exploitation of conventional algorithms or models, yet they sacrifice the topological structure of the point cloud. PointNet [29] is one of the early deep learning methods that learn point cloud per se. It applies the pointwise network on every point to guarantee permutation invariance. PointNet++ [31] incorporates sampling and local grouping mechanisms to learn local features in multiple resolutions. However, it requires the whole PointNet structure to learn the locality in different resolutions, which is computationally expensive.

Adopting convolution is a plausible choice to learn the local features efficiently. Early works of 3D convolution rely on voxelized input and cost heavily in memory due to dense 3D filter grids [24, 30, 43]. Minkowski Engine [4] utilizes

a sparse tensor to overcome memory inefficiency, but it still requires discretization of input space. A convolution method using a point cloud explicitly requires a mechanism to weigh irregular samples in continuous space. PointConv [42] computes a linear filter matrix by an MLP and scales input features by density estimation. KPconv [33] defines its filter function by a combination of independent grids where weights measure the correlation between input and fixed anchor coordinates. Ummenhofer *et al*. [35] maps input coordinates within a ball region to a grid and computes filter by interpolating grid cell values. It is also worth mentioning that many works have recently adopted Transformer [37] and its learning techniques in point cloud analysis [17, 44, 47].

In summary, the aforementioned methods have learned the local features effectively by applying advanced architectures toward point cloud. However, they may struggle with the inference of symmetry in input space unless trained with data augmentation. Some previous works suggest extracting rotation-invariant features from local space partitioned with reference vectors [45, 46], projecting features onto learned orientation vectors which respond equivariantly against rotations [11,23], or building a mathematical framework that averages each result of the whole model acted by sampled group elements [28]. However, we will focus on methods that explicitly utilize or extend a convolution with group theory.

2.2 Group convolution on 3D data:

Cohen and Welling present G-CNN [6], which parametrizes filter grids defined over discretized group space. They also suggest Steerable CNN [10] that retains the equivariance from symmetric filters defined by the sum of orthogonal basis representation. These are early works of group convolution using two major strategies respectively, but they are confined to 2D images and finite groups.

In terms of 3D data, learning group equivariance usually aims to learn SO(3) or SE(3) equivariance. One of the most representative works is Spherical CNN [7], which generalizes the Fourier transform and utilizes the convolution theorem to replace convolution over a spherical surface. Additionally, Weiler *et al*. [39] and Cesa *et al*. [1] extend Steerable CNN to continuous group spaces, SO(3) and E(n), respectively. While these methods effectively compute convolution equivariant under continuous group actions, they are limited to handling discretized formats, such as spherical images or voxels. Tensor Field Network [34] adopts harmonic analysis similar to the above but available for point clouds, and SE(3)-Transformer [14] adds a scale dot-product attention on the Tensor Field Network to rescale value messages. Despite their strict equivariance, the computation of harmonic bases makes them extremely expensive in time complexity.

In contrast, the following methods utilize group space discretization to 3D data. Earlier works sample a discrete subset of 3D rotation group [40, 41] or change the domain to a discretized manifold [5] to approximate 3D-rotation equivariant convolution. These works, however, are yet applicable only to discretized data. EPN [2] is one of the representative roto-translation equivariant networks available for handling point cloud. It leverages a convolution strategy

of KPConv [33] while factorizing SE(3) group into discrete rotation and translation groups. E2PN [48] adapts EPN to operate convolution on the quotient space and use symmetric kernels to enhance both time and memory efficiency. Since EPN and E2PN conduct convolution on points and explicit integration on group space, we consider them as baselines for our convolution module.

3 Preliminaries

3.1 Problem setup

The point cloud is defined as $\mathcal{P} = \{(\mathbf{x}_i, f_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^3$ is the coordinate and $f_i = f(\mathbf{x}_i)$ is the feature. We denote each sample with an index, but $\mathcal{P}$ is a set without any order among samples. Thus a point cloud model $\mathcal{M} : \mathcal{P} \to \mathbb{R}^d$ is desired to be permutation invariant. Our work mainly addresses rotation-invariant tasks, which maintain identical labels against 3D rotations over input: $\forall g \in SO(3) \quad \mathcal{M}(\mathcal{P}) = \mathcal{M}(g\mathcal{P}) = \mathcal{M}(\{(g\mathbf{x}_i, f_i)\}_{i=1}^N)$. In addition, we assume that the centroid of object, or geometric center, is lying on the origin. Since a translation symmetry is practically satisfied by shifting point clouds as their average coordinate values, such assumption is easily expandable to a SE(3) symmetry. Thus the projection of points in 3D Euclidean space onto the surface of 3D sphere, or S^2, is computed with simple L2 normalization.

3.2 Group theory for extending convolution

A group is a set with a binary operation $(G, \cdot)$, where the operation is associative, the identity element exists, and every element is invertible. The significance of the group comes from that it delineates the system of symmetry, a transformation that preserves the shape of objects [6]. Such transformation is called *action*, which is a function of how the group acts on the object. We use a left action by default: the action of $g \cdot h$ for $g, h \in G$ acts the action of h first.

Proceeding from the general definition, we can decompose a group into subspaces useful to formulate the mathematical framework of convolution in 3D space. A *quotient space* is the base space of group G where the subgroup $H \subset G$ is lied on [9], denoted as G/H. The important property of G/H is that it is the set of *cosets*, and *coset* gH is the set referencing $g \in G$ associated by H: $gH = \{gh \mid h \in H\}$. For instance, SO(2) *coset* for $g \in$ SO(3) is the set of rotations sharing the same rotation axis. Thus a *quotient space* SO(3)/SO(2) corresponds to the set of rotation axes and is equivalent to the unit sphere S^2.

Moreover, a *quotient space* can be partitioned into disjoint orbits by another subgroup H_2. One of these orbits is called *double coset* $H_2\backslash g/H$, and their set is named *double coset space* $H_2\backslash G/H$. Back to the example, when H_2 is an another $SO(2)$, the rotation around the Z-axis, then a *double coset* corresponds to the arc on the sphere, orthogonal to the Z-axis. These arcs can be specified with their location on the line linking the north and south poles; thus this line is equivalent to the *double coset space* SO(2)\SO(3)/SO(2). See Appendix A.1. for more strict definitions of the above concepts. We can utilize them to implement symmetric filters, available for convolution on the quotient space [8].

4 Method

This section describes the actualization of point convolution equivariant to the continuous group action. We start off with the mathematical framework to define SO(3) equivariant convolution on S^2. Then we elaborate on the implementation of CSEConv, such as the approximation of integration and learning a filter function. Finally, we organize task-specific models with our module, suggesting classification and place recognition models.

4.1 SO(3) equivariant convolution framework

The group convolution manifests the equivariance toward general transformations, extended from the translation symmetry of convolution in real space [9]. It can be defined upon a measurable group space and the existence of the integrable filter functions. Thankfully, the group space of our interest, such as SO(3), always has a measure called the Haar measure $\mu_G(\cdot)$. The Haar measure is known to be invariant to group actions and uniquely exists in G [8, 13]. Then group convolution is defined as Equation 1:

$$(f * \kappa)(g) = \int_{h \in G} f(h^{-1} \cdot g))\kappa(h)\ d\mu_G(h), \tag{1}$$

where $f(\cdot)$ is a feature and $\kappa(\cdot)$ is a filter.

However, leveraging a group as the coordinate per se is intractable in 3D data. For example, it needs at least 4 dimensions in the Euclidean space to map coordinates isomorphically towards SO(3). Thus, the actual data representation, uniquely embedded in the 3D space, causes a redundant dimension if one naively maps real coordinates to group space. It implies the integration over the continuous SO(3) space is intractable, and problematic for our setups. Fortunately, convolution on group can be rewritten as the equivalent convolution defined on its quotient space [8, 9, 48] as Equation 2:

$$(f * \kappa)(g) = \int_{h \in S^2} f(h)\kappa(s(g)^{-1}h)\ dh, \tag{2}$$

where S^2 is 2-Sphere, $g, h \in S^2$, and $s : S^2 \to \mathrm{SO}(3)$ denotes a *section map*. By mapping input point on S^2 into SO(3) rotation by *section map*, $\kappa(\cdot)$ retains its property as a function responding to difference upon S^2 space. This reformulation is also tractable because the projection of 3D coordinates to a spherical surface is unambiguous. Nevertheless, Equation 2 is not SO(3) equivariant by itself. In favor of equivariance, the filter function $\kappa(\cdot)$ should satisfy the follows:

$$\forall g\ \in S^2\ \forall h \in \mathrm{SO}(2) \quad \kappa(g) = \kappa(h \cdot g), \tag{3}$$

which denotes that the filter function is invariant to SO(2) [48]. See the Appendix A.2 for the detailed proof of the constraint.

We have a geometrical explanation of the Equation 3 in Figures 2a and 2b. Given two input sets, $\{g, h\}$ and $\{g', h'\}$, which have identical topology but

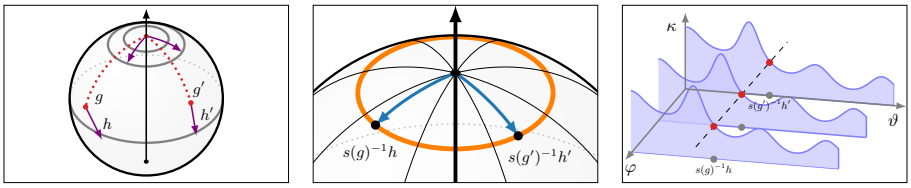

Fig. 2: The depiction of *section map* transformation in S^2 convolution and how CSEConv actualize the filter invariance. (a) Convolution on S^2 transforms point sets by the inverse of *section map*, aligning centroids to the reference frame (north pole). (b) When enlarging the north pole, transformed coordinates, $s(g)^{-1}h$ and $s(g')^{-1}h'$, lie on the arc around the pole axis, or *double coset*. (c) Our filter function $\kappa(\cdot)$ varies only by ϑ, whose domain is equivalent to the line of longitude, or *double coset space*. Thus $\kappa(\cdot)$ responds uniquely to *double coset*, guaranteeing SO(2) invariance.

different orientations, a section map $s(\cdot)$ aligns only the centroid coordinates g and g' to the identical reference point. Neighbor points such as h or h' are projected onto the identical *double coset*, which is bundled by the rotation around the Z-axis. Thus the filter function should be invariant to SO(2) actions. This condition will be highlighted later for the filter function design.

4.2 Implementation of CSEConv

This section explains how we employ Equation 2 into CSEConv, capable of conducting SO(3) equivariant convolution over a set of irregular points.

Approximated integration: A point cloud is a set of discrete and finite samples per se, where an integration is not applicable without approximating it with discretized summation. Following the inductive bias of the locality, we only consider the neighborhood of the input coordinate [12,16]. It is known that integration over the neighbor region remains equivariant in probability against the group with Haar measure [12], which validates our approximated convolution on S^2. Then Equation 2 is approximated as follows:

$$(f * \kappa)(g) = \frac{1}{|\mathcal{N}_g|} \sum_{h \in \mathcal{N}_g} f(h) \cdot \kappa(s(g)^{-1}h), \tag{4}$$

where $\mathcal{N}_g$ is actually determined by the K-nearest neighbors algorithm in the 3D space. Since the neighbors are determined by distances, it remains invariant under arbitrary rotations: $d(x, y) = d(gx, gy),\ \forall\ x, y \in S^2\ \forall g \in \mathrm{SO}(3)$ [12]. This property of invariance extends to the approximated integration 4.

Although we sample neighbors in Euclidean space, the coordinate values, g or h themselves, should be on S^2. Thanks to our assumption in Section 3.1, we can project them onto S^2 by normalizing their coordinate values. The computations of section map $s(\cdot)$ or filter $\kappa(\cdot)$ are done with these normalized values.

SO(3) equivariance constraint: We need to restrict the filter to fulfill the constraint 3 for the SO(3) equivariance, and it can be accomplished by confining

the domain of kernel functions. As Figure 2 (b), the inverse action of the section map sends the input element g to the north pole: the rotation around the Z-axis corresponds to our SO(2). If $g \in S^2$ is rotated around Z-axis by γ, its spherical coordinate becomes $(\varphi + \gamma, \vartheta)$. Thus the filter is invariant to SO(2) if we confine the domain of filter to ϑ as Equation 5:

$$\begin{aligned} \kappa : S^2 \to \mathbb{R}^n : g \mapsto \varkappa(\vartheta_g), \\ \text{s.t. } \varkappa : \mathbb{R} \to \mathbb{R}^n, \;\; g \equiv (\sin\vartheta_g \cos\varphi_g, \sin\vartheta_g \sin\varphi_g, \cos\vartheta_g). \end{aligned} \tag{5}$$

After all, the SO(3) equivariance of CSEConv originates from the domain of $\kappa(\cdot)$ free from SO(2) actions. We have the depiction of our filter in Figure 2c.

Filter learning by neural networks: We exploit the above condition to implement an efficient filter function while preserving the equivariance and the continuity. Since our filter function should be defined on a continuous ϑ, we substitute it to a neural network instead of tensor-shaped parameters. Fourier feature mapping [32] is leveraged to encode coordinate values in ϑ, effective for capturing high-frequency information in coordinate-based networks and learning filter functions in CNN [26]. This filter is named $\kappa_{\text{FF}}(\cdot)$ and expressed as:

$$\kappa_{\text{FF}}(g;\theta) = \text{MLP}(\text{FF}(\vartheta_g);\theta), \;\; \text{s.t. FF}(\vartheta) = [\sin(2\pi\mathbf{W}\vartheta), \cos(2\pi\mathbf{W}\vartheta)], \tag{6}$$

where $\mathbf{W}$ is a frozen parameter initialized from $\mathcal{N}(\mathbf{0}, \sigma^2)$. $\text{FF}(\cdot)$ maps univariate ϑ_g into multivariate feature vectors, and $\text{MLP}(\cdot)$ is a neural network parametrized by θ. Later, we show this method is more effective in time and memory complexity than existing methods using grid parameters.

Finally, the implementation of CSEConv can be written as follows:

$$(f * \kappa_{\text{FF}})(g;\theta) = \frac{1}{|\mathcal{N}_g|} \sum_{h \in \mathcal{N}_g} \kappa_{\text{FF}}(s(g)^{-1}h;\theta) \cdot f(h)^\top, \tag{7}$$

where $f(\cdot)$ is a feature vector in $\mathbb{R}^{D_{\text{in}}}$ for every neighbor point and $\kappa_{\text{FF}}(\cdot)$ maps coordinate differences to filter matrices in $\mathbb{R}^{D_{\text{out}} \times D_{\text{in}}}$. Thus the output of the convolution is a feature vector in $\mathbb{R}^{D_{\text{out}}}$ for an input point. In the next section, we introduce the architecture of whole task models using CSEConv layers.

4.3 Rotation-invariant CSEConv models

We implement two downstream task models for object classification and retrieval with CSEConv. Our models share the encoder hierarchy as Figure 3 before the downstream task module. During the preprocessing, rotation-invariant local features are extracted per every point by the projection onto the unit sphere, while projected S^2 points are treated as coordinate values to compute filter values. These local features and coordinates are propagated to the layers of convolution blocks, composed of CSEConv, activation, and batch normalization in order. The CSEConv layers sample down the number of points, if in need, by farthest point sampling and conduct convolution only on sampled points. Finally,

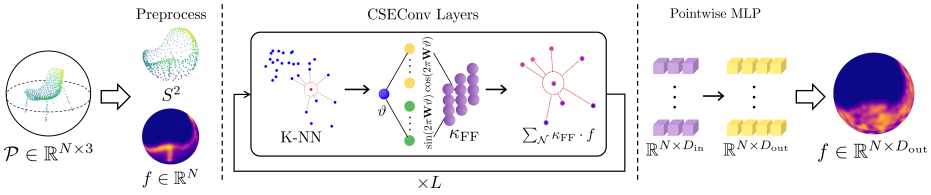

Fig. 3: The hierarchy of local feature encoder leveraging CSEConv layers. It receives a point cloud in $\mathbb{R}^{N\times 3}$ and returns a set of local features in $\mathbb{R}^{N\times D_{\mathrm{out}}}$. In this work, only the distance from the centroid ($\|x\|_2$) is assigned as the initial local feature.

a weight-sharing MLP, adapting implementation of PointNet [29], maps each local feature into higher dimensional space. These local features are aggregated along the dimension of points by max-pooling, which outputs a rotation-invariant global feature to solve downstream tasks.

Point cloud classification: A classification model has a simple classifier architecture, which consists of linear layers and ReLU activation, after the encoder hierarchy. It maps a global feature to a logit vector in class dimensions. Lastly, a softmax layer computes the predicted likelihood of each class and the cross entropy loss is minimized with Adam optimizer and cosine annealing scheduler [22].

Point cloud retrieval: A retrieval model also has a simple MLP to map global features to the metrizable space, normalizing with L2-norm. The triplet loss is optimized by Adam and cosine annealing to learn the model, sampling 2 positive samples and 8 negative samples distanced by Euclidean metric. Meanwhile, we empirically found that the model diverges at the beginning if the model parameter has to learn every parameter from scratch. This was mitigated by utilizing the encoder from the classification model as the frozen pre-trained model and only fine-tuning the last MLP to learn a desired metric space for a given task.

5 Experiments

5.1 Analysis on CSEConv

Equivariance error of CSEConv: In this experiment, we compare how much approximation error occurs from CSEConv when SO(3) actions are acted on point cloud. Though the mathematical framework guarantees the equivariance of our model, the implementation is not perfectly equivariant due to the discretized integration or the locality assumption. This necessitates the verification of how strictly rotation-equivariant is our module. We adapted the experiment setups from [7] to prove the robustness of our method against SO(3) actions. We sampled $N = 500$ point cloud samples $\{\mathcal{P}_i\}_{i=1}^N$ and random rotation matrices $\{R_i\}_{i=1}^N$ as the dataset. The difference metric Δ from [7] is also adopted to measure the equivariance error as follows: $\Delta = \frac{1}{N}\sum_{i=1}^N \mathrm{std}(R_i\mathcal{M}(\mathcal{P}_i) - \mathcal{M}(R_i \cdot \mathcal{P}_i))/\mathrm{std}(\mathcal{M}(\mathcal{P}_i))$, where $\mathcal{M}(\cdot)$ is an output tensor and the standard deviation $\mathrm{std}(\cdot)$ is computed on whole elements of $\mathcal{M}(\mathcal{P}_i)$. It measures the variance of output by SO(3) actions while revising the absolute size of the metric identically.

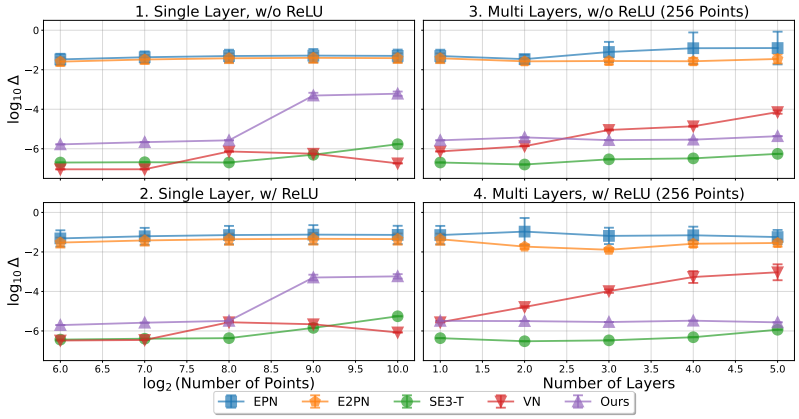

Fig. 4: The measurement of equivariance error Δ on rotation-equivariant models. We also denote the standard deviation of 500 samples composing Δ on each point as error bars. The number of points grows exponentially from 64 to 1024, and the number of layers increases from one to five. Beware that Δ is log-scaled.

We measured Δ on CSEConv and baseline rotation-equivariant methods. Among many works on learning group equivariant representation of point cloud, we choose methods applicable in layer-wise, including SE(3)-Transformer [14] (SE3-T), Vector Neurons [11] (VN), EPN [2], and E2PN [48]. Every model is randomly initialized and its dimension of output feature is configured to 10. Experiments are conducted by varying the number of points in a point cloud, the number of layers, and the usage of activation functions, as following 4 situations:

1. Single layer, increasing number of points.
2. Single layer with ReLU, increasing number of points.
3. Increasing number of layers, 256 points.
4. Increasing number of layers with ReLU, 256 points.

As shown in Figure 4, EPN and E2PN show higher equivariance error due to a group discretization. Conversely, our method maintains comparable error with strictly equivariant methods (VN, SE3-T) when the number of layers varies, though the error also has a correlation with the number of sample points. To confirm that integration approximation and locality assumption are factors in this tendency, we conduct an ablation experiment on the number of KNN neighbors in Figure 5. It shows that Δ asymptotically decreases as the neighborhood size is doubled, supporting the equivariance of CSEConv in probability.

Time and memory complexity of CSEConv: This section compares the cost of CSEConv with baselines in terms of time and memory consumption. The experiment gauge the memory and time cost of classification models on ModelNet40 [43] during the training and evaluation phases. The memory occupation in GPU and the average number of batches computed per second are measured, following the identical methods used in [48]. The only difference is that we had

Table 1: The time and memory efficiency table of baselines. We experiment with DGCNN [38]-variants of Vector Neurons (VN) and Frame Averaging [28] (FA).

Models (Batch Size = 12)		Ours	SE3-T	VN	FA	EPN	E2PN
Memory (GB)↓	train	**3.03**	8.50	6.10	15.19	13.40	3.94
	test	2.21	**0.25**	2.00	3.38	6.38	2.44
Speed (#batch/sec)↑	train	**27.46**	1.71	4.37	5.09	2.09	8.64
	test	**51.91**	5.75	9.09	12.04	3.30	17.72

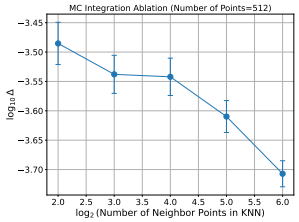

Fig. 5: The equivariance error ablation to the number of neighbor points.

fixed the size of mini-batches to 12 during both training and evaluation. Every experiment run on a single NVIDIA GeForce RTX 3090 GPU.

The result in Table 1 demonstrates the superior efficiency of CSEConv compared to group equivariant baselines. The memory consumption of CSEConv is the most efficient during the training phase. Especially, its time complexity exceeds existing methods multiple times faster. This is possible due to our filter function being realized as small-sized weight-sharing neural networks instead of large tensor with gradient operations.

5.2 Object classification experiments

We compare CSEConv with baselines for point cloud analysis through the ModelNet40 [43] and ScanObjectNN [36] classification tasks, standard benchmarks experimented in multiple roto-translation equivariant research. ModelNet40 is composed of synthetic point clouds with 40 classes, and ScanObjectNN, in contrast, comprises point clouds sampled from real-world scans. During training, parameters that achieve the best performance in validation under SO(3) actions are selected for evaluation in every method.

Classification of ModelNet40: Adopting the configuration from [48], we evaluate point cloud models in 4 situations whether random SO(3) augmentation is conducted during the training or test phase. However, we do not augment point clouds with noise processes such as jittering, translation, and random dropout. We choose both equivariant, introduced formerly at Section 5.1, and non-equivariant methods, such as PointNet++ [31], KPConv [33], and Point Cloud Transformer [17] (PCT), as baselines.

The result of every model and case for ModelNet40 classification task is in Table 2. The most notable result is that CSEConv outperforms every baseline when SO(3) augmentation is applied only during at the evaluation. It is apparent that non-equivariant models suffer drastic degradation in such situations, and they also undergo a performance decline when augmented with respect to an aligned evaluation set, as aforementioned in Section 1.

Meanwhile, equivariant methods maintain robust performance under SO3 actions, except for EPN and E2PN. These methods, which discretize the group

Table 2: The total accuracy (%) of baselines for ModelNet40 classification task. I: no rotations applied, SO(3): applying random SO(3) rotations.

Augmentation	Continuous Group				Discrete Group		Non-Equivariant		
Train/Test	Ours	VN	FA	SE3-T	EPN	E2PN	PointNet++	KPConv	PCT
I/I	83.79	78.27	82.25	71.60	91.45	**91.58**	89.13	91.25	91.25
I/SO(3)	**83.75**	78.27	82.25	71.60	31.08	44.47	9.58	14.56	15.84
SO(3)/I	83.83	78.23	82.01	73.01	86.60	**89.47**	81.16	84.84	86.39
SO(3)/SO(3)	83.75	78.23	82.01	73.01	86.93	**88.58**	80.29	83.39	84.20

Table 3: The total accuracy (%) of equivariant baselines for ScanObectNN classification task. Same notations in Table 2 are used to designate augmentation conditions.

Equivariance Type	Model	I/I	I/SO(3)	SO(3)/I	SO(3)/SO(3)
Discrete Group	EPN	71.87	47.07	74.37	73.37
	E2PN	**82.70**	60.70	**78.53**	**77.03**
Continuous Group	VN	56.92	56.82	59.39	59.39
	FA	71.94	71.94	70.05	70.05
	SE3-T	59.21	59.21	61.27	61.27
	Ours	<u>72.29</u>	**72.29**	70.40	70.40

space, undergo deterioration over 40pp by SO(3) rotations when augmentation is not applied. Since they are reported to reach an accuracy over 90% against rotations sampled from the discretized SO(3) [48], the bias toward discrete group is inevitable in such methods. In contrast, CSEConv maintains consistent performance in every case thanks to its equivariance. It also achieves the best performance among continuous group equivariant methods, comparable to the accuracy of non-equivariant models trained with SO(3) augmentation.

Classification of ScanObjectNN: This experiment is conducted only with equivariant methods to compare their scalability on a real-world dataset. We minimize the difference in model architecture to the one for ModelNet40 experiment, only fine-tuning the model hyperparameter and optimizer configuration. Though ScanObjectNN provides diverse variants with respect to noisiness, we only used the **OBJ_ONLY** variant where only object related points are segmented from the original point cloud scan.

The result in Table 3 is mostly consistent with ModelNet40 experiment, but the performance gap between baselines narrows down than before. Since the dataset domain is shifted to noisy real-world and model configurations are not altered much, every model undergoes an accuracy decrease. Nonetheless, CSEConv still achieves the best performance among continuous group equivariant methods and reaches second place in non-augmented setup (I/I). These outcomes support the scalability of our method, proving its potential to be applicable to real-world rotation-invariant tasks.

Table 4: The mean average precision (mAP) of baselines for ModelNet40 retrieval.
†: mAP from [2], obtained by augmentation during training.

Models	Retrieval (mAP)
Ours	**79.66**
PointNet++	70.3†
KPConv	77.5†
EPN	**79.7**†

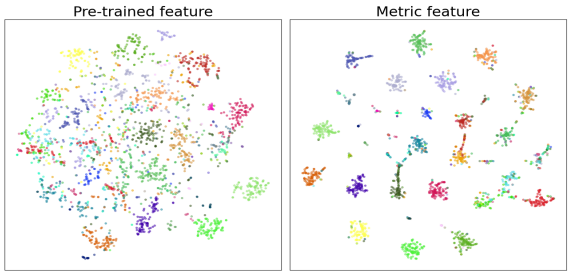

Fig. 6: The t-SNE visualization of ModelNet40 features before and after mapping into metric space.

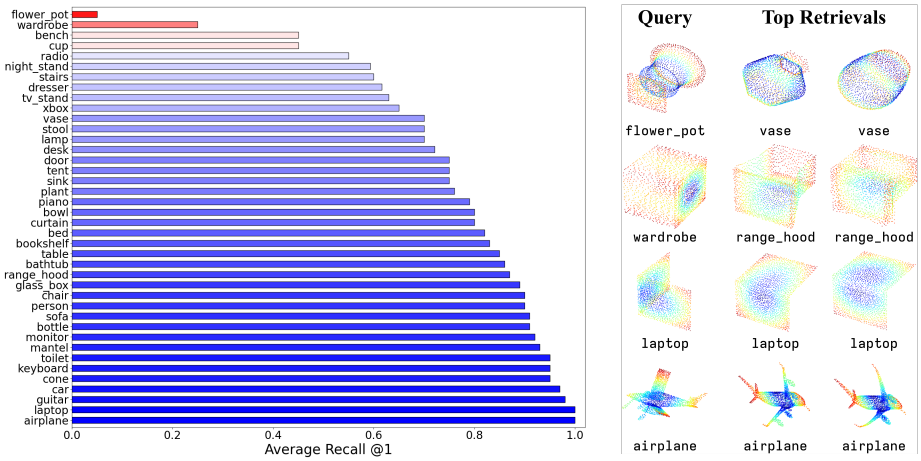

Fig. 7: The visualization of retrieval model performance. (a) The average recall at top-1 measured per class. (b) The examples of top-2 worst (`flower_pot`, `wardrobe`) and best (`airplane`, `laptop`) classes retrieved by ours.

5.3 Object retrieval experiment

We conduct the retrieval task on ModelNet40 by following [2, 43]. First, a retrieval model maps every point cloud in the test set into metrizable vectors. Then these vectors are distanced from every other vector, and we retrieve the top-k closest vectors for each query. While piling metrizable vectors for the evaluation, every point cloud is augmented by random SO(3) actions. However, such augmentation is not applied during the training of our method, in contrast to the baselines that utilized the augmented dataset. We also visualize features before and after the last MLP using t-SNE and assess the learned metric space qualitatively. As shown in Figure 6, the learned metric space vividly separates features from encoders into clusters, enabling features more metrizable with a L2 distance.

The measured mean average precision (mAP) of baselines is in Table 4, which demonstrates that our method outperforms the non-equivariant methods trained with the augmentation, reaching the comparable performance to EPN. In Figure 7a, we also evaluate the average recall at top-1 per every object class. It implies that the most of performance degradation occurs from a few classes. To verify such false positive cases, we visualize query and retrieved point clouds from top-2 best and worst classes in Figure 7b. It reveals that our method proficiently classifies objects with globally distinct shapes but struggles to learn vague differences between similar classes, such as `flower_pot` and `vase`.

6 Discussions and Limitations

To summarize results in Section 5, CSEConv achieves the decent performance in rotation-invariant tasks, comparable to non-equivariant methods leveraging the rotation augmentation. It verifies again the shortcomings of reliance on the 3D rotation augmentation. The proposed method also indicates its superior scalability amidst the adequately equivariant methods thanks to its performance and exceptional efficiency. Thus we claim that CSEConv has the potential to replace existing methods in rotation-invariant tasks without any augmentation.

In spite of such scalability, CSEConv also reveals its current limitations. Our method is relatively deficient in learning features variable to local or subtle geometries, as shown in Figure 7b. This may explains its insufficient model capacity compared to group discretization or non-equivariant methods when every data orientation is biased. Besides, our work only deals with rotation-invariant tasks so far, since it defines group actions on the feature space as identity for simplicity. However, it misses other significant rotation-equivariant tasks such as normal or pose estimations. Manifesting group actions applicable in the arbitrary feature space could be focal points for future equivariance studies.

7 Conclusion

This work suggests CSEConv, which is a convolution network formulated on S^2 and equivariant towards continuous SO(3). The mathematical structure of CSEConv is founded on the group theory framework, which provides an important constraint to the filter for maintaining SO(3) equivariance. Upon the theoretical support, we implement the weight-sharing neural network which maps coordinate differences in the *double coset* to filter values. The comparative analyses empirically show that CSEConv is equivariant to continuous SO(3) and surpasses other equivariant methods with respect to time efficiency. Moreover, our method achieves the best performance among the methods equivariant to continuous group, performing equivalently to models exploiting the rotation augmentation in benchmark rotation-invariant tasks. Though it still has drawbacks in model capacity and task diversity, we propose the prospect of CSEConv in substituting the reliance on rotation augmentation with our method.

Acknowledgements

This work was partly supported by the IITP (RS-2021-II212068-AIHub/10%, RS-2021-II211343-GSAI/15%, 2022-0-00951-LBA/15%, 2022-0-00953-PICA/20%), NRF (RS-2024-00353991-SPARC/20%, RS-2023-00274280/10%), and KEIT (RS-2024-00423940/10%) grant funded by the Korean government.

References

1. Cesa, G., Lang, L., Weiler, M.: A program to build e(n)-equivariant steerable CNNs. In: International Conference on Learning Representations (2022)
2. Chen, H., Liu, S., Chen, W., Li, H., Hill, R.: Equivariant point network for 3d point cloud analysis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14514–14523 (2021)
3. Chen, X., Läbe, T., Milioto, A., Röhling, T., Vysotska, O., Haag, A., Behley, J., Stachniss, C.: OverlapNet: Loop Closing for LiDAR-based SLAM. In: Proceedings of Robotics: Science and Systems (RSS) (2020)
4. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3075–3084 (2019)
5. Cohen, T., Weiler, M., Kicanaoglu, B., Welling, M.: Gauge equivariant convolutional networks and the icosahedral cnn. In: International conference on Machine learning. pp. 1321–1330. PMLR (2019)
6. Cohen, T., Welling, M.: Group equivariant convolutional networks. In: International conference on machine learning. pp. 2990–2999. PMLR (2016)
7. Cohen, T.S., Geiger, M., Köhler, J., Welling, M.: Spherical cnns. In: International Conference on Learning Representations (2018)
8. Cohen, T.S., Geiger, M., Weiler, M.: Intertwiners between induced representations (with applications to the theory of equivariant neural networks). arXiv preprint arXiv:1803.10743 (2018)
9. Cohen, T.S., Geiger, M., Weiler, M.: A general theory of equivariant cnns on homogeneous spaces. Advances in neural information processing systems **32** (2019)
10. Cohen, T.S., Welling, M.: Steerable cnns. arXiv preprint arXiv:1612.08498 (2016)
11. Deng, C., Litany, O., Duan, Y., Poulenard, A., Tagliasacchi, A., Guibas, L.J.: Vector neurons: A general framework for so (3)-equivariant networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12200–12209 (2021)
12. Finzi, M., Stanton, S., Izmailov, P., Wilson, A.G.: Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In: International Conference on Machine Learning. pp. 3165–3176. PMLR (2020)
13. Folland, G.B.: A course in abstract harmonic analysis, vol. 29. CRC press (2016)
14. Fuchs, F., Worrall, D., Fischer, V., Welling, M.: Se (3)-transformers: 3d roto-translation equivariant attention networks. Advances in neural information processing systems **33**, 1970–1981 (2020)
15. Gerken, J., Carlsson, O., Linander, H., Ohlsson, F., Petersson, C., Persson, D.: Equivariance versus augmentation for spherical images. In: International Conference on Machine Learning. pp. 7404–7421. PMLR (2022)
16. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
17. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: Point cloud transformer. Computational Visual Media **7**, 187–199 (2021)

18. Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M.: Deep learning for 3d point clouds: A survey. IEEE transactions on pattern analysis and machine intelligence **43**(12), 4338–4364 (2020)
19. Ioannidou, A., Chatzilari, E., Nikolopoulos, S., Kompatsiaris, I.: Deep learning advances in computer vision with 3d data: A survey. ACM computing surveys (CSUR) **50**(2), 1–38 (2017)
20. Kim, G., Kim, A.: Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4802–4809. IEEE (2018)
21. Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D.L., Han, S.: Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 2774–2781. IEEE (2023)
22. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
23. Luo, S., Li, J., Guan, J., Su, Y., Cheng, C., Peng, J., Ma, J.: Equivariant point cloud analysis via learning orientations for message passing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18932–18941 (2022)
24. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 922–928. IEEE (2015)
25. Müller, P., Golkov, V., Tomassini, V., Cremers, D.: Rotation-equivariant deep learning for diffusion mri. arXiv preprint arXiv:2102.06942 (2021)
26. van der Ouderaa, T., Romero, D.W., van der Wilk, M.: Relaxing equivariance constraints with non-stationary continuous filters. Advances in Neural Information Processing Systems **35**, 33818–33830 (2022)
27. van der Ouderaa, T.F., van der Wilk, M.: Sparse convolutions on lie groups. In: NeurIPS Workshop on Symmetry and Geometry in Neural Representations. pp. 48–62. PMLR (2023)
28. Puny, O., Atzmon, M., Smith, E.J., Misra, I., Grover, A., Ben-Hamu, H., Lipman, Y.: Frame averaging for invariant and equivariant network design. In: International Conference on Learning Representations (2022)
29. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
30. Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L.J.: Volumetric and multi-view cnns for object classification on 3d data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5648–5656 (2016)
31. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)
32. Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 7537–7547. Curran Associates, Inc. (2020)
33. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6411–6420 (2019)

34. Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., Riley, P.: Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. arXiv preprint arXiv:1802.08219 (2018)
35. Ummenhofer, B., Prantl, L., Thuerey, N., Koltun, V.: Lagrangian fluid simulation with continuous convolutions. In: International Conference on Learning Representations (2020)
36. Uy, M.A., Pham, Q.H., Hua, B.S., Nguyen, T., Yeung, S.K.: Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1588–1597 (2019)
37. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
38. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (tog) **38**(5), 1–12 (2019)
39. Weiler, M., Geiger, M., Welling, M., Boomsma, W., Cohen, T.S.: 3d steerable cnns: Learning rotationally equivariant features in volumetric data. Advances in Neural Information Processing Systems **31** (2018)
40. Winkels, M., Cohen, T.S.: 3d g-CNNs for pulmonary nodule detection. In: Medical Imaging with Deep Learning (2018)
41. Worrall, D., Brostow, G.: Cubenet: Equivariance to 3d rotation and translation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 567–584 (2018)
42. Wu, W., Qi, Z., Fuxin, L.: Pointconv: Deep convolutional networks on 3d point clouds. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 9621–9630 (2019)
43. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1912–1920 (2015)
44. Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., Lu, J.: Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19313–19322 (2022)
45. Zhang, Z., Hua, B.S., Chen, W., Tian, Y., Yeung, S.K.: Global context aware convolutions for 3d point cloud understanding. In: 2020 International Conference on 3D Vision (3DV). pp. 210–219. IEEE (2020)
46. Zhang, Z., Hua, B.S., Rosen, D.W., Yeung, S.K.: Rotation invariant convolutions for 3d point clouds deep learning. In: 2019 International conference on 3d vision (3DV). pp. 204–213. IEEE (2019)
47. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 16259–16268 (2021)
48. Zhu, M., Ghaffari, M., Clark, W.A., Peng, H.: E2pn: Efficient se (3)-equivariant point network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1223–1232 (2023)