Neurocomputing 660 (2026) 131832

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Equivariant feature extraction: Enhancing 3D point cloud analysis with robust rotational equivariance

Qianwei Tang [a, b], Baile Xu [a, b], Jian Zhao [a, c], Furao Shen [a, b, *]

[a] National Key Laboratory for Novel Software Technology, Nanjing University, 163 Xianlin Avenue, Nanjing, 210023, Jiangsu, China
[b] School of Artificial Intelligence, Nanjing University, 163 Xianlin Avenue, Nanjing, 210023, Jiangsu, China
[c] School of Electronic Science and Engineering, Nanjing University, 163 Xianlin Avenue, Nanjing, 210023, Jiangsu, China

HIGHLIGHTS

- We propose a novel and generalizable model for extracting $SO(3)$ equivariant features. The model uses a small number of learnable parameters and can be easily integrated into classic models, endowing them with equivariance while maintaining computational efficiency.
- We introduce group representations and spherical harmonics into the model, and design a series of equivariant operators, providing a new perspective for the future development of equivariant feature extraction in 3D data processing.
- We conduct experiments and demonstrate that, by adopting our model, classic models can achieve equivariance and perform excellently on rotated test sets, significantly enhancing their robustness to rotational transformations.

ARTICLE INFO

Communicated by B. Fan

Keywords:
Point cloud
Rotation equivariance
3D vision
Representation learning

ABSTRACT

Recent advancements in 3D point cloud representation techniques have significantly facilitated research in downstream tasks such as classification and segmentation. Despite the impressive capabilities of existing methods, they often struggle with rotated data due to a lack of rotation equivariance. This paper introduces the Equivariant Feature Extractor (EFE), a feature extraction method based on equivariant representations and spherical harmonics. EFE encodes 3D position data using equivariant group representations and extracts high-quality equivariant features through a series of equivariant operators. This method can be seamlessly integrated into classic neural networks, ensuring rotation equivariance while introducing only a small number of learnable parameters. Experimental results demonstrate that integrating EFE into mainstream point cloud models achieves outstanding performance, particularly on rotated test sets. This study highlights EFE's ability to extract equivariant features and its direct applicability to classic models, contributing significantly to improving downstream task performance and advancing point cloud feature extraction techniques.

1. Introduction

In recent years, the rapid development of high-precision sensors, such as LiDAR [1] and Kinect [2], has revolutionized the acquisition of 3D data, making it more accessible and accurate than ever before. This technological advancement has enabled the widespread application of 3D data across diverse fields [3], including autonomous driving, robotics, augmented reality, and 3D reconstruction [4,5]. As a commonly used 3D data representation, point clouds offer exceptional spatial expressiveness due to their ability to directly capture the geometric structure of objects in a discrete yet detailed manner. Consequently, point clouds have emerged as a primary data format for representing the 3D world and a critical research tool in 3D graphics tasks. This has spurred a surge in related studies, with point cloud classification and segmentation standing out as particularly popular and active areas of research [6]. These tasks are foundational for enabling machines to understand and interact with 3D environments, paving the way for advancements in scene understanding and object recognition.

Recent models, such as PointNet [7] and DGCNN [8], have leveraged convolutional operations to extract point cloud features, effectively addressing the challenge of permutation invariance inherent in point

* Corresponding author at: National Key Laboratory for Novel Software Technology, Nanjing University, 163 Xianlin Avenue, Nanjing, 210023, Jiangsu, China.
Email addresses: qweit@smail.nju.edu.cn (Q. Tang), xubaile@nju.edu.cn (B. Xu), jianzhao@nju.edu.cn (J. Zhao), frshen@nju.edu.cn (F. Shen).

https://doi.org/10.1016/j.neucom.2025.131832
Received 2 July 2025; Received in revised form 27 September 2025; Accepted 15 October 2025
Available online 16 October 2025

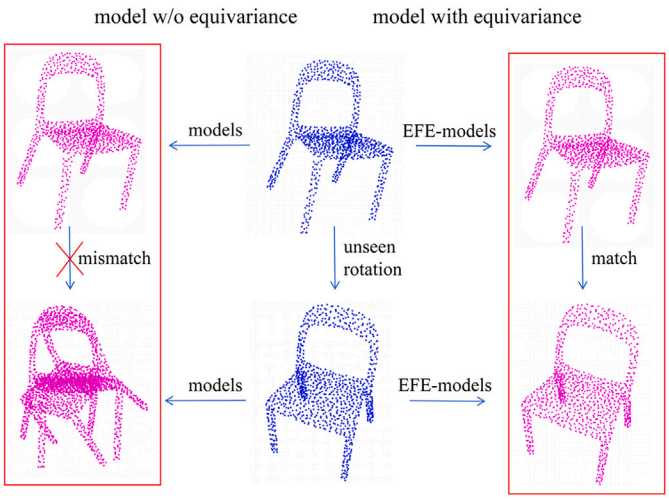


Fig. 1. Diagram of equivariance. The blue represents the original and rotated data, while the purple indicates the output of these data through the model. Models with equivariance perform well with rotated data, maintaining consistency in feature representation, while models lacking equivariance show significantly poorer performance under rotational transformations.

cloud data. These methods have achieved remarkable success in generating high-quality features for various downstream tasks. Moreover, E-CNN [9] enhances performance by integrating image preprocessing with an improved SqueezeNet architecture. S^2ANet [10] addresses the limitation of insufficient local feature extraction in point cloud processing by fusing spectral and spatial domain features. SparseVoxNet [11] improves performance through sparsely connected 3D dense blocks, surface feature extraction, and a hybrid CNN-RF model. However, despite their strengths, the features extracted by these approaches often exhibit limited generalization ability and robustness, particularly when subjected to rotational transformations. This limitation arises from the lack of $SO(3)$ invariance or equivariance, which are critical properties for ensuring consistent performance across different orientations of 3D data. Fundamentally, this is due to the use of conventional neural network components—such as linear layers and activation functions—that do not possess rotational equivariance. Models with equivariance exhibit stability under rotation operations, as their feature representations transform predictably with the input data. Invariance, a special case of equivariance, ensures that features remain unchanged under transformations, and can be easily derived from equivariant features, for instance, by taking the modulus or applying equivariant linear layers. This study primarily focuses on $SO(3)$ equivariance in Euclidean space, a property closely tied to the geometric nature of point cloud data. As illustrated in Fig. 1, models with equivariance maintain consistent performance on rotated data, while those lacking equivariance suffer significant performance degradation, often leading to reduced accuracy in tasks like classification and segmentation.

Several recent point cloud studies have explored equivariant methods for feature extraction and network construction to address these challenges [12,13]. These approaches typically employ low-order equivariant representations to capture rotational symmetries in 3D data. However, such representations are often insufficient to fully capture the intricate geometric properties of point clouds, resulting in features with poor semantic quality and low accuracy in downstream tasks. Furthermore, these methods struggle to integrate effectively with existing point cloud models, limiting their practical applicability and compatibility with well-established frameworks.

To address the aforementioned issues, we draw inspiration from the concept of equivariant group representations in physics, utilizing spherical harmonics to encode 3D directional vectors and related information into high-dimensional spaces. This approach enables us to obtain robust equivariant representations and further derive high-quality equivariant features for point cloud data. In this work, we define l as the degree of the spherical harmonics used for encoding, which corresponds to the order of the group representation. The parameter l serves as a hyperparameter that controls the complexity and expressiveness of the encoded features, allowing for flexible adaptation to different tasks and datasets. Previous works by Dym and Maron [14], as well as Joshi [15], have demonstrated that applying the Clebsch-Gordan (CG) tensor product [16] on higher-order group representations ($l > 1$) can significantly enhance feature expressiveness. Building on this insight, our study adopts this approach to extract more distinctive geometric features from point clouds, thereby improving their utility in downstream tasks such as classification and segmentation.

This paper introduces an Equivariant Feature Extractor (EFE), a model meticulously designed to extract equivariant features of points based on strict $SO(3)$ equivariance properties. The extracted equivariant features can be seamlessly integrated into classic neural networks, enabling their application to a wide range of downstream tasks across various scenarios, including object recognition and scene understanding. We conducted extensive experiments on traditional point cloud classification and segmentation tasks using benchmark datasets ModelNet40 [17] and ShapeNet [18], and compared the results with previous equivariant methods. The experimental results demonstrate that our approach achieves outstanding performance while maintaining excellent compatibility with conventional methods, thus providing a practical and efficient solution for point cloud processing. We summarize the main contributions of this work as follows:

- We propose a novel and generalizable model for extracting $SO(3)$ equivariant features. The model uses a small number of learnable parameters and can be easily integrated into classic models, endowing them with equivariance while maintaining computational efficiency.
- We introduce group representations and spherical harmonics into the model, and design a series of equivariant operators, providing a new perspective for the future development of equivariant feature extraction in 3D data processing.
- We conduct experiments and demonstrate that, by adopting our model, classic models can achieve equivariance and perform excellently on rotated test sets, significantly enhancing their robustness to rotational transformations.

2. Related works

Deep learning architectures such as PointNet [7], PointNet++ [19], DGCNN [8], PointCNN [20], and others have achieved promising results in point cloud processing tasks. However, these classic methods lack rotational robustness, which has sparked interest in the research of rotation-invariant and equivariant approaches. In recent years, deep learning methods targeting rotation-invariant and equivariant 3D geometric features have developed rapidly. The following subsections will briefly review these methods.

2.1. Rotation-equivariant methods

Rotation equivariance not only provides rotational robustness to the model, ensuring that the extracted features are unaffected by rotational transformations, but also allows the observation of transformations in high-dimensional coordinates based on 3D transformations, offering a broader application scope for future 3D tasks. For example, after applying a rotation R to the point cloud coordinates in three-dimensional space, an equivariant model extracts the original high-dimensional features from the untransformed data, while from the rotated data it extracts features that are transformed by a corresponding high-dimensional rotation R'. The low-dimensional rotation R and the high-dimensional rotation R' are mutually corresponding, and we

will provide a formal definition of this relationship in the subsequent sections.

Currently, there are relatively few equivariant methods applied to point cloud data. TFN [21] is a pioneering work in the molecular domain that applies spherical harmonics and Clebsch-Gordan (CG) tensor products [16] to achieve $SO(3)$ equivariance in models. Following this, a series of improved methods based on these two tools have emerged, such as SE(3)-Transformer [22] and Equiformer [23]. However, these methods are computationally expensive and difficult to integrate into classic point cloud models. A noteworthy work is Vector Neurons [24], which employs vector neurons and rewrites common neural network functions to induce rotation equivariance, making it easier to transfer to traditional models. Additionally, OrientedMP [25] introduces a novel equivariant message-passing framework by learning point-wise orientations, decoupling global rotations, and achieving competitive performance in point cloud analysis and physical modeling tasks.

2.2. Rotation-invariant methods

Rotation equivariance means that when an object in Euclidean space is rotated, its corresponding features in the feature space also rotate in a consistent way. Rotation invariance can be seen as a special case of equivariance, where the features do not change at all under rotation. In other words, equivariant features preserve information about how the object is rotated, while invariant features discard this information and only retain properties that remain the same under rotation. For example, in molecular property prediction tasks, the predicted molecular energy (a scalar) should be rotation-invariant, since its value must remain unchanged regardless of the molecule's orientation. In contrast, the predicted molecular forces (vectors) should be rotation-equivariant, as their directions change consistently with the rotation of the molecule. Furthermore, invariant features can be readily derived from equivariant ones, for instance, by applying equivariant layers to reduce dimensionality or by directly computing the vector norm. In practical applications, rotation equivariance or invariance is indispensable, as standard neural networks are not inherently robust to rotations. Ideally, data that differ only by a rotational transformation should yield consistent outputs; however, networks lacking rotation equivariance often produce inconsistent predictions with degraded performance.

In recent years, numerous rotation-invariant approaches have emerged. GC-Conv [26] relies on PCA-based multiscale reference frames to construct rotation invariance. RI-Framework [27] and LGR-Net [28] pair local invariant information with global context. RIConv [29] achieves stable rotation-invariant representations by considering the relationships between key points and their neighbors, as well as the intrinsic connections among neighboring points, utilizing local reference axes. Furthermore, PaRot [30] proposes a patch-wise rotation-invariant network that disentangles shape content and orientation features using Siamese training, embedding geometric relations to restore relative pose information, enhancing rotation-invariant learning for classification and segmentation.

3. Preliminary

In this section, we provide a concise introduction to equivariance and invariance, followed by a brief overview of the mathematical tools used in our model to achieve $SO(3)$ equivariance for point cloud analysis.

3.1. Equivariance and invariance

For a group G, a function $f : X \to Y$ is equivariant under G if the following holds:

$$f(D_X(g)x) = D_Y(g)f(x), \forall g \in G, x \in X, \tag{1}$$

where $D_X(g)$ and $D_Y(g)$ are the representations of the group element g acting on the input space X and output space Y, respectively. In simpler terms, equivariance means that applying a transformation g to the input x and then computing f yields the same result as first computing $f(x)$ and then applying the corresponding transformation to the output.

Invariance is a particular and straightforward case of equivariance. Rotation invariant methods aim to produce the same or closely similar results for inputs with different poses:

$$f(x) = f(D_X(g)x), \forall g \in G, x \in X. \tag{2}$$

In this paper, we focus on $SO(3)$ equivariance, which pertains to 3D rotations—a critical property for point clouds, as their geometric features often need to transform consistently under rotations.

3.2. Irreducible representations

For a group element $g \in SO(3)$, its irreducible representations (ireps) are given by $(2l+1) \times (2l+1)$ Wigner-D matrices $D^l(g)$, which act on $(2l+1)$-dimensional vector spaces, where $l \geq 0$ is the degree of the representation. Vectors in these spaces are called type-l vectors; for example, scalars are type-0 vectors, and Euclidean vectors are type-1 vectors. Each type-l vector has $2l+1$ components, indexed by the order m, where $-l \leq m \leq l$, representing different orientations within the vector space.

To form $SO(3)$-equivariant features, we concatenate multiple type-l vectors. For each degree l (where $0 \leq l \leq l_{\max}$), we include C_l type-l vectors, with C_l denoting the number of channels for that degree. Features across different channels of the same degree l are parameterized by distinct weights but transform identically under the same Wigner-D matrix $D^l(g)$.

3.3. Spherical harmonics

Spherical harmonics $Y^l_m : S^2 \to \mathbb{R}$ map points on the unit sphere S^2 to real-valued scalars, where $l \geq 0$ is the degree and $-l \leq m \leq l$ is the order. These functions are used to project 3D Euclidean vectors into $SO(3)$-equivariant type-l vectors while maintaining strict equivariance [31,32]. For a Euclidean vector $\vec{r} \in \mathbb{R}^3$, we compute a type-l vector f^l as $f^l = Y^l(\frac{\vec{r}}{||\vec{r}||})$, where $Y^l(\frac{\vec{r}}{||\vec{r}||}) = [Y^l_{-l}(\frac{\vec{r}}{||\vec{r}||}), Y^l_{-l+1}(\frac{\vec{r}}{||\vec{r}||}), \dots, Y^l_l(\frac{\vec{r}}{||\vec{r}||})]$ is a $(2l+1)$-dimensional vector. The structure of this sequence of vectors is illustrated in Fig. 2. This projection satisfies $SO(3)$ equivariance:

$$D^l(g)f^l = Y^l\left(\frac{D^1(g)\vec{r}}{||D^1(g)\vec{r}||}\right). \tag{3}$$

We use real-valued spherical harmonics for computational convenience. In our model, we apply spherical harmonics to relative positions $\vec{r}_{ij}$

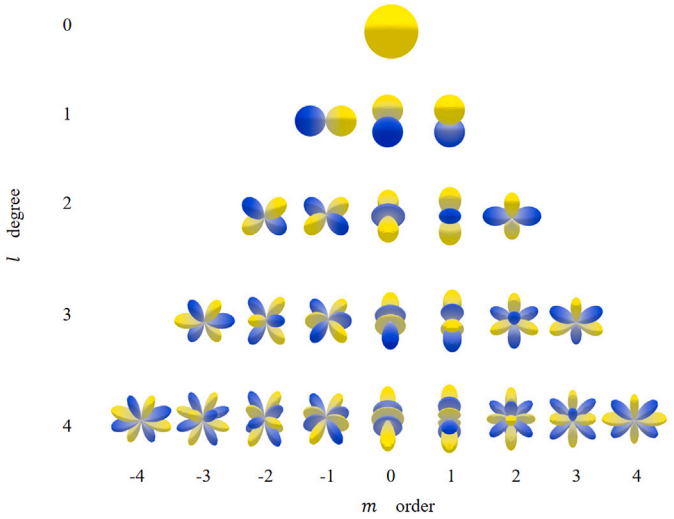


Fig. 2. Illustration of spherical harmonics up to $l = 4$.

between points to generate initial irreps features, which are then propagated through equivariant operators like tensor products.

3.4. Tensor product

The tensor product [21] is a key operation for combining $SO(3)$ representations in an equivariant manner. For type-l_1 vector f^{l_1} and type-l_2 vector g^{l_2}, their tensor product produces a type-l vector h^l using Clebsch-Gordan (CG) coefficients:

$$h^l_m = (f^{l_1} \otimes g^{l_2})_m = \sum_{m_1=-l_1}^{l_1} \sum_{m_2=-l_2}^{l_2} C^{(l,m)}_{(l_1,m_1),(l_2,m_2)} f^{l_1}_{m_1} g^{l_2}_{m_2}, \tag{4}$$

where $\otimes$ represents the tensor product, m_1 and m_2 index the components of f^{l_1} and g^{l_2}, respectively, and $C^{(l,m)}_{(l_1,m_1),(l_2,m_2)}$ are CG coefficients, which are non-zero only when $|l_1 - l_2| \le l \le |l_1 + l_2|$.

Each distinct non-trivial combination under the tensor product $l_1 \otimes l_2 \to l$ is referred to as a path, and each path independently preserves equivariance. The triangle inequality $|l_1 - l_2| \le l \le |l_1 + l_2|$ must be satisfied. For example, when $l_1 = 1$ and $l_2 = 0$, the tensor product $l_1 \otimes l_2$ can only yield type-1 features, since $|1 - 0| \le l = 1 \le |1 + 0|$. In contrast, if $l_1 = 1$ and $l_2 = 1$, the tensor product can produce type-0, type-1, and type-2 features, because $|1 - 1| \le l = 0, 1, 2 \le |1 + 1|$. We assign a learnable weight to each path, enabling the tensor product to be extended to irreducible representation features with multiple channels of type-l vectors, thereby facilitating flexible and equivariant feature interactions. These weights can also be conditioned on quantities such as relative distances.

Now consider the tensor product of two tensors x and y, where x consists of two scalars (with $l = 0$, 1-dimensional) and three directional vectors (with $l = 1$, 3-dimensional), i.e., $x := 2 * 0e + 3 * 1e$. y, on the other hand, consists of five scalars (with $l = 0$, 1-dimensional), seven directional vectors (with $l = 1$, 3-dimensional), and one type-2 vector (with $l = 2$, 5-dimensional), i.e., $y := 5 * 0e + 7 * 1e + 1 * 2e$. Here, the symbol e serves as a formal notation representing an irrep element, and it also indicates that the tensor products in our work are considered only over the real domain.

The fully connected tensor product can be represented as:

$$2 * 0e + 3 * 1e \otimes 5 * 0e + 7 * 1e + 1 * 2e \to 31 * 0e + 53 * 1e + 26 * 2e + 3 * 3e, \tag{5}$$

with the computational **path** shown in Fig. 3 and Table 1. Based on this result, the fully connected tensor product of x and y, denoted as $x \otimes y$, is composed of 31 scalars (with $l = 0$, 1-dimensional), 53 directional vectors (with $l = 1$, 3-dimensional), 26 type-2 vectors (with $l = 2$, 5-dimensional), and 3 Type-3 vectors (with $l = 3$, 7-dimensional). By selecting specific degrees (l) and unifying the number of channels (C_l), the equivariant result can be uniformly represented, ensuring consistent dimensionality for the features of each point in the 3D point cloud. As long as the tensor product is computed along this correct path, in accordance with physical principles [16], the process guarantees equivariance across different channels for each order l.

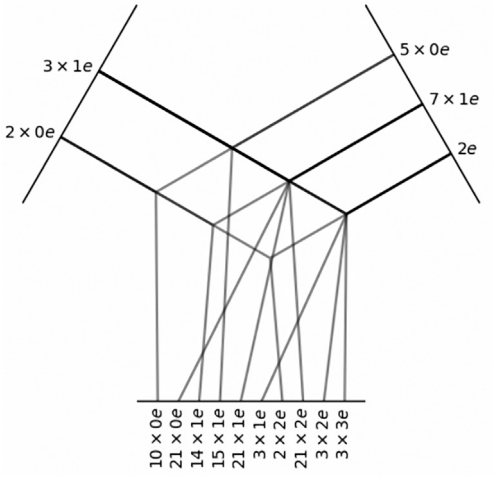


Fig. 3. The CG tensor product path diagram for $x(l_1 = 0, 1)$ and $y(l_2 = 0, 1, 2)$. The path is controlled by the output degree l, which must satisfy the inequality $|l_1 - l_2| \le l \le (l_1 + l_2)$.

Table 1
Decomposition of the tensor product $2 * 0e + 3 * 1e \otimes 5 * 0e + 7 * 1e + 1 * 2e$. For each pair (l_1, l_2), the valid output degrees l are determined by the triangle inequality $|l_1 - l_2| \le l \le l_1 + l_2$, and the contribution to each l is given by the product of channel counts $C_{l_1} \times C_{l_2}$.

l_1	C_{l_1} (in x)	l_2	C_{l_2} (in y)	Valid l	Contribution
0	2	0	5	$l = 0$	$2 \times 5 = 10$
0	2	1	7	$l = 1$	$2 \times 7 = 14$
0	2	2	1	$l = 2$	$2 \times 1 = 2$
1	3	0	5	$l = 1$	$3 \times 5 = 15$
1	3	1	7	$l = 0, 1, 2$	$3 \times 7 = 21$ each
1	3	2	1	$l = 1, 2, 3$	$3 \times 1 = 3$ each

4. Method

In this section, we introduce the structure of the Equivariant Feature Extractor (EFE), a model designed to extract robust and expressive $SO(3)$-equivariant features from 3D point clouds. EFE takes 3D spatial coordinates as input, performs point embedding, and processes directional information using spherical harmonics to project the data into a high-dimensional equivariant space. After passing through K Equivariant Blocks with skip connections, the resulting features can be transformed into invariant features through equivariant linear layers or by taking the norm. These features are then aligned dimensionally for integration into classic point cloud models.

4.1. Embedding

For each sampled point p_i from the input point cloud $p \in \mathbb{R}^{N \times 3}$, where N denotes the number of sampled points, we construct a series of type-l vectors as the initial embedding x_0, following the rules of spherical harmonics. It should be noted in advance that, under the hyperparameter l_{max}, the feature x_K at the K-th layer of the network is obtained by applying K successive Equivariant Blocks to the initial embedding x_0. The resulting feature has dimensionality $N \times \sum_{l=0}^{l_{max}} [(2 * l + 1) \times C_l]$, where C_l denotes the number of channels associated with order l. Since for the maximum degree l_{max} and the spherical harmonic rule $0 \le l \le l_{max}$, each type-l vector consists of $(2l + 1)$ basis functions, and each basis function vector can be further assigned C_l channels. Therefore, the overall feature dimensionality becomes $N \times \sum_{l=0}^{l_{max}} [(2 * l + 1) \times C_l]$.

At initialization, we set the hyperparameter $l_{max} = 1$. This implies that the embedding consists of one type-0 vector and one type-1 vector, resulting in a dimensionality of $N \times (1 \times C_0 + 3 \times C_1)$. The type-0 vector is initialized as an all-ones tensor of shape $N \times 1 \times C_0$, while the type-1 vector is initialized as a tensor of shape $N \times 3 \times C_1$, where the second dimension is filled with the local normal vector of the neighborhood around each point and then broadcasted. The local normal vector is computed as follows: for each point, we first select its m nearest neighbors in terms of Euclidean distance (m is set to 20 by default to align with subsequent experimental configurations). We then compute the covariance matrix of these neighboring points and take the eigenvector corresponding to the smallest eigenvalue as the local normal vector of the point, which has a dimensionality of 3. The resulting tensor of shape $N \times 3$ is subsequently broadcast along the channel dimension by replicating it C_1 times, yielding the final initialization of the type-1 vector.

Finally, for each point p_i, we identify its neighboring points p_j according to their Euclidean distances. The criteria for selecting neighboring points and defining neighborhoods follow the same conventions as in the embeddings described earlier. Initially, the m points closest in Euclidean space are chosen (with m set to 20 by default). During subsequent feature update steps, however, the m nearest points are selected based on feature space distances, following the approach used in DGCNN [8]. Let the embedding of p_i be denoted as $x_{i,0}$ and that of p_j as $x_{j,0}$, both of shape $1\times C_0+3\times C_1$. The coordinate difference between the two points is given by the displacement vector $r_{ij} = p_j - p_i$. Passing r_{ij} through the spherical harmonics of degree $l = 0$ and $l = 1$, yields a 1-dimensional vector and a 3-dimensional vector, respectively. After broadcasting them to C_0 and C_1 channels, the dimensionalities align with those of the embeddings, enabling the subsequent tensor product operation.

4.2. Equivariant operator

Here, we describe the equivariant operators used in EFE and explain how they preserve $SO(3)$ equivariance. Before introducing the equivariant operators, we first provide some definitions. In this section, let the tensor feature of points at the K-th layer be denoted as x_K, and let the maximum equivariant order of the current layer be $l_{max,K}$. The dimensionality of x_K is then given by $\sum_{l=0}^{l_{max,K}}[(2l+1)\times C_l]$, where C_l denotes the number of channels corresponding to order l. The feature x_K thus consists of multiple type-l vectors, and our equivariant operators are defined separately for each order l.

We denote by $x_K = [x_K^0; x_K^1; \dots; x_K^{l_{max,K}}]$ where x_K^l is the extraction of the type-l vector from x_K, which has dimensionality $(2l+1)\times C_l$. Processing features by order refers to separating the type-l vectors for different values of l, and applying various neural network operations within the same order. In addition, illustrative diagrams of the equivariant operators introduced in this section are provided in Fig. 5.

4.2.1. Linear

The equivariant linear layer performs linear operation on the entire irreducible representation (irreps) features by separately transforming different type-l vectors x^l. Specifically, independent linear operators are applied to each group of type-l vectors. Bias terms $\mathbf{b}$ are only applied to type-0 vectors, as introducing bias terms for type-l vectors with $l > 0$ could potentially break equivariance.

The linear layer function is written as

$$\mathbf{Linear}(x) = [x^0\mathbf{W}^0 + \mathbf{b}; x^1\mathbf{W}^1; \dots; x^l\mathbf{W}^l], \tag{6}$$

where the superscript l denotes the degree of group representation and type-l vector x^l is the extraction of the type-l vector from x, which has dimensionality $(2l+1)\times C_l$. $\mathbf{W}^l \in \mathbb{R}^{C_l\times C_l'}$ is the parameterized matrix and different numbers of channels C_l correspond to distinct matrices $\mathbf{W}^l$.

For Linear layer, we treat the different type-l vectors separately, extracting the type-l component from the feature tensor, processing it within its respective order l, and then concatenating it back into the original tensor, so that for each degree l, the output type-l vector is a linear combination of other input type-l vectors, which have the same transformation matrix $D_X(g)$. This means that the output vector has the same matrix $D_Y(g) = D_X(g)$, and this satisfies Equation (1).

4.2.2. Layer normalization

Similar to the linear layer, when applying equivariant layer normalization, we independently process different type-l vectors. For type-0 vector x^0, we calculate the linear transformation of normalized input as

$$\mathbf{LN}(x^0) = \left(\frac{x^0 - \mu_{C_0}}{\sigma_{C_0}}\right)\otimes \mathbf{W}^0 + \mathbf{b}, \tag{7}$$

where C_0 denotes the channels of type-0 vector and μ_{C_0}, σ_{C_0} are mean and standard deviation of x^0 along the channel dimension, $\mathbf{W^0}, \mathbf{b}$ are learnable parameters, and their dimensions are aligned with x^0. $\otimes$ denotes element-wise product.

For type-l vectors with $l > 0$, we remove the mean and bias terms, and replace the standard deviation with the root mean square (**RMS**) of the **L2-norm** of the type-l vectors along the channel dimension. This yields the formula for performing layer normalization on irreducible representation (irreps) features. Specifically, for type-l vector x^l, the Eq. (7) becomes:

$$\mathbf{LN}(x^l) = \left(\frac{x^l}{\mathbf{RMS}(\mathbf{norm}(x^l))}\right)\otimes \mathbf{W}^l, \tag{8}$$

where $\otimes$ denotes element-wise product. **norm** calculates the **L2-norm** of type-l vectors and **RMS** calculates the root mean square value. $\mathbf{W}^l$ is a learnable scaling parameter that allows the network to retain sufficient representational capacity after normalization, and its dimension is aligned with x^l; otherwise, LayerNorm would merely enforce zero mean and unit variance, which may constrain the expressive power of the model.

For Layer Normalization, the type-0 part of the irreps is always the same regardless of $SO(3)$ transformations. For non-scalar parts ($l > 0$), the **L2-norm** of the type-l vectors is always invariant to $SO(3)$ group, leading to the **RMS** being invariant. Multiplying an equivariant feature by an invariant scalar results in an equivariant feature, thus ensuring that the operation is equivariant.

4.2.3. Gate activation

Activation functions [33] play a crucial role in the network architecture, as they introduce non-linearity, integrate linear outputs, and enhance the expressive power of the model. For type-0 vectors, standard activation functions can be used and here we choose SiLU [34,35]:

$$\mathbf{Gate}(x^0) = \mathbf{SiLU}(x^0\mathbf{W}^0). \tag{9}$$

For type-l vectors with $l > 0$, we cannot directly apply activation functions, as the non-linearity of different elements would inevitably disrupt the equivariant properties. Therefore, we design a gate activation function that combines these vectors with type-0 vectors, ensuring equivariance while simultaneously enhancing the expressiveness:

$$\mathbf{Gate}(x^l) = \mathbf{Sigmoid}(x^0\mathbf{W'}^0)\otimes x^l. \tag{10}$$

where $\otimes$ denotes element-wise product and $\mathbf{W'}^0$ is different from $\mathbf{W}^0$. In this manner, we apply the Sigmoid activation function to the type-0 vectors to obtain non-linear weights. Each type-l vector is then multiplied by its corresponding non-linear weight. Since the non-linear weights are invariant, multiplying the equivariant features by these weights results in equivariant features, thereby preserving the equivariance.

4.3. Overall architecture

The architecture of EFE is illustrated in Fig. 4 (a). The input 3D point cloud is first processed by the Embedding module to produce initial embeddings $x_{i,0}$ for each point p_i. These embeddings are then passed through K Equivariant Blocks, with two skip connections (denoted by $\oplus$) adding the input embeddings to the intermediate outputs to enhance feature propagation.

The detailed structure of an Equivariant Block is shown in Fig. 4 (b). For each point p_i, the block takes as input its current embedding $x_{i,K}$, the embeddings $x_{j,K}$ of its neighboring points $p_j \in N_i$, and the direction vector $r_{ij} = p_j - p_i$. The direction vector r_{ij} is projected into type-l vectors using spherical harmonics, producing initial irreps features. These features are combined with the embeddings $x_{i,K}$ and $x_{j,K}$ via a tensor product, yielding intermediate irreps features f_{ij}^l. The type-0 features f_{ij}^0 are processed through a SiLU activation, while the type-l features ($l > 0$) are processed through a gate activation and additional tensor product operations. The outputs are then normalized and passed through linear layers, with the final output $x_{i,K+1}$ serving as the input to the next block.

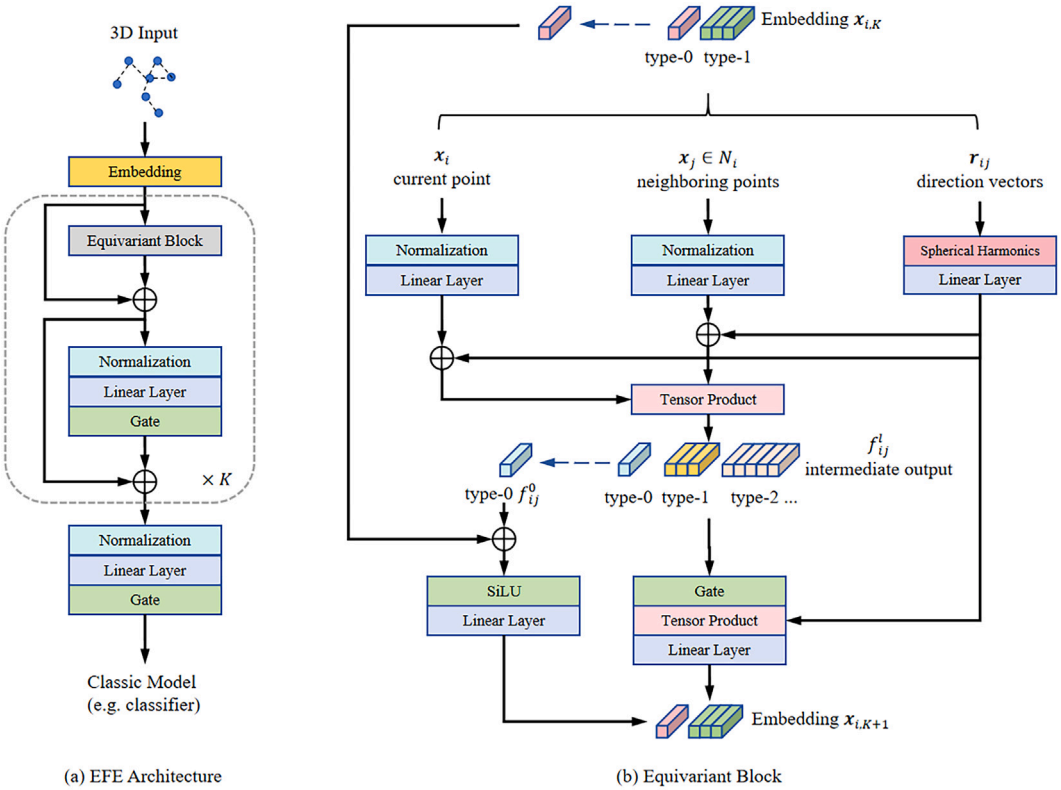


Fig. 4. Architecture of EFE. The 3D data is passed through the Embedding module and then fed into K Equivariant Blocks, with two skip connections linking the input and output. The detailed structure of the Equivariant Block is shown in (b), where the symbol $\oplus$ denotes the addition of corresponding type-l channels, and f^0_{ij}, f^l_{ij} represent the irreducible representation (irreps) features obtained from the intermediate output after the tensor product.

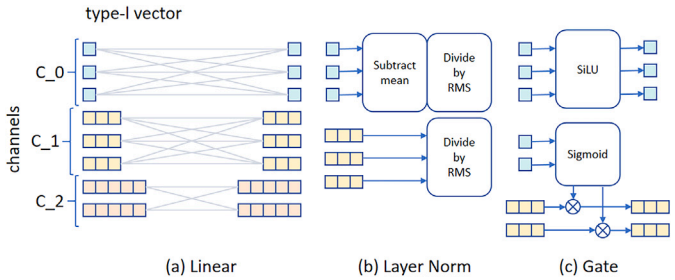


Fig. 5. Equivariant operations used in EFE. (a) Each gray line between input and output irreps features contains one learnable weight. (b) "RMS" denotes the root mean square value along the channel dimension. For simplicity, we have removed multiplying by weight **W** here. (c) Gate layers are equivariant activation functions where non-linearly transformed scalars are used to gate non-scalar irreps features.

After K Equivariant Blocks, the final embeddings $x_{i,K}$ are passed through an equivariant linear layer to produce the output equivariant features, which transform consistently under rotations in 3D space, providing rich information for potential future applications. These features can be dimensionally aligned, for example, by adjusting channel dimensions, to facilitate integration into classical point cloud models for downstream tasks. For the classification and segmentation tasks in this study, we focus on leveraging the invariant components of these equivariant features, which remain unchanged under rotational transformations. To achieve this, the equivariant features are converted into invariant features by extracting type-0 vectors, computing their norms, or using an equivariant linear layer that maps $l > 0$ to $l = 0$. These invariant features are then fed into standard point cloud classifiers to perform the classification and segmentation tasks.

5. Experiments

We evaluate our method on two core tasks in point cloud processing: 3D shape classification and part segmentation. Following the approach of Esteves et al. [45], we adopt three different train/test settings: training and testing the network under rotations around the z-axis (z/z); training under rotations around the z-axis and testing under arbitrary rotations ($z/SO(3)$); and training and testing under arbitrary rotations ($SO(3)/SO(3)$). Here, z refers to data augmentation involving rotations only around the z-axis, while $SO(3)$ denotes arbitrary rotations. All rotation matrices are dynamically generated during training, where random z-axis rotations or full $SO(3)$ rotations are applied to the data to construct the training set, and the corresponding models are trained accordingly. At test time, the trained models are evaluated on data that are similarly transformed with either random z-axis rotations or $SO(3)$ rotations, yielding the z/z, $z/SO(3)$, and $SO(3)/SO(3)$ results. A comparison of these results highlights the distinction between intrinsic equivariance and equivariance achieved through data augmentation.

The experimental setup in this study follows a standard template designed for evaluating rotation-invariant and rotation-equivariant methods. In contrast, classic classification and segmentation methods that do not focus on rotational properties typically do not undergo rotation testing. Consequently, the results reported in the z/z column of Table 2 for these methods correspond to experiments conducted under the I/I setting, where I denotes the identity transformation (i.e., no transformations are applied to either the training or test sets).

In the classification and segmentation experiments, we integrate the EFE with classic point cloud network architectures, introducing only a

Table 2
Test classification accuracy (%) on the ModelNet40 dataset in three train/test scenarios. We have listed methods that are variant, invariant, and equivariant under rotation operations.

Methods	z/z	z/SO(3)	SO(3)/SO(3)
rotation-variant methods			
PointNet [7]	89.2	16.4	75.5
DGCNN [8]	92.9	20.6	81.1
PCNN [36]	92.3	11.9	85.1
ShellNet [37]	**93.1**	19.9	87.8
PointNet++ [19]	90.7	28.6	85.0
PointCNN [20]	92.5	41.2	84.5
rotation-invariant methods			
RI-Conv [29]	86.5	86.4	86.4
SPHNet [38]	87.7	86.6	87.6
ClusterNet [39]	87.1	87.1	87.1
GC-Conv [26]	89.0	89.1	89.2
RI-Framework [27]	89.4	89.4	89.3
PaRot [30]	90.9	90.9	90.8
TetraSphere [40]	90.5	90.5	90.5
RI [41]	90.4	90.4	90.4
RotInv-PCT [42]	91.1	91.1	91.1
Shen [43]	92.8	90.6	90.6
LocoTrans [44]	91.6	**91.6**	**91.5**
rotation-equivariant methods			
Spherical-CNN [45]	88.9	76.7	86.9
SFCNN [46]	91.4	84.8	90.1
TFN [21]	88.5	85.3	87.6
VN-PointNet [24]	77.5	77.5	77.2
VN-DGCNN [24]	89.5	89.5	90.2
VN-Transformer [47]	–	90.8	–
OrientedMP [25]	88.4	88.4	88.9
Equivariant-Conv [48]	86.9	87.0	89.0
EFE-PointNet	86.8	86.8	86.5
EFE-DGCNN	91.2	91.2	<u>91.2</u>
EFE-PointTransformer	91.4	<u>91.4</u>	91.0

small number of learnable parameters without modifying the original network structure. This results in a network architecture with rotation equivariance, achieving promising performance. Considering the stability and influence of various classic networks, we primarily choose DGCNN as the base architecture for the hybrid model due to its excellent stability, robustness, and widespread impact. In addition, our training settings are consistent with those of DGCNN and VN-DGCNN, including 250 epochs for classification, 200 epochs for segmentation, 1,024 sampled points, and the use of the same optimizer, learning rate, and other hyperparameters. Each configuration was evaluated through five independent runs, with the mean performance reported as the final result.

5.1. 3D object classifcation

5.1.1. Synthetic dataset

We first evaluate the model's performance on the synthetic ModelNet40 dataset [17], which consists of CAD models from 40 categories, such as airplanes, bottles, chairs, dressers, vases, and so on. We use the preprocessed data from PointNet [7], which includes 9,843 models for training and 2,468 models for testing. In this experiment, we use point clouds of size 1024. Each point is represented by (x, y, z) coordinates in the Euclidean space.

We report and compare the performance of our hybrid model in Table 2. Compared to non-equivariant networks, the EFE-models consistently achieve good results across all three settings, demonstrating their robustness to rotations. Notably, in the $z/SO(3)$ setting, where the test set contains unseen rotations not present in the training set, classic methods perform poorly, whereas our method remains stable. Even in the $SO(3)/SO(3)$ setting, where extensive data augmentation is applied during training, the performance of rotation-sensitive networks still falls short of that achieved by the EFE-equivariant network, which proves the effectiveness of our method.

Table 3
Test classification accuracy (%) on the ScanObjectNN dataset(PB_T50_RS) in three train/test scenarios. We have listed methods that are variant, invariant, and equivariant under rotation operations. Our method falls into the equivariant category and achieves outstanding results.

Methods	z/z	z/SO(3)	SO(3)/SO(3)
rotation-variant methods			
PointNet [7]	68.2	17.1	42.2
DGCNN [8]	78.1	16.1	63.4
PointNet++ [19]	77.9	15.8	60.1
PointCNN [20]	**78.5**	14.9	51.8
rotation-invariant methods			
RI-Conv [29]	68.1	68.3	68.3
GC-Conv [26]	69.8	69.8	70.0
RI-Framework [27]	–	70.1	–
PaRot [30]	74.2	74.2	74.6
rotation-equivariant methods			
OrientedMP [25]	68.4	68.4	68.9
EFE-PointNet	67.8	67.8	67.5
EFE-DGCNN	75.0	**75.0**	**75.0**

5.1.2. Real dataset

Real-world point cloud data often contain missing points, occlusions, and non-uniform density. ScanObjectNN [49] is a commonly used benchmark dataset, captured by RGB-D cameras, to evaluate the robustness of methods on noisy and deformed 3D objects with non-uniform surface density. This dataset comprises 2,902 incomplete point clouds across 15 categories. In our evaluation, we utilize the preprocessed files and select the most challenging subset, PB_T50_RS, which includes 50 % bounding box translation, rotation around the gravity axis, and random scaling, resulting in rotated and partially missing data. We sample 1,024 points under the z/z, $z/SO(3)$ and $SO(3)/SO(3)$ settings. We report the performance of our hybrid model in Table 3 and our approach demonstrates superior results compared to classic methods.

5.2. 3D part segmentation

Shape part segmentation is a more challenging task compared to object classification. We conduct experiments on the ShapeNet [18] dataset to evaluate its performance on part segmentation tasks. This dataset consists of 16,881 models from 16 categories, annotated with 50 parts. Additionally, there is no overlap between the training and testing sets, and 2048 points with (x, y, z) coordinates are sampled as model inputs.

Table 4 presents the segmentation results for different methods in the $z/SO(3)$ and $SO(3)/SO(3)$ scenarios. Classic methods, such as PointNet and DGCNN, demonstrate vulnerability to rotations. It further confirms that despite applying rotation data augmentation, these methods, which lack inherent rotational invariance or equivariance still perform poorly. In contrast, we achieve outstanding results with our equivariant model.

To provide a more intuitive observation of the advantages of the equivariance method on rotated datasets, we visualize the object part segmentation results, as shown in Fig. 6. We select representative object categories from the ShapeNet dataset (specifically, airplanes and cars) for visualization and compare the classic models with our invariant model. The experiments are conducted under the previously mentioned configurations of rotation around the z-axis and arbitrary spatial rotations ($SO(3)$). The results reveal that classic models perform poorly on the rotated test sets, while our method is on a par with the original test set, highlighting the equivariance property and superior performance of our approach.

Table 4
ShapeNet part segmentation. The results are reported as the overall average category mean IoU in two train/test scenarios. We have listed methods that are variant, invariant, and equivariant under rotation operations. Our method falls into the equivariant category and achieves outstanding results.

Methods	z/SO(3)	SO(3)/SO(3)
rotation-variant methods		
PointNet [7]	38.0	62.3
DGCNN [8]	49.3	78.6
PCT [50]	38.5	75.2
ShellNet [37]	47.2	77.1
PointNet++ [19]	48.3	76.7
PointCNN [20]	34.7	71.4
rotation-invariant methods		
RI-Conv [29]	75.3	75.3
GC-Conv [26]	77.3	77.2
RI-Framework [27]	79.2	79.4
PaRot [30]	79.2	79.5
TetraSphere [40]	82.0	82.0
RI [41]	**84.5**	**84.5**
RotInv-PCT [42]	82.3	82.3
Shen [43]	82.8	81.5
LocoTrans [44]	84.0	83.8
rotation-equivariant methods		
VN-PointNet [24]	72.4	72.8
VN-DGCNN [24]	81.4	81.4
EFE-PointNet	78.4	78.3
EFE-DGCNN	83.4	83.4

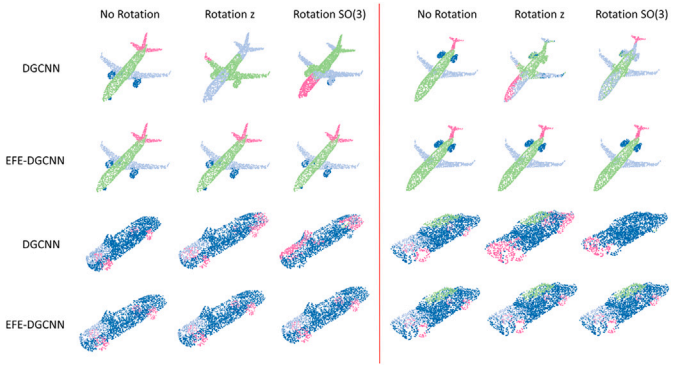


Fig. 6. The comparative visualization results of object part segmentation experiments were conducted by testing the dataset under three conditions: no rotation, rotation around the z-axis, and arbitrary spatial rotation. We selected representative object categories from the ShapeNet dataset (specifically, airplanes and cars) and compared the performance of classic models with that of our combined model.

5.3. Ablation study

In this section, we conduct ablation studies to evaluate the contributions of key components in our Equivariant Feature Extractor (EFE), including the equivariant operators, the model's lightweight design, and the impact of the equivariant order l. All experiments are performed on the ModelNet40 dataset under the $z/SO(3)$ setting, where the training set is aligned along the z-axis, and the test set includes arbitrary $SO(3)$ rotations.

5.3.1. Effectiveness of equivariant operators

We validate the effectiveness of a series of equivariant operators in EFE through classification experiments under $z/SO(3)$ setting. Starting with the complete EFE model with DGCNN, we progressively ablate the equivariant linear layers **L**, equivariant layer normalization **LN**, gate activation functions **Gate**, and tensor products **TP**, replacing them with standard linear layers, standard layer normalization, standard activation functions, and regular matrix multiplication, respectively. The experimental results, as shown in Table 5, demonstrate that these equivariant operators play a critical role in preserving equivariance of the model. Removing them can significantly reduce accuracy and may even result in the loss of equivariance.

Table 5
Ablation study on ModelNet40. We list the equivariant operators employed in our method and progressively ablate them. The results demonstrate the effectiveness of these operators.

Model	**L**	**LN**	**Gate**	**TP**	z/SO(3)
A	✓	✓	✓	✓	91.2
B		✓	✓	✓	89.6
C	✓		✓	✓	30.7
D	✓	✓		✓	45.5
E	✓	✓	✓		38.8
DGCNN					20.6

Table 6
Ablation Study on Lightweight. We compared the learnable parameters of classic models and their corresponding equivariant variants, including VN and EFE. The results demonstrate that our method is more lightweight, introducing only a small increase in parameters while maintaining seamless integration with classic models.

Methods	Params.	Relative Params.
PointNet	0.696 M	–
DGCNN	1.810 M	–
RI-Conv	4.189 M	–
RI-Framework	2.363 M	–
RotInv-PCT	7.645 M	–
LocoTrans	6.273 M	–
VN-PointNet	2.201 M	216.2 %
VN-DGCNN	2.899 M	59.6 %
EFE-PointNet	0.702 M	0.85 %
EFE-DGCNN	1.819 M	0.50 %

5.3.2. Lightweight and portability

Unlike other fixed architectures, our approach is lightweight and highly versatile, making it easy to integrate with classic methods to achieve rotation equivariance. The VN method [24], similar to our work, is an equivariant approach that can be conveniently applied to classic models. We evaluate the model size, presented in Table 6, which shows that our method introduces only a small increase in the number of learnable parameters compared to classic models. Specifically, the EFE models corresponding to PointNet and DGCNN only increase by 0.85 % and 0.50 % respectively, while their corresponding VN models show a significantly larger increase in parameter numbers. In contrast, our approach offers a more lightweight design while seamlessly integrating with classic models.

5.3.3. Impact of equivariant order

We investigate the impact of the maximum equivariant order $l_{\max}$ on the performance of EFE. The equivariant order determines the complexity of the $SO(3)$ representations used in the model, with higher $l_{\max}$ capturing more detailed rotational information but increasing computational cost. We test $l_{\max}$ values from 0 to 3, using EFE integrated with DGCNN, and evaluate the classification accuracy under the z/SO(3) setting on ModelNet40. The results are reported in Table 7.

When $l_{\max} = 0$, the model only uses scalar (type-0) features, effectively reducing to rotation-invariant features rather than equivariant ones. This leads to a slightly lower accuracy. However, when $l_{\max}$ reaches 3, the accuracy slightly decreases, likely due to overfitting caused by the increased complexity of the representations. These results underscore

Table 7
Ablation study on the impact of equivariant order l_{max} on ModelNet40 under the z/SO(3) setting. We report the classification accuracy (%) for different l_{max} values.

l_{max}	z/SO(3) (%)
0	89.7
1	90.5
2	91.2
3	90.2

the importance of selecting an appropriate l_{max} to balance rotational expressiveness and generalization ability.

6. Conclusion

In this paper, we propose a novel approach for extracting $SO(3)$ rotation-equivariant features from 3D point clouds. We utilize group representations and spherical harmonics to encode 3D positional information, projecting it into a high-dimensional equivariant space. A series of equivariant operators is then designed to process these features, facilitating the exchange and update of equivariant information. This results in features that are both highly expressive and robust to rotations, enabling seamless integration with classical point cloud models to construct rotation-equivariant architectures.

Our method demonstrates strong performance in classification and segmentation tasks, particularly in rotation test sets, where it achieves excellent rotational consistency compared to existing methods that often struggle with varying rotation types. However, our approach has certain limitations. It exhibits average performance on large-scale point clouds due to computational constraints, and its fitting ability is limited on datasets without rotational variations, as the model is primarily optimized for rotation-equivariant feature extraction. In future work, we aim to address these shortcomings by exploring more efficient equivariant feature extraction methods, optimizing computational scalability for large-scale point clouds, and enhancing the model's generalization ability on non-rotated datasets.

CRediT authorship contribution statement

Qianwei Tang: Writing – original draft, Visualization, Validation, Methodology, Investigation, Data curation, Conceptualization. **Baile Xu:** Writing – review & editing, Supervision. **Jian Zhao:** Writing – review & editing, Supervision. **Furao Shen:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially supported by the STI 2030-Major Projects of China (Grant No. 2021ZD0201300), the National Natural Science Foundation of China (Grant Nos. 62276127, 62495094), and the Fundamental Research Funds for the Central Universities (Grant No. 2024300394) of Nanjing University. The authors gratefully acknowledge these supports.

Data availability

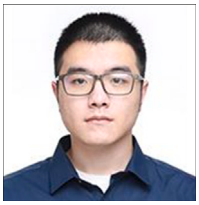
Data will be made available on request.

References

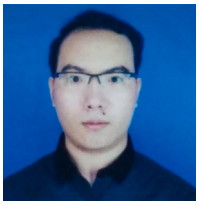
[1] R. Wang, J. Peethambaran, D. Chen, Lidar point clouds to 3-d urban models : a review, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 11 (2) (2018) 606–627.
[2] J. Smisek, M. Jancosek, T. Pajdla, 3D with Kinect, Consumer Depth Cameras For Computer Vision: Research Topics And Applications (2013) 3–25.
[3] M.A. Guerroudji, K. Amara, M. Lichouri, N. Zenati, M. Masmoudi, A 3D visualization-based augmented reality application for brain tumor segmentation, Comput. Animat. Virtual Worlds 35 (1) (2024) e2223.
[4] F. Sattler, B. Carrillo-Perez, S. Barnes, K. Stebner, M. Stephan, G. Lux, Embedded 3D reconstruction of dynamic objects in real time for maritime situational awareness pictures, Vis. Comput. 40 (2) (2024) 571–584.
[5] X. Zhu, X. Yao, J. Zhang, M. Zhu, L. You, X. Yang, J. Zhang, H. Zhao, D. Zeng, Tmsdnet: transformer with multi-scale dense network for single and multi-view 3D reconstruction, Comput. Animat. Virtual Worlds 35 (1) (2024) e2201.
[6] X. Huang, G. Mei, J. Zhang, R. Abbas, A comprehensive survey on point cloud registration, arXiv preprint arXiv:2103.02690 (2021).
[7] C.R. Qi, H. Su, K. Mo, 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 (CVPR), 2017, pp. 652–660.
[8] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph CNN for learning on point clouds, ACM Trans. On Graph (TOG) 38 (5) (2019) 1–12.
[9] S. Safa Aldin, N.B. Aldin, M. Aykac, Enhanced image classification using edge CNN (E-CNN), Vis. Comput. 40 (1) (2024) 319–332.
[10] L.I.U. Yujie, S.U.N. Xiaorui, S.H.A.O. Wenbin, Y.U.A.N. Yafu, S2anet: combining local spectral and spatial point grouping for point cloud processing, Virtual Real. Intell. Hardw. 6 (4) (2024) 267–279.
[11] A. Karambakhsh, B. Sheng, P. Li, H. Li, J. Kim, Y. Jung, C.L.P. Chen, SparseVoxNet: 3-D object recognition with sparsely aggregation of 3-D dense blocks, IEEE Trans. Neural Netw. Learn. Syst. 35 (1) (2022) 532–546.
[12] T. Cohen, M. Welling, Group equivariant convolutional networks, in: International Conference on Machine Learning (IMCL), PMLR, 2016, pp. 2990–2999.
[13] D. Worrall, G. Brostow, Cubenet: equivariance to 3D rotation and translation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 567–584.
[14] N. Dym, H. Maron, On the universality of rotation equivariant point cloud networks, arXiv preprint arXiv:2010.02449 (2020).
[15] C.K. Joshi, C. Bodnar, S.V. Mathis, T. Cohen, P. Lio, On the expressive power of geometric graph neural networks, in: International Conference on Machine Learning (ICML), PMLR, 2023, pp. 15330–15355.
[16] D.J. Griffiths, D.F. Schroeter, Introduction to Quantum Mechanics, Cambridge university press, 2019.
[17] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3D shapenets: a deep representation for volumetric shapes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1912–1920.
[18] A.X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al., Shapenet: an information-rich 3d model repository, arXiv preprint arXiv:1512.03012 (2015).
[19] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: deep hierarchical feature learning on point sets in a metric space, Adv. Neural Inf. Process. Syst. 30 (2017).
[20] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, B. Chen, PointCNN: convolution on x-transformed points, Adv. Neural Inf. Process. Syst. 31 (2018).
[21] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, P. Riley, Tensor field networks: rotation-and translation-equivariant neural networks for 3d point clouds, arXiv preprint arXiv:1802.08219 (2018).
[22] F. Fuchs, D. Worrall, V. Fischer, M. Welling, Se (3)-transformers: 3D roto-translation equivariant attention networks, Adv. Neural Inf. Process. Syst. 33 (2020) 1970–1981.
[23] Y.-L. Liao, T. Smidt, Equiformer: equivariant graph attention transformer for 3d atomistic graphs, arXiv preprint arXiv:2206.11990 (2022).
[24] C. Deng, O. Litany, Y. Duan, A. Poulenard, A. Tagliasacchi, L.J. Guibas, Vector neurons: a general framework for so (3)-equivariant networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 12200–12209.
[25] S. Luo, J. Li, J. Guan, Y. Su, C. Cheng, J. Peng, J. Ma, Equivariant point cloud analysis via learning orientations for message passing, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 18932–18941.
[26] Z. Zhang, B.-S. Hua, W. Chen, Y. Tian, S.-K. Yeung, Global context aware convolutions for 3d point cloud understanding, in: 2020 International Conference on 3D Vision (3DV), IEEE, 2020, pp. 210–219.
[27] X. Li, R. Li, G. Chen, C.-W. Fu, D. Cohen-Or, P.-A. Heng, A rotation-invariant framework for deep point cloud analysis, IEEE Trans. Vis. Comput. Graph. 28 (12) (2021) 4503–4514.
[28] C. Zhao, J. Yang, X. Xiong, A. Zhu, Z. Cao, X. Li, Rotation invariant point cloud classification: where local geometry meets global topology, arXiv preprint arXiv:1911.00195 (2019).
[29] Z. Zhang, B.-S. Hua, D.W. Rosen, 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.
[30] D. Zhang, J. Yu, C. Zhang, W. Cai, Parot: Patch-wise rotation-invariant network via feature disentanglement and pose restoration, in: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), vol. 37, 2023, pp. 3418–3426.
[31] T.S. Cohen, M. Geiger, J. Köhler, M. Welling, Spherical cnns, arXiv preprint arXiv:1801.10130 (2018).
[32] M. Geiger, T. Smidt, e3nn: Euclidean neural networks, arXiv preprint arXiv:2207.09453 (2022).
[33] M. Weiler, M. Geiger, M. Welling, W. Boomsma, T.S. Cohen, 3D steerable CNNs: learning rotationally equivariant features in volumetric data, Adv. Neural Inf. Process. Syst. 31 (2018).

[34] S. Elfwing, E. Uchibe, K. Doya, Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, Neural Networks 107 (2018) 3–11.
[35] P. Ramachandran, B. Zoph, Q.V. Le, Searching for activation functions, arXiv preprint arXiv:1710.05941 (2017).
[36] M. Atzmon, H. Maron, Y. Lipman, Point convolutional neural networks by extension operators, arXiv preprint arXiv:1803.10091 (2018).
[37] Z. Zhang, B.-S. Hua, S.-K. Yeung, Shellnet: efficient point cloud convolutional neural networks using concentric shells statistics, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1607–1616.
[38] A. Poulenard, M.-J. Rakotosaona, Y. Ponty, M. Ovsjanikov, Effective rotation-invariant point CNN with spherical harmonics kernels, in: 2019 International Conference on 3D Vision (3DV), IEEE, 2019, pp. 47–56.
[39] C. Chen, G. Li, R. Xu, T. Chen, M. Wang, L. Lin, Clusternet: deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4994–5002.
[40] P. Melnyk, A. Robinson, M. Felsberg, M. Wadenbäck, Tetrasphere: a neural descriptor for O (3)-invariant point cloud analysis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 5620–5630.
[41] S. Luo, W. Gao, A general framework for rotation invariant point cloud analysis, in: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2024, pp. 3665–3669.
[42] C. He, Z. Zhao, X. Zhang, H. Yu, R. Wang, Rotinv-pct: Rotation-invariant point cloud transformer via feature separation and aggregation, Neural Networks 185 (2025) 107223.
[43] K. Shen, J. Zhao, M. Xie, A novel equivariant self-supervised vector network for three-dimensional point clouds, Algorithms 18 (3) (2025) 152.
[44] Y. Chen, L. Duan, S. Zhao, C. Ding, D. Tao, Local-consistent transformation learning for rotation-invariant point cloud analysis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 5418–5427.
[45] C. Esteves, C. Allen-Blanchette, A. Makadia, K. Daniilidis, Learning so (3) equivariant representations with spherical CNNs, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 52–68.
[46] Y. Rao, J. Lu, J. Zhou, Spherical fractal convolutional neural networks for point cloud recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 452–460.
[47] S. Assaad, C. Downey, R. Al-Rfou, N. Nayakanti, B. Sapp, Vn-transformer: rotation-equivariant attention for vector neurons, arXiv preprint arXiv:2206.04176 (2022).
[48] L. Weijler, P. Hermosilla, Efficient continuous group convolutions for local se (3) equivariance in 3d point clouds, arXiv preprint arXiv:2502.07505 (2025).
[49] M.A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, S.-K. Yeung, 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 (ICCV), 2019, pp. 1588–1597.
[50] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R.R. Martin, S.-M. Hu, Pct: Point cloud transformer, Comput. Vis. Media 7 (2021) 187–199.

Author biography

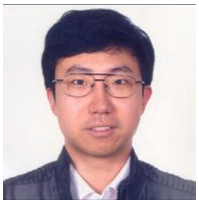


Qianwei Tang received B.E. degree in June 2024 from Nanjing University. Currently, he pursuing a master's degree of School of Artificial Intelligence in Nanjing University as a member of RINC Group, led by professor Furao Shen. His research interests include AI for Science and computer vision.



Baile Xu received the bachelor's degree from Department of Software Engineering, Shandong University. Currently he is a Ph.D. student at Nanjing University. His research interests include artificial neural networks and machine learning.

Furao Shen received the B.Sc. and M.Sc. degrees in mathematics from Nanjing University, Nanjing, China, in 1995 and 1998, respectively, and the Ph.D. degree from the Tokyo Institute of Technology, Tokyo, Japan, in 2006. He is currently a Full Professor of computer science and technology with Nanjing University. His current research interests include neural computing and robotic intelligence.



Jian Zhao received the B.S. degree from Nanjing University, Nanjing, China, in 2001, the M.Sc. degree from Hamburg University of Technology, Hamburg, Germany, in 2004, and the Ph.D. degree in electrical engineering from Swiss Federal Institute of Technology (ETH) Zurich, Switzerland, in 2010. Since December 2010, he has been with the Institute for Infocomm Research, Singapore. His research interests include optimization techniques in wireless communications, multiuser MIMO communications, and cooperative communications. Dr. Zhao has received a number of awards, including the DAAD-Siemens Asia 21st Century Scholarship, IEEE Globecom 2008 Best Paper Award, and Chinese Government Award for Outstanding Self-Financed Students Abroad.