Efficient Continuous Group Convolutions for Local SE(3) Equivariance in 3D Point Clouds

Lisa Weijler
TU Wien, Austria

Pedro Hermosilla
TU Wien, Austria

Abstract

Extending the translation equivariance property of convolutional neural networks to larger symmetry groups has been shown to reduce sample complexity and enable more discriminative feature learning. Further, exploiting additional symmetries facilitates greater weight sharing than standard convolutions, leading to an enhanced network expressivity without an increase in parameter count. However, extending the equivariant properties of a convolution layer comes at a computational cost. In particular, for 3D data, expanding equivariance to the SE(3) group (rotation and translation) results in a 6D convolution operation, which is not tractable for larger data samples such as 3D scene scans. While efforts have been made to develop efficient SE(3) equivariant networks, existing approaches rely on discretization or only introduce global rotation equivariance. This limits their applicability to point clouds representing a scene composed of multiple objects. This work presents an efficient, continuous, and local SE(3) equivariant convolution layer for point cloud processing based on general group convolution and local reference frames. Our experiments show that our approach achieves competitive or superior performance across a range of datasets and tasks, including object classification and semantic segmentation, with negligible computational overhead. The code for our implementation is available at this repository.

1. Introduction

In 3D vision, point clouds are the most commonly used representation to process 3D data given that they are relatively cheap to capture and process. This representation is composed of multiple point coordinates, with additional attributes such as color or normal vector, from samples on the surface of 3D objects. In the past years, several neural network architectures have been proposed to process such data [2, 6, 17, 24, 26, 31]. Approaches learning directly from 3D data often take inspiration from the success in 2D vision and address two of the main challenges in such data representation, order invariance and translation equivariance. Yet, 3D data entails more variations and complex group transformations due to an increased number of degrees of freedom (DoF); for the roto-translation group SE(3), DoF $= 6$. Objects in 3D space do not have a predefined canonical orientation and many rotational variances are present.

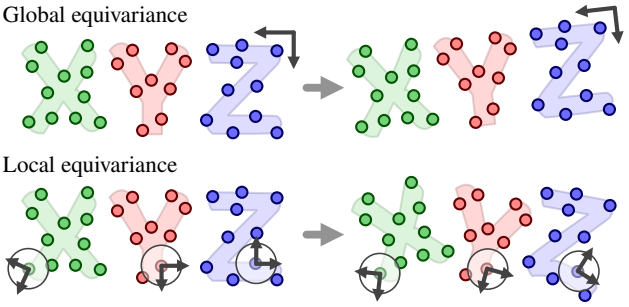

Figure 1. While **global equivariant** designs ensure robustness to whole-scene rotations, they fail with randomly rotated scene parts or elements. In contrast, **local equivariant** operations maintain robustness by handling local geometry rotations around each point.

Equivariance is the property of an operator that allows the prediction of the transformation of the output given an input transformation, while group-invariant operators produce identical features under various group transforms of the input. The latter can be seen as an information loss; they struggle to differentiate between unique instances with internal symmetries, e.g., "8" vs. "∞". Baking SE(3)-equivariance into the network architecture can thus be beneficial since equivariant features maintain information about the input group transform across neural layers, making them more expressive and generalizable by capturing the variance that is present in the data.

Traditionally, to obtain such properties, data augmentation techniques are used, but this requires neural networks to store latent orientations of the objects, limiting the network capacity. Whilst this might be a viable solution for network architectures for 2D images, neural networks for 3D point clouds usually require large amounts of memory, limiting the number of parameters of the models and

making it impossible to achieve such equivariance from the data. Recent advances have been made to address this problem [13, 23], where several neural network architectures can match or even surpass the performance of the standard architectures relying on data augmentations.

Unfortunately, many of those solutions only address the problem of global rotation equivariance, *i.e*. rotations of a single 3D object or scene as a whole. 3D objects or scenes are composed of multiple parts or objects that can have arbitrary orientations w.r.t. each other, see Fig. 1. The relative orientations of different objects in the scene cannot be captured by global equivariance as obtained by existing architectures or by data augmentation techniques.

Group convolution is an operation that is, per definition, equivariant to a specific group and, hence, capable of coping with such problems. These operations aggregate information from neighboring samples, not only from the translation group T(3) as standard convolutions, but from the rotation and translation group SE(3). By restricting the receptive field of these operations, they become rotation equivariant w.r.t. the local geometry inside the receptive field, allowing them to be equivariant to the relative rotations of different parts of the scene. To successfully compute such operations in the continuous domain a complex integral over the full group needs to be solved (6D convolution), which makes such operations not practical for large networks since it has large memory and computational burden. Further, defining a grid on SE(3) is not trivial, where recent works try to address these problems by using Monte Carlo (MC) integration [14] or by discretizing the SO(3) group [8, 38]. However, as we will show, these approximations limit the performance of the network.

In this paper, we propose using a finite subset $\mathcal{F}(x) \subset$ SE(3), referred to as a frame, to solve the group equivariant integral, which allows for exact equivariance (as opposed to approaches based on MC sampling or discretization), while reducing the computational burden. The elements $g \in \mathcal{F}(x)$ can be seen as Local Reference Frame (LRF), that together with their corresponding point x build a grid on SE(3), where the integral of the group convolution can be computed efficiently. Further, our approach stochastically samples $g \in \mathcal{F}(x)$ during training with only a few samples, two or even one, which reduces the additional computational and memory burden significantly and for the case of one sample to almost zero. Our extensive experiments show that such group convolution is able to achieve local rotation equivariance, surpassing other local equivariant designs by a large margin. Moreover, our experiments also show that a network constructed using such convolutions as building blocks is able to be robust to local transformations not seen during training, where popular global equivariant frameworks fail.

2. Related work

This section gives an overview of networks that can process unstructured data, such as point clouds, with a focus on specific point architectures that are equivariant to rotations.

Point-based neural networks. The first neural network architecture specifically designed to process point clouds was PointNet [24]. The idea of directly processing point clouds was followed by several works incorporating concepts and designs from convolutional neural networks for images into the continuous domain. Atzmon *et al*. [2], Thomas *et al*. [26], and Boulch *et al*. [6] propose a convolution operation based on a set of points located inside a receptive field and a correlation function as the kernel function. Another line of research, including Hermosilla *et al*. [17] and Wu *et al*. [31] uses a convolution operation with a kernel function represented by an Multi-Layer Perceptron (MLP), that takes the relative position between points as input. In this work, similarly, we use an MLP as our kernel, but with the relative orientation in addition to the relative position between points as input.

Rotation equivariant point networks. Equivariance or invariance to SE(3) can be achieved by modifying the model's input through data augmentation or by adapting the internal operations of the model to have such equivariance by construction. Several works have designed equivariant network architectures by aligning the input point cloud to a reference frame before being processed by the model. Gojcic *et al*. [16] align local patches of a point cloud to their LRF defined by the normal to solve the task of keypoint matching. Xiao *et al*. [33] uses Principal Component Analysis (PCA) to build a frame to transform the input point cloud and an attention network to aggregate features over the different transformations. Other works, instead, adopt a different approach, in which they compute invariant local features and use these as input to a convolution operation. Zhang *et al*. [35] use angles and distances as input to a local PointNet architecture to achieve rotation invariance. Later, Zhang *et al*. [37] extended this work with additional local features. Yu *et al*. [34] also used distances and angles to achieve global equivariance. However, none of these approaches achieved the goal of our work, local SE(3) equivariance. Recently, Puny *et al*. [23] suggested a general framework to achieve equivariance on any neural network by averaging the model's output over a subset of the group elements. Concurrently with our work, Atzmon *et al*. [3] introduced a piecewise E(3) equivariant approach applying [23] to multiple parts proposed by a partition prediction model, where the locality is determined by the partition of the objects. In this work we use the concepts of [23] to achieve local SE(3) equivariance efficiently without the need for a partition prediction.

Another line of work achieves local equivariance by making the model's internal features *steerable* [10], i.e.,

the feature values transform predictably as the input transforms. In 3D, these works usually rely on the theory of spherical harmonics to obtain *steerable* features [15, 27, 29]. Vector Neurons [13] also uses higher-order features, representing each feature as a 3D vector, to achieve global SO(3) equivariance. Unfortunately, this increases the memory consumption of the models and restricts the kernel representation used.

More related to our work is the concept of group convolutions [9]. These operations generalize the concept of convolutions and extend the equivariance to the translation group of standard convolution to any group. These ideas have been applied to the SE(3) group for voxelized representations, where the group has a finite number of elements [30], and to point clouds in the continuous domain by discretizing the continuous SE(3) group using the icosahedral group [8, 38]. Although the computation of such group convolution can be implemented with permutation matrices, a large number of group elements requires a significant computational burden. To address this issue, Chen *et al*. [8] proposed a separable convolution allowing for fast computation of the group convolution. Zhu *et al*. [38] instead proposes to use the SO(2) group as the stabilizer subgroup to form spherical quotient feature fields. Unfortunately, these discretizations require lifting the feature representation to the size of the discrete group used, increasing the memory requirement of the model by a factor equal to the size of the discrete group. Recent works have suggested solving the group convolution integral by using MC sampling on the continuous group [14, 18]. These works randomly sample the group and use farthest point sampling to select a subset from which the integral is approximated with MC integration. However, this approach might require large samples to obtain a reasonable estimation of the integral and hence suffer from substantial memory load.

In this work, we also suggest using group convolutions on the continuous domain to achieve local rotation equivariance. However, our sampling strategy allows us to solve this integral only with a few samples on the SO(3) group, rendering group convolutions a viable solution to achieve equivariance on standard deep point-based architectures with negligible computational or memory requirements.

3. Methods

In this section, we describe our proposed approach. First, the reader is introduced to the concept of group equivariant convolutions. Then, our efficient continuous group convolution is described in detail.

3.1. Group equvivariant convolution

An intuitive way of thinking about convolutions is the notion of template matching, where a kernel k is shifted over a feature map f to detect patterns. In the continuous case, we consider a feature map $f : X \to \mathbb{R}^c$ as a multi-channel scalar field and $\mathcal{X} = (\mathrm{L}^2(X))^c$ as the space of feature maps over some space X. A more formal definition of a convolution layer is then given as a learnable kernel operator $\Phi : \mathcal{X} \to \mathcal{Y}$ that transforms feature maps f as follows

$$[\Phi f](y) = (f \star k)(y) = \int_X f(x)k(x-y)dx, \qquad (1)$$

with $X = Y = \mathbb{R}^d$, where $d = 3$ for point clouds. (Note that the definition given is cross-correlation instead of convolution since this aligns better with template-matching.) It is well known that convolution layers are translation equivariant due to the shifted kernel, i.e., the kernel is only dependent on relative distances: if the input feature map is shifted, the output feature map follows the same transformation. Yet, since relative distances hold directional information that changes under rotations, it is self-evident that a convolution layer is not equivariant to rotations. One solution is to use $\|x - y\|$ as input to the kernel at the cost of losing the capacity to capture directional features.

We say that an operator Φ is equivariant to a specific Group G if it commutes with group representations on the input and output feature maps, meaning $\forall g \in G : \rho^{\mathcal{Y}}(g) \circ \Phi = \Phi \circ \rho^{\mathcal{X}}(g)$, where $\rho^{\mathcal{X}}(g)$ is the regular group representation of g that transforms a function $f \in \mathcal{X}$ by shifting its domain via g^{-1}. If the output feature map is left unaltered, Φ is G-invariant. Various important works in the field of equivariant deep learning [4, 11, 19] show or conclude that a linear operator Φ that maps between feature maps on homogeneous spaces X, Y of a group G, is G-equivariant iff it is a kernel operator (also often called integral operator) with a single-valued kernel (only dependent on relative values). Further, considering $Y = G/H$ as quotient space with $H = \{g \in G | gy_0 = y_0\}$ as the stabilizer subgroup $\mathrm{Stab}_G(y_0)$, which consists of group elements that leave a chosen origin $y_0 \in Y$ unchanged, the kernel of a G-equivariant Φ must be invariant towards elements of H (invariance constraint).

When looking at the concrete example $Y = \mathbb{R}^d$, $G = \mathrm{SE}(d)$, we say $\mathbb{R}^d \equiv \mathrm{SE}(d)/\mathrm{SO}(d)$ is a quotient space with stabilizer subgroup $\mathrm{SO}(d)$; an intuition is given in the following. Since $\mathbb{R}^d$ is a homogeneous space of $\mathrm{SE}(d)$, every point $x \in \mathbb{R}^d$ can be reached from the origin $\mathbf{0} \in \mathbb{R}^d$ by a group element, a roto-translation, $(\mathrm{t}, \mathrm{R}) \in \mathrm{SE}(d)$. In fact there exist several group elements such that $x = (\mathrm{t}, \mathrm{R})\mathbf{0} = \mathrm{R}\mathbf{0} + \mathrm{t}$, namely any group element with $\mathrm{t} = x$ regardless of the rotation part as any rotation $\mathrm{R} \in \mathrm{SO}(d)$ leaves the origin unchanged. Hence if $Y = \mathbb{R}^d$ and $G = \mathrm{SE}(d)$, the kernel of an $\mathrm{SE}(d)$-equivariant Φ must be $\mathrm{SO}(d)$-invariant, meaning one could only use isotropic kernels, which severely limits the expressivity of patterns that can be detected e.g. using $\|x - y\|$ as discussed above. In order to not limit the representation power of the kernel while achieving $\mathrm{SE}(d)$ equiv-

ariance, the feature maps need to be lifted to the group itself $Y = G$ since then the stabilizer subgroup only consists of the trivial element $H = \{\mathbf{e}\}$, and the kernel is no longer constrained. Note that Y and X do not necessarily have to be the same space. Consequently, to extend the translation equivariance of convolution layers to arbitrary affine Lie groups three types of layers can be used [4]:

- **Lifting layer** ($X = \mathbb{R}^d, Y = G, H = \{\mathbf{e}\}$):

$$(f \star k)(g) = \int_{R^d} f(x) k(g^{-1}x) dx \tag{2}$$

  For G =SE(d), this can be viewed as a template matching various rotated versions of the kernel, creating a feature map for different positions and rotations.
- **Group convolution layer** ($X = G, Y = G, H = \{\mathbf{e}\}$):

$$(f \star k)(g) = \int_{G} f(g') k(g^{-1}g') d\mu(g') \tag{3}$$

  This layer constitutes the convolution on the full group, e.g., it conducts template matching over all possible combinations of positions and rotations from the input and output feature map.
- **Projection layer** ($X = G, Y = \mathbf{R}^d, H = \text{Stab}_G(\mathbf{0})$):

$$(f \star k)(x) = \int_{H} f(x, h') d\mu(h') \tag{4}$$

  For tasks like point-wise classification, the final prediction must be invariant, so feature maps or rotations are projected to their corresponding point in $\mathbb{R}^d$. This layer is omitted for tasks like pose estimation.

3.2. Efficient group convolution

Since group convolution layers map between higher dimensional feature maps and must compute the integral over the entire group, they can introduce a computational bottleneck. In the case of 3D point clouds and the affine group SE(3) $= \mathbb{R}^3 \rtimes$ SO(3), Eq. (3) turns into a 6D convolution $(f \star k)(g)$ with $g = (\text{x}, \text{R}) \in$ SE(3), which can be written as a double integral

$$\int_{\mathbb{R}^3} \int_{\text{SO(3)}} f(\text{t}, \text{R'}) k(\text{R}^{-1}(\text{t} - \text{x}), \text{R}^{-1}\text{R'}) d\text{t} d\mu(\text{R'}), \tag{5}$$

with $\mu(\cdot)$ being the Haar measure on SO(3).

In addition to the computational burden of a 6D convolution, another difficulty lies in how to define a grid on SE(3) or, more specifically, on the SO(3) part to compute the integral of Eq. (5). Previous works such as [8, 38] have relied on the discretization of SO(3) using platonic solids that assign to each spatial component the same finite grid on SO(3) to make it tractable, yet at the loss of continuity and exact equivariance. To stay in the continuous domain, similarly to the work of Finzi *et al.* [14], one can use MC approximation for both the spatial and rotational part to solve the double integral

$$\sum_j \frac{1}{|H'_j|} \sum_{(\text{t}, \text{R'}) \in H'_j} f(\text{t}, \text{R'}) k(\text{R}^{-1}(\text{t} - \text{x}), \text{R}^{-1}\text{R'}), \tag{6}$$

where j are the indices of the points $x_j \in \mathbb{R}^3$ of the point cloud and $H'_j = \{(\text{t}, \text{R'}) | \text{t} = x_j, \text{R'} \in \text{SO(3)}\}$ is the set of SE(3) group elements that result form lifting points x_j to SE(3) by repeating the point coordinate with uniformly sampled rotations. Note that the point cloud is treated as a sparse feature map that defines the sampling of the spatial component.

Using MC approximation can be thought of as defining a random grid on SE(3). Hence, the approximation quality of this integral depends on the number of sampled group elements or, more precisely, on the number of rotations $|H'_j| = O$ sampled per point x_j; the approximation error converges towards zero for $O \to \infty$. However, sampling O rotations per point increases the model's memory by a factor of O. Moreover, the required computations for the convolution also increase by a factor of O^2. Hence, using MC results in a trade-off between computational efficiency and preciseness of equivariance property, showing that an efficient grid on SE(3) that allows for exact equivariance with finite rotation elements is crucial to make continuous group convolutions practical for point-based networks.

Efficient grid on SE(3). To achieve exact equivariance with tractable computational load, we propose a carefully constructed grid $\mathcal{F}(x_j) \subset SE(3)$ specific to each point $x_j \in \mathbb{R}^3$. Note that while H_j in Eq. (6) was also dependent on x_j, the grid was still the same for each point, namely the entire group, where the dependency merely came from approximation by sampling.

We call $\mathcal{F}(x) : \mathbb{R}^3 \to 2^{\text{SE(3)}}$ a frame, which is a set-valued function and maps a point in space to a set of group elements such that $\forall (\text{t}, \text{R}) \in \mathcal{F}(x) : x = \text{t}$. A frame is called G-equivariant if $\forall g \in G : g\mathcal{F}(x) = \mathcal{F}(gx)$. Using $\mathcal{F}(x)$ as grid, we define a 3D sparse point cloud group convolution layer $\Phi_{\mathcal{F}}$ as

$$\sum_j \frac{1}{|\mathcal{F}(x_j)|} \sum_{(\text{t}, \text{R'}) \in \mathcal{F}(x_j)} f(\text{t}, \text{R'}) k(\text{R}^{-1}(\text{t} - \text{x}), \text{R}^{-1}\text{R'}). \tag{7}$$

$\Phi_{\mathcal{F}}$ thus transforms feature maps $f : X \to \mathbb{R}^c$, defined on the domain $X = \{\mathcal{F}(x) | x \in \mathbb{R}^3\}$. Using those definitions, we can formulate the following.

Theorem 1. *Let $\mathcal{F}$ be an SE(3)-equivariant frame. Then, $\Phi_{\mathcal{F}}$ is SE(3)-equivariant.*

Proof. See suppl. mat. □

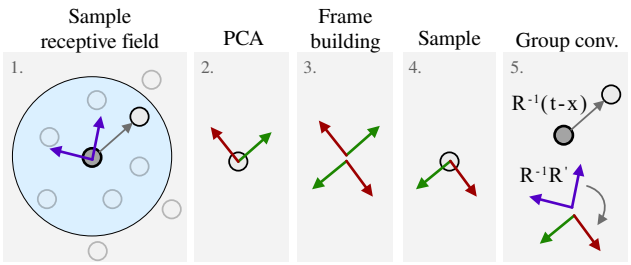

Figure 2. Overview of our convolution operation. Given a central point with an orientation, first, we sample neighboring points. For each point, we use PCA to build a frame from it. Then, we sample an orientation from the frame. Then, the input to the group convolution kernel is the relative position plus the relative orientations between points.

Since $\mathcal{F}(x)$ can be constructed with local PCA, as explained below, it only consists of a few elements and the amount of computations is significantly reduced.

Frame Construction. We compute PCA over a region around the point to construct $\mathcal{F}(x)$. Due to the ambiguity of PCA w.r.t. the direction of the different axes, we follow Xiao *et al*. [33] and Puny *et al*. [23] and construct 4 different LRF by inverting the sign of the different directions. Given the eigenvectors $[v_1, v_2, v_3]$ of the covariance matrix C of the point coordinates, the frame can be defined as $\mathcal{F}(x) = \{([\alpha_1 v_1, \alpha_2 v_2, \alpha_3 v_3], t) | \alpha_i \in \{1, -1\}\}$. To be equivariant to the SE(3) group, $\mathcal{F}(x)$ is restricted to orthogonal, positive rotation matrices. This results in a frame with a finite number of elements, $|\mathcal{F}(X)| = 2^{3-1} = 4$.

Stochastic Approximation. Although $\mathcal{F}(x)$ only has 4 elements, this might still be restrictive for modern state-of-the-art deep architectures used to process large 3D scenes. Therefore, we propose to perform a stochastic approximation of Eq. (7) during training by only sampling a subset of the elements of $\mathcal{F}(x)$ for input and output domains of the feature maps. In particular, we propose randomly sampling two or even only one element of $\mathcal{F}(x)$ for each point x in the point cloud where the convolution will be computed. Then, during the computation of Eq. (7), only the sampled elements for points x_j are used to approximate the SO(3) integral. Our approach is illustrated in Fig. 2.

While using all elements of $\mathcal{F}(x)$ would increase the memory consumption of a standard model by a factor of 4 and the number of computations by a factor of 16, sampling 2 elements would only increase the memory by a factor of 2 and the computations by a factor of 4. More importantly, randomly sampling only 1 element will maintain the memory consumption and computations equal to the model with standard convolutions. During testing, since large batches are not necessary, we can use the full frame $\mathcal{F}(x)$ to compute Eq. (7). The error introduced by stochastic approximation by subsampling 2 or 1 element instead of using all 4 is discussed in the supplementary materials.

Local vs Global Equivariance. In practice, the locality of the kernel is enforced by calculating the convolution for a local neighborhood $N_x = \{x_j \in \mathbb{R}^3 | \|x_j - x\| < r\}$ of x only. Equivariance of Eq. (6) and Eq. (7) is ensured on a scale that depends on the receptive field used. Since we only consider a small receptive field around each point, our operations become equivariant w.r.t. rotations of the local geometry within this receptive field. By incorporating several layers in our architectures with increasing receptive fields, the model is able to capture patterns at different scales in an equivariant manner. Ultimately, the whole model also covers the global equivariance scale since the last layers have an effective receptive field covering the entire scene.

4. Experiments

We conduct experiments on object classification, and semantic segmentation to validate our methods. Due to space constraints, additional experiments, ablation studies, detailed dataset description and implementation are provided in the supplementary materials.

4.1. Baselines.

In our main experiments, we compare our convolution operation, Ours, to the same model where the integral is solved using MC [14], MC, and a model using standard convolutions, STD. Moreover, we also compare to additional rotation equivariant networks, relying on global and local equivariant designs.

4.2. Shape classification

We use the task of shape classification to measure the equivariant capabilities of the models w.r.t. global rotations. For this task, predictions must be invariant of the rotation applied to the model. We use a global pooling operation as the projection layer (Eq. (4)) at the end of our encoder to transform the equivariant features into invariant ones.

Dataset. We use the ModelNet40 dataset [32] since this is a standard benchmark for rotation equivariant networks [13]. Our model only takes as input point coordinates, and performance is measured with overall accuracy.

Experimental setup. We provide different configuration setups in our experiments. All models are evaluated when trained and tested without any rotation, I / I. Further, we evaluate all models trained without any rotation but random rotations during testing, I / SO(3). Lastly, we evaluate our models with random rotations during training and testing. Additionally, to compare to other state-of-the-art methods, we take the commonly used setup where random rotations are applied along the up vector during training and random rotations on SO(3) during testing, z / SO(3). Although this setup is less challenging than I / SO(3), it allows us to compare to additional rotation equivariant models.

Main results. In our main results, we compare our method, Ours, to MC and STD for different samples taken during training and testing.

Table 1 presents the results of this experiment. As expected, we can see that the standard method STD achieves good accuracy for I / I. Ours and MC, as it is typical for rotation equivariant networks in this setup, achieve competitive performance but are below STD. However, when we look at the more challenging setup, I / SO(3), we can see that Ours is able to maintain similar accuracy as in the I / I setup, $86.9\,\%$, a drop by only one point in accuracy, while STD achieves $12.3\,\%$. MC, although it can also achieve competitive performance, for most of the cases, the drop in performance is significant compared to the I / I results. When we look at the SO(3) / SO(3) setup, all three methods achieve good performance; MC and Ours are able to outperform STD, while Ours achieves the best accuracy.

Analyzing the effect of different samples used to compute the integral over SO(3) for training and testing, we can see that Ours, even with 1 sample, can achieve similar results than when using 4 samples. With only 2 samples, our method is able to match or even surpass the accuracy of using the full frame, 4 samples. Moreover, using only 1 or 2 samples appears to be more robust than using the full frame, 4 samples, when tested with different numbers of samples. We hypothesize that training with random 1 or 2 samples, rather than using the full frame, introduces stochasticity that acts as a regularizer, enhancing robustness to errors in SO(3) integral estimation. In contrast, MC is more sensitive to the number of samples, exhibiting significant performance degradation with 1 or 2 samples.

Comparison to other methods. First, we compare our model to existing non-equivariant point-based network architectures, architectures that rely on global equivariance, and models like ours that use group convolutions to achieve local equivariance. In Tab. 2, we can see that our model achieves the best performance among the group convolution tested by a large margin in the I/SO(3) setting. This is due to the discretization of the group SO(3) used by the EPN [8] and E2PN methods [38]. Also, in the z / SO(3) and SO(3) / SO(3) settings, we outperform all local rotation equivariant networks. When compared to global equivariant networks, our method falls behind in the I / SO(3) setup and achieves similar performance on the z / SO(3) and SO(3) / SO(3) setup. However, as we will show later in the segmentation task, while some global equivariant networks only slightly outperform ours on this task, they fail to solve tasks requiring local rotation equivariance.

4.3. Semantic segmentation

In semantic segmentation, incorporating symmetries like SE(3) equivariance is key for generalization, especially due to the varying orientations and part compositions in point clouds. We evaluate our method on body part segmentation and scene understanding.

4.3.1 Human body parts

For semantic segmentation of human body parts, the local equivariance property is essential to distinguish correctly between parts undergoing diverse SE(3) transformations within the kinematic tree. Due to the additional symmetry information, we show that our models can generalize to unseen, out-of-distribution poses.

Dataset. For training and testing, we use two subsets of the AMASS meta-dataset [21], DFAUST [5] and PosePrior [1], respectively. The PosePrior dataset consists of challenging poses significantly divergent from those executed in DFAUST, which we use to test our model for generalization to unseen, out-of-distribution poses.

Experimental setup. To assess the ability of our method to generalize to local transformations, we adopt a setup in which we do not use any rotation during training or testing. Since the testing data is composed of rare poses not seen during training, the models must become invariant to transformations of the different local parts.

Main results. Table 3 presents the results of the main experiment. We can see that STD struggles to generalize to these out-of-distribution poses, achieving a mAcc of 85.3 and mIoU of 74.5. Ours, on the other hand, achieves better performance with 95.0 mAcc and 90.8 mIoU. MC can also achieve competitive performance, but, as in the classification task, this is lower than our proposed approach.

When evaluating the model robustness to the number of samples in the SO(3) integral, Ours outperforms MC in all cases except when trained on 4 samples but tested on one, as seen in the classification task.

Comparison to other methods. In Tbl. 4, we present the results of comparing our method to other global and local equivariant point-based networks. We can see that Ours achieves an impressive performance of 95.0 mAcc and 90.8 mIoU. Contrary to the task of shape classification, global equivariant models struggle to generalize to out-of-distribution local transformations not seen during training. Fig. 3 depicts predictions for different models tested on the dataset. The results show that global equivariant methods such as VN or FA struggle with out-of-distribution models, confusing legs and arms and right and left. The same is true for our non-equivariant version, STD. The training data contain mostly upright positions, i.e., feet are, on average, further down on the z-axis. In contrast, the hands and the head are further up, leading to generalization errors in those models, e.g., the handstand pose as shown in Fig. 3. Ours, on the other hand, achieves predictions comparable to the ground truth annotations despite never seen those extreme poses during training. MC also achieves remarkable per-

Table 1. Results for different configurations for the classification task on the ModelNet40 dataset. The results show that using our sampling approach increases the performance significantly, leading to better results with fewer samples.

Method	# samp.	I / I			I / SO(3)			SO(3) / SO(3)		
	train ↓/ test →	1	2	4	1	2	4	1	2	4
MC	1	85.4	84.6	83.1	78.8	74.1	70.1	86.5	85.6	84.4
	2	86.2	87.0	87.1	80.3	82.3	82.3	87.1	87.0	87.0
	4	84.2	87.4	87.5	78.4	85.6	86.2	85.4	88.3	88.2
Ours	1	86.9	86.8	86.7	85.5	85.3	85.3	88.7	88.5	88.5
	2	87.9	87.9	87.7	86.6	**86.9**	86.8	88.9	88.7	88.7
	4	73.2	87.6	87.8	61.4	85.7	86.5	59.7	**89.0**	88.7
STD			**90.7**			12.3			87.5	

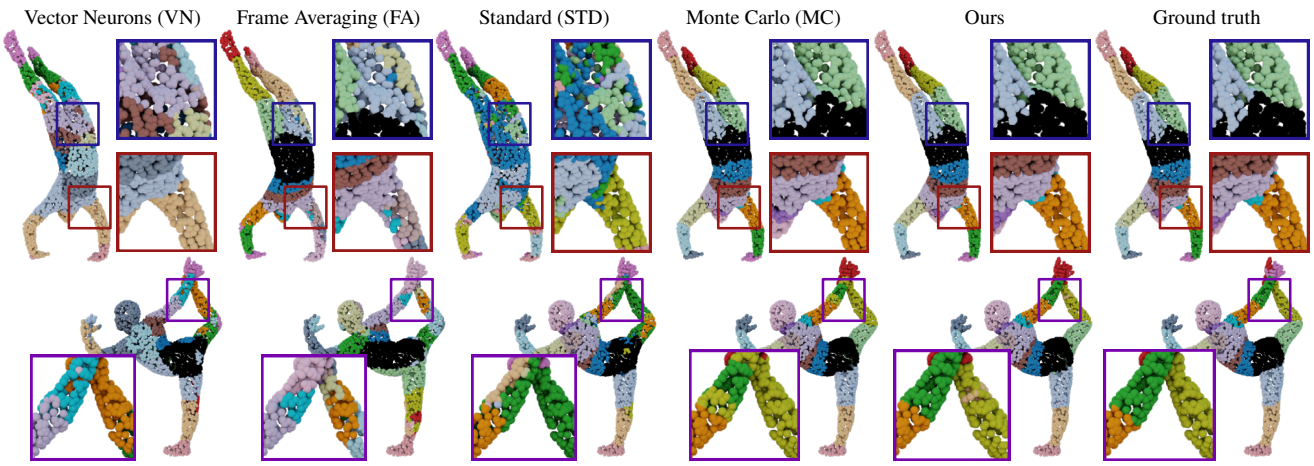

Figure 3. **Qualitative results.** Global equivariant methods such as VN, or FA struggle with out-of-distribution models. Our method, on the other hand, achieves almost perfect predictions. Lastly, MC also achieves good performance but falls behind our method, which better approximates the group convolution integral.

formance but performs several prediction mistakes due to inefficient sampling of Frame elements as Tbl. 3 indicates.

When comparing to current state-of-the-art local equivariant methods, we can see that while they also outperform global equivariant methods by a large margin, our method gives superior results, with E2PN [38] reaching a slightly lower performance.

Tbl. 5 compares a forward pass of a single convolution layer using 1024 points and 256 input and output features. We can see that using only one sample to approximate the integral over SO(3) has approximately similar memory consumption and frames per second (FPS) as the non-SO(3) equivariant version of our model. This shows that with our method, we can introduce the equivariant property without extra costs, demonstrating the efficiency of our proposed model. When we analyze the two-sample version of our group convolution, we can see that memory and computation increase by a factor of 2, still making it suitable for its applicability. When using 4 samples, the memory and computations increase significantly. Compared to other state-of-the-art local rotation equivariant methods, E2PN [38] and EPN [8], the computational resources needed for our approach are significantly lower even when using 4 samples.

4.3.2 Scene understanding

Scenes consist of multiple parts or objects with arbitrary orientations, making local equivariance essential for generalizing to unseen configurations.

Dataset. We test our method on ScanNet [12], a dataset composed of several indoor 3D scene scans, to show its applicability to real-world scenarios.

Experimental Setup. Since our surroundings have a notion of an up orientation, we fix the z-axis and conduct our experiments for SO(2). We sample only one orientation from the frame for all experiments, which does not pose additional memory or computational burden on the model. This is a crucial property for processing such large point clouds, making it intractable for the other methods to run

Table 2. Comparison to equivariant models on the classification task of ModelNet40 for different setups.

Equiv.	Method	I / SO(3)	z / SO(3)	SO(3) / SO(3)
None	PointNet [24]	–	19.6	**84.9**
	PointNet++ [25]	13.8	28.4	**84.9**
	DGCNN [28]	**17.3**	33.8	84.8
	PointCNN [20]	–	**41.2**	84.8
	KPConv [26]	12.7	–	81.2
Global	GC-Conv [36]	–	89.1	89.2
	FA-PointNet [23]	85.9	85.5	85.8
	FA-DGCNN [23]	88.4	88.9	88.5
	VN-PointNet [13]	77.2	77.5	77.2
	VN-DGCNN [13]	**90.0**	**89.5**	**90.2**
Local	TFN [27]	–	85.3	87.6
	ClusterNet [7]	–	86.4	86.4
	RI-Conv [35]	–	86.4	86.4
	SPHNet [22]	–	86.6	87.6
	EPN [8]	32.3	–	87.8
	E2PN [38]	44.4	–	88.6
	Ours	**86.9**	**87.0**	**89.0**

Table 3. Semantic segmentation results for different models trained on DFAUST and tested on PosePrior. By using our sampling approach, mAcc, and mIoU increase significantly with only a few samples of the frame.

Method	# samp.	mAcc			mIoU		
	train ↓/ test →	1	2	4	1	2	4
MC	1	93.1	93.0	92.7	87.7	87.6	87.2
	2	93.8	93.9	93.8	88.8	89.0	88.7
	4	93.4	94.2	94.4	87.9	89.3	89.7
Ours	1	93.8	93.9	93.9	88.9	88.9	89.0
	2	94.3	94.4	94.5	89.7	89.9	89.9
	4	32.6	92.4	**95.0**	21.6	86.8	**90.8**
STD			85.3			74.5	

Table 4. Comparison of our method to other rotation equivariant models on the segmentation task for out-of-distribution poses.

Equiv.	Method	mAcc	mIoU
Global	FA-PointNet [23]	77.4	64.7
	FA-DGCNN [23]	81.7	71.0
	VN-PointNet [13]	63.1	47.5
	VN-DGCNN [13]	61.1	46.6
Local	EPN [8]	89.9	82.3
	E2PN [38]	94.8	90.7
	Ours	**95.0**	**90.8**

reasonable-sized networks for this task.

Main Results. Tbl. 6 shows that our method outperforms STD in all three configurations, underlining the benefits of baking SE(3) equivariance in the model architecture. Compared to MC, we can see that our approach obtains better predictions in all but one configuration.

Table 5. Computational and memory resources of a single convolution layer for our approach and state-of-the-art methods.

Method	# samp.	Mem. (Mb) ↓	FPS ↑
STD		37.1	704.2
Ours	1	37.1	581.4
	2	76.9	432.9
	4	165.2	255.8
E2PN [38]		1211.6	45.0
EPN [8]		1636.4	10.2

Table 6. Results for the semantic segmentation task on ScanNet20 show that using our sampling approach increases the performance.

Method	I / I		I / SO(2)		SO(2) / SO(2)	
	mAcc	mIoU	mAcc	mIoU	mAcc	mIoU
MC	73.4	64.5	**74.1**	65.2	74.2	65.7
Ours	**73.6**	**65.6**	72.7	**65.4**	**75.6**	**67.5**
STD	73.0	64.4	70.9	63.5	74.5	66.4

5. Conclusions

This paper presents an instance of group convolutions on the continuous domain, which is equivariant to SE(3). Using a carefully constructed subset of group elements makes our operation computationally and memory efficient, obtaining competitive performance when only one sample is taken to solve the integral over the SO(3) group and, therefore, not requiring additional resources over a standard convolution. Moreover, by restricting the receptive field of our convolution, our operation becomes local equivariant, allowing us to be robust to local transformations. Our extensive evaluation presents our approach as a viable solution to incorporate local equivariance in deep network architectures for point clouds without significant additional cost.

References

[1] Ijaz Akhter and Michael J Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1446–1455, 2015. 6

[2] M. Atzmon, H. Maron, and Y. Lipman. Point convolutional neural networks by extension operators. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 2018. 1, 2

[3] Matan Atzmon, Jiahui Huang, Francis Williams, and Or Litany. Approximately piecewise e (3) equivariant point networks. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[4] Erik J Bekkers. B-spline cnns on lie groups. In *International Conference on Learning Representations*, 2020. 3, 4

[5] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J Black. Dynamic faust: Registering human bodies in motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6233–6242, 2017. 6

[6] Alexandre Boulch. Convpoint: Continuous convolutions for point cloud processing. *Computers & Graphics*, 2020. 1, 2

[7] Chao Chen, Guanbin Li, Ruijia Xu, Tianshui Chen, Meng Wang, and Liang Lin. Clusternet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 8

[8] Haiwei Chen, Shichen Liu, Weikai Chen, Hao Li, and Randall Hill. Equivariant point network for 3d point cloud analysis. pages 14514–14523, 2021. 2, 3, 4, 6, 7, 8

[9] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016. 3

[10] Taco S Cohen and Max Welling. Steerable cnns. *arXiv preprint*, 2016. 2

[11] Taco S Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant cnns on homogeneous spaces. *Advances in neural information processing systems*, 32, 2019. 3

[12] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *CVPR*, 2017. 7

[13] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12200–12209, 2021. 2, 3, 5, 8

[14] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. *International Conference on Machine Learning*, 2020. 2, 3, 4, 5

[15] Fabian B. Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2020. 3

[16] Zan Gojcic, Caifa Zhou, Jan Dirk Wegner, and Wieser Andreas. The perfect match: 3d point cloud matching with smoothed densities. In *CVPR*, 2019. 2

[17] Pedro Hermosilla, Tobias Ritschel, Pere-Pau Vazquez, Alvar Vinacua, and Timo Ropinski. Monte carlo convolution for learning on non-uniformly sampled point clouds. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2018)*, 2018. 1, 2

[18] Michael J Hutchinson, Charline Le Lan, Sheheryar Zaidi, Emilien Dupont, Yee Whye Teh, and Hyunjik Kim. Lietransformer: Equivariant self-attention for lie groups. In *International Conference on Machine Learning*, pages 4533–4543. PMLR, 2021. 3

[19] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International conference on machine learning*, pages 2747–2755. PMLR, 2018. 3

[20] Yangyan Li, Rui Bu, Mingchao Sun, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2018. 8

[21] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 6

[22] A. Poulenard, M. Rakotosaona, Y. Ponty, and M. Ovsjanikov. Effective rotation-invariant point cnn with spherical harmonics kernels. In *International Conference on 3D Vision (3DV)*, 2019. 8

[23] Omri Puny, Matan Atzmon, Edward J. Smith, Ishan Misra, Aditya Grover, Heli Ben-Hamu, and Yaron Lipman. Frame averaging for invariant and equivariant network design. In *International Conference on Learning Representations*, 2022. 2, 5, 8

[24] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 1, 2, 8

[25] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Conference on Neural Information Processing Systems (NIPS)*, 2017. 8

[26] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. *ICCV*, 2019. 1, 2, 8

[27] 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 preprin*, 2018. 3, 8

[28] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019. 8

[29] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. In *Pro-*

ceedings of the 32nd International Conference on Neural Information Processing Systems, 2018. 3

[30] Daniel Worrall and Gabriel Brostow. Cubenet: Equivariance to 3d rotation and translation. In *ECCV*, 2018. 3

[31] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, 2019. 1, 2

[32] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of 28th IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 5

[33] Z. Xiao, H. Lin, R. Li, L. Geng, H. Chao, and S. Ding. Endowing deep 3d models with rotation invariance based on principal component analysis. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, 2020. 2, 5

[34] Ruixuan Yu, Xin Wei, Federico Tombari, and Jian Sun. Deep positional and relational feature learning for rotation-invariant point cloud analysis, 2020. 2

[35] Zhiyuan Zhang, Binh-Son Hua, David W. Rosen, and Sai-Kit Yeung. Rotation invariant convolutions for 3d point clouds deep learning. In *International Conference on 3D Vision (3DV)*, 2019. 2, 8

[36] Zhiyuan Zhang, Binh-Son Hua, Wei Chen, Yibin Tian, and Sai-Kit Yeung. Global context aware convolutions for 3d point cloud understanding. In *2020 International Conference on 3D Vision (3DV)*, 2020. 8

[37] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. Riconv++: Effective rotation invariant convolutions for 3d point clouds deep learning. *International Journal of Computer Vision*, 130(5):1228–1243, 2022. 2

[38] Minghan Zhu, Maani Ghaffari, William A Clark, and Huei Peng. E2pn: Efficient se (3)-equivariant point network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1223–1232, 2023. 2, 3, 4, 6, 7, 8