# Bundle Geodesic Convolutional Neural Network for Multishell DWI Voxel Classification

First Author[1 [0000−1111−2222−3333]], Second Author[2,3 [1111−2222−3333−4444]], and Third Author[3 [2222−−3333−4444−5555]]

**Abstract.** This work presents a Riemannian Deep Learning framework for extracting features with rotational invariance. Our main application is classification of multishell Magnetic Resonance Diffusion Weighted Imaging (DWI) at single voxel level. To that extent, we extend a G-CNN learning architecture, first in a generic way on a Riemannian manifold. It mainly consists of three layers: a lifting layer that locally represents and convolves data on tangent spaces to produce a family of functions defined on the rotation groups of the tangent spaces, i.e., a *section* of a bundle of rotational functions on the manifold; a group convolution layer that convolves this section with rotation kernels to produce a new section; and a projection by maximisation to collapse this local data to form new manifold based functions. We propose an instantiation on the 2-dimensional sphere where the DWI orientation data is in general represented, and we use it for voxel classification. We show that it provides a grey matter vs. non-grey matter classification that is competitive with reported results in the literature.

**Keywords:** DWI · Geodesic CNN · Classification.

## 1 Introduction

This work focuses on building a neural network (NN) for data on manifolds with some form of orientation invariance, and here we take Diffusion Weighted Imaging as the main application. Our goal is to be able to understand spherical patterns up to rotations. There are series of proposals to generalise a $\mathbb{R}^2$ convolutional neural network to curved spaces. In general, to define convolution, the underlying space must have a group structure, or be an homogeneous space of a group. This is not always the case for a curved space. But even when it is, this usually imposes a certain type of filters. In the case of a sphere, this means a rotationally symmetric filter. In our case, rotational invariance is a desirable property we wish freedom in design. We propose a general architecture for extracting and filtering local orientation information of data defined on a manifold. The architecture allows us to learn similar orientation structures which can appear at different locations on the manifold. Reasonable manifolds have local orientation structures – rotations on tangent spaces. Our architecture lifts data to these structures and performs local filtering on them, before collapsing them back to obtained filtered features on the manifold. This provides both rotational invariance and flexibility in design, without having to resort to

complex embeddings in Euclidean spaces. We provide an explicit construction of the architecture for multishell DWI data and show very promising results for this case.

**Related work** The importance of extraction of rotationally invariant features beyond Fractional Anisotropy [1] has been recognized in series of DWI works. [5] developed invariant polynomials of spherical harmonic (SH) expansion coefficients, and discussed their application in population studies. [16] proposed a related construction using eigenvalue decomposition of SH operators. [14] and [22] argued their usefulness for understanding microstructures in relation to DWI. However, there has been little exploration from the view point of deep learning (DL).

There is though a vast growth in literature on DL for non-flat data. [13] proposed a NN on surfaces that extracts local rotationally invariant features. A non rotationally invariant modification was proposed in [3]. On the other hand, convolution generalises to more group actions than just translation, and this has led to group-convolution neural networks for structures where these operations are supported, especially Lie groups themselves and their homogeneous spaces [10, 8, 20, 12, 2]. Global equivariance is often sought but proved complicated or even elusive in many cases when the underlying geometry is non-trivial [7]. An elementary construction on a general manifold is proposed in [15] via a fixed choice of paths used to transport filters between points on the manifold, ignoring the effects of path dependency (holonomy). Removing this dependency can be obtained by summarising local responses over local orientations, this is what is done in [13]. To explicitly deal with holonomy, [17] proposed a convolution construction on manifolds based on stochastic processes via the frame bundle, but it is at this point still very theoretical.

In this work we are interested in rotationally invariant features, so we take a path close to [15, 13], but we add an extra local group convolution layer before summarising the data and eliminating path dependency.

**Organisation** We introduce the construction in the next section, first in a general setting, then in our case of interest, the sphere $\mathbb{S}^2$. We present experiments and results in section 3. Discussion and conclusion are presented in section 4.

## 2 Bundle Geodesic Convolutional Neural Network

Bekkers *et al.* [2] used the fact that $SE(2)$ acts on $\mathbb{R}^2$ to lift 2D (vector-values) images to $\mathbb{R}^2 \times \mathbb{S}^1$ via *correlation kernels*. This is not in general the case when $\mathbb{R}^2$ is replaced by an oriented Riemannian manifold $\mathcal{M}$ as there is no roto-translation group defined on a general manifold. An alternative construction is however possible by combining [2] and [13], to obtain a 3-component layer architecture: i) the **lifting layer**, ii) the **group correlation layer**, and iii) **the projection layer**. In practical applicaitons, one or more of these multilayers can

be used and a fully connected layer is built upon the last one. In this section, we focus only on the Riemannian part.

We refer the readers to [4] for the Riemannian geometric constructions. In the sequel, a base point $\boldsymbol{x}_0$ is chosen on $\mathcal{M}$. A piecewise smooth path $\gamma$ joining $\boldsymbol{x}_0$ and $\boldsymbol{x} \in \mathcal{M}$ is a continuous curve which may fail to be smooth at a finite number of points. With such a curve, there is a *parallel transport* $P_\gamma$ between $T_{\boldsymbol{x}_0}\mathcal{M}$ and $T_{\boldsymbol{x}}\mathcal{M}$. This is an orientation preserving isometry between tangent spaces. A tangent *kernel* at $\boldsymbol{x}_0$ is a function $\kappa : T_{\boldsymbol{x}_0}\mathcal{M} \to \mathbb{R}^N$. We assume it has a "small support". A rotational kernel at $\boldsymbol{x}_0$ is a function $K : SO(\boldsymbol{x}_0) \to \mathbb{R}^M$, where, $SO(\boldsymbol{x})$ denotes the rotation group of $T_{\boldsymbol{x}}\mathcal{M}$.

### 2.1 Layer definitions

As it is usually the case that correlation replaces convolution in convolutional neural networks (CNN). The first two layers will be defined via correlation.
**Lifting layer.** The correlation $f\tilde{\star}_\gamma\kappa$ of $f \in L^2(\mathcal{M}, \mathbb{R}^N)$ is defined as the function on $SO(\boldsymbol{x})$

$$f\tilde{\star}_\gamma\kappa(S) = \sum_{i=1}^{N} \int_{T_{\boldsymbol{x}}\mathcal{M}} \kappa_i(P_\gamma^{-1}S^{-1}v) f_i(\mathrm{Exp}_{\boldsymbol{x}}(v))\, dv \tag{1}$$

We assume that $\kappa \circ P_\gamma^{-1}$, the support of $\kappa$, is sufficiently small so that the exponential map is injective. For any other path $\delta$ between $\boldsymbol{x}_0$ and $\boldsymbol{x}$, it is easy to show that there exists a rotation $R \in SO(T_{\boldsymbol{x}}\mathcal{M})$ that only depends on $P_\gamma$ and $P_\delta$ with $f\tilde{\star}_\delta\kappa(S) = f\tilde{\star}_\gamma\kappa(RS)$. For any point $\boldsymbol{x}$ and a path $\gamma_{\boldsymbol{x}}$ between $\boldsymbol{x}_0$ and $\boldsymbol{x}$, this filters/lifts $f$ to functions $F_{\boldsymbol{x}} : SO(\boldsymbol{x}) \to \mathbb{R}$. Using an input $f^{(\ell-1)} : \mathcal{M} \to \mathbb{R}^{N_{\ell-1}}$ and $N_\ell$ $\boldsymbol{x}_0$-kernels $\boldsymbol{\kappa}^{(\ell)} = \left(\kappa_1^{(\ell)}, \ldots, \kappa_{N_\ell}^{(\ell)}\right)$, $\kappa_i^{(\ell)}$ $\mathbb{R}^{N_{\ell-1}}$-valued at layer $\ell - 1$,

$$\forall \boldsymbol{x} \in \mathcal{M}, \quad F_{\boldsymbol{x}}^{(\ell)} = \left(f^{(\ell-1)}\tilde{\star}_{\gamma_{\boldsymbol{x}}}\kappa^1, \ldots, f^{(\ell-1)}\tilde{\star}_{\gamma_{\boldsymbol{x}}}\kappa_{N_\ell}\right) \tag{2}$$

The output $F^{(\ell)}$ is not a function defined on $\mathcal{M}$, but a *section*, in general non smooth, of the *function bundle* $\mathbb{L}^2\left(SO(\mathcal{M}), \mathbb{R}^{N_\ell}\right) = \sqcup_{\boldsymbol{x}} L^2(SO(\boldsymbol{x}), \mathbb{R}^{N_\ell})$.
**Group correlation layer.** if $F$ is a function $SO(\boldsymbol{x}) \to \mathbb{R}^M$, we define $F \star_\gamma K$ as the *classical* group correlation

$$F \star_\gamma K(S) = \sum_{i=1}^{M} \int_{SO(\boldsymbol{x})} F_i(U) K_i(P_\gamma^{-1}S^{-1}UP_\gamma)\, dU. \tag{3}$$

This construction provides a new family of functions $\bar{F}_{\boldsymbol{x}} : SO(\boldsymbol{x}) \to \mathbb{R}$. Differing from [2], translations are in general not defined in $\mathcal{M}$ and rotations are only local. If $F_\gamma = (f_i\tilde{\star}_\gamma\kappa_i)_{i=1}^M$ and $F_\delta = (f_i\tilde{\star}_\delta\kappa_i)_{i=1}^M$ then it can be easily shown using the bi-invariance of the Haar measure on $SO(n)$ that $\varphi(F_\delta)\star_\delta K(S) = \varphi(F_\gamma)\star_\gamma K(SR)$ where $R$ depends only on paths $\gamma$ and $\delta$, and $\varphi$ is any real function (typically a rectified linear unit (ReLU)). With input $F^{(\ell-1)} \in \mathbb{L}^2\left(SO(\mathcal{M}), \mathbb{R}^{N_{\ell-1}}\right)$ with

$N_{\ell-1}$ channels at layer $\ell-1$ and $\boldsymbol{x}_0$-rotation kernels $\boldsymbol{K}^{(\ell)} = \left(K_1^{(\ell)}, \ldots, K_{N_\ell}^{(\ell)}\right)$, each with $N_{\ell-1}$ channels, one obtains $F^{(\ell)} \in \mathbb{L}^2(SO(\mathcal{M}), \mathbb{R}^{N_\ell})$ as

$$F_{\boldsymbol{x}}^{(\ell)} = \left(F^{(\ell-1)} \star_{\gamma_{\boldsymbol{x}}} K_1^{(\ell)}, \ldots, F^{(\ell-1)} \star_{\gamma_{\boldsymbol{x}}} K_{N_\ell}^{(\ell)}\right) \tag{4}$$

**Projection layer.** A family $F^{(\ell-1)} \in \mathbb{L}^2\left(SO(\mathcal{M}), \mathbb{R}^{N_{(\ell-1)}}\right)$ is projected to a function $f : \mathcal{M} \to \mathbb{R}^{N_{(\ell-1)}}$ as

$$f_i^{(\ell)}(x) = \max_{S \in SO(\boldsymbol{x})} F_{i\,\boldsymbol{x}}^{(\ell-1)}(S), \quad i = 1 \ldots N_{\ell-1} \tag{5}$$

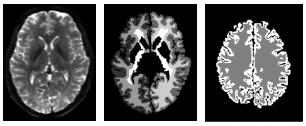
This actually removes the path dependency thanks to the change of path property which was described above. See fig. 1a for illustration.

Biases are added per kernel. Nonlinear transformations of ReLU type are applied after each of these layers. Note that without them, a lifting followed by group correlation would actually factor in a new lifting transformation.

### 2.2 Discretisation and implementation in the case $\mathcal{M} = \mathbb{S}^2$

In this work, the manifold of interest is $\mathbb{S}^2$. Spherical functions $f : \mathbb{S}^2 \to \mathbb{R}^N$ are typically given at a number of points and interpolated using a Watson kernel [11], which also serves as our choice. We use a very simple discretisation of $\mathbb{S}^2$ via the vertices of a regular icosahedron. Tangent kernels are defined over these vertices, sampled along the rays of a polar coordinate system respecting the vertices of the icosahedron. This is illustrated in fig. 1b.

## 3 Experiments & Results

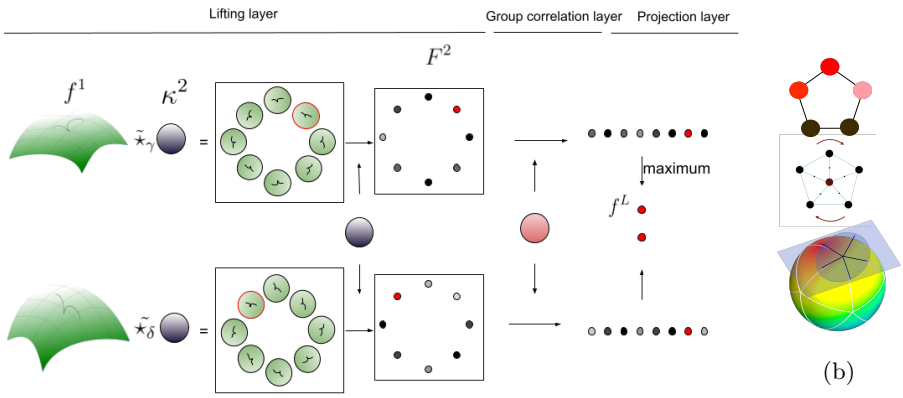


(a) (b) (c)

Fig. 2: Left to right: original diffusion data, the ground-truth segmentation, and the binary ground-truth that we are going to learn from.

We evaluate our kernels on DWI data from the human connectome project [19]. We train a network using our framework on the individual voxels containing signals on $\mathbb{S}^2$. Our goal is a voxel-wise classification of grey matter and the rest of the brain.

We used the pre-processed DWI data [18] and normalised each DWI scan for all 3 b-values (shells) with the voxel-wise average of the $B_0$. The labels provided with the T1-image was transformed to the DWI using a nearest neighbour interpolation (fig. 2). We estimated the partial-volume error from the relabeling to be roughly 6% thus the expected maximum accuracy was estimated to be 94%. We set up a binary classification task using the kernels we have defined - segmenting the grey matter and the rest of the brain. See fig. 2c.




(a) (b)

Fig. 1: In fig. 1a, the top row shows the lifting kernel $\kappa^{(2)}$ applied at a point on the manifold, resulting in an image $F^{(2)}$ defined on $SO(2)$ as in eq. (1). The function is first mapped onto the tangent space of the point of interest via the exponential map, and $\kappa^{(2)}$ is convolved with the mapped function to get $F^2$. In the figure we rotate the tangent space instead of the kernel as eq. (1), but they are equivalent constructions. We get rotationally invariant responses from the projection layer. The bottom row shows the same process but with a different kernel parallel transport, illustrating that the responses of the convolutional layers are simply rotated. In fig. 1b, the bottom row shows $\mathbb{S}^2$ with a regular icosahedric tessellation and a tangent plane at one of the vertices and 5 sampled directions. The disk represents the kernel support. The middle row shows the actual discrete kernel used, with the $2\pi/5$ rotations and the top row is represents the lifted function on the discrete rotation group.

### 3.1 Experimental setup

After getting the responses from our proposed layers, we feed them into a fully connected network to perform our classification task. We design our experiments with respect to 2 criteria: kernel describability and the number of scan shells.

**Describability** We used 2 different network architectures to test the describability of our kernels. We first design a baseline experiment where the network directly connects the stacked responses from our kernels to an output layer without hidden layers in between to evaluate the describability of our kernels. Then, to explore the full capacity of the method for the classification task, we designed a full model experiment that includes hidden layers to compare with the baseline. Each layer, as well as the lifting and group convolution layer, is followed by a ReLU activation layer.

**Number of Scan Shells.** We further modified the architectures to include different numbers of shells ($s \in 1, 2, 3$) of the scans in the input data to see its impact on the results. For those experiments with multiple shells, we defined

independent kernels for each shell, after which we stack the responses from all shells and feed the resulting vector to the fully connected network.

In addition to the criteria described above, we have also experimented with the impact of the resolution of the kernel definition on model fitting. There are 2 kinds of resolution that we analysed: kernel orientation resolution and kernel location resolution. Kernel orientation resolution is the density of samples that a kernel covers. It has 2 factors: the number of rays ($nR$) of the circular kernel and the number of samples per ray ($nS$). We test with 2 setups of the kernel resolution: $nR = 5, nS = 2$ and $nR = 30, nS = 10$. Kernel location resolution is the density of locations that we define the kernels on, which can be achieved by subdividing the icosahedron to get a higher resolution polyhedron. Having explored different configurations of resolution (both orientation and location resolution), we observed that subdividing the icosahedron to have a higher density of kernel locations introduces significant model overfitting. This, however, is expected since there are only 90 values (signals) on $\mathbb{S}^2$ for each voxel, and a high density of kernel locations will results in unnecessary complexity on the model, which leads to model overfitting. Increasing the orientation resolution of single kernels without subdivision of the icosahedron does not boost the performance either (it even leads to overfitting in some cases), which can be explained by the sparsity of signals on $\mathbb{S}^2$ in our data as well.

Therefore, we use the icosahedron structure as kernel locations with low orientation resolution of the kernels - 5 rays per kernel, and 2 sample points per ray. The radius of the kernels should guarantee that the kernel coverage of 2 adjacent icosahedron vertices will overlap with each other, therefore we choose 0.6 as our radius. We use 10 kernels for both lifting and group convolution layers.

We have different dimensions of the input layer of the fully connected network within each experiment setup due to the different numbers of shells used in the data. The network structures for both experiment setups are given in the top table in table 1.

| Shells / Experiment | Baseline | Full model |
|---|---|---|
| 1 | 1200,2 | 1200,500,200,100,50,10,2 |
| 2 | 2400,2 | 2400,1000,500,200,100,50,10,2 |
| 3 | 3600,2 | 3600,1500,1000,500,200,100,50,10,2 |

| | | |
|---|---|---|
| 1 (Accuracy/Recall/Dice) | 0.805/0.820/0.773 | 0.805/0.778/0.785 |
| 2 | 0.847/0.835/0.828 | 0.861/**0.865**/0.840 |
| 3 | 0.850/0.826/0.837 | **0.864**/0.862/**0.845** |

Table 1: The top table shows the illustration of numbers of neurons in each fully connected layer for experiment setups, and the bottom table shows the results of each experiment setups.

### 3.2 Results

Our dataset contains 53 scans from the human connectome project [19]. We use 2 scans for training, 1 scan for validation, and 50 scans for testing.

We train each network presented in table 1 for 5 epochs with batch size 100 on an Ubuntu 20.04.2 LTS machine with an Intel Xeon(R) Silver 4210 CPU @ 2.20GHz × 40 processor and a GEFORCE RTX 3090 graphics card. Our framework is implemented in Python 3.6 and Pytorch 1.7. According to the experiment setup, the runtime for 1 epoch varies from ca. 3 min (baseline with 1 shell) to ca. 8 min (full model with 3 shells), and the memory usage varies from 1295 MiB to 1481 MiB. The statistical results (accuracy, recall and Dice score) are shown in the bottom table in table 1. It is easy to see the correlation between the number of shells included and the increase in performance of the method. This is of no surprise since it is axiomatic that more information in the data aids the learning. Also, increasing the capacity of the network improves performance, however, not to the same extent as the inclusion of additional shells. This demonstrates that our kernels are able to encode the data well. The results in table 1 are aggregated from all subject scans in the test set combined, therefore we also analyse the test statistics across individual scans. As shown in fig. 3, we see that the distributions of all 3 evaluation criteria are reasonably tight, meaning that we have stable performances across individual subjects. This demonstrates that even trained on only 2 patients, our model generalises very well to a large scale of different patients. See fig. 4 for examples of predictions from our model (full model with 3 shells).

## 4 Discussion and Conclusion

The proposed method is a simple extension of CNN to Riemannian Manifolds which learns rotationally invariant features. The Bundle G-CNN capability has been demonstrated on a simple non-flat manifold, $\mathbb{S}^2$, and been used to build a voxel-wise classification of DWI data to recognise grey matter, with an accuracy of 86% which is on par with standard frameworks [9]. After correcting for the uncertainty of the propagated labels we achieve close to 92% correct classification. Our results compare well with existing methods [21, 6]. This is to the best of our knowledge, the first learning technique that explicitly learns from multishell input, and it has promising applications in understanding patterns of pathology. We have so far only tested it on $\mathbb{S}^2$, however, an extension to other surfaces appears easy, though the choice of a grid might be important. An extension to dimension 3 will require efficient $SO(3)$ convolutions, potentially using generalised spectral theory for compact Lie groups.

## 5 Acknowledgements

Acknowledgements go there.

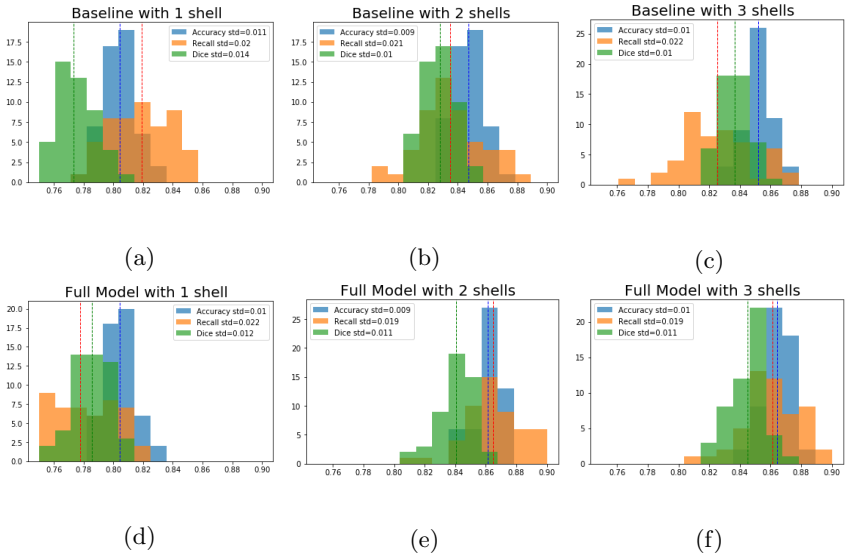



Fig. 3: Histograms of accuracy, recall and Dice score across individual scans in the test set. The mean and standard deviation (std) are shown in the figure as well.

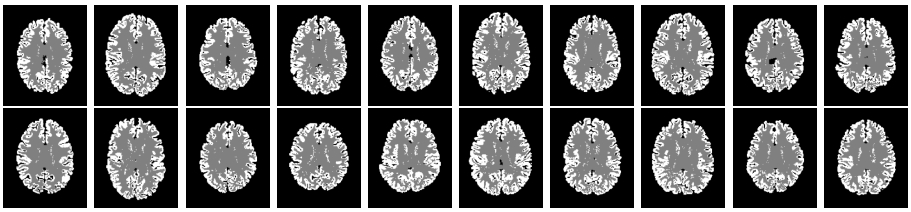


Fig. 4: Examples of predictions of grey/white matter of patients in the test set. These predictions are from the full model with 3 shells included in the data.

## References


1. Basser, P., Mattiello, J., LeBihan, D.: Mr diffusion tensor spectroscopy and imaging. Biophys. J. **66**(1), 259–267 (1994)
2. Bekkers, E., Veta, M.L.M., Eppenhof, K., Pluim, J., Duits, R.: Roto-translation covariant convolutional networks for medical image analysis. In: Proc. MICCAI 2018. pp. 440–448 (2018)
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. Carmo, M.P.D.: Riemannian Geometry. Birkhauser (1992)
5. 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)
6. Cheng, H., Newman, S., M. Afzali, S.S.F., Garyfallidis, E.: Segmentation of the brain using direction-averaged signal of dwi images. Magnetic Resonance Imaging **69**, 1–7 (2020)
7. Cohen, T., Weiler, M., Kicanaoglu, B., Welling, M.: Gauge equivariant convolutional networks and the icosahedral cnn. In: Proc. ICML. pp. 1321–1330 (2019)
8. Cohen, T., Welling, M.: Group equivariant convolutional neural networks. In: Int. Conf. Machine Learning. pp. 2990–2999 (2016)
9. Eggert, L.D., Sommer, J., Jansen, A., Kircher, T., Konrad, C.: Accuracy and reliability of automated gray matter segmentation pathways on real and simulated structural magnetic resonance images of the human brain. PloS one **7**(9), e45081 (2012)
10. Gens, R., Domingos, P.: Deep symmetry networks. In: Advances in neural information processing systems. pp. 2537—2545 (2014)
11. 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)
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. Masci, J., Boscaini, D., Bronstein, M., Vandergheynst, P.: Geodesic convolutional neural networks on riemannian manifolds. In: Proceeding of 3dRR (2015)
14. Novikov, D., Veraart, J., Jelescu, I., Fieremans, E.: Rotationally-invariant mapping of scalar and orientational metrics of neuronal microstructure with diffusion mri. NeuroImage **174**, 518–538 (2018)
15. Schonsheck, S.C., Dong, B., Lai, R.: Parallel transport convolution: A new tool for convolutional neural networks on manifolds (2018)
16. Schwab, E., Cetingül, H.E., Asfari, B., Vidal, E.: Rotational invariant features for hardi. In: Proc. IPMI (2013)
17. Sommer, S., Bronstein, A.: Horizontal flows and manifold stochastics in geometric deep learning. IEEE Trans. PAMI (2020)
18. Sotiropoulos, S., Moeller, S., Jbabdi, S., Xu, J., Andersson, J.L., Auerbach, E.J., Yacoub, E., Feinberg, D., Setsompop, K., Wald, L., Behrens, T.E.J., Ugurbil, K., Lenglet, C.: Effects of image reconstruction on fiber orientation mapping from multichannel diffusion mri: Reducing the noise floor using sense. Magnetic Resonance in Medicine **70**(6), 1682 – 1689 (2013)

19. Van Essen, D.C., Smith, S.M., Barch, D.M., Behrens, T.E., Yacoub, E., Ugurbil, K.: The wu-minn human connectome project: An overview. NeuroImage **80**, 62 – 79 (2013)
20. Worrall, D., Garbin, S., Turmukhambetov, D., Brostow, G.: Harmonic networks: Deep translation and rotation equivariance (2017)
21. Yap, P.T., Zhang, Y., Shen, D.: Brain tissue segmentation based on diffusion mri using 0 sparse-group representation classification. In: Proc. MICCAI - III. pp. 132–139 (2015)
22. 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)