Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

SE(3) Group Convolutional Neural Networks and a Study on Group Convolutions and Equivariance for DWI Segmentation

Renfei Liu (renfei.liu@di.ku.dk)
University of Copenhagen
Francois Lauze (francois@di.ku.dk)
University of Copenhagen
Erik Bekkers (e.j.bekkers@uva.nl)
University of Amsterdam
Kenny Erleben (kenny@di.ku.dk)
University of Copenhagen
Sune Darkner (darkner@di.ku.dk)
Department of Computer Science, University of Copenhagen

Article

Keywords:

DOI: https://doi.org/

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: There is **NO** Competing Interest.

$SE(3)$ Group Convolutional Neural Networks and a Study on Group Convolutions and Equivariance for DWI Segmentation

Renfei Liu[1*], Francois Lauze[1], Erik J. Bekkers[2], Kenny Erleben[1] and Sune Darkner[1]

[1]Department of Computer Science, University of Copenhagen, Universitetsparken 1, Copenhagen, Denmark.
[2]Department of Computer Science, University of Amsterdam, Science Park 904, Amsterdam, Netherlands.

*Corresponding author(s). E-mail(s): renfei.liu@di.ku.dk;
Contributing authors: francois@di.ku.dk; e.j.bekkers@uva.nl; kenny@di.ku.dk; darkner@di.ku.dk;

Abstract

We present an ***SE*(3)** Group Convolutional Neural Network along with a series of networks with different group actions for segmentation of Diffusion Weighted Imaging data. These networks gradually incorporate group actions that are natural for this type of data, in the form of convolutions that provide equivariant transformations of the data. This knowledge provides a potentially important inductive bias and may alleviate the need for data augmentation strategies. We study the effects of these actions on the performances of the networks by training and validating them using the diffusion data from the Human Connectome project. Unlike previous works that use Fourier-based convolutions, we implement direct convolutions, which are more lightweight. We show how incorporating more actions - using the ***SE*(3)** group actions - generally improves the performances of our segmentation while limiting the number of parameters that must be learned.

Keywords: Geometric deep learning, Group action, Homogeneous spaces GCNN, Image Segmentation, Diffusion Weighted Imaging

1 Introduction

In this work, we study the influence of group actions on data and how they may impact the architecture and performances of neural networks, especially convolutional neural networks (CNN). CNNs rely on assumed translational symmetries in data and have shown very robust performance in imaging tasks, especially medical imaging ones, and they are highly parameter-efficient thanks to their weight-sharing property. When data offer more structure than simply translation, this can be used to build generalized CNNs. This is especially the case for the task at hand - classification and segmentation of Diffusion Weighted Imaging (DWI) data. These Group and Geometric CNNs (GCNN) have been studied intensively and applied in many situations in the few past years ([1–5] to cite a few).

DWI is a non-invasive image modality that provides local information about water diffusion in tissues by means of measuring spins displacement [6]. It provides 3-dimensional diffusion information at each location x that can be encoded as a function f_x on the 2-dimensional sphere $\mathbb{S}^2$. A field of these functions, on a given domain, can be represented as a function $f : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$. If a sample is rotated and translated, the acquired signal should reflect, up to the limitations of acquisition protocol, this transformation. The group in question is the group of 3D rigid motions, $SE(3)$, and the space $\mathbb{R}^3 \times \mathbb{S}^2$ is a *homogeneous space* under the action of $SE(3)$: a point in $\mathbb{R}^3 \times \mathbb{S}^2$ can be transformed in any other point by a rigid transformation. This notion of homogeneous space is at the heart of the extension of CNNs to GCNNs [5, 7].

Our task at hand is the classification/segmentation of diffusion data. The inductive bias provided by the knowledge of these transformations may prove important for our task, especially when the amount of annotated data is limited. The problem boils down to how to incorporate this knowledge. The most classical approach is to use data augmentation, reflecting the expected symmetries in the data, in the hope that the network will be able to learn it during the training phase, learning symmetry-aware kernels.

Incorporating, on the other hand, some information about the symmetries of the data in the model has been shown to boost the performances of these networks [4]. But how much of this information is needed for a given task? To provide an answer, for the DWI segmentation task, we propose several networks, which gradually incorporate these symmetries in their architecture and study their performances. In addition, instead of performing convolution on non-Euclidean data in a spectral fashion using Fourier-type transformations, we implement convolution in all our experiments in a direct way, as is usually done in the image analysis community. In other words, we use regular representations of groups to encode the group actions in the models, instead of irreducible representations. Our experiments, in some sense, perform a *group action ablation study*. We start with a "naive" CNN, then incorporate spherical symmetries, resulting in a $SO(3)$-GCNN, discarding the spatial aspect of the data. The spatial aspect is then added in the form of a standard CNN coupled with spherical symmetries and then we build a network where

roto-translational transformations are used in almost all steps. This work demonstrates empirically the improvement in performances. The results are, however, not always clear-cut. The GCNN built from 3D-translations on one hand and rotations on the other hand seems to perform better than a $SE(3)$-GCNN. However, the $SE(3)$-network generalizes better to unseen rotated data than the previous one. The reason may lie in the particular type of data used - our DWI scans come from the Human Connectome Project (HCP) [8] are highly preprocessed, including a form of alignment – and this may impact the results. Nevertheless, for every model we propose, we also experiment training them with data augmentation to compare with our equivariant networks. We show that the more equivariance we incorporate into the model, the better the model resists the inconsistency of distributions between training and testing data.

This work is an extension of our previous work [9] with detailed theoretic formulation of the proposed method and an ablation study of different group actions in different spaces and the combinations of these actions with additional experiments using data augmentation, as well as comparison to [10], which, to our knowledge, is the only other existing work that does tissue classification from DWI data using SE(3) group convolutions.

In the rest of this paper, we review related work, both around CNN and DWI classification problem. Then we introduce the theoretical setup of GCNN and build several networks. Thereafter we study and discuss their performances.

2 Related Work

Deep Learning (DL) for non-flat data, or using more complex group actions than just translations, is currently getting more attention from the research field. When it comes to non-flat data, such as the point-wise spherical signals in DWI, particularly relevant related works are the following. [1] proposed a NN on surfaces that extracts local rotationally invariant features. A non-rotationally invariant modification was proposed by [3]. The above provide methods for DL-based processing of data on arbitrary manifolds. When the manifold, however, is a homogeneous space, i.e., there is a group action by which any two points on the manifolds can be reached, theory simplifies via a natural generalization of classical convolutions in group convolution neural networks (GCNNs), as was presented in [4, 11, 12]. GCNNs guarantee global equivariance. However, global equivariance can be complicated and elusive when the underlying geometry is non-trivial, which was discussed in [13]. An elementary construction on a general manifold is proposed by [14] via a fixed choice of geodesic paths used to transport filters between points on the manifold, ignoring the effects of path dependency, i.e. holonomy when paths are geodesics. The removal of this path dependency can be obtained by summarizing local responses over local orientations, which is what was done by [1]. To explicitly deal with holonomy, [15] proposed a theoretical breakthrough using

convolution construction on manifolds based on stochastic processes via the frame bundle.

On the other hand, [11] lifted spherical functions to the 3D-rotation group $SO(3)$ and used a generalization of Fourier transform on it to perform convolution. [16] proposed an equivariant spherical deconvolution method to learn the orientation distribution function (ODF). [17] generalized convolution to manifold-valued convolutions using Volterra Series, preserving its equivariance. With the generalization of convolution to more complex group actions than translation, several authors [2, 4, 12, 18–26] explored the group convolution path for Lie groups and the homogeneous spaces of these groups. [27] proposed a separable convolution setup on Lie groups. The relation between group actions, principal bundles and related vector bundles, and convolutional architectures is currently explored [5, 13, 28]. The latter elucidates important relations between differential geometry of bundles and Reproducible Kernel Hilbert Spaces. Links between partial differential equations, symmetries and GCNN is studied in [29]. A unifying framework for equivariant DL on manifolds, connecting both the bundle and homogeneous space viewpoint, is given in [30] through a notion of coordinate indepencent convolutions.

Most CNNs approach for the processing of DWI signals discard its specific structure. For instance, [31] built multi-layer perceptrons in q-space for kurtosis and NODDI mappings. However, the importance of spherical equivariant or invariant structure has been acknowledged for some years now. The importance of the extraction of rotationally invariant features beyond Fractional Anisotropy [32] has been recognized in series of DWI works. For instance, [33] developed invariant polynomials of spherical harmonic (SH) expansion coefficients, and discussed their application in population studies. [34] proposed a related construction using eigenvalue decomposition of SH operators. [35] and [36] argued their usefulness for understanding microstructures in relation to DWI.

[23] proposed a rotation equivariant construction inspired by [11] for disease classification. The same authors [37] used a $\mathbb{S}^2 \times \mathbb{R}^+$ CNN using SHORE function representation for classification in Parkinson Disease. [38] used a spherical U-Net for f-ODF estimation The same authors [39] used a spherical CNN for microstructure parameter estimation, using spherical harmonics representations. [10] proposed a sixth-D, 3D space and q-space NNs with roto-translation/rotation equivalence properties, targeted at DWI data. [40] reviewed several implementations of $SE(3)$ neural networks and showcased a comparison among these networks. In their work, steerable CNNs generalize better than group CNNs while dealing with inconsistent distributions between training and testing data for 3D images. In our experiments, in comparison to [10] which uses steerable filter bases, we found out, however, that our direct convolution implementation of $SE(3)$ GCNN does not perform inferior to its steerable alternative.

3 Method

The networks we present will be built from the principle of expanding CNNs to groups and their homogeneous spaces, on which they act by extending convolution operations to functions on groups and their homogeneous spaces. For the rotation group $SO(3)$ and the sphere $\mathbb{S}^2$ as $SO(3)$-homogeneous space, the common path for implementing convolutions/correlations is to use irreducible representations [41]. We do not follow that path here.

An action of a Lie group G on a space $\mathcal{M}$ is a smooth mapping $G\times\mathcal{M} \to \mathcal{M}$, $(g, m) \to g.m$ such that for each g, $m \to g.m$ is a diffeomorphism of $\mathcal{M}$ and such that $g.(g'.m) = (gg').m$. The neutral element of G acts as the identity. The orbit of $m \in \mathcal{M}$ is the set $G.m = \{g.m, g \in G\}$. The stabilizer G_m of an element m is the set of transformations that lets m fixed, $G_m = \{g \in G, g.m = m\}$. It is a subgroup of G. $\mathcal{M}$ is a *G-homogeneous space* if it contains only one orbit, i.e, if for any $m, m' \in \mathcal{M}$, there exists $g \in G$, with $g.m = m'$. Given a base point m_0 (for instance, the north pole if $\mathcal{M}$ is a sphere) in the homogeneous space $\mathcal{M}$, there is an isomorphism $G/G_{m_0} \simeq \mathcal{M}$, called the orbit map. G/G_{m_0} is the quotient space of G by G_{m_0} and consists of the *left cosets* gG_{m_0} of G_{m_0}. The inverse of the point m by the orbit map is a coset gG_{m_0}, with $g.m_0 = m$, called the *fiber* above m.

3.1 Standard convolution operations

A group G acting on a space $\mathcal{M}$ via $(g, m) \mapsto g.m$ also acts on functions on $\mathcal{M}$ by the *left translation*

$$(L_g f)(m) = f(g^{-1}m). \tag{1}$$

We assume that each homogeneous space is endowed with a G-invariant measure that allows integration, and that each G is endowed with a left-invariant Haar measure.

3.1.1 Lifting layer

A function $f : \mathcal{M} \to \mathbb{R}^N$ can be *lifted* to the group G via a kernel $\kappa : \mathcal{M} \to \mathbb{R}^K$ by

$$\kappa * f(g) = \sum_{i=1}^{K} \int_{\mathcal{M}} f(m)\kappa_i(g^{-1}m)\, dm \tag{2}$$

This operation is *equivariant*: $\kappa * L_g f = L_g(\kappa * f)$.

3.1.2 Group convolution layer

A feature function $F : G \to \mathbb{R}^N$ can be transformed by a convolution kernel $K : G \to K$ by

$$K * F(g) = \sum_{i=1}^{N} \int_{G} F(h)K_i(h^{-1}g)dh. \tag{3}$$

This operation is equivariant: $K * (L_g F) = L_g(K * F)$.

3.1.3 Projection Layer

If needed, feature map $F : G \to \mathbb{R}^n$ can be projected to a function $f : \mathcal{M} \to \mathbb{R}^n$ by summarizing on the fibers

$$\overline{F}(m) = \max_{h \in G_{m_0}} F(gh), \quad \text{for any } g \text{ with } g.m_0 = m, \tag{4}$$

where the max is computed component-wise. This operation is equivariant: $\overline{L_k F} = L_k \overline{F}$.

3.1.4 Activation Functions and Separable Kernels

A point-wise activation function α, such as ReLU, is trivially equivariant $L_g(\alpha f) = \alpha(L_g f)$. On manifolds with an underlying product structure, $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$ - this includes homogeneous spaces and groups - one can choose separable kernels $\kappa = \kappa_{\mathcal{M}_1} \otimes \kappa_{\mathcal{M}_2}$, and activation functions can be introduced in (2) and (3). For instance, lifting (2) can be replaced by

$$\kappa *^{\alpha} f(g) = \sum_{i=1}^{K} \int_{\mathcal{M}_1} \alpha\left(\int_{\mathcal{M}_2} f(m_1, m_2) \kappa_2(g^{-1} m_2)\, dm_2 \right) \kappa_1(g^{-1} m_1)\, dm_1, \tag{5}$$

which preserves equivariance. Having separable kernels increases the efficiency of the model since it increases weight sharing. For example, instead of having kernels defined in $\mathbb{R}^3 \times \mathbb{S}^2$, we have kernels defined in $\mathbb{R}^3$ and in $\mathbb{S}^2$. In this way, all voxels in $\mathbb{R}^3$ share the same spherical kernels. This is used in this work.

The spaces used in this work are $\mathbb{R}^3$, the sphere $\mathbb{S}^2$ and the product space $\mathbb{R}^3 \times \mathbb{S}^2$. The groups that we consider are the group of translations of $\mathbb{R}^3$, $\mathbb{T}^3 \simeq \mathbb{R}^3$, the group $SO(3)$ or 3D rotations, the direct product $\mathcal{G} = \mathbb{T}^3 \times SO(3)$ and the special Euclidean group $SE(3) = SO(3) \ltimes \mathbb{T}^3$. Note that though $\mathcal{G}$ and $SE(3)$ are isomorphic as manifolds, they are not as groups: in $\mathcal{G}$, $(\vec{t}, R).(\vec{s}, S) = (\vec{t} + \vec{s}, RS)$ while in $SE(3)$, $(R, \vec{t}).(S, \vec{s}) = (RS, \vec{t} + R\vec{s})$. This is also reflected in their respective actions in Table (1), which shows the different combinations of spaces and groups.

3.2 Discretization of spherical signals

The way spherical signals are numerically handled have major implications for our networks. A DWI signal is treated as a discretization of a signal $f : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$. DWIs are acquired, for each voxel, at N fixed directions $p_1, \ldots, p_N$ on $\mathbb{S}^2$ (here $N = 90$). These are represented in two different ways.

G \ $\mathcal{M}$	$\mathbb{R}^3, x$	$\mathbb{S}^2, \vec{v}$	$\mathbb{R}^3 \times \mathbb{S}^2, (x, \vec{v})$
$\mathbb{T}^3, \vec{t}$	$x + \vec{t}$		
$SO(3), R$		$R\vec{v}$	
$\mathbb{T}^3 \times SO(3), (\vec{t}, R)$	$x + \vec{t}$	$R\vec{v}$	$(x + \vec{t}, R\vec{v})$
$SE(3), (R, \vec{t})$	$Rx + \vec{t}$	$R\vec{v}$	$(Rx + \vec{t}, R\vec{v})$

Table 1: The groups and homogeneous spaces in this work. For each group and each homogeneous space, typical elements are provided, as well as the action of the group element on the space element. Entries left empty are not used or fail to be homogeneous spaces for standard group actions on them.

- Type 1. Ignoring the spherical structure, at each voxel x, we get a measurement vector $f(x) = (f(x, p_1) \ldots, f(x, p_N)) \in \mathbb{R}^N$. Thus an image is a mapping $I : \mathbb{R}^3 \to \mathbb{R}^N$.
- Type 2. A signal at voxel x is interpolated as a proper spherical function $f(x, \vec{v}) = W(v;\ v_1, \ldots, v_N)$ where W is a Watson kernel [42]. An image from this type is a mapping $I : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$.

3.3 Direct convolution and discretization of groups

Unlike existing methods that use generalized Fourier-type transforms to perform convolution on spheres [2, 4, 11, 12, 18, 20–25], we implement the convolution for spheres directly as in classical 2D CNNs in the image analysis field. We first discretize the sphere $\mathbb{S}^2$ using an icosahedron. To lift the function from the sphere to the $SO(3)$ group, we define a star-shaped kernel $k : \mathbb{S}^2 \mapsto \mathbb{R}$ with a limited support. The kernel then moves around the discretized sphere, and convolves with signals at each vertex of the icosahedron. It rotates 5 times at each icosahedral vertex according to the 5 edges each vertex has, and collects convolutional responses from all 5 rotations. In this way, the spherical function is lifted to $SO(3)$, which is discretized by $I_{SO(3)}$ - the 60 rotational symmetries of an icosahedron. This is shown in fig. 1 A. For the $SO(3)$ group convolution layer, the kernel is defined on $SO(3)$, which is represented by the icosahedral symmetries. Here we specially design the kernel in the way that the support of it covers exactly a fiber. Therefore, we rotate (permute) the kernel at each fiber and convolve the rotated kernels with the fiber, and move the kernel to the next fiber. This is shown in fig. 1 B.

3.4 Generic Networks used in this work

We present 4 constructions in which gradual levels of complexity in group actions are introduced. This can be seen as a group-action ablation study. The precise description of each network will be provided in section 4.

3.4.1 $\mathbb{T}^3$

The $\mathbb{S}^2$-structure of the signal is ignored, using the Type 1 discretization. The group being $\mathbb{T}^3$, we just obtain a standard CNN, ignoring rotational information. An illustration can be found in fig. 2.

3.4.2 $SO(3)$

This time the spatial structure is ignored, and each voxel provides a spherical data point. Type 2 discretization is used. The GCNN takes as input a spherical function, and will classify it by performing $SO(3)$-lifting, $SO(3)$-convolutions and summarization. The convolved function on $SO(3)$ is then projected back to $\mathbb{S}^2$ by this summarization. It is illustrated in fig. 1 A and B. This model is a fully equivariant implementation of $SO(3)$ group convolution followed by the work in [43], which does not hold global equivariance.

3.4.3 $\mathbb{T}^3 \times SO(3)$

Spatial and spherical structures are decoupled. This implies a standard spatial CNN dealing with only voxel translations, and a $SO(3)$-GCNN part for the directional signal. Type 2 discretization is used for spherical signals. The decoupled $\mathbb{R}^3$-layer and $\mathbb{S}^2$-layer are with group actions $\mathbb{T}^3$ and $SO(3)$ respectively. The illustration for the $\mathbb{S}^2$-layer can be found in fig. 1 A and B, and the illustration for the $\mathbb{R}^3$-layer can be regarded as only one Conv3D operation in fig. 1 C without the rotations. Note that since the spatial convolution does not incorporate rotational equivariance, it does not reflect equivariance of the DWI measurements. I.e., one can expect that when the brain rotates, the spatial patterns rotate as well as their spherical diffusion signals. This model takes rotation into account in the spherical part of the signal, but not the spatial part. The projection at the end collapses the function in the group back to $\mathbb{R}^3$ by summarizing - in this case, maximizing - over $SO(3)$, and the resulting feature map is fed into a fully connected layer to perform the classification task.

3.4.4 $SE(3)$

Type 2 discretization is used and the network uses the full interplay between spatial roto-translations and corresponding rotations of the spherical signal and is thus fully equivariant to $SE(3)$ transformations on the DWI data. fig. 1 A and B shows the kernels of the $\mathbb{S}^2$-layer. When the kernel moves from one vertex to another, it follows a specific rotation that maps the one-ring neighborhood of the source vertex to the one-ring neighborhood of the target vertex. At each vertex, the kernel has an $SO(2)$ symmetry group structure discretized by 5 rotations. fig. 1 C shows the kernel for the $\mathbb{R}^3$-layer. It is rotated with the same rotation matrices that moved the $\mathbb{S}^2$-kernel as in fig. 1 A and B. Since the spatial kernels are cube-shaped grids, interpolation is required while rotating them. Here we use linear interpolation, which can be easily implemented.

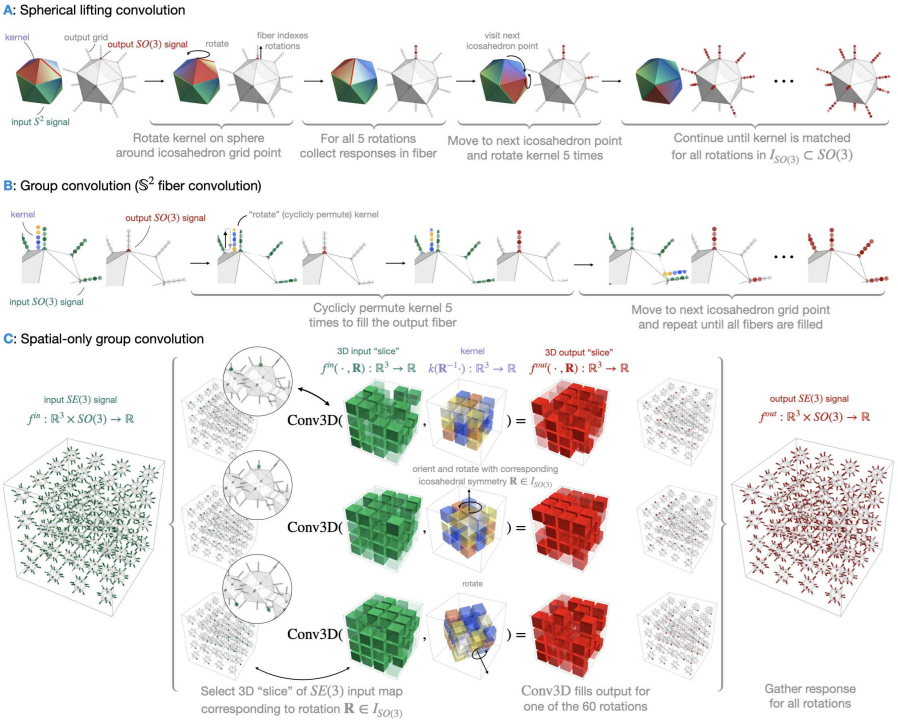


Fig. 1: The three group convolution operators used in this paper. Fig. A shows the spherical part of the separable lifting convolution. The star-shaped kernel translates (in this case translation is equivalent to rotation) to the 12 icosahedron vertices like a spider crawling on a sphere. At each vertex location, the kernel rotates 5 times aligned with the edges of the icosahedron and gets 5 responses from all the orientations. Therefore, at each vertex, the output is a fiber consisting of 5 elements. There are in total 60 responses from all 12 vertices, and thus 60 rotation matrices to translate the kernel, assembling a discretization of $SO(3)$ - $I_{SO(3)}$. Fig. B shows the spherical part of the separable group convolution. The kernel is then defined at each fiber, and is rotated (permuted) again for 5 times to get the responses of different orientations, as in the lifting convolution. Fig. C shows the spatial part of the separable convolution (the spatial convolution is the same in the lifting and group convolution, thus we only show one). The spatial kernel is a 3D grid. The grid is rotated to convolve with all 60 spherical responses. The kernel is rotated 60 times, using the same icosahedral symmetry rotations as those on which the input is sampled.

To perform the segmentation task, the projection layer collapses the function on $SE(3)$ back to $\mathbb{R}^3$ by summarizing - again, maximizing - over $SO(3)$.

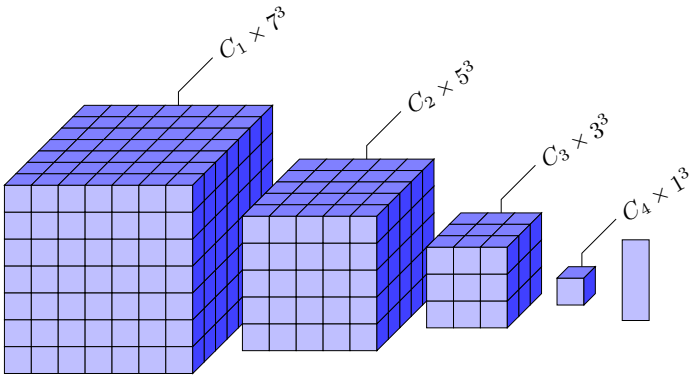


Fig. 2: Illustration of the classical CNN. In the grids shown above, which assembles the dimensions of feature maps in the later experiments. Each voxel in the *ith* layer contains C_i values, indicating the numbers of channels. C_1 here is the number of signal values each voxel from the original scan, thus 90. Due to striding, the grid shrinks to 1 voxel after 3 convolutional layers, and then is fed into a fully connected layer for classification.

4 Experiments and Results

In this section, we first list all the detailed network setups, after which we present the results of the experiments. We evaluate our method on the DWI brain dataset from the human connectome project (HCP) [8]. We classify the human brains into 4 regions - cerebrospinal fluid (CSF), subcortical, white matter (WM), and grey matter (GM). An illustration of the task can be found in fig. 3.

We use the pre-processed DWI data [8] and normalize each DWI scan for the b-1000 images with the voxel-wise average of the b_0. We use the brain masks provided in the dataset to obtain the voxels of interest, while background is ignored. The labels provided with the T1-image are transformed to the DWI using nearest neighbor interpolation (fig. 3). The resolution of the DWI images is $145 \times 174 \times 145$, and the resolution of the T1-images is $260 \times 311 \times 260$. Focal Loss [44] is used to counter the class imbalance of the 4 brain regions. For Focal Loss, all experiments use $\gamma = 2$ and use

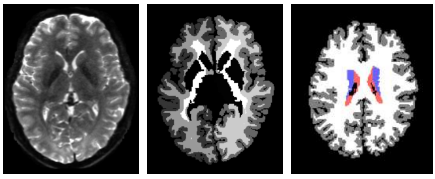


Fig. 3: Left to right: original diffusion data, the ground-truth segmentation, and the processed ground-truth that we are going to learn from. The label colors for CSF, subcortical, white matter and grey matter are red, blue, white and grey respectively. The figures only illustrate the data, they are not necessarily from the same slice of the same scan.

$\alpha = (0.35, 0.35, 0.15, 0.15)$ for CSF, subcortical, WM, and GM respectively. For the Watson Kernel, all experiments that used this interpolation (Type 2 discretization) have $\kappa = 10$. Batch size for all experiments is 100.

4.1 Experiment setup

To reduce the computational burden, as inputting a full DWI volume is intractable, we use spatial windows of N^3 voxels, with $N = 1$ for the $SO(3)$-action network and $N = 7$ for the rest. In addition, due to the effect of striding in spatial convolution, the 7^3 grid of voxels shrinks to 1^3 after 3 spatial convolutions. Therefore, a separable convolution layer (for both $\mathbb{T}^3 \times SO(3)$ and $SE(3)$ actions) is equivalent to a single $SO(3)$ convolution layer when the grid shrinks to 1^3, since the spatial convolution becomes trivial. $\mathbb{S}^2$ is discretized by a regular icosahedron. $SO(3)$ is discretized as the icosahedral rotation group with 60 elements. Each vertex of the icosahedron is fixed by 5 rotations, isomorphic to the subgroup of $SO(2)$ consisting of rotations of angle $2k\pi/5$, $k = 0 \dots 4$. This is, of course, the discretization used for $SO(2)$.

To validate the proposed $SE(3)$ network, we first provide an ablation study of our proposed 4 types of networks based on different group actions. Then, we compare it with [10], which implements an $SE(3)$-GCNN using *irreducible representations*.

For the ablation study, based on the networks we introduced above and on top of the networks presented in [9], we design our experiments for them. For each experiment, in order to explore the impact of model capacity on the performance, we construct 2 models with high and low capacities respectively, denoted by the superscription + and -. We choose the architectures for the models with low capacity by trying out different complexities and depths and picking the one with the lowest capacity with the same level of performance. Then for the models with high capacity, we simply increase the numbers of kernels in each layer of the models with low capacity.

Detailed descriptions of all the experiments are reported below, and a summary of the experiments can be found in table 2.

4.2 Ablation study

$\mathbb{T}^3$-*Classical CNN*

The architecture we use is $ReLU(\mathbb{R}^3$ conv$) - ReLU(\mathbb{R}^3$conv$) - ReLU(\mathbb{R}^3$conv$)-$FC with network setups of a low capacity and a high capacity. FC here is a fully connected layer. We label the small network $(90 - 5 - 5 - 5 - 4)$ Classical$^-$ and the big network $(90 - 120 - 120 - 90 - 4)$ Classical$^+$.

$SO(3)$-*Baseline*

In the experiments, we use the $ReLU$(lift) $-ReLU$(gconv)$-$project$-$FC architecture as was used in [43], but with true $SO(3)$-convolution. The projection layer takes the maximum of the 5 rotations to collapse the function back to the

sphere. We experimented various sizes of the network ($10-20-proj.-4$ and $20-40-proj.-4$), in addition to the setup used in [43] ($1-5-proj.-4$). The network that has the biggest size did not seem to improve the second biggest one, thus we omit it in this paper. Based on the size of the experiments, we call the small network Baseline$^-$ and the big network Baseline$^+$.

$\mathbb{T}^3 \times SO(3)$-***OursDecoupled***

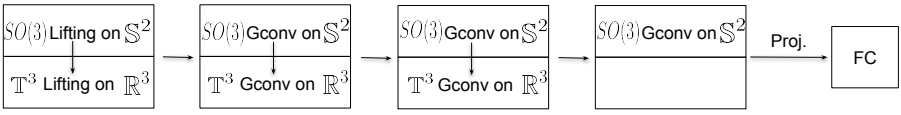


Fig. 4: Architecture of the network with group action $\mathbb{T}^3 \times SO(3)$. Each block is a convolutional layer split into 2 separable layers. The vertical arrows in each block shows the separable convolutions. First the spherical convolution is applied, followed by the spatial convolution. The last block before the FC layer is equivalent to a single $\mathbb{S}^2$-layer as explained in section 4.1. Illustrations of ReLU actions are omitted for visualization simplicity.

We use the architecture $ReLU(\text{lift}) - ReLU(\text{gconv}) - ReLU(\text{gconv}) - ReLU(\text{gconv})$ $-\text{project}-\text{FC}$. Using separability discussed in section 3.1.4, a convolution layer (including lifting) is split into 2, and ReLU activation is added between separable layers as well. An illustration of the architecture can be found in fig. 4.

We again experiment with 2 sizes of the network - a small one and a big one. The small network has $5-5-5-5-5-5-5-proj.-4$ kernels for each layer, while the big network has $10-20-20-40-40-20-10-proj.-4$. We label them OursDecoupled$^-$ and OursDecoupled$^+$.

$SE(3)$-***Ours***

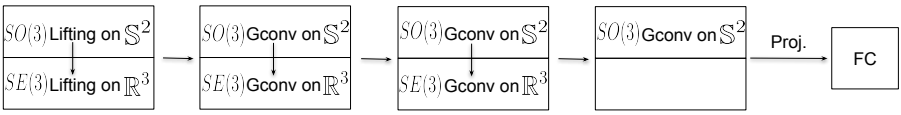


Fig. 5: Architecture of the network with group action $SE(3)$.

Here too we use the separable setup described in section 3.1.4. Thus a layer is again split into 2 layers - an $\mathbb{S}^2$-layer and an $\mathbb{R}^3$-layer, both for lifting and group convolution. The $\mathbb{S}^2$-layer is defined as shown in fig. 1 A and B. We rotate the $\mathbb{R}^3$ kernels and the $\mathbb{S}^2$ kernels using the same actions. The rotational actions of the kernels can be represented by 60 rotation matrices, and is equivalent to the discretization of the $SO(3)$ rotation group using the

Experiment	G	#Params	#Epochs
$I : \mathbb{R}^3 \to \mathbb{R}^N$			
Classical$^-$	$\mathbb{T}^3$	13539	34
ClassicalAug$^-$			66
Classical$^+$		972694	19
ClassicalAug$^+$			67
$I : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$			
Baseline$^-$	$SO(3)$	286	31
BaselineAug$^-$			45
Baseline$^+$		2104	31
BaselineAug$^+$			54
OursDecoupled$^-$	$\mathbb{T}^3 \times SO(3)$	2514	41
OursDecoupledAug$^-$			80
OursDecoupled$^+$		59914	15
OursDecoupledAug$^+$			54
OursPart$^-$	$SE(3)*$	2514	41
OursPartAug$^-$			49
OursPart$^+$		59914	15
OursPartAug$^+$			48
OursFull$^-$	$SE(3)$	2514	41
OursFullAug$^-$			86
OursFull$^+$		59914	15
OursFullAug$^+$			42

Table 2: Criteria and properties of experiments. $SE(3)*$ indicates the rotations in the spatial part are only a part of the rotations used in the spherical part.

icosahedral symmetry group, as shown in fig. 1 C. As in section 4.2, we use the $ReLU(\text{lift}) - ReLU(\text{gconv}) - ReLU(\text{gconv}) - ReLU(\text{gconv}) - \text{project} - \text{FC}$ architecture. After the separation of the layers, the illustration is showcased in fig. 5. As in section 4.2, ReLU activations are added between separable layers as well.

In addition, we intend to explore the impact of the equivariance we imposed in $\mathbb{R}^3$ in this section. As was explained above, we align the rotations of the $\mathbb{R}^3$ kernel with the ways the $\mathbb{S}^2$ kernel moved on the sphere, which is discretized by the 60 rotation symmetries of an icosahedron. At a vertex $x_i, i \in 1, ..., 12$ of an icosahedron, there exists a stabilizer $SO(3)_{x_i}$ discretized by 5 equally divided rotations that keep x_i unchanged. Therefore, we also experiment a partial equivariance in the $\mathbb{R}^3$ roto-translational convolution. This means at each vertex x_i of the icosahedron, we only take 1 out of the 5 rotations that discretized $SO(3)_{x_i}$ instead of using all of them to rotate the spatial kernel. Note that the partially equivariant models are only fully $SE(3)$-equivariant when the kernels have a sub-group $SO(2)$ symmetry in them [7, Thm 1], which we do not impose and thus equivariance is not guaranteed.

Again, we experiment with 2 sizes of the network with $5 - 5 - 5 - 5 - 5 - 5 - 5 - proj. - 4$ and $10 - 20 - 20 - 40 - 40 - 20 - 10 - proj. - 4$ kernels respectively. Therefore, we generate 4 experiments for this section: OursFull$^-$, OursPart$^-$, OursFull$^+$, and OursPart$^+$.

4.2.1 Data augmentation experiments

To validate the robustness of GCNNs against data variation modeled by group actions, we train all the proposed models with augmented data as well. Each data sample (grid of 7^3 or 1^3) is randomly rotated on the fly before being fed into the model. To prevent interpolation, the rotations used to transform the data are sampled from a octohedral symmetry group. For DWI data that have directional signals in each voxel, the directions of the signals (b-vectors) in each voxel rotate with the voxel grid. In order to guarantee the signal values in each voxel are from the same orientations after augmentation, we interpolate the function values at the orientations-of-interest using the rotated b-vectors. Therefore, for Type 1 discretization, we interpolate function values at the original b-vectors, and for Type 2 discretization, we interpolate at the pre-defined icosahedron as demonstrated above.

4.2.2 Results

As was done in [43], we trained all networks using **1** scan, validated using **1** scan, and tested using **50** scans. We evaluate the accuracies and Dice scores of the classification of the 4 regions respectively, and the overall classification accuracy across all test scans. We have also tried training models with more scans (5 or 10), it does not seem to improve the results significantly. Therefore, we choose to use 1 scan for training. For each class, the accuracy is calculated by $\frac{\#CorrectPredictions}{\#ClassSamples}$, and the Dice score is calculated by $\frac{2TP}{2TP+FP+FN}$ for the class. The overall accuracy is calculated by $\frac{\#CorrectPredictions}{\#AllSamples}$.

We trained all models until they converge and before overfitting, thus models of different capacities and different setups are stopped at different epochs. Each model is trained with both original data and augmented data. Details can be found in table 2.

Experiment \ Class	CSF	Subcortical	WM	GM
$I : \mathbb{R}^3 \to \mathbb{R}^N$				
Classical$^-$	0.756 ± 0.07	0.376 ± 0.043	0.834 ± 0.011	0.839 ± 0.02
ClassicalAug$^-$	0.625 ± 0.11	0.128 ± 0.021	0.77 ± 0.017	0.806 ± 0.017
$I : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$				
Baseline$^-$	0.75 ± 0.073	0.185 ± 0.04	0.801 ± 0.012	0.83 ± 0.011
BaselineAug$^-$	0.741 ± 0.074	0.232 ± 0.048	0.805 ± 0.014	0.835 ± 0.011
OursDecoupled$^-$	$\mathbf{0.817} \pm 0.051$	$\mathbf{0.705} \pm 0.033$	$\mathbf{0.867} \pm 0.009$	$\mathbf{0.909} \pm 0.007$
OursDecoupledAug$^-$	0.775 ± 0.063	0.639 ± 0.038	0.851 ± 0.01	0.886 ± 0.009
OursPart$^-$	0.807 ± 0.048	0.658 ± 0.037	0.865 ± 0.009	0.899 ± 0.008
OursPartAug$^-$	0.78 ± 0.06	0.643 ± 0.037	0.849 ± 0.01	0.886 ± 0.009
OursFull$^-$	0.769 ± 0.06	0.621 ± 0.038	0.854 ± 0.01	0.891 ± 0.008
OursFullAug$^-$	0.772 ± 0.061	0.637 ± 0.037	0.846 ± 0.01	0.884 ± 0.009

Table 3: Statistics of dice scores from experiments using models of low capacity.

Experiment / Class	CSF	Subcortical	WM	GM	Overall
$I : \mathbb{R}^3 \to \mathbb{R}^N$					
Classical$^-$	0.792 ± 0.08	0.415 ± 0.053	$\mathbf{0.879} \pm 0.024$	0.789 ± 0.034	0.806 ± 0.017
ClassicalAug$^-$	0.662 ± 0.105	0.088 ± 0.017	0.808 ± 0.042	0.801 ± 0.039	0.761 ± 0.014
$I : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$					
Baseline$^-$	0.742 ± 0.082	0.145 ± 0.04	0.804 ± 0.024	0.85 ± 0.016	0.788 ± 0.011
BaselineAug$^-$	0.785 ± 0.074	0.202 ± 0.055	0.793 ± 0.028	0.858 ± 0.018	0.791 ± 0.012
OursDecoupled$^-$	$\mathbf{0.844} \pm 0.061$	0.741 ± 0.033	0.833 ± 0.02	$\mathbf{0.934} \pm 0.013$	$\mathbf{0.878} \pm 0.009$
OursDecoupledAug$^-$	0.769 ± 0.087	0.716 ± 0.04	0.854 ± 0.023	0.87 ± 0.023	0.853 ± 0.01
OursPart$^-$	0.787 ± 0.068	0.717 ± 0.032	0.848 ± 0.019	0.906 ± 0.016	0.868 ± 0.009
OursPartAug$^-$	0.772 ± 0.081	$\mathbf{0.752} \pm 0.036$	0.848 ± 0.021	0.87 ± 0.022	0.852 ± 0.01
OursFull$^-$	0.81 ± 0.065	0.692 ± 0.029	0.857 ± 0.022	0.874 ± 0.019	0.856 ± 0.01
OursFullAug$^-$	0.783 ± 0.077	0.711 ± 0.054	0.855 ± 0.023	0.864 ± 0.021	0.85 ± 0.01

Table 4: Statistics of classification accuracy from all experiments using models of low capacity.

The Dice scores and accuracies of models of low capacity can be found in table 3 and table 4, while the Dice scores and accuracies of models of high capacity can be found in table 5 and table 6. The numbers shown in all the tables are the average value and standard deviation across 50 test scans. Examples of predictions compared with the ground-truth can be found in fig. 9a.

The impact of data augmentation

As we can see from the table 3, table 4, table 5, and table 6, models trained with augmented data do not perform better than their counterparts trained with just original data, if not worse. Unlike 2D image datasets in the computer vision community that have various backgrounds and objects in their images, the HCP dataset is very uniform. The distribution of the original training data is expected to be the same as the test set data. However, after augmentation, the distribution of the training data changed and it differs from the test data. Therefore, in this case data augmentation does not help any of the models since the augmentation does not represent the diversity in this dataset. One extreme would be Classical$^-$ vs. ClassicalAug$^-$ that can be found in table 3 and table 4, the augmented data confused the model in terms of the subcortical region - a somewhat mixture of white and grey matter. Therefore, from now on, if not specified, we mainly discuss the models and results trained without data augmentation.

The impact of the $\mathbb{R}^3$ spatial component

It is easy to observe that the the Baseline experiments perform worst among all. This is an anticipated outcome since it is usually the case that neighboring information is an essential type of local features.

Type 1 discretization vs Type 2 discretization

The classical CNNs use Type 1 discretization while Type 2 discretization is used for the rest of the models. The classical CNNs do not perform as well as

models that take into account the spherical geometry with spatial information, but performs better than Baseline. However, Classical$^-$ is not much better than Baseline$^+$ while having far more parameters to train, and Classical$^+$ performs even worse than OursDecoupled$^-$, OursPart$^-$, or OursFull$^-$, which have much less training parameters.

The results of the two extreme cases - Baseline that only takes into account spherical geometry but ignore any spatial information and Classical that only looks into the spatial part and discards spherical geometry - show that the voxel geometry and neighboring voxel correlation can both capture some decent amount of information to deal with the segmentation task, but they both have something that the other one cannot grasp, and combining the spherical geometry and the spatial correlation can boost the performance to a promising extent.

The impact of adding an $\mathbb{R}^3$ part to Baseline

On top of the Baseline, the easiest way to add spatial information to the purely voxel-based framework is what was done in OursDecoupled section 4.2 - a GCNN on $\mathbb{S}^2$ to learn the geometric signals in individual signals and a regular classical CNN to take into account the local spatial information. We can see from the results that this setup immediately boosted the performance compared to the Baseline. We can also see that OursDecoupled$^+$ performs better than OursDecoupled$^-$, for the sake of model capacity.

Experiment / Class	CSF	Subcortical	WM	GM
$I : \mathbb{R}^3 \to \mathbb{R}^N$				
Classical$^+$	0.804 ± 0.053	0.583 ± 0.036	0.856 ± 0.011	0.893 ± 0.009
ClassicalAug$^+$	0.752 ± 0.069	0.407 ± 0.044	0.828 ± 0.011	0.849 ± 0.017
$I : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$				
Baseline$^+$	0.754 ± 0.069	0.334 ± 0.037	0.805 ± 0.013	0.841 ± 0.012
BaselineAug$^+$	0.748 ± 0.072	0.311 ± 0.037	0.796 ± 0.016	0.845 ± 0.011
OursDecoupled$^+$	0.827 ± 0.047	0.716 ± 0.044	0.878 ± 0.009	0.903 ± 0.01
OursDecoupledAug$^+$	0.79 ± 0.053	0.721 ± 0.033	0.87 ± 0.009	0.902 ± 0.007
OursPart$^+$	$\mathbf{0.834} \pm 0.045$	$\mathbf{0.752} \pm 0.034$	$\mathbf{0.878} \pm 0.009$	$\mathbf{0.914} \pm 0.007$
OursPartAug$^+$	0.789 ± 0.059	0.736 ± 0.035	0.872 ± 0.009	0.902 ± 0.008
OursFull$^+$	0.788 ± 0.05	0.746 ± 0.034	0.877 ± 0.008	0.909 ± 0.006
OursFullAug$^+$	0.792 ± 0.051	0.737 ± 0.031	0.873 ± 0.009	0.907 ± 0.007

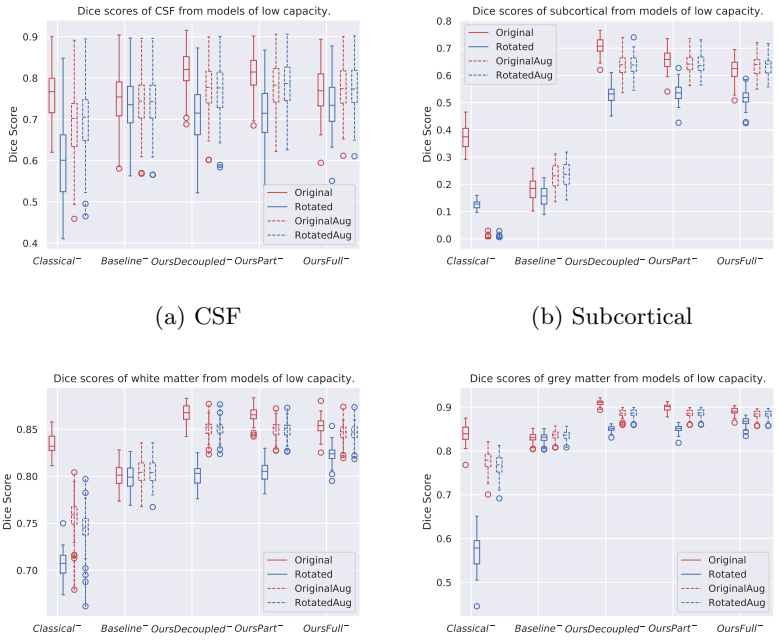
Table 5: Statistics of dice scores from experiments using models of high capacity.

The argument for OursFull not performing the best

For models of low capacity, however, we can observe from table 3 and table 4 that our proposed method performs worse than OursDecoupled$^-$. Additionally, for models of high capacity, even though we can see that OursFull$^+$

Experiment / Class	CSF	Subcortical	WM	GM	Overall
$I : \mathbb{R}^3 \to \mathbb{R}^N$					
Classical$^+$	0.815 ± 0.061	0.702 ± 0.026	0.834 ± 0.022	0.89 ± 0.011	0.854 ± 0.012
ClassicalAug$^+$	0.687 ± 0.088	0.42 ± 0.04	0.863 ± 0.031	0.818 ± 0.038	0.812 ± 0.015
$I : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$					
Baseline$^+$	0.778 ± 0.07	0.379 ± 0.065	0.784 ± 0.024	0.848 ± 0.02	0.792 ± 0.013
BaselineAug$^+$	0.776 ± 0.076	0.351 ± 0.067	0.749 ± 0.029	0.875 ± 0.017	0.789 ± 0.014
OursDecoupled$^+$	0.865 ± 0.061	0.783 ± 0.035	0.867 ± 0.017	0.902 ± 0.019	0.879 ± 0.011
OursDecoupledAug$^+$	0.821 ± 0.066	0.759 ± 0.052	0.876 ± 0.02	0.891 ± 0.018	0.876 ± 0.008
OursPart$^+$	0.819 ± 0.065	0.816 ± 0.031	0.845 ± 0.019	$\mathbf{0.936} \pm 0.011$	$\mathbf{0.888} \pm 0.009$
OursPartAug$^+$	0.756 ± 0.084	0.816 ± 0.033	$\mathbf{0.876} \pm 0.017$	0.888 ± 0.017	0.877 ± 0.009
OursFull$^+$	$\mathbf{0.896} \pm 0.042$	$\mathbf{0.826} \pm 0.023$	0.857 ± 0.017	0.912 ± 0.014	0.883 ± 0.008
OursFullAug$^+$	0.864 ± 0.048	0.78 ± 0.031	0.866 ± 0.019	0.905 ± 0.016	0.88 ± 0.008

Table 6: Statistics of classification accuracy from all experiments using models of high capacity.



(a) CSF

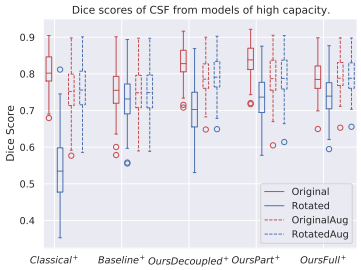
(b) Subcortical

(c) White matter

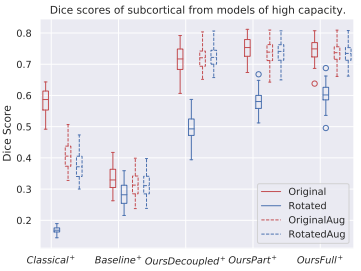
(d) Grey matter

Fig. 6: Comparison of Dice scores of the 4 classes from low-capacity models trained with original data and augmented data, tested with original and rotated test set.

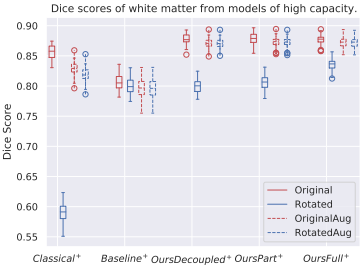
and OursPart$^+$ improve from their low capacity counterparts more than OursDecoupled$^+$, OursFull$^+$ does not perform as well as OursPart$^+$ as shown in table 5 and table 6. This differs from our expectation since models with full



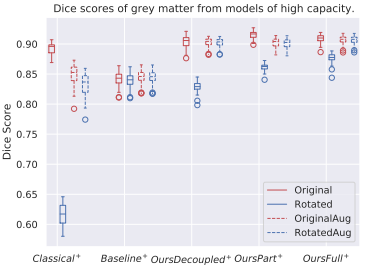
(a) CSF



(b) Subcortical

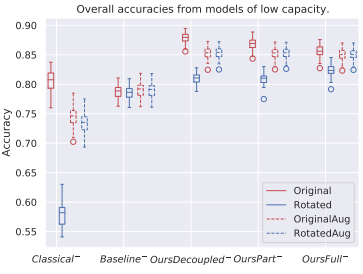


(c) White matter

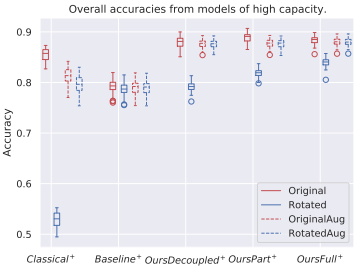


(d) Grey matter

Fig. 7: Comparison of Dice scores of the 4 classes from high-capacity models trained with original data and augmented data, tested with original and rotated test set.



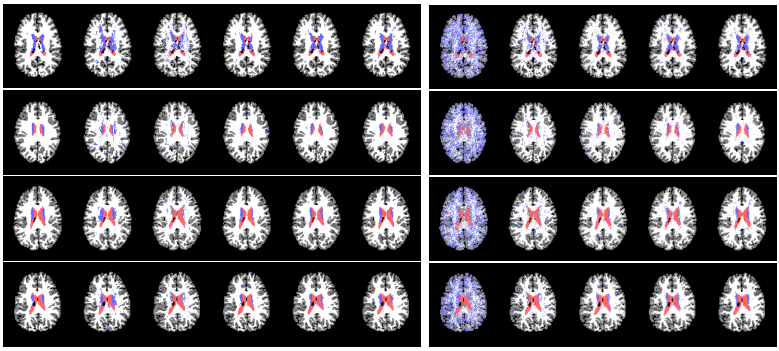
(a) Low capacity models



(b) High capacity models

Fig. 8: Comparison of overall accuracies from the original and rotated data.

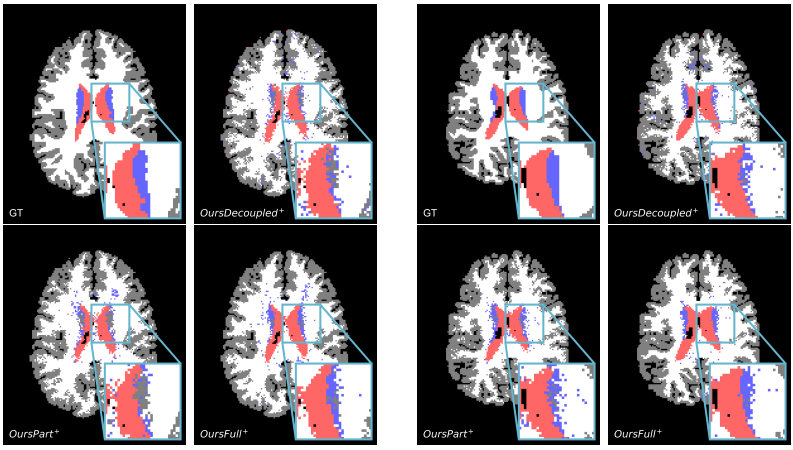
roto-translational equivariance should be more capable of handling variances in data, thus should have better performance. Recall that the HCP dataset [8] contains scans that are preprocessed and aligned with axes, thus there is little



(a) Predictions from test set using original data.

(b) Predictions from test set using rotated data.

Fig. 9: Examples of predictions. fig. 9a shows the predictions from the original test set, and fig. 9b shows the predictions from the augmented (rotated) test set. In fig. 9a, from left to right are ground-truth, Classical$^+$, Baseline$^+$, OursDecoupled$^+$, OursPart$^+$, and OursFull$^+$. In fig. 9b, from left to right are Classical$^+$, Baseline$^+$, OursDecoupled$^+$, OursPart$^+$, and OursFull$^+$. The colors of CSF, subcortical, WM and GM are red, blue, white, and grey respectively.



(a) A test scan slice.

(b) Another test scan slice.

Fig. 10: Showcases of zoom-in regions from predictions of the rotated test set. For both scan slices presented, from left to right, top to bottom, are the ground-truth, prediction from OursDecoupled$^+$, OursPart$^+$, and OursFull$^+$.

variance in rotation. In this case, enforcing $SE(3)$ equivariance in the model can be futile and be even confusing for the model.

Experiment / Class	CSF	Subcortical	WM	GM
$I : \mathbb{R}^3 \to \mathbb{R}^N$				
Classical$^-$	0.631 ± 0.097	0.101 ± 0.014	0.696 ± 0.019	0.558 ± 0.044
ClassicalAug$^-$	0.678 ± 0.094	0.117 ± 0.025	0.775 ± 0.018	0.813 ± 0.019
$I : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$				
Baseline$^-$	0.735 ± 0.076	0.158 ± 0.037	0.799 ± 0.013	0.829 ± 0.011
BaselineAug$^-$	0.741 ± 0.074	0.237 ± 0.047	0.804 ± 0.014	0.834 ± 0.011
OursDecoupled$^-$	0.708 ± 0.073	0.531 ± 0.033	0.801 ± 0.012	0.851 ± 0.006
OursDecoupledAug$^-$	0.771 ± 0.065	0.641 ± 0.036	$\mathbf{0.851} \pm 0.01$	0.886 ± 0.009
OursPart$^-$	0.714 ± 0.069	0.536 ± 0.035	0.804 ± 0.011	0.851 ± 0.008
OursPartAug$^-$	$\mathbf{0.784} \pm 0.059$	$\mathbf{0.642} \pm 0.036$	0.849 ± 0.01	$\mathbf{0.887} \pm 0.009$
OursFull$^-$	0.737 ± 0.065	0.517 ± 0.033	0.823 ± 0.01	0.867 ± 0.009
OursFullAug$^-$	0.774 ± 0.061	0.636 ± 0.036	0.846 ± 0.01	0.884 ± 0.009

Table 7: Statistics of dice scores from experiments using rotated data and models of low capacity.

Experiment / Class	CSF	Subcortical	WM	GM	Overall
$I : \mathbb{R}^3 \to \mathbb{R}^N$					
Classical$^-$	0.643 ± 0.106	0.24 ± 0.047	0.767 ± 0.051	0.421 ± 0.048	0.563 ± 0.023
ClassicalAug$^-$	0.677 ± 0.105	0.08 ± 0.02	0.811 ± 0.044	0.811 ± 0.043	0.767 ± 0.016
$I : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$					
Baseline$^-$	0.733 ± 0.085	0.12 ± 0.035	0.802 ± 0.024	0.852 ± 0.016	0.786 ± 0.011
BaselineAug$^-$	0.786 ± 0.074	0.21 ± 0.057	0.793 ± 0.029	0.856 ± 0.018	0.79 ± 0.012
OursDecoupled$^-$	0.755 ± 0.076	0.528 ± 0.037	0.779 ± 0.02	$\mathbf{0.871} \pm 0.013$	0.81 ± 0.008
OursDecoupledAug$^-$	0.765 ± 0.09	0.72 ± 0.038	0.853 ± 0.023	0.871 ± 0.023	$\mathbf{0.853} \pm 0.01$
OursPart$^-$	0.69 ± 0.084	0.599 ± 0.033	0.791 ± 0.02	0.852 ± 0.018	0.809 ± 0.009
OursPartAug$^-$	0.778 ± 0.081	$\mathbf{0.745} \pm 0.038$	0.849 ± 0.021	0.87 ± 0.021	0.853 ± 0.01
OursFull$^-$	$\mathbf{0.79} \pm 0.067$	0.591 ± 0.026	0.835 ± 0.023	0.84 ± 0.022	0.823 ± 0.01
OursFullAug$^-$	0.785 ± 0.077	0.707 ± 0.053	$\mathbf{0.854} \pm 0.023$	0.865 ± 0.021	0.85 ± 0.01

Table 8: Statistics of classification accuracy from experiments using rotated data and models of low capacity.

To verify this theory, we evaluated all models on the rotated test set. Taking the N^3 ($N = 1$ for Baseline models and $N = 7$ for the rest) grids of voxels we extracted from the test scans, we randomly rotate each grid using a rotation sampled from the octahedral symmetry group to create a new rotated test set. In this way, we do not need to interpolate while rotating, and the rotations are not aligned with the ones we used in our models to rotate the kernels while still resemble a discretization of the $SO(3)$ group. Hence we have 2 categories of models as well as 2 categories of the test set: models trained with original data vs. models trained with augmented data, and original test set vs. the randomly rotated test set.

Models trained with data augmentation tested with rotated test set

We see that all models trained with augmented training set have very similar performance results to the same models tested with the original test set, and they all perform better in this task than their counterparts trained with the

Experiment / Class	CSF	Subcortical	WM	GM
$I : \mathbb{R}^3 \to \mathbb{R}^N$				
Classical$^+$	0.549 ± 0.106	0.124 ± 0.007	0.535 ± 0.014	0.59 ± 0.022
ClassicalAug$^+$	0.768 ± 0.066	0.445 ± 0.038	0.82 ± 0.015	0.857 ± 0.014
$I : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$				
Baseline$^+$	0.733 ± 0.076	0.282 ± 0.036	0.799 ± 0.013	0.839 ± 0.012
BaselineAug$^+$	0.748 ± 0.072	0.311 ± 0.037	0.796 ± 0.016	0.844 ± 0.011
OursDecoupled$^+$	0.702 ± 0.075	0.497 ± 0.037	0.8 ± 0.011	0.829 ± 0.009
OursDecoupledAug$^+$	$\mathbf{0.794} \pm 0.054$	0.723 ± 0.033	0.87 ± 0.009	0.902 ± 0.007
OursPart$^+$	0.734 ± 0.063	0.58 ± 0.033	0.806 ± 0.011	0.862 ± 0.006
OursPartAug$^+$	0.791 ± 0.058	$\mathbf{0.736} \pm 0.034$	0.872 ± 0.009	0.901 ± 0.008
OursFull$^+$	0.74 ± 0.06	0.604 ± 0.034	0.835 ± 0.01	0.877 ± 0.008
OursFullAug$^+$	0.79 ± 0.051	0.735 ± 0.03	$\mathbf{0.872} \pm 0.009$	$\mathbf{0.907} \pm 0.007$

Table 9: Statistics of dice scores from experiments using rotated data and models of high capacity.

Experiment / Class	CSF	Subcortical	WM	GM	Overall
$I : \mathbb{R}^3 \to \mathbb{R}^N$					
Classical$^+$	0.632 ± 0.097	0.452 ± 0.02	0.434 ± 0.018	0.5 ± 0.03	0.471 ± 0.015
ClassicalAug$^+$	0.71 ± 0.088	0.517 ± 0.033	0.811 ± 0.038	0.85 ± 0.034	0.812 ± 0.015
$I : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$					
Baseline$^+$	0.769 ± 0.074	0.307 ± 0.059	0.782 ± 0.024	0.846 ± 0.02	0.786 ± 0.013
BaselineAug$^+$	0.776 ± 0.076	0.356 ± 0.068	0.749 ± 0.029	0.873 ± 0.017	0.788 ± 0.014
OursDecoupled$^+$	0.756 ± 0.082	0.597 ± 0.034	0.797 ± 0.019	0.81 ± 0.019	0.791 ± 0.01
OursDecoupledAug$^+$	0.819 ± 0.067	0.761 ± 0.051	0.876 ± 0.019	0.891 ± 0.018	0.876 ± 0.008
OursPart$^+$	0.716 ± 0.078	0.635 ± 0.033	0.78 ± 0.021	0.876 ± 0.012	0.819 ± 0.008
OursPartAug$^+$	0.762 ± 0.085	$\mathbf{0.811} \pm 0.032$	$\mathbf{0.878} \pm 0.018$	0.886 ± 0.017	0.877 ± 0.009
OursFull$^+$	$\mathbf{0.88} \pm 0.048$	0.659 ± 0.028	0.83 ± 0.019	0.868 ± 0.018	0.84 ± 0.009
OursFullAug$^+$	0.862 ± 0.049	0.78 ± 0.031	0.865 ± 0.019	$\mathbf{0.904} \pm 0.016$	$\mathbf{0.88} \pm 0.008$

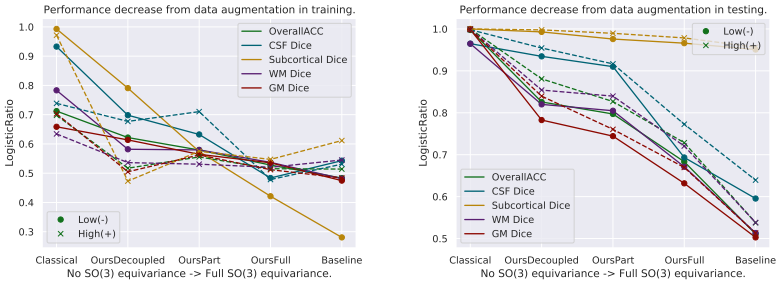
Table 10: Statistics of classification accuracy from experiments using rotated data and models of high capacity.

original training set. This checks with our statement in section 4.2.2 that the consistency of data distributions of the training and test sets boosts test performance. In this case, we used the same kind of rotations while augmenting the training set and test set, therefore the consistency of data distributions is maintained. However, this can never be guaranteed in real life. We can see this from table 7, table 8, table 9, table 10, fig. 6, fig. 7, and fig. 8.

Models trained with original data tested with rotated test set

In this section, only models trained without data augmentation are compared and discussed. For models with both low and high capacity, OursFull models have the best performance among other models. OursFull$^-$ remains 0.823 accuracy, decreased from 0.856 while OursFull$^+$ decreased from 0.883 to 0.84. This is illustrated in table 8 and table 10. In terms of Dice scores, OursFull$^-$ performs the best for all classes but the subcortical class, and OursFull$^+$ has the best results for **all** classes, as shown in table 7 and table 9.

Comparison figures of the 4 classes for all models can be found in fig. 6 and fig. 7, while comparisons of overall accuracies can be found in fig. 8. We can see again from the model with no spatial equivariance (OursDecoupled), the model with partial spatial equivariance (OursPart), and the model with full spatial equivariance (OursFull) that the gap between the performances on original data and rotated data shrink. It is worth noticing that Baseline models



(a) Model performance decrease while trained with data augmentation. aug- mentation.

(b) Model performance decrease while applied with rotated test set.

Fig. 11: Logistic map of the ratio of two criteria to evaluate the proposed models. One criterion is for the models trained with augmented data compared to their counterparts trained with original data. For models trained both with original and augmented data, fig. 11a shows the decrease of test results while trained with data augmentation and tested with the original test set as shown in table 3, table 4, table 5, and table 6. The second criterion is for the models trained with original data only. It is the decrease of performance while tested with rotated data, shown in fig. 11b.

almost do not suffer from performance drop while applied with rotated data. It is an $SO(3)$-network that preserves rotational equivariance on $\mathbb{S}^2$. For a single-voxel input, the network is very resistant to variations, but the performance of this model is limited due to the lack of spatial interaction and thus in general worse than models with spatial interplay.

Examples of predictions using the rotated test set can be found in fig. 9b. It is easily observed that the classical CNN does not generalize well to the data variation, while models with rotational symmetry (either $SO(3)$, $\mathbb{T}^3 \times SO(3)$, or $SE(3)$) generate better results. However, it is also noticeable that for a challenging minority class, subcortical region, OursFull$^+$ performs better than the others while other models with some rotational equivariance do not predict a concentrated subcortical region. Zoom-in examples can be found in fig. 10. Predictions from Baseline are omitted from fig. 10 since it does not have the same level of performance.

Augmentation in training data vs. augmentation in testing data

We have experimented models trained with both the original training set and augmented training set, and models tested with both the original test set and randomly rotated test set. The random rotations applied to the test set can be seen as augmentation too. As was discussed above, data augmentation changes the distribution of the dataset, which creates inconsistency between the training and testing set. However, augmentation in the training set enables the models to see more data and thus even tested with the original test set, the performance of any model does not go far off, since the model has seen the type of data in the test set. The performance of models trained with data augmentation is worse than that of models trained with the original training set, though, due to the inconsistency of distributions between the training set and test set when only one of them is augmented. fig. 11a shows, for models tested with the original test set only, the decrease of model performance from models trained with the original training set to models trained with data augmentation. The y-axis shows the logistic map of the ratio of the performance decrease, and is calculated by $L(x) = \frac{1}{e^{-\alpha x}}$ with $\alpha = 20$, $x = \frac{C_{original}}{C_{augmented}}$, and $C_{original}$ and $C_{augmented}$ are the numbers indicating the performance (in this case, either dice score or accuracy as shown in the figure) of models tested with only the original test set but trained with the original ($C_{original}$) or augmented ($C_{augmented}$) training set. We can see from fig. 11a that the performance of the equivariant models we propose decrease less. This shows, from one perspective, the resistance of equivariant models to inconsistency of data distributions between training and testing data. On the other hand, having data augmentation only in the test set becomes a big problem for models without equivariance. fig. 11b shows, for models trained with the original training set only, the performance decrease from models tested with the original test set to those tested with rotated data. The y-axis values are calculated the same as the formula above, but the $C_{original}$ and $C_{augmented}$ become the numbers indicating the performance of models trained with the original training set only but tested with the original ($C_{original}$) or rotated ($C_{augmented}$) test set. We can see clearly from fig. 11b as well that the performance of classical CNN decreases the most using rotated data, and the decrease of performance goes down when we enforce more spatial equivariance in the model. Baseline models decrease the least, but again, the performance is limited due to the lack of information in $\mathbb{R}^3$. Furthermore, the $SE(3)$-equivariance is implemented separately for the spatial and spherical parts, and is with interpolation in the spatial part, thus there are some errors introduced to it. Therefore, OursFull models always perform the best when there is variation in the test data.

Rotational invariance for Type 1 discretization

Further more, we have also experimented with networks that have some rotational invariance but in the classical CNN setup - viewing the DWI images as $I : \mathbb{R}^3 \to \mathbb{R}^N$. Taking the classical CNN setup we have in section 4.2, we rotate the CNN kernels in each layer using the same rotations as in section 4.2 to

Rotations	Data Type	CSF Dice	Subcortical Dice	WM Dice	GM Dice	Overall ACC
$90-5-5-5-FC$, #Param 13539						
Part(12)	Original	0.798 ± 0.058	0.425 ± 0.052	0.843 ± 0.01	0.875 ± 0.01	0.838 ± 0.011
	Rotated	0.71 ± 0.074	0.306 ± 0.042	0.755 ± 0.014	0.796 ± 0.014	0.75 ± 0.013
Full(60)	Original	0.754 ± 0.065	0.485 ± 0.059	0.823 ± 0.014	0.848 ± 0.02	0.818 ± 0.016
	Rotated	0.75 ± 0.063	0.479 ± 0.059	0.813 ± 0.013	0.838 ± 0.02	0.809 ± 0.016

Table 11: Augmented CNN tested with original and rotated data.

discretize $SO(3)$. As was done above, we use the 60 rotations from the icosahedral symmetry group as well as only 12 of them (1 at each rotation axis) to act on the CNN kernels. In each layer, one rotation of the kernel is only convolved with the response of the corresponding rotation from the last layer, thus this network is in fact 60 (or 12) independent networks, in which they share the same weights of different rotations. At the end, we take the average of the 60 (or 12) responses from all the rotations. With a small trial, we discovered that, as expected, even though this type of network does not perform as well as our spatial-directional GCNN as a whole, the performance decreases little in the full icosahedral group case with 60 rotations when tested with augmented data, and decreases more when only a subset (12) of the group is used to rotate the kernels. See table 11.

This further demonstrates that having rotational equivariance in the model makes it much more robust to variance in the data - which, with no need of explanation, is inevitable when dealing with real-world raw data. Averaging rotational copies of a classical CNN achieves the goal of dealing with variance in data, but for nonlinear data like DWI, for which signals in voxels have some geometric structure, our full $SE(3)$-GCNN provides the best solution.

4.3 Comparison to Müller et al. [10]

We now compare our method to the approach of [10]. They used DWI data with q-space encoding in the diffusion part and the spatial part of the data is referred to as p-space, and these two parts of the data resemble the $\mathbb{S}^2$ and $\mathbb{R}^3$ spaces in our formulation. We use the b-vectors from the HCP dataset as the input to the q-space. In their case, the input of the network is a whole DWI scan, not a series of extracted patches like we do, and we cannot fit an entire HCP scan into the model without exceeding the memory limit of a 24 GB GPU. After discussion and agreement with one of the authors (V. Golkov), we decided to use a modified architecture of their network to get an as fair as possible comparison: 1) we provide their network with patches of the same size as ours ($7 \times 7 \times 7$), but with DWI signals that are only normalized by $b0$ instead of interpolated spherical functions in each voxel like we did in our method. 2) The best performing model hyper-parameters they provided in the paper (with 4 and 5 layers in totals) are optimized for receptive fields that are much larger than ours, we use instead their 3-layer network, which has almost the same level of performance. 3) We have also disabled padding in their network in order to to cancel biases introduced in the networks. After 3 p-spatial layers, the output of their network without padding has spatial dimensions $1 \times 1 \times 1$.

Their method and ours thus perform the same task: voxel-wise classification. We used the Focal Loss [44] using the same parameters as all the experiments above. We used the suggested structure of their network with fully connected layers in the radial basis, which reportedly has better performance than ones without them. To make the comparison fair, we use a network whose hyper-parameters are different from what was presented in [9] such that the number of trainable parameters is similar to that of [10].

4.3.1 Network architectures

For [10], we use the $1(pq)+1(q-reduction)+2(p)$ layer structure with the $TP\pm$ 1 basis presented in their paper, and channels $(5,3,0,0)$, $(5,3,0,0)$, $(10,5,0,0)$, $(4,0,0,0)$ as presented in the appendix section E.1 in their paper, except that we changed the output channel to 4 to fit our multi-class classification task, and changed the p-space kernel sizes to 3 to ensure that the receptive field of the network is $7\times7\times7$, as we discussed with the author. For our method, we use a $ReLU(lift)-ReLU(gconv)-ReLU(gconv)-project-FC$ architecture such that there are 3 spatial layers as in [10]. With each layer split into 2, we use $10-10-20-40-20-10-proj.-4$ as our layer structure such that we have similar numbers of parameters as [10]. Our method has 34964 parameters, while [10] has 34781 parameters.

4.3.2 Results

Experiment / Class	CSF	Subcortical	WM	GM	Overall
Accuracy					
Ours	**0.804** $\pm$ 0.073	**0.754** $\pm$ 0.033	**0.871** $\pm$ 0.018	**0.908** $\pm$ 0.011	**0.882** $\pm$ 0.008
Müller's	0.583 $\pm$ 0.123	0.442 $\pm$ 0.176	0.83 $\pm$ 0.036	0.834 $\pm$ 0.033	0.805 $\pm$ 0.015
Dice score					
Ours	**0.799** $\pm$ 0.053	**0.722** $\pm$ 0.034	**0.877** $\pm$ 0.008	**0.908** $\pm$ 0.006	
Müller's	0.655 $\pm$ 0.086	0.41 $\pm$ 0.105	0.813 $\pm$ 0.015	0.849 $\pm$ 0.016	

Table 12: Statistics of results from both our method and Müller's method

The results are shown in table 12. We can see that our method performs better than [10]. To test the equivariance of both methods, we again test both models with the randomly rotated test set as presented above, and the results can be found in table 13.

We can see from the numbers that the performance of [10] does not drop much either while tested with unseen rotated test set, similar to our method. As we can see from fig. 12, overall, [10] lost less in percentage of the Dice scores of Subcortical, White matter, and overall accuracy, but more in CSF

Experiment / Class	CSF	Subcortical	WM	GM	Overall
Accuracy					
Ours	**0.725** ± 0.083	**0.596** ± 0.036	**0.834** ± 0.02	**0.874** ± 0.013	**0.838** ± 0.008
Müller's	0.445 ± 0.1	0.337 ± 0.146	0.823 ± 0.036	0.789 ± 0.031	0.771 ± 0.014
Dice score					
Ours	**0.742** ± 0.067	**0.593** ± 0.032	**0.832** ± 0.009	**0.875** ± 0.006	
Müller's	0.426 ± 0.055	0.343 ± 0.104	0.787 ± 0.015	0.813 ± 0.015	

Table 13: Statistics of results from both our method and Müller's method tested with rotated test set.

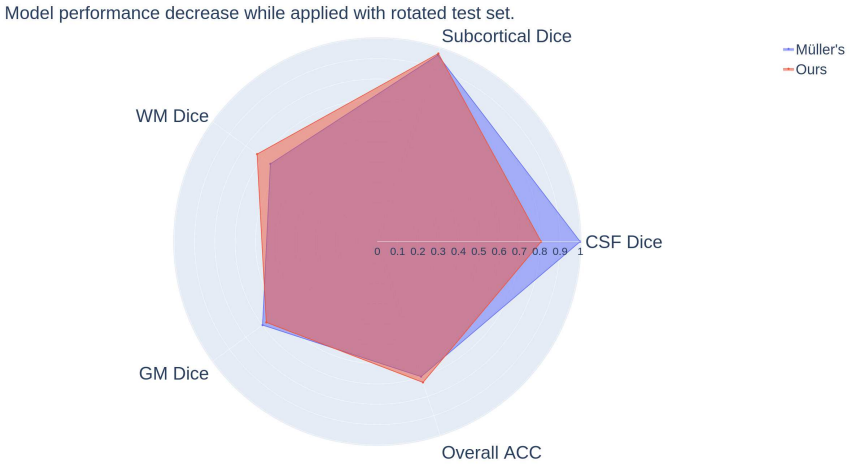


Fig. 12: Comparison of model performance decrease while applied with rotated test set between our method and Müller's. The radial axis indicates the decrease, and it is the logistic map of the ratio calculated by the same scheme used in fig. 11.

Dice score. Both equivariant methods are more resistant to variations in the distributions of the training and test set than the non-equivariant models presented above. Moreover, since the overall performance decrease of [10] while tested with rotated data is lower than our fully equivariant model, [10] actually has better equivariance than all models we presented even though their prediction accuracies and dice scores are lower.

5 Discussion

The resistance to data variation that has been shown by our fully equivariant network was demonstrated on synthetically augmented data - with 90-degree

rotations. Even though this synthetic augmentation did not cost any loss of signals or any interpolation-caused inaccuracy, it is desirable to verify the robustness of more complex group actions in CNNs using data with real-world variations (e.g. subjects scanned in different positions, affine variations in shapes). Acquiring this type of data is another challenge. On the other hand, data augmentation seems to be very robust against the variations in the rotated test set. However, this is because the augmentations applied in the training set and the test set are identical, they modeled exactly the same distribution in the data. Our proposed equivariant methods deal with inconsistent distributions between the training set and the test set much better, which is usually the case in real world. In addition, our method outperforms [10] with the same amount of information given to the models. Even though both methods show similar resistance to variations in the distributions of the training and test set, our model has a more light-weight implementation using regular group representation with separable kernels.

6 Conclusion

We presented a systematic study of GCNNs of various group actions with the application to DWI segmentation. We interpreted images of DWI scans ($I : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$) as functions in the homogeneous spaces of groups with different complexities of symmetries and provided a detailed analysis of how different levels of complexities of these symmetries impact the performance of the network. From the experiments, we conclude that 1) exploiting the spatial-directional interactions in the data is crucial for efficient learning of the features; 2) incorporating complex group actions of 3D rigid motions - SE(3) - might not be essential for highly aligned and preprocessed data like the human connectome project (HCP) [8], but it shows significantly higher resistance to variations in data. For real-world raw data in which positions of subjects are not perfectly aligned as in [8], our proposal shows great potential.

Acknowledgements

We would like to thank Dr. Vladimir Gorkov for his efforts and insights in helping us setting up experiments with their model [10].

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 801199. This paper only contains the author's views. The Research Executive Agency and the Commission are not responsible for any use that may be made of the information it contains. Data were provided [in part] by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University. This project is also partially funded by 3Shape A/S,

as well as by the research programme VENI (grant number 17290), financed by the Dutch Research Council (NWO).

References

[1] Masci, J., Boscaini, D., Bronstein, M., Vandergheynst, P.: Geodesic Convolutional Neural Networks on Riemannian Manifolds. In: Proceeding of 3dRR (2015)

[2] Cohen, T.S., Welling, M.: Group equivariant convolutional neural networks. In: Int. Conf. Machine Learning, pp. 2990–2999 (2016)

[3] Boscaini, D., Masci, J., Rodolà, E., Bronstein, M.: Learning Shape Correspondence With Anisotropic Convolutional Neural Networks. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 29 (2016)

[4] Bekkers, E.J., Veta, M.L.a.M., Eppenhof, K.A.J., Pluim, J.P.W., Duits, R.: Roto-Translation Covariant Convolutional Networks for Medical Image Analysis. In: Proc. MICCAI 2018, pp. 440–448 (2018)

[5] Cohen, T., Geiger, M., Weller, M.: A General Theory of Equivariant CNNs on Homogeneous Spaces. Advances in Neural Information Processing Systems (NeurIPS 2019) **32**, 9142–9153 (2020)

[6] Tuchs, D.S.: Q-Ball Imaging. Magnetic Resonance in Medicine **52**, 1358–1372 (2004)

[7] Bekkers, E.J.: B-spline cnns on lie groups. In: International Conference on Learning Representations (2019)

[8] Van Essen, D.C., Smith, S.M., Barch, D.M., Behrens, T.E.J., Yacoub, E., Ugurbil, K.: The WU-Minn Human Connectome Project: An Overview. NeuroImage **80**, 62–79 (2013)

[9] Liu, R., Lauze, F.B., Bekkers, E.J., Erleben, K., Darkner, S.: Group convolutional neural networks for DWI segmentation. In: Geometric Deep Learning in Medical Image Analysis (2022). https://openreview.net/forum?id=7S1l2zzUZFI

[10] Müller, P., Golkov, V., Tomassini, V., Cremers, D.: Rotation-Equivariant Deep Learning for Diffusion MRI (2021)

[11] Cohen, T.S., Geiger, M., Köhler, J., Welling, M.: Spherical CNNs. In: International Conference on Learning Representations (2018)

[12] Kondor, R., Trivedi, S.: On the Generalization of Equivariance and Convolution in Neural Networks to the Action of Compact Groups. In: Proc. ICML, pp. 2747–2755 (2018)

[13] T.S. Cohen and M. Weiler and B. Kicanaoglu and M. Welling: Gauge equivariant convolutional networks and the icosahedral cnn. In: Proc. ICML, pp. 1321–1330 (2019)

[14] Schonsheck, S.C., Dong, B., Lai, R.: Parallel Transport Convolution: A New Tool for Convolutional Neural Networks on Manifolds (2018)

[15] Sommer, S., Bronstein, A.M.: Horizontal Flows and Manifold Stochastics in Geometric Deep Learning. IEEE Trans. PAMI (2020)

[16] Elaldi, A., Dey, N., Kim, H., Gerig, G.: Equivariant spherical deconvolution: Learning sparse orientation distribution functions from spherical data. In: Feragen, A., Sommer, S., Schnabel, J., Nielsen, M. (eds.) Information Processing in Medical Imaging, pp. 267–278. Springer, Cham (2021)

[17] Bouza, J.J., Yang, C.-H., Vaillancourt, D., Vemuri, B.C.: A higher order manifold-valued convolutional neural network with applications to diffusion mri processing. In: Feragen, A., Sommer, S., Schnabel, J., Nielsen, M. (eds.) Information Processing in Medical Imaging, pp. 304–317. Springer, Cham (2021)

[18] Gens, R., Domingos, P.M.: Deep Symmetry networks. In: NIPS, pp. 2537–2545 (2014)

[19] Weiler, M., Hamprecht, F., Storath, M.: Learning Steerable Filters for Rotation Equivariant Cnns. In: Proc. CVPR, pp. 849–858 (2018)

[20] Weiler, M., Geiger, M., Welling, M., Boomsma, W., Cohen, T.S.: 3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data. In: Proc. NIPS (2018)

[21] Worrall, D.E., Garbin, S.J., Turmukhambetov, D., Brostow, G.J.: Harmonic Networks: Deep Translation and Rotation Equivariance (2017)

[22] Andrearczyk, V., Fageot, J., Depeursinge, A.: Local Rotation Invariance in 3D CNNs. Medical Image Analysis **65** (2020)

[23] Chakraborty, R., Banerjee, M., Vemuri, B.C.: A CNN for Homogeneous Riemannian Manifolds with Application to NeuroImaging (2018)

[24] Chakraborty, R., Banerjee, M., Vemuri, B.C.: H-cnns: Convolutional neural networks for riemannian homogeneous spaces. arXiv preprint

arXiv:1805.05487 **1** (2018)

[25] Chakraborty, R., Bouza, J., Manton, J., Vemuri, B.C.: Manifoldnet: A deep neural network for manifold-valued data with applications. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)

[26] Graham, S., Epstein, D., Rajpoot, N.: Dense steerable filter cnns for exploiting rotational symmetry in histology images. IEEE Transactions on Medical Imaging **39**(12), 4124–4136 (2020). https://doi.org/10.1109/TMI.2020.3013246

[27] Knigge, D.M., Romero, D.W., Bekkers, E.J.: Exploiting redundancy: Separable group convolutional networks on lie groups. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (eds.) Proceedings of the 39th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 162, pp. 11359–11386. PMLR, ??? (2022). https://proceedings.mlr.press/v162/knigge22a.html

[28] Aronsson, J.: Homogeneous Vector Bundles and $\mathcal{G}$-Equivariant Convolutional Neural Networks (2021)

[29] Smets, B.M.N., Portegies, J., Bekkers, E.J., Duits, R.: PDE-based Group Equivariant Convolutional Neural Networks (2021)

[30] Weiler, M., Forré, P., Verlinde, E., Welling, M.: Convolutional networks–isometry and gauge equivariant convolutions on riemannian manifolds. arXiv preprint arXiv:2106.06020 (2021)

[31] Golkov, V., Dosovitskit, A., Sperl, J.I., Menzel, M.I., Czisch, M., Särmann, P., Brox, T., Cremers, D.: q-Space Deep Learning: Twelve-Fold Shorter and Model-Free Diffusion MRI Scans. IEEE Trans. Med. Im. **35**(5), 1344–1351 (2016)

[32] Basser, P.J., Mattiello, J., LeBihan, D.: MR Diffusion Tensor Spectroscopy and Imaging. Biophys. J. **66**(1), 259–267 (1994)

[33] Caruyer, E., Verma, R.: On Facilitating the Use of HARDI in Population Studies by Creating Rotation-Invariant Markers. Medical Image Analysis **20**(1), 87–96 (2015)

[34] Schwab, E., Cetingül, H.E., Asfari, B., Vidal, E.: Rotational Invariant Features for HARDI. In: Proc. IPMI (2013)

[35] Novikov, D.S., Veraart, J., Jelescu, I.O., Fieremans, E.: Rotationally-Invariant Mapping of Scalar and Orientational Metrics of Neuronal Microstructure with Diffusion MRI. NeuroImage **174**, 518–538 (2018)

[36] Zucchelli, M., Deslauriers-Gauthier, S., Deriche, R.: A Computational Framework for Generating Rotation Invariant Features and its Application in Diffusion MRI. Medical Image Analysis **60** (2020)

[37] Banerjee, M., Chakraborty, R., Archer, D., Vaillancourt, D., Vemuri, B.C.: DMR-CNN: A CNN Tailored for DMR Scans with Applications to PD Calssifiaction. In: Proceedings of International Symposium on Biomedical Imaging (2019)

[38] Sedlar, S., Papadopoulo, T., Deriche, R., Deslauriers-Gauthier, S.: Diffusion MRI fiber orientation distribution function estimation using voxel-wise spherical U-net. In: International MICCAI Workshop 2020 - Computational Diffusion MRI, Lima, Peru (2020). https://hal.archives-ouvertes.fr/hal-02946371

[39] Sedlar, S., Alimi, A., Papadopoulo, T., Deriche, R., Deslauriers-Gauthier, S.: A Spherical Convolutional Neural Network for White Matter Structure Imaging via dMRI. In: de Bruijne, M., Cattin, P.C., Cotin, S., Padoy, N., Speidel, S., Zheng, Y., Essert, C. (eds.) Medical Image Computing and Computer Assisted Intervention – MICCAI 2021, pp. 529–539. Springe, ??? (2021)

[40] Poulenard, A., Ovsjanikov, M., Guibas, L.J.: Equivalence Between SE(3) Equivariant Networks via Steerable Kernels and Group Convolution. arXiv (2022). https://doi.org/10.48550/ARXIV.2211.15903. https://arxiv.org/abs/2211.15903

[41] Cohen, T.S., Welling, M.: Steerable CNNs. arXiv e-prints (2016)

[42] Jupp, P.E., Mardia, K.V.: A Unified View of the Theory of Directional Statistics, 1975-1988. International Statistical Review / Revue Internationale de Statistique **57**(3), 261–294 (1989)

[43] Liu, R., Lauze, F., Erleben, K., Darkner, S.: Bundle geodesic convolutional neural network for dwi segmentation from single scan learning. In: Cetin-Karayumak, S., Christiaens, D., Figini, M., Guevara, P., Gyori, N., Nath, V., Pieciak, T. (eds.) Computational Diffusion MRI, pp. 121–132. Springer, Cham (2021)

[44] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal Loss for Dense Object Detection (2018)