Equivariant Point Network for 3D Point Cloud Analysis Supplementary Material

1. Proofs of equivariance

In this section, we provide proofs of SE(3) equivariance to the convolution introduced in the main text. Recall that the SE(3) space can be factorized into the space of 3D rotation $\{R|R \in \text{SO(3)}\}$ and 3D translation $\{\mathcal{T}|\mathcal{T} \in \mathbb{R}^3\}$. A convolution operator equivariant to SE(3) must therefore satisfy:

$$\begin{aligned} \forall \mathcal{R} \in \text{SO(3)}, \mathcal{R}(\mathcal{F} * h)(x, g) = (\mathcal{R}\mathcal{F} * h)(x, g), \\ \forall \mathcal{T} \in \mathbb{R}^3, \mathcal{T}(\mathcal{F} * h)(x, g) = (\mathcal{T}\mathcal{F} * h)(x, g). \end{aligned} \tag{1}$$

Theorem 1. The continuous convolution operator

$$\begin{aligned} &(\mathcal{F} * h)(x, g) \\ &= \int_{x_i \in \mathbb{R}^3} \int_{g_j \in \text{SO(3)}} \mathcal{F}(x_i, g_j) h(g^{-1}(x - x_i), g_j^{-1} g) \end{aligned} \tag{2}$$

is equivariant w.r.t. rotation $\mathcal{R} \in \text{SO(3)}$ and translation $\mathcal{T} \in \mathbb{R}^3$

Proof. Firstly, we prove that Eq.(2) is equivariant to 3D rotation. For convenience of notation, let $x_i' = \mathcal{R}^{-1} x_i$, and $g_j' = \mathcal{R}^{-1} g_j$.

$$\begin{aligned} &\mathcal{R}(\mathcal{F} * h_1)(x, g) = (\mathcal{F} * h_1)(\mathcal{R}x, \mathcal{R}g) \\ &= \int_{x_i \in \mathbb{R}^3} \int_{g_j \in \text{SO(3)}} \\ &\qquad \mathcal{F}(x_i, g_j) h((\mathcal{R}g)^{-1}(\mathcal{R}x - x_i), g_j^{-1}\mathcal{R}g) \\ &= \int_{x_i \in \mathbb{R}^3} \int_{g_j \in \text{SO(3)}} \\ &\qquad \mathcal{F}(x_i, g_j) h(g^{-1}(x - \mathcal{R}^{-1}x_i), (\mathcal{R}^{-1}g_j)^{-1} g) \\ &= \int_{x_i' \in \mathbb{R}^3} \int_{g_j' \in \text{SO(3)}} \\ &\qquad \mathcal{F}(\mathcal{R}x_i', \mathcal{R}g_j') h(g^{-1}(x - x_i'), g_j'^{-1} g) \\ &= (\mathcal{R}\mathcal{F} * h_1)(x, g). \end{aligned}$$

Then, we prove that Eq.(2) is equivariant to 3D translation. Let $x_i' = \mathcal{T}^{-1} x_i$. Because $\mathcal{T}(x - x_i) = x - x_i$:

$$\begin{aligned} &\mathcal{T}(\mathcal{F} * h_1)(x, g) = (\mathcal{F} * h_1)(\mathcal{T}x, g) \\ &= \int_{x_i \in \mathbb{R}^3} \int_{g_j \in \text{SO(3)}} \\ &\qquad \mathcal{F}(x_i, g_j) h(g^{-1}(\mathcal{T}x - x_i), g_j^{-1} g) \\ &= \int_{x_i \in \mathbb{R}^3} \int_{g_j \in \text{SO(3)}} \\ &\qquad \mathcal{F}(x_i, g_j) h(g^{-1}\mathcal{T}(x - \mathcal{T}^{-1}x_i), g_j^{-1} g) \\ &= \int_{x_i' \in \mathbb{R}^3} \int_{g_j \in \text{SO(3)}} \\ &\qquad \mathcal{F}(\mathcal{T}x_i', g_j) h(g^{-1}(x - x_i'), g_j^{-1} g) \\ &= (\mathcal{T}\mathcal{F} * h_1)(x, g). \end{aligned}$$

The continuous convolution operator is therefore SE(3) equivariant. Given a finite point set $\mathcal{P}$ and a finite rotation group G, the SE(3) separable convolution consists of two discrete convolution operators:

$$(\mathcal{F} * h_1)(x, g) = \sum_{x_i \in \mathcal{P}} \mathcal{F}(x_i) h_1(g^{-1}(x - x_i), g) \tag{3}$$

$$(\mathcal{F} * h_2)(x, g) = \sum_{g_j \in G} \mathcal{F}(g_j) h_2(x, g_j^{-1} g) \tag{4}$$

For convenience, we use an equivalent definition in the following proof:

$$(\mathcal{F} * h_1)(x, g) = \sum_{x_i \in \mathcal{P}} \mathcal{F}(x_i, g) h_1(g^{-1}(x - x_i)) \tag{5}$$

$$(\mathcal{F} * h_2)(x, g) = \sum_{g_j \in G} \mathcal{F}(x, g_j) h_2(g_j^{-1} g) \tag{6}$$

Theorem 2. The discrete convolution operators given in Eq.(5),(6) are equivariant w.r.t. rotation $\mathcal{R} \in G$ and translation $\mathcal{T} \in \mathbb{R}^3$

Again, we first prove that the two operators are equivariant to 3D rotations in the rotation group G. Following the notations used in the previous proof, let $\mathcal{P}_\mathcal{R} = \{x_i' | x_i' = \mathcal{R}x, x \in \mathcal{P}\}$, $G_\mathcal{R} = \{g_j' | g_j' = \mathcal{R}^{-1} g, g \in G\}$:

$$
\begin{aligned}
&\mathcal{R}(\mathcal{F} * h_1)(x, g) = (\mathcal{F} * h_1)(\mathcal{R}x, \mathcal{R}g) \\
&= \sum_{x_i \in \mathcal{P}} \mathcal{F}(x_i, \mathcal{R}g) h_1((\mathcal{R}g)^{-1}(\mathcal{R}x - x_i)) \\
&= \sum_{x_i \in \mathcal{P}} \mathcal{F}(x_i, \mathcal{R}g) h_1(g^{-1}(x - \mathcal{R}^{-1}x_i)) \\
&= \sum_{x_i' \in \mathcal{P}_{\mathcal{R}}} \mathcal{F}(\mathcal{R}x_i', \mathcal{R}g) h_1(g^{-1}(x - x_i')) \\
&= (\mathcal{R}\mathcal{F} * h_1)(x, g).
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{R}(\mathcal{F} * h_2)(x, g) = (\mathcal{F} * h_2)(\mathcal{R}x, \mathcal{R}g) \\
&= \sum_{g_j \in G} \mathcal{F}(\mathcal{R}x, g_j) h_2(g_j^{-1}\mathcal{R}g) \\
&= \sum_{g_j' \in G_{\mathcal{R}}} \mathcal{F}(\mathcal{R}x, \mathcal{R}g_j') h_2(g_j'^{-1}g) \\
&= (\mathcal{R}\mathcal{F} * h_2)(x, g).
\end{aligned}
$$

We then prove that the two operators are equivariant to 3D translation. Let $x_i' = \mathcal{T}^{-1}x_i$:

$$
\begin{aligned}
&\mathcal{T}(\mathcal{F} * h_1)(x, g) = (\mathcal{F} * h_1)(\mathcal{T}x, g) \\
&= \sum_{x_i \in \mathcal{P}} \mathcal{F}(x_i, g) h_1(g^{-1}(\mathcal{T}x - x_i)) \\
&= \sum_{x_i \in \mathcal{P}} \mathcal{F}(x_i, g) h_1(g^{-1}\mathcal{T}(x - \mathcal{T}^{-1}x_i)) \\
&= \sum_{x_i \in \mathcal{P}} \mathcal{F}(x_i, g) h_1(g^{-1}(x - \mathcal{T}^{-1}x_i)) \\
&= \sum_{x_i' \in \mathcal{T}^{-1}\mathcal{P}} \mathcal{F}(\mathcal{T}x_i', g) h_1(g^{-1}(x - x_i')) \\
&= (\mathcal{T}\mathcal{F} * h_1)(x, g).
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{T}(\mathcal{F} * h_2)(\mathcal{T}x, g) = (\mathcal{F} * h_2)(\mathcal{T}x, g) \\
&= \sum_{g_j \in G} \mathcal{F}(\mathcal{T}x, g_j) h_2(g_j^{-1}g) \\
&= (\mathcal{T}\mathcal{F} * h_2)(x, g).
\end{aligned}
$$

Since both operators are SO(3) equivariant and translation equivariant, we have:

$$
\begin{aligned}
\mathcal{R}((\mathcal{F} * h_1) * h_2)(x, g) &= (\mathcal{R}(\mathcal{F} * h_1) * h_2)(x, g) \\
&= ((\mathcal{R}\mathcal{F} * h_1) * h_2)(x, g), \\
\mathcal{T}((\mathcal{F} * h_1) * h_2)(x, g) &= (\mathcal{T}(\mathcal{F} * h_1) * h_2)(x, g) \\
&= ((\mathcal{T}\mathcal{F} * h_1) * h_2)(x, g).
\end{aligned}
$$

Thus, the SE(3) separable convolution is equivariant w.r.t. rotation $\mathcal{R} \in G$ and translation $\mathcal{T} \in \mathbb{R}^3$, which approximates equivariance to SE(3).

2. Network Architecture and Parameters

The network architecture used in both experiments is illustrated in Figure 1. Input points ($\mathcal{P} \in \mathbb{R}^{N \times 3}$) are first lifted to features that are defined in the SE(3) space ($\mathcal{F}(x_i, g_j) : \mathbb{R}^3 \times G \to \mathbb{R}$), by assigning rotation group to each point and setting its associated features to be constant 1s (denoting occupied space). Therefore, in the first layer, the network learns to differentiate different input points by the kernel correlation function (Equation 7 in the main text). The layer after the separable convolutional layers is an MLP layer with a symmetry function (average function) that aggregates features in the spatial dimension. We have introduced this layer as a function with implicit kernel formulation (see Equation 8 in the main text). Before the fully connected layers, a separate branch of unitary convolution takes the spatially pooled feature defined in SO(3), and outputs the attention confidence (see Section 3.3 in the main text). The output feature of the network can be further processed by a softmax layer in the classification task, or an l_2 normalization in the shape matching task.

3. More Implementation Details

In the implementation of SE(3) point convolution, we follow the design principles in [5] to compute a spatially hierarchical local structure of the points, by subsampling the input points with furthest point sampling and obtaining spatial local neighborhood by the ball searching algorithm. For the explicit point kernel function, we select a kernel size of 24 with kernel points evenly distributed inside a ball $\mathcal{B}_r^3$. The radius r of grouping operator is set as $r^2 = d\sigma$, where d is a parameter related to the density of the input points, σ is a parameter used in the correlation function $\kappa(y, \tilde{y})$ described in Section 3.1 in the main text. The kernel radius r_k is set as $r_k = 0.7r$.

To achieve more effective rotation group convolution, inspired by [1], we choose to sample the rotation group from an axis-aligned 3D model of regular icosahedral. Each face normal of the icosahedral provides the α and β angles. We additionally sample three γ angles for each face normal, each separated by 120 degrees. The cardinality of the rotation set is thus $20 \times 3 = 60$. Thanks to the icosahedral symmetry, the set of rotation forms a rotation group G with closure, associativity, identity and invertibility. For band-limited filters, a 12-element subgroup is chosen, which transforms each element of G to its SO(3) neighbors.

It is thus worth noting that while we maintain a sparse representation for the spatial dimension of the point set, which takes online computation to find its local structure in the spatial dimension, the rotation group naturally possesses a closed grid-like structure. This greatly facilities the computation for the band-limited group convolution.

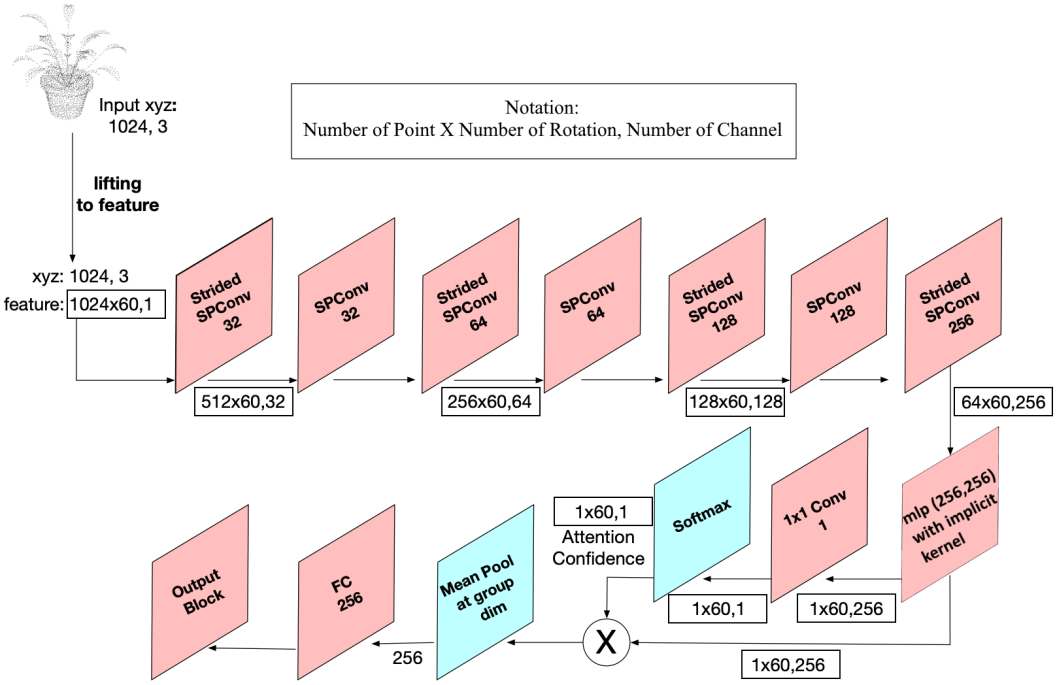

Figure 1: An illustration of the network architecture used in both ModelNet and 3DMatch experiments.

4. Details on Experiment Setup

4.1. Details on ModelNet Training

For each training object, we randomly sample 1,024 points from the input point cloud. We train our network with Adam optimizer. The learning rate is set to 0.001 and the batch size is 16. The model is trained for 150 epochs with an exponential decay of learning rate by half for every 50 epochs.

4.2. Details on 3DMatch Training

The training set of 3DMatch consists of RGB-D images in sequences from 62 indoor scenes. We denote a fused sequence of RGB-D images and its converted point cloud as a fragment of a scene. To generate training examples, we follow [4] to first fuse the RGB-D images into fragments. Then we convert the fragments to point clouds and select pairs that have more than 30% overlapping region, given the ground-truth camera transformations. Therefore, each pair of the fragments comes from the same indoor scene. The point cloud patches used as input to the network are generated by gathering N=1,024 points within a support region whose radius is set to 0.4m.

We train a Siamese network that extracts features from the source and target point clouds in parallel. Inspired by [2], the network learns from a Batch Hard (BH) triplet loss, where negative examples are target patches ($\mathcal{X}_i^p$) in a minibatch that do not correspond to the source patch ($\mathcal{X}_i^a$):

$$\mathcal{L}_{BH}(\mathcal{X}) = \frac{1}{\mathcal{X}} \sum_{i=1}^{|\mathcal{X}|} \max(0, ||f(\mathcal{X}_i^a) - f(\mathcal{X}_i^p)||_2 - \min_{\substack{j=1\ldots|\mathcal{X}| \\ j\neq i}} ||f(\mathcal{X}_i^a) - f(\mathcal{X}_j^p)||_2 + m),$$

where m is the margin for the triplet loss. We use a batch size of 16 for a mini-batch, which contains pairs of point cloud patches from the same pair of partially overlapped fragments. The model is trained for 30 epochs with an exponential decay of learning rate by half for every 6 epochs. The choice of optimizer and all other hyperparameters remain consistent with the classification network.

4.3. Inference speed.

We compare our model used in the experiment to the baseline models that employ similar equivariant structures regarding the inference time. Specifically, we evaluated our 20-anchor model to align with the settings in [1, 3]. Among the selected baselines, [1, 3] are multi-view image networks that are SO(3) equivariant; TFN [6] is an example of "non-separable" SE(3) equivariant network. Our network is found to be faster than all of the baselines selected, and it is significantly faster than the SE(3) equivariant framework that is not separable.

Method	OURS-20	EMVN-20	RotationNet	TFN
Time	**35.4ms**	35.9ms	108.0ms	302.9ms

References

[1] Carlos Esteves, Yinshuang Xu, Christine Allen-Blanchette, and Kostas Daniilidis. Equivariant multi-view networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1568–1577, 2019. 2, 3

[2] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5545–5554, 2019. 3

[3] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5010–5019, 2018. 3

[4] Lei Li, Siyu Zhu, Hongbo Fu, Ping Tan, and Chiew-Lan Tai. End-to-end learning local multi-view descriptors for 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1919–1928, 2020. 3

[5] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 2

[6] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018. 3