3D-Rotation-Equivariant Quaternion Neural Networks: Supplementary Materials

Wen Shen[2,*], Binbin Zhang[2,*], Shikun Huang[2,*], Zhihua Wei[2], and Quanshi Zhang[1,†]

[1]Shanghai Jiao Tong University, [2]Tongji University, Shanghai
{wen_shen,0206zbb,hsk,zhihua_wei}@tongji.edu.cn,zqs1022@sjtu.edu.cn

A Quaternion Operations

A quaternion $\mathbf{q} = q_0 + q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k} \in \mathbb{H}$ is a hyper-complex number with a real part (q_0) and three imaginary parts ($q_1\boldsymbol{i}, q_2\boldsymbol{j}, q_3\boldsymbol{k}$), where $q_0, q_1, q_2, q_3 \in \mathbb{R}$; $\mathbb{H}$ denotes the algebra of quaternions. The products of basis elements $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$ are defined by $\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1$ and $\boldsymbol{ij} = \boldsymbol{k}, \boldsymbol{jk} = \boldsymbol{i}, \boldsymbol{ki} = \boldsymbol{j}, \boldsymbol{ji} = -\boldsymbol{k}, \boldsymbol{kj} = -\boldsymbol{i}$, and $\boldsymbol{ik} = -\boldsymbol{j}$.

Just like complex numbers, given two quaternions $\mathbf{p} = p_0 + p_1\boldsymbol{i} + p_2\boldsymbol{j} + p_3\boldsymbol{k}$ and $\mathbf{q} = q_0 + q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k}$, a series of quaternion operations can be defined as follows.

Addition: $\mathbf{p} + \mathbf{q} = (p_0 + q_0) + (p_1 + q_1)\boldsymbol{i} + (p_2 + q_2)\boldsymbol{j} + (p_3 + q_3)\boldsymbol{k}$.

Scalar multiplication: $\lambda\mathbf{q} = \lambda q_0 + \lambda q_1\boldsymbol{i} + \lambda q_2\boldsymbol{j} + \lambda q_3\boldsymbol{k}$.

Element multiplication: $\mathbf{pq} = (p_0q_0 - p_1q_1 - p_2q_2 - p_3q_3) + (p_0q_1 + p_1q_0 + p_2q_3 - p_3q_2)\boldsymbol{i} + (p_0q_2 - p_1q_3 + p_2q_0 + p_3q_1)\boldsymbol{j} + (p_0q_3 + p_1q_2 - p_2q_1 + p_3q_0)\boldsymbol{k}$.

Note that multiplication of two quaternions is non-commutative, because $\mathbf{qp} = (q_0p_0 - q_1p_1 - q_2p_2 - q_3p_3) + (q_0p_1 + q_1p_0 + q_2p_3 - q_3p_2)\boldsymbol{i} + (q_0p_2 - q_1p_3 + q_2p_0 + q_3p_1)\boldsymbol{j} + (q_0p_3 + q_1p_2 - q_2p_1 + q_3p_0)\boldsymbol{k} \neq \mathbf{pq}$.

Norm: $\|\mathbf{q}\|=\sqrt{\mathbf{q}\overline{\mathbf{q}}} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$, where $\overline{\mathbf{q}} = q_0 - q_1\boldsymbol{i} - q_2\boldsymbol{j} - q_3\boldsymbol{k}$ is the conjugation of $\mathbf{q}$.

B From Layerwise Rotation-Equivariance Property to the Rotation-Equivariance Property of the REQNN

Let $\mathbf{x} \in \mathbb{H}^n$ and $\mathbf{y} = \mathbf{\Phi}(\mathbf{x}) = \Phi_L(\Phi_{L-1}(\cdots\Phi_1(\mathbf{x}))) \in \mathbb{H}^C$ denote the input and the output of the REQNN, respectively. Let $\mathbf{f}_l = \Phi_l(\mathbf{f}_{l-1}) \in \mathbb{H}^d$ denote the output of the l-th layer. We prove that the layerwise rotation equivariance can ensure the rotation-equivariance property of the REQNN. *I.e.* if $\Phi_l(\mathbf{f}_{l-1}^{(\theta)}) = \mathbf{R}\circ\Phi_l(\mathbf{f}_{l-1})\circ\overline{\mathbf{R}}$, *s.t.* $\mathbf{f}_{l-1}^{(\theta)} \triangleq \mathbf{R}\circ\mathbf{f}_{l-1}\circ\overline{\mathbf{R}}$, $\forall l \in [1, 2, ..., L]$, then $\mathbf{\Phi}(\mathbf{x}^\theta) = \mathbf{R}\circ\mathbf{y}\circ\overline{\mathbf{R}}$.

* Wen Shen, Binbin Zhang and Shikun Huang have equal contributions.
† Quanshi Zhang is the corresponding author.

$$\begin{aligned}\boldsymbol{\Phi}(\mathbf{x}^{\theta}) =& \mathit{\Phi}_L(\mathit{\Phi}_{L-1}(\cdots \mathit{\Phi}_1(\mathbf{R}\circ\mathbf{x}\circ\overline{\mathbf{R}})))\\ =& \mathit{\Phi}_L(\mathit{\Phi}_{L-1}(\cdots \mathbf{R}\circ\mathit{\Phi}_1(\mathbf{x})\circ\overline{\mathbf{R}}))\\ &\cdots\\ =& \mathit{\Phi}_L(\mathbf{R}\circ\mathit{\Phi}_{L-1}(\cdots \mathit{\Phi}_1(\mathbf{x}))\circ\overline{\mathbf{R}})\\ =& \mathbf{R}\circ\mathit{\Phi}_L(\mathit{\Phi}_{L-1}(\cdots \mathit{\Phi}_1(\mathbf{x})))\circ\overline{\mathbf{R}}\\ =& \mathbf{R}\circ\mathbf{y}\circ\overline{\mathbf{R}}\end{aligned} \tag{1}$$

C Proofs of Layerwise Rotation Equivariance

To prove that a layerwise operation $\mathit{\Phi}(\cdot)$ is rotation equivariant, $\mathit{\Phi}(\cdot)$ should satisfy $\mathit{\Phi}(\mathbf{f}^{(\theta)}) = \mathbf{R}\circ\mathit{\Phi}(\mathbf{f})\circ\overline{\mathbf{R}}$, where $\mathbf{f}^{(\theta)}$ indicates the quaternion feature that was obtained by rotating $\mathbf{f}$ around an axis $\mathbf{o} = o_1\boldsymbol{i}+o_2\boldsymbol{j}+o_3\boldsymbol{k}$ with an angle θ. Such a rotation can be represented using quaternion $\mathbf{R} = \cos\frac{\theta}{2}+\sin\frac{\theta}{2}(o_1\boldsymbol{i}+o_2\boldsymbol{j}+o_3\boldsymbol{k})$ and its conjugation $\overline{\mathbf{R}} = \cos\frac{\theta}{2} - \sin\frac{\theta}{2}(o_1\boldsymbol{i}+o_2\boldsymbol{j}+o_3\boldsymbol{k})$, *i.e.* $\mathbf{f}^{(\theta)} = \mathbf{R}\circ\mathbf{f}\circ\overline{\mathbf{R}}$. We only use symbol θ to represent a rotated quaternion feature and omit the symbol $\mathbf{o}$ for convenience.

C.1 Convolution Operation

Let $\mathbf{f} = [\mathbf{f}_1, \ldots, \mathbf{f}_d]^{\top} \in \mathbb{H}^d$ denote the quaternion feature of a point in the point cloud. Note that 3D point cloud processing usually uses the specific convolution with 1×1 kernels. Let us take this specific convolution as the example to prove that the revised convolution operation is rotation equivariant. The revised convolution operation,

$$\begin{aligned}Conv(\mathbf{f}) &= w\otimes\mathbf{f}\\ &= \begin{bmatrix} w_{11} & \cdots & w_{1d}\\ \vdots & \ddots & \vdots\\ w_{D1} & \cdots & w_{Dd}\end{bmatrix}\otimes\begin{bmatrix}\mathbf{f}_1\\ \vdots\\ \mathbf{f}_d\end{bmatrix}\\ &= \begin{bmatrix}(w_{11}\mathbf{f}_1+\cdots+w_{1d}\mathbf{f}_d)\\ \vdots\\ (w_{D1}\mathbf{f}_1+\cdots+w_{Dd}\mathbf{f}_d)\end{bmatrix},\end{aligned} \tag{2}$$

is rotation-equivariant, because $Conv(\mathbf{f}^{(\theta)}) = \mathbf{R} \circ Conv(\mathbf{f}) \circ \overline{\mathbf{R}}$. The proof is given as follows.

$$\begin{aligned}
Conv(\mathbf{f}^{(\theta)}) &= w \otimes (\mathbf{R} \circ \mathbf{f} \circ \overline{\mathbf{R}}) \\
&= \begin{bmatrix} w_{11} & \cdots & w_{1d} \\ \vdots & \ddots & \vdots \\ w_{D1} & \cdots & w_{Dd} \end{bmatrix} \otimes (\mathbf{R} \circ \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_d \end{bmatrix} \circ \overline{\mathbf{R}}) \\
&= \begin{bmatrix} (w_{11}(\mathbf{R}\mathbf{f}_1\overline{\mathbf{R}}) + \cdots + w_{1d}(\mathbf{R}\mathbf{f}_d\overline{\mathbf{R}})) \\ \vdots \\ (w_{D1}(\mathbf{R}\mathbf{f}_1\overline{\mathbf{R}}) + \cdots + w_{Dd}(\mathbf{R}\mathbf{f}_d\overline{\mathbf{R}})) \end{bmatrix} \\
&= \begin{bmatrix} (\mathbf{R}(w_{11}\mathbf{f}_1)\overline{\mathbf{R}} + \cdots + \mathbf{R}(w_{1d}\mathbf{f}_d)\overline{\mathbf{R}}) \\ \vdots \\ (\mathbf{R}(w_{D1}\mathbf{f}_1)\overline{\mathbf{R}} + \cdots + \mathbf{R}(w_{Dd}\mathbf{f}_d)\overline{\mathbf{R}}) \end{bmatrix} \\
&= \mathbf{R} \circ (w \otimes \mathbf{f}) \circ \overline{\mathbf{R}} \\
&= \mathbf{R} \circ Conv(\mathbf{f}) \circ \overline{\mathbf{R}}.
\end{aligned} \tag{3}$$

C.2 ReLU Operation

The revised ReLU operation, $ReLU(\mathbf{f}_v) = \frac{\|\mathbf{f}_v\|}{\max\{\|\mathbf{f}_v\|, c\}} \mathbf{f}_v$, is rotation-equivariant, because $ReLU(\mathbf{f}_v^{(\theta)}) = \mathbf{R} ReLU(\mathbf{f}_v)\overline{\mathbf{R}}$. The proof is given as follows.

$$\begin{aligned}
ReLU(\mathbf{f}_v^{(\theta)}) &= \frac{\|\mathbf{R}\mathbf{f}_v\overline{\mathbf{R}}\|}{\max\{\|\mathbf{R}\mathbf{f}_v\overline{\mathbf{R}}\|, c\}} (\mathbf{R}\mathbf{f}_v\overline{\mathbf{R}}) \\
&= \frac{\|\mathbf{f}_v\|}{\max\{\|\mathbf{f}_v\|, c\}} (\mathbf{R}\mathbf{f}_v\overline{\mathbf{R}}) \\
&= \mathbf{R}(\frac{\|\mathbf{f}_v\|}{\max\{\|\mathbf{f}_v\|, c\}} \mathbf{f}_v)\overline{\mathbf{R}} \\
&= \mathbf{R} ReLU(\mathbf{f}_v)\overline{\mathbf{R}}.
\end{aligned} \tag{4}$$

Note that $\|\mathbf{R}\mathbf{f}_v\overline{\mathbf{R}}\| = \|\mathbf{f}_v\|$, because rotating a quaternion will not change its norm.

C.3 Batch-Normalization

The revised batch-normalization, $norm(\mathbf{f}_v^{(i)}) = \frac{\mathbf{f}_v^{(i)}}{\sqrt{\mathbb{E}_j[\|\mathbf{f}_v^{(j)}\|^2] + \epsilon}}$, is rotation-equivariant, because $norm((\mathbf{f}_v^{(i)})^{(\theta)}) = \mathbf{R} norm(\mathbf{f}_v^{(i)})\overline{\mathbf{R}}$. The proof is given as follows.

$$
\begin{aligned}
norm((\mathbf{f}_v^{(i)})^{(\theta)}) &= \frac{\mathbf{R}\mathbf{f}_v^{(i)}\overline{\mathbf{R}}}{\sqrt{\mathbb{E}_j[\|\mathbf{R}\mathbf{f}_v^{(j)}\overline{\mathbf{R}}\|^2] + \epsilon}} \\
&= \frac{\mathbf{R}\mathbf{f}_v^{(i)}\overline{\mathbf{R}}}{\sqrt{\mathbb{E}_j[\|\mathbf{f}_v^{(j)}\|^2] + \epsilon}} \\
&= \mathbf{R}\frac{\mathbf{f}_v^{(i)}}{\sqrt{\mathbb{E}_j[\|\mathbf{f}_v^{(j)}\|^2] + \epsilon}}\overline{\mathbf{R}} \\
&= \mathbf{R}norm(\mathbf{f}_v^{(i)})\overline{\mathbf{R}}.
\end{aligned} \tag{5}
$$

C.4 Max-Pooling Operation

The revised max-pooling operation, $maxPool(\mathbf{f}) = \mathbf{f}_{\hat{v}}$, s.t. $\hat{v} = \arg\max_{v=1,\ldots,d.}[\|\mathbf{f}_v\|]$, is rotation-equivariant, because $maxPool(\mathbf{f}^{(\theta)}) = \mathbf{R}maxPool(\mathbf{f})\overline{\mathbf{R}}$. The proof is given as follows.

$$
\begin{aligned}
maxPool(\mathbf{f}^{(\theta)}) &= \mathbf{R}\mathbf{f}_{\hat{v}}\overline{\mathbf{R}} \quad \text{s.t.} \quad \hat{v} = \underset{v=1,\ldots,d.}{\arg\max}[\|\mathbf{R}\mathbf{f}_v\overline{\mathbf{R}}\|] \\
&= \mathbf{R}\mathbf{f}_{\hat{v}}\overline{\mathbf{R}} \quad \hat{v} = \underset{v=1,\ldots,d.}{\arg\max}[\|\mathbf{f}_v\|] \\
&= \mathbf{R}maxPool(\mathbf{f})\overline{\mathbf{R}}.
\end{aligned} \tag{6}
$$

D Element-Wise Max-Pooling Operation

In point cloud processing, a special element-wise max-pooling operation is widely used for aggregating a set of neighboring points' features into a local feature. Let $f \in \mathbb{R}^{D\times K}$ denote the features of K neighboring points. Each element of f, *i.e.* $f_k \in \mathbb{R}^D$, denotes the feature of a specific neighboring point. Let $f^{\text{upper}} \in \mathbb{R}^D$ denote the output local feature. The element-wise max-pooling operation is formulated as follows.

$$
\begin{aligned}
\mathbf{MAX}(f) &= \mathbf{MAX}\begin{bmatrix} f_{11} & \cdots & f_{1K} \\ \vdots & \ddots & \vdots \\ f_{D1} & \cdots & f_{DK} \end{bmatrix} \\
&\overset{\textbf{define}}{=\!=\!=} < \max_{k=1,\ldots,K} f_{1k}, \ldots, \max_{k=1,\ldots,K} f_{Dk} >^\top
\end{aligned} \tag{7}
$$

To extend this special max-pooling operation to be suitable for quaternion operations, we replace each max operation $\max_{k=1,\ldots,K} f_{dk}$ in Equation (7) by $\max_{k=1,\ldots,K}[\|\mathbf{f}_{dk}\|]$, where $\mathbf{f}_{dk} \in \mathbb{H}$ is a quaternion.