GAUGE EQUIVARIANT CONVOLUTIONAL NEURAL NETWORKS FOR DIFFUSION MRI

Uzair Hussain
Centre for Functional and Metabolic Mapping, Robarts Research Institute, Western University, London, Canada
ughussain@gmail.com

Ali R. Khan[*] [†]
Centre for Functional and Metabolic Mapping, Robarts Research Institute, Western University, London, Canada
Department of Medical Biophysics, Schulich School of Medicine and Dentistry, Western University, London, Canada
Western Institute for Neuroscience, Western University, London, Canada
ali.khan@uwo.ca

June 9, 2023

ABSTRACT

Diffusion MRI (dMRI) is an imaging technique widely used in neuroimaging research, where the signal carries directional information of underlying neuronal fibres, based on the diffusivity of water molecules. One of the shortcomings of dMRI is that numerous images, sampled at gradient directions on a sphere, must be acquired to achieve a reliable angular resolution for model-fitting, which translates to longer scan times, higher costs, and barriers to clinical adoption. In this work we introduce gauge equivariant convolutional neural network (gCNN) layers that overcome the challenges associated with the dMRI signal being acquired on a sphere with antipodal points identified, by making it equivalent to the real projective plane, $\mathbb{R}P^2$, which is a non-euclidean and a non-orientable manifold. This is in stark contrast to a rectangular grid which typical convolutional neural networks (CNNs) are designed for. We apply our method to upsample angular resolution for predicting diffusion tensor imaging (DTI) parameters from just six diffusion gradient directions. The symmetries introduced allow gCNNs the ability to train with fewer subjects and are general enough to be applied to many dMRI related problems.

[*]Corresponding author
[†]Mailing address: 100 Perth Dr, London, ON N6A 5K8, Canada

1 Introduction

Diffusion weighted imaging (DWI) or diffusion MRI (dMRI) is a type of imaging that conveys microstructural information of underlying tissue, by tracking the diffusion rate of water particles along different directions [20]. Applied to brain imaging, this gives dMRI the unique ability to map the integrity and direction of white matter tracts [16] and also the orientation and dispersion of neurites in grey matter [26]. Diffusion tensor imaging (DTI) is the most commonly used model applied to dMRI datasets, and it aims to predict the diffusion tensor at each location of the image [1]. The eigenvalues of the tensor provide useful rotationally-invariant micro-structural metrics, such as fractional anisotropy (FA) and mean diffusivity (MD). The first eigenvector of the tensor (V1), provides an estimate of the underlying direction of white matter fibres, and can be used to perform tractography for mapping white matter tracts in the brain.

One of the drawbacks of dMRI and analysis techniques like DTI is that many images are required, one for each direction we want measure the diffusion rate along. Although theoretically the diffusion tensor only requires six directions to be fully determined mathematically, in practice more directions are needed to reduce noise, which translates into longer scan times and higher costs. Recently, many machine learning based methods for dMRI have been proposed to address this issue and reduce the noise generated from using fewer directions [18, 21, 14, 11, 3]. The earliest of these models, q-space deep learning (q-DL) [11] was able to show that a simple three-layer multi-layer perceptron (MLP) was able to accurately estimate diffusion kurtosis and neurite orientation dispersion and density imaging (NODDI) [26] parameters with fewer directions thereby reducing scan time by twelve-fold. However, this MLP approach did not take advantage of the inherent spatial correlation between voxels in doing so. Convolutional neural networks (CNNs) are a highly-effective means to leverage spatial correlations present in images. For a single scalar 3D image, one can easily employ 3D convolutions, but for dMRI, at each voxel, we have many signals, one for each direction the signal is sampled from in q-space. Thus, at each voxel the signal is realized on a unit sphere, with directions uniformly distributed on the sphere. Further, because of the inherent antipodal symmetry in the diffusion signals, the domain for the signal is actually the set of all unoriented lines from the origin of $\mathbb{R}^3$, i.e., the real projective plane, $\mathbb{R}P^2$ [10].

Standard CNNs, developed for computer vision tasks, make use of flat, orientable manifolds, and thus are not readily applicable to the non-euclidean and non-orientable nature of the real projective plane. To circumvent these obstacles, one approach taken by authors is to perform convolutions in the spectral space [18]. This approach has the advantage of easily implementing rotational equivariance by employing Wigner matrices acting on spherical harmonics [22], however, a downside is that moving to spectral space introduces an extra level of processing. Another approach is to limit the domain to six predefined directions, then apply regular 3d convolutions while taking the six directions as different channels [21]. Although this method has the advantage of readily using available 3D convolution toolboxes, the directions sampled must be fixed to these six directions, which is rarely the case in practice. Further, this approach does not have the rotational equivariance property. Another promising approach is to use models developed for natural language processing (NLP) like the attention based transformer architecture [14, 24]. However, this requires

substantially more subjects for training, which is infeasible in many research or clinical applications. Also, similar to [21], it requires six predefined directions and lacks rotational equivariance.

In this work we apply the theory of gauge equivariant CNNs (gCNNs) [5, 4], to introduce for the first time, convolutional layers for the real projective plane, $\mathbb{RP}^2$, and apply these to the problem of denoising DTI metrics derived from six directions. Using gCNNs allows us to perform convolutions on non-euclidean manifolds, and by modelling $\mathbb{R}P^2$ as the top half of an icosahedron we are able to take convolutions on a non-orientable manifold as well. In addition, we are not limited to a predefined and fixed set of diffusion gradient directions, as our approach is general and can be applied with any set of uniformly distributed directions. Further, our method is not limited to DTI, and the gCNN layers developed here with the accompanying open-source implementation are general and can be applied to any dMRI related problem or architecture where the signal is acquired on a unit sphere with antipodal symmetry. A particularly beneficial side-effect of using gCNNs is a higher learning efficiency brought about by manually rotating and reflecting convolution kernels, as opposed to learning these transformations, and as a result we are able to reach similar accuracy levels with significantly fewer training subjects.

2 Theory

The theory of gCNNs is developed in [4], which is then elegantly applied to the icosahedron in [5]. For completeness we review some aspects of the theory relevant to this work, and also show how it can be applied to the real projective plane, $\mathbb{R}P^2$, which is the manifold of interest for dMRI.

2.1 Gauge transformations

A gauge usually means a choice of some reference point. E.g., in a literal gauge like a speedometer, we calibrate the needle to a zero on the markings, and measure speed against that zero. This idea also applies when viewing coordinate systems as a reference, and then coordinate transformations are a type of gauge transformation. One simple example of a gauge transformation is a rotation of the 2d Cartesian plane by some angle.

More formally, following the notation and definitions of [5], we define a gauge as a position-dependent invertible linear map $w_p : \mathbb{R}^d \to T_pM$ where T_pM is the tangent space of a d-dimensional manifold M at point p. If $\{e_a\}$ is the standard Cartesian frame of $\mathbb{R}^d$ then $\{w_p(e_a)\}$ is the frame in T_pM. A gauge transformation, $g_p \in GL(d, \mathbb{R})$, is a position-dependent mapping where $GL(d, \mathbb{R})$ is the general linear group (the group of invertible $d \times d$ matrices). To apply the gauge transformation, one composes w_p with g_p, i.e., $w_p \to w_p g_p$. In addition, components of a vector, v, in the basis $\{w_p(e_a)\}$, transform as $v \to g_p^{-1} v$, leaving the vector $(w_p g_p)(g_p^{-1} v) = w_p v \in T_pM$ invariant. We can see this invariance in the previous example of the 2D plane and the standard Cartesian basis, now w_p is the identity map, and g_p is a rotation matrix, R. Let $e_1 = \hat{x}$ and $e_2 = \hat{y}$. Then we have $V = V^x \hat{x} + V^y \hat{y}$. Under a coordinate rotation we have $\hat{x}' = R\hat{x}$, and $\hat{y}' = R\hat{y}$, in the new basis $V = V^{x'} \hat{x}' + V^{y'} \hat{y}' = (R^{-1}V)^x R\hat{x} + (R^{-1}V)^y R\hat{y} = V^x \hat{x} + V^y \hat{y}$.

In general, at a point p on M, we have other "fibres" of which d-dimensional vectors in T_pM are a special case. For the purposes of neural networks, a natural object to consider is a feature field f with C channels. If under a gauge transformation, f transforms as $f(p) \to \rho(g_p^{-1})f(p)$, we will call f a ρ-field where $\rho : G \to GL(C, \mathbb{R})$ is a representation of G. For example, consider a change of coordinate systems in d-dimensional Cartesian space by a constant translation, $x^i \to x^i + t^i$. Then a vector V^i transforms as $V^i \to \sum_j J^i_j V^j$, where J is the Jacobian matrix. On the other hand, a complex valued function $f(x^i) = e^{ik_jx^j}$, transforms as $f(x) \to e^{ik_j(x^j-t^j)} = e^{-ik_jt^j}f(x)$. Here the Jacobian matrix is an element of $GL(d, \mathbb{R})$ but $\rho = e^{-ik_jt^j}$ is an element of $\mathbb{C}$ (in this particular example we have to extend our field of numbers from real to complex).

To arrive at the definition of gauge equivariant convolutions we need one more ingredient, the exponential map. Let $\gamma_V : \mathbb{R} \to M$ be a geodesic such that $\gamma(0) = p$ and the geodesic tangent is $\dot{\gamma}(0) = V$. Then, the exponential map is defined simply by $\exp_p V := \gamma_V(1)$. This map walks a point along the geodesic starting at p with velocity $||V||$ for one unit of time arriving at $q = \exp_p V$ [17].

2.2 Gauge equivariant convolution

Suppose we have a feature map which is a C_{in}-dimensional ρ_{in}-field, f. We define the following convolution [5]:

$$(K \star f)(p) = \int_{\mathbb{R}^d} K(v)\rho_{\text{in}}\left(g_{p\leftarrow q_v}\right) f(q_v)dv, \tag{1}$$

where $v \in T_pM$, $K : \mathbb{R}^d \to \mathbb{R}^{C_{\text{out}}\times C_{\text{in}}}$ is the kernel, $q_v = \exp_p w_p(v)$ and $\rho_{\text{in}}(g_{p\leftarrow q_v}) \in \mathbb{R}^{C_{\text{in}}\times C_{\text{in}}}$ is the transformation that $f(q_v)$ undergoes as it is transported to p. Under a gauge transformation these objects transform as,

$$v \to g_p^{-1}v, \qquad f(q_v) \to \rho_{\text{in}}(g_{q_v}^{-1})f(q_v), \tag{2}$$

$$w_p \to w_pg_p, \qquad g_{p\to q_v} \to g_p^{-1}g_{p\leftarrow q_v}g_{q_v}. \tag{3}$$

In addition, we require that $K \star f$ be a ρ_{out}-field, i.e., it should transform as, $(K \star f)(p) \to \rho_{\text{out}}(g_p^{-1})(K \star f)(p)$. This is achieved if and only if,

$$K(g^{-1}v) = \rho_{\text{out}}(g^{-1})K(v)\rho_{\text{in}}(g) \tag{4}$$

for all $g \in G$.

2.3 Group convolutions of hexagonal filters

In this study we will be using group convolutions with hexagonal filters [12]. We fill first briefly review group convolutions for flat manifolds and then show how it connects with Equation 1. For a group G, group convolution is abstractly defined as[4, 25],

$$(\zeta \circledast \Psi)(g) = \int_G \zeta(h)\Psi(h^{-1}g)d\lambda(h), \tag{5}$$

where $g, h \in G$, $\zeta : G \to \mathbb{R}$ is a feature map, $\Psi : G \to \mathbb{R}$ is a filter, and λ is a Haar measure. For hexagonal filters the group G is the dihedral group, D_6. As we will see below, since the manifold we are using is the real projective plane, $\mathbb{R}P^2$, our group, D_6, includes reflections in addition to rotations, giving 12 elements. Writing these elements out, we have,

$$D_6 = \{1, R, R^2, \ldots, R^5, S, RS, \ldots, R^5 S\}, \tag{6}$$

where R is a rotation by $\Delta\theta = 2\pi/6$. Letting $\theta_i = \Delta\theta i$ for $i \in \{0, 5\}$, we then have $R_{\theta_i} = R^i$. We take S to be the reflection along the $\theta_i = 0$ axis; this implies that RS is a reflection along the $\theta_1 = \Delta\theta/2$ axis, $R^2 S$ is a reflection along the $\theta_2 = 2\Delta\theta/2$ axis, and so on. To match the notation of the rotations we will denote reflections as $S_{\theta_i} = R^i S$, but note that this does not mean that the reflection is along the θ_i axis, but rather half of that. Consider the inverses of these operations, $R_{\theta_i}^{-1} = R_{-\theta_i} = R_{\Delta\theta(-i \mod 6)}$. For the inverse of reflections we have $S_{\theta_i}^{-1} = (R_{\theta_i} S)^{-1} = S R_{-\theta_i}$. This can be simplified further, noting that $(R_{\theta_i} S)^2 = 1 \implies S R_{\theta_i} S = R_{\theta_i}^{-1} \implies R_{\theta_i} S = S^{-1} R_{\theta_i}^{-1} = S R_{-\theta_i}$. The first equality comes from the fact that if you apply reflection twice it's the same as identity, then the rest follows. This means that $S_{\theta_i}^{-1} = (R_{\theta_i} S)^{-1} = S R_{-\theta_i} = R_{\theta_i} S = S_{\theta_i}$.

With $x = (x^1, x^2) \in \mathbb{R}^2$, let $I_c(x)$ be the c-th channel of the input and $\Psi_{\hat{c}c}^{(1)}(x)$ be the hexagonal filter for the first layer [12, 5]. Group operations on the filter are denoted as $R_{\theta_i}\Psi = \Psi\left(R_{\theta_i}^{-1} x\right)$ (omitting indices for clarity), and $S_{\theta_i}\Psi = \Psi\left(S_{\theta_i}^{-1} x\right)$. Then the group convolution for the input (scalar) layers is given by [6],

$$\begin{aligned} y_{\hat{c}}^{(1)}\left(x, \theta_i^R\right) &= \sum_{c=1}^{C_{\text{in}}} \left(I_c * R_{\theta_i} \Psi_{\hat{c}c}^{(1)}\right)(x) \\ y_{\hat{c}}^{(1)}\left(x, \theta_i^S\right) &= \sum_{c=1}^{C_{\text{in}}} \left(I_c * S_{\theta_i} \Psi_{\hat{c}c}^{(1)}\right)(x) \end{aligned} \tag{7}$$

where $*$ denotes a typical spatial convolution, $\theta_j^R \in D_6$ denotes the application of rotation, and $\theta_j^S \in D_6$ denotes the application of a reflection. After the convolution, we add the bias, $\beta_{\hat{c}}$, and apply a non-linearity σ,

$$\zeta_{\hat{c}}^{(1)}(x, g) = \sigma\left(y_{\hat{c}}^{(1)}(x, g) + \beta_{\hat{c}}\right) \tag{8}$$

where $g \in \{\theta_i^R, \theta_i^S\} \in D_6$. Layers after the first one are called group-convolutional, or "regular" layers, since the input now has both a spatial argument and rotation/reflection argument. This means that the filters will also have rotation/reflection arguments, $\Psi^{(l)}(x, g)$. For these layers, consider first the convolution where we restrict the filter only over rotations,

$$y_{\hat{c}}^{(l)}(x, \theta_i^R) = \sum_{c=1}^{C_{\text{in}}} \left(\zeta_c^{(l-1)} \circledast_R \Psi_{\hat{c}c}^{(l)}\right)(x, \theta_i^R) + \sum_{c=1}^{C_{\text{in}}} \left(\zeta_c^{(l-1)} \circledast_S \Psi_{\hat{c}c}^{(l)}\right)(x, \theta_i^R) \tag{9}$$

where,

$$\begin{aligned}\left(\zeta_c^{(l-1)} \circledast_R \Psi_{\hat{c}c}^{(l)}\right)(x,\theta_i^R) &= \sum_{\phi_j^R} \int_{\mathbb{R}^2} \zeta_c^{(l-1)}\left(u,\phi_j^R\right) \Psi_{\hat{c}c}^{(l)}\left((u,R_{\phi_j})^{-1}(x,\theta_i^R)\right) du \\ &= \sum_{\phi_j^R} \int_{\mathbb{R}^2} \zeta_c^{(l-1)}\left(u,\phi_j^R\right) \Psi_{\hat{c}c}^{(l)}\left(R_{\phi_j}^{-1}(x-u), R_{\phi_j}^{-1}\theta_i^R\right) du \\ &= \sum_{\phi_j^R} \int_{\mathbb{R}^2} \zeta_c^{(l-1)}\left(u,\phi_j^R\right) R_{\phi_j}\Psi_{\hat{c}c}^{(l)}\left(x-u,\theta_i^R-\phi_j^R\right) du \\ &= \sum_{\phi_j^R} \zeta_c^{(l-1)}\left(\cdot,\phi_j^R\right) * \mathcal{R}_{\phi_j}\Psi_{\hat{c}c}^{(l)}\left(\cdot,\theta_i^R\right) du\end{aligned} \tag{10}$$

Going from the first line to the second we have used the group multiplication rule, i.e., $(u, R_{\phi_j})^{-1}(x, \theta_i^R) = (R_{\phi_j}^{-1}(x - u), \theta_i^R - \phi_j^R)$. The third line follows from our notation for rotated filters, and in the fourth line, $*$ denotes spatial convolution and we introduce the notation,

$$\mathcal{R}_{\phi_j}\Psi(x,\theta_i^R) := R_{\phi_j}\Psi(x,\theta_i^R-\phi_j^R). \tag{11}$$

Following similar steps we can show that,

$$\left(\zeta_c^{(l-1)} \circledast_S \Psi_{\hat{c}c}^{(l)}\right)(x,\theta_i^R) = \sum_{\phi_j^S} \zeta_c^{(l-1)}\left(\cdot,\phi_j^S\right) * \mathcal{S}_{\phi_j}\Psi_{\hat{c}c}^{(l)}\left(\cdot,\theta_i^R\right), \tag{12}$$

where,

$$\mathcal{S}_{\phi_j}\Psi(x,\theta_i^R) := S_{\phi_j}\Psi(x,\phi_j^S-\theta_i^S). \tag{13}$$

Restricting the filter argument over reflections we get,

$$\begin{aligned}\left(\zeta_c^{(l-1)} \circledast_R \Psi_{\hat{c}c}^{(l)}\right)(x,\theta_i^S) &= \sum_{\phi_j^R} \zeta_c^{(l-1)}\left(\cdot,\phi_j^R\right) * \mathcal{R}_{\phi_j}\Psi_{\hat{c}c}^{(l)}\left(\cdot,\theta_i^S\right) \\ \left(\zeta_c^{(l-1)} \circledast_S \Psi_{\hat{c}c}^{(l)}\right)(x,\theta_i^S) &= \sum_{\phi_j^S} \zeta_c^{(l-1)}\left(\cdot,\phi_j^S\right) * \mathcal{S}_{\phi_j}\Psi_{\hat{c}c}^{(l)}\left(\cdot,\theta_i^S\right),\end{aligned} \tag{14}$$

where,

$$\mathcal{R}_{\phi_j}\Psi(x,\theta_i^S) := R_{\phi_j}\Psi(x,\theta_i^S-\phi_j^S), \tag{15}$$

$$\mathcal{S}_{\phi_j}\Psi(x,\theta_i^S) := S_{\phi_j}\Psi(x,\phi_j^R-\theta_i^R). \tag{16}$$

Note that in the Equation 9 we inevitably have two summations, one over the input channels and one over the group elements. To make the connection with the gauge equivariant convolution, Equation 1 we will stack these summations

into one channel dimension[5]. Then we have at point, p,

$$I_c : \mathbb{R}^2 \to \mathbb{R}^{C_{\text{in}}} \qquad \Psi^{(1)}_{\hat{c}c} : \mathbb{R}^2 \to \mathbb{R}^{12C_{\text{out}} \times C_{\text{in}}} \tag{17}$$

$$\zeta^{(l)}_c : \mathbb{R}^2 \to \mathbb{R}^{12C_{\text{in}}} \qquad \Psi^{(l)}_{\hat{c}c} : \mathbb{R}^2 \to \mathbb{R}^{12C_{\text{out}} \times 12C_{\text{in}}} \tag{18}$$

This stacking allows us to express the group convolution in the form of Equation 1 with $\rho_{\text{in}}(g_{p \leftarrow q_v}) \in \mathbb{R}^{12C_{\text{in}} \times 12C_{\text{in}}}$. In Figure 1 we see this stack of filters, where the first column shows how a filter, $\Psi^{(1)}$ is expanded for scalar inputs, i.e, rotations/reflections of the the filter. For regular layers, the treatment is different as now we also have an extra dimension θ, denoted by different colors in Figure 1. The rows show the action of operators $\mathcal{R}_{\phi_j}$ and $\mathcal{S}_{\phi_j}$. Notice how θ_R and θ_S permute within their blocks with actions of ϕ_R, but the blocks as a whole also permute with action of ϕ_S. This is because a rotation followed by a reflection produces a reflection, and two consecutive reflections produces a rotation.

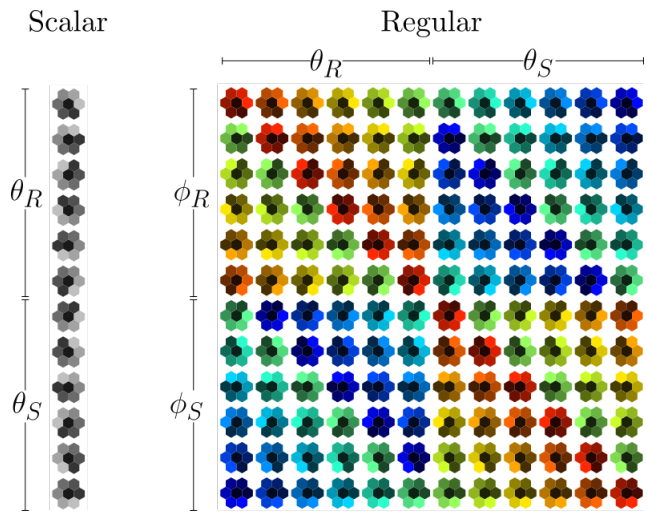

Figure 1: This is an illustration of the group operations on an input layer filter, i.e., scalar layer, and deeper (regular) layers. For the scalar layer we see a simple rotation and reflection of the hexagonal filter. On the other hand, for regular layers, we have an extra dimension, θ for previously applied group operations. This is depicted horizontally with the different colors. The vertical axis shows the effect of the operations $\mathcal{R}_\phi$ and $\mathcal{S}_\phi$. Note how a previously applied rotation followed by a reflection produces a reflection, and a previously applied reflection followed by a reflection produces a rotation.

2.4 The half icosahedron and padding

In dMRI the signal is assumed to have the following form [19],

$$S_k(\vec{x}) = S_0 e^{-b\vec{g}_k^T D(\vec{x}) \vec{g}_k}, \tag{19}$$

where S_0 is a constant, $D(\vec{x})$ is the diffusion tensor field, $\vec{g}_k$ is a unit vector, and b is the b-value. In a typical dMRI acquisition there will be many images, $k \in \{1, \ldots, N_{\text{directions}}\}$, such that the unit vectors $\vec{g}_k$ are spread evenly over a sphere. We can see from Equation 19 that the signal S_k has antipodal symmetry, i.e., it is invariant under the transformation $\vec{g}_k \to -\vec{g}_k$. Hence, we assume that the data from dMRI is a function of all unoriented lines from the

origin of $\mathbb{R}^3$, i.e., the real projective plane, $\mathbb{R}P^2$. A topological model of $\mathbb{R}P^2$ can be constructed by a sphere with antipodal sides points identified and then extracting the top hemisphere of such a sphere [10]. This top hemisphere can then be flattened in a manner analogous to the treatment in [5]. This procedure is illustrated in Figure 2.

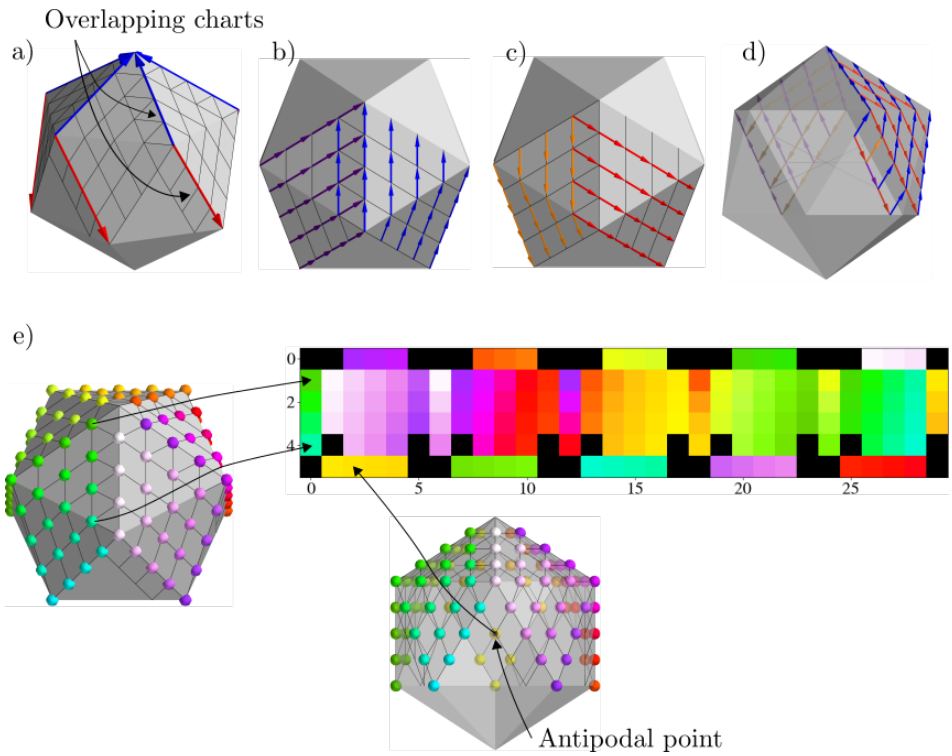

Figure 2: This figure illustrates how a signal and feature maps on the icosahedron are padded, and mapped to a rectangular grid. In a) we see the five square charts with the blue arrow showing the y-axis and the red arrow showing the x-axis. Also labelled in a) are the regions where the charts overlap, the bottom strip overlaps due to antipodal identification. In b) and c) we see how the unit vectors, blue and red, are related to the unit vectors purple and orange, respectively, of the chart on the left by counter-clockwise rotation of $60°$. In d) we see an example of how antipodal identification on the bottom strip leads to a reflection that swaps the x and y axes. In e) we see how a scalar signal is mapped to the rectangular grid. Note that since there is a strip added on the bottom and left of each 5×5 grid, the resulting grid is 6×30. The padding on these added strips is also illustrated in e).

Following the procedure of [5], a hexagonal grid on the half icosahedron $\mathcal{I}$ can be constructed from series of grid-refinement steps. Let $\mathcal{H}_0$ be the grid consisting of only the corners of $\mathcal{I}$. We may then take each triangular face and subdivide it into four smaller triangles which introduces three new points on the mid-points of the edges of the original larger triangles. This process can be repeated r times to obtain a refined grid, $\mathcal{H}_r$.

The half icosahedron, $\mathcal{I}$, has an atlas that consists of five overlapping charts. A chart is an invertible map $\varphi_i : U_i \to V_i$, where $U_i \subset \mathcal{H}_r \subset \mathcal{I}$ and $V_i \subset \mathbb{Z}^2$. For each chart we define an exterior $\overline{V}_i \subset V_i$, which consists of border pixels, and an interior $V_i^\circ \subset V_i$, consisting of all pixels that have neighbours in the same chart i. To ensure that each pixel (except for the 11 corners) in $\mathcal{H}_r$ is occupied by a chart, a strip is added added to the left and bottom of each chart as shown in Figure 2. Note in Figure 2a) that some points in the bottom strip are outside the half icosahedron, $\mathcal{I}$, but due to antipodal identification in our topological model they are identified with pixels on $\mathcal{I}$.

For a point $p \in U_i$ we choose a gauge, w_p^i, which is illustrated with red (x-axis) and blue (y-axis) in Figure 2a). Let $e_1 = (1, 0)$ and $e_2 = (0, 1)$ be the orthonormal basis vectors for $\mathbb{Z}^2$, then w_p^i is given by the Jacobian of $\varphi_i^{-1}(p)$, $\mathrm{d}\varphi_i^{-1}(p) : \mathbb{R}^2 \to T_p\mathcal{I}$. The basis vectors $w_p^i(e_1)$ and $w_p^i(e_2)$ make an angle of $2 \cdot 2\pi/6$ and are aligned with the grid.

Certain pixels, $p \in U_i \cap U_j$, are occupied by multiple charts, for example, in the edges of each chart, either due to sharing an edge or due to antipodal identification. Another example is points on the added strips which are clearly in an overlap of charts, or when some of these points are in multiple charts due to antipodal identification. Note that the basis vectors at such a point p may not coincide, i.e, $w_p^i(e_a)$ may not be the same as $w_p^j(e_a)$ for $a \in \{1, 2\}$. To switch from one chart to another we have to apply a gauge transformation $g_{ij}(p)$. In Figure 2b) and c) we see how the x-axis (red) and y-axis (blue) vectors of a chart compare with the those of a chart on its left. The gauge transformation in this case is a counter clockwise rotation of 60° on the left charts vectors to align them on the overlap. This gauge transformation is conveniently an element of the group D_6. Figure 2d) shows how vectors transform as we move to an antipodal point: the x-axis (orange) and y-axis (purple) swap. This gauge transformation, a reflection, is the S_{θ_2} element of the group D_6.

In accordance with these gauge transformations, we can pad the input and the activation maps between convolutions. Figure 2e) shows the padding for a scalar input. There are five square charts, each with dimension 5×5. Adding the padding strips on the left and bottom, we have dimension 6×6, and finally stacking them gives dimensions 6×30. The left strips are padded with neighbouring charts, and the bottom strips are padded with antipodal points. Note that after the first layer, activation maps will have dimensions $6 \times 30 \times 12$. In this case the padding is similar, but we have to pad from the appropriate θ dimension. For example, when padding the left strip of a particular chart in the θ_i dimension we take values from the $R\theta_i$ dimension, when padding the top we take values from the $R^{-1}\theta_i$, and when padding the bottom and right we take values form the $S_{\theta_2}\theta_i$ dimension. Note that we pad the right column of each of the five charts after each convolution since the neighbouring column to that is a padding strip for the next chart.

3 Methods

3.1 Data

We use dMRI data from the Human Connectome Project (HCP) Young Adult 3T study, WU-Minn S1200 release [23]. The dMRI acquisition is multi-shell with b-values of 0, 1000, 2000, and 3000 s/mm^2 with approximately an equal number of directions on each non-zero b-shell and an echo spacing of 0.78 ms. The resolution is 1.25 mm and the voxels are isotropic. We chose 15 random subjects for training and 25 random subjects for testing. We also calculate a brain mask that excludes the cerebrospinal fluid (CSF) using Freesurfer's `mri_binarize` with flags `--all-wm` and `--gm` (`https://surfer.nmr.mgh.harvard.edu`) [9, 7, 8]. In addition to the dMRI data we also utilize the structural T_1 and T_2 weighted images which are downsampled with cubic spline interpolation to match the resolution of the diffusion images at 1.25mm.

3.2 Network

The fundamental network architecture we follow is a residual network [15]. As shown in Figure 3, our network is similar to the work of [21] with some differences. The input of the network is $16 \times 16 \times 16$ blocks of T_1,T_2, S_0, and six images $S_1, \ldots, S_6$ corresponding to six gradient directions from the shell with b-value 1,000 s/mm^2. The directions are simply taken to be the first six in the acquisition; this is done to mimic an early stop of the scan session. This is possible since the HCP protocol made use of an incremental acquisition schemes, which are designed to make sure an aborted scan would result in a reasonably uniform coverage of the sampling domain [2]. S_0 is the first image taken without diffusion gradients applied. Given the high memory consumption, we limit our batch size to one. These blocks are then passed through three 3D convolutional layers with `ReLU` non-linearities, filters of size $3 \times 3 \times 3$ and 6×64 channels. This gives 64 channels for each of the six gradient directions. The output of the 3D convolutions is then projected to the icosahedron. This is done by using inverse distance weighting interpolation to sample points on the icosahedron from the six directions for each of the channels. This is followed by averaging antipodal points and then flattening to the top half of the icosahedron as shown in Figure 2. During this flattening, the 16^3 voxels are moved to the batch dimension with the chosen $h = 6$ and $w = 30$ parameters. This is then passed on to four layers of gauge equivariant convolutions with hexagonal filters [12], i.e., seven weights per channel, 64 channels, followed by group batch normalization layers and `ReLU` non-linearity. In the final gauge equivariant layer, we use an orientation pooling layer to collapse the rotation/reflection direction [25]. Next, the batch dimension is put back into the original 16^3 shape. Since this is a residual network to get the final output we have to add the inputs. This is done by taking the signal for the six directions, $S_1 \ldots, S_6$, and flattening it in the same way as the outputs of the 3d convolutions. Similar to the approach of [21], we train the model to predict the signal as computed with the tensor model from the fully-sampled data Equation 19, and not the actual acquired fully-sampled signal. We fit the parameters for this equation by using FSL's `dtifit` `https://fsl.fmrib.ox.ac.uk/` [13]. This gives us the diagonalized form of the diffusion tensor D, i.e, we get three eigenvectors V_1, V_2 and V_3 and their corresponding eigenvalues λ_1, λ_2 and λ_3. The metrics that are of particular interest are the first eigenvector, V_1 and fractional anisotropy (FA), which can be computed from the eigenvalues and is a measure of how anisotropic the diffusion process is. This approach of fitting to a modelled signal rather than the acquired signal results in more accurate V_1 and FA values after `dtifit` is performed on the predicted signal. We generate training labels in the following way, let X_{flat} be the projection of the signal at the six directions on to icosahedron followed by flattening to a rectangle as described above. Then we use the dMRI signal for 90 directions with b-value of 1000 s/mm^2, and perform `dtifit` on it. This allows us to directly compute the signal on the icosahedron mesh using Equation 19, letting this quantity be Y_{flat}. The final training label is given by $Y_{\text{flat}} - X_{\text{flat}}$.

Training was performed with 5, 10 and 15 subjects, using a learning rate of 1×10^{-4}, reduced by a factor of 0.5 on plateaus, number of epochs as 20 with a batch size of one due to memory constraints, and the loss function as mean squared error. Two V100 GPUs were used, for the maximum 15 subjects training lasted approximately two days. Predictions were performed on 25 test subjects. Once the outputs were computed, a single shell dMRI volume was created with a b-value 1000s/mm^2, with directions taken to be the nodes of the icosahedron mesh. Once we have this

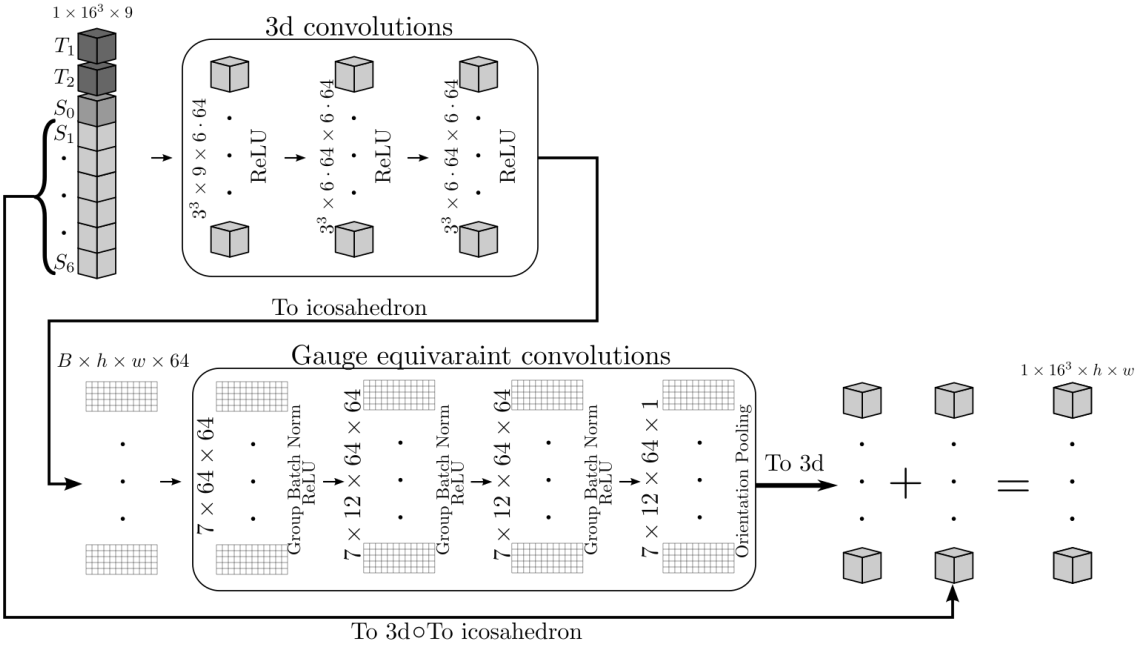

Figure 3: This figure shows the architecture we used. The input to the network is T_1, T_2, S_0 and S_i for $i \in \{1, \ldots, 6\}$ which are the volumes corresponding to six diffusion gradient directions which are simply taken to be the first six in the acquisition to predict an early stop. These nine volumes are broken into $16 \times 16 \times 16$ patches and passed through three layers of 3d convolutions, with each layer containing 6×64 filters with a kernel size of $3 \times 3 \times 3$, followed by ReLU activation functions. Then the six directions are projected onto the top icosahedron mesh which is then flattened. This gives us 64 channels with the $16 \times 16 \times 16$ voxels stacked into the batch dimension. This is then followed by four layers gauge equivariant convolutions with 64 filters each. The first layer has hexagonal filters containing seven weights followed by group convolutional layers with 7×12 weights, each layer is followed by group batch norm and a ReLU nonlinearity [5]. At the last layer we perform orientation pooling to collapse the rotation/reflection dimension [6]. Finally we put the batch dimension back into its 3d shape and add the inputs, S_i after projection to the icosahedron, giving us volumes with shape $16^3 \times h \times w$ where h and w are the height and width of the flattened icosahedron.

volume we perform `dtifit` to obtain FA and V_1. To evaluate the performance of the network in predicting FA and V_1, we compute the mean absolute difference (MAD) between ground truth values and predicted values where the mean is taken over all voxels within the mask that excluded CSF. This is also done for FA and V_1 values computed from the six direction volume without the enhancement from the network.

4 Results and discussion

Figure 4 shows the result of one of the test subjects out of the 25, where training is performed with 15 subjects. In the first row we have the ground truth FA values, as calculated with 90 gradient directions, followed by the FA as calculated from six directions and then the FA values calculated from the output of the network. In the second row we see the residual image, with respective MAD values denoted below, calculated on the mask that excludes CSF. We see that there is 76% reduction in the FA MAD value. In the third row we have the ground truth primary eigenvector, V_1, followed by the eigenvector as calculated from six directions, and then the eigenvector as calculated from the output of the network. In these images the direction of the vector is encoded by color as is typical, where red represents left to right, green

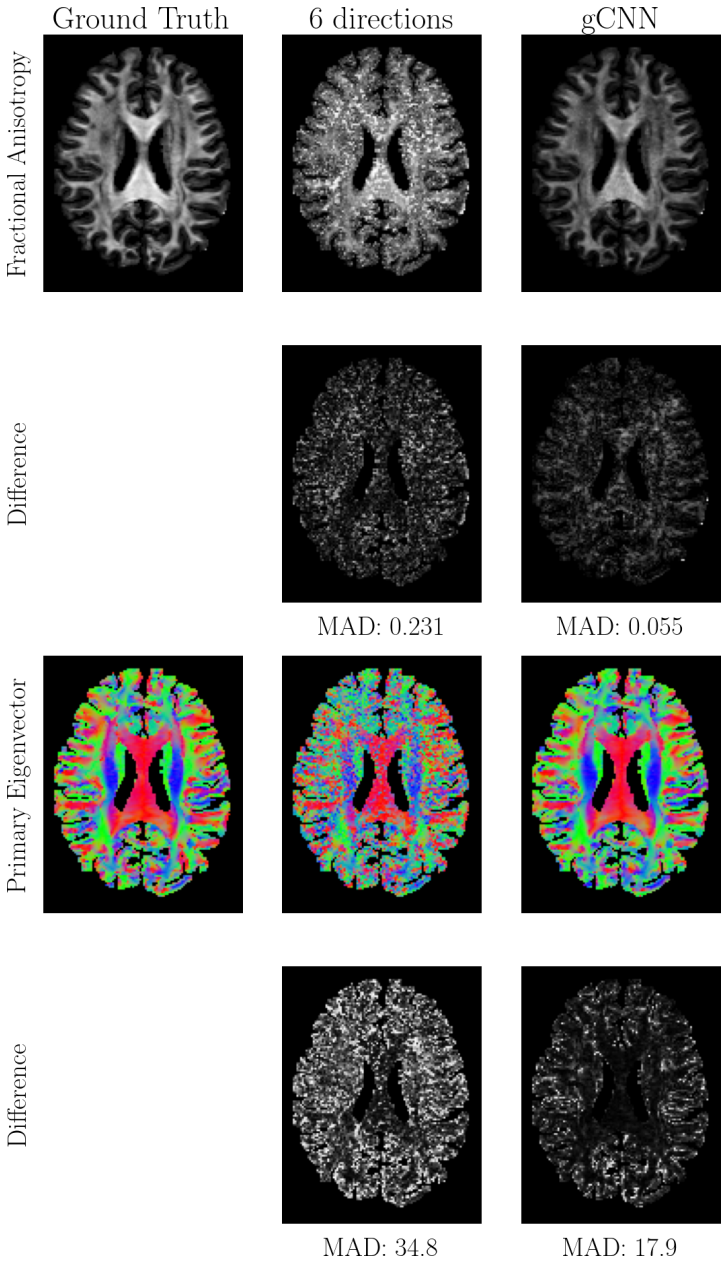

Figure 4: This figure summarizes the results from the network for one subject. The first row shows the FA values, with the first panel showing the ground truth, second showing FA as computed from 6 directions and the last showing FA computed from the output of the network. The second row shows the difference with ground truth with the MAD values computed from the CSF exclusive mask shown below each panel. The third and fourth row are similar but for the primary eigenvector V_1. The colors encode the direction of the vectors, red represents left to right, green anterior to posterior and blue superior to inferior.

anterior to posterior and blue superior to inferior. In the fourth row we show the angle difference between ground truth values and the respective image, also denoted the MAD values below, as calculated on the mask that excludes CSF. We see that there is a reduction of 49% in the V_1 MAD value. In Figure 5 we see the effect that training with 5, 10 and 15 subjects (x-axis) has on the MAD values (y-axis) for FA (Figure 5 a)) and V_1 (Figure 5 b)). Each point is the mean of the MAD values averaged over the CSF exclusive mask and the error bars show the standard deviation. Our work is

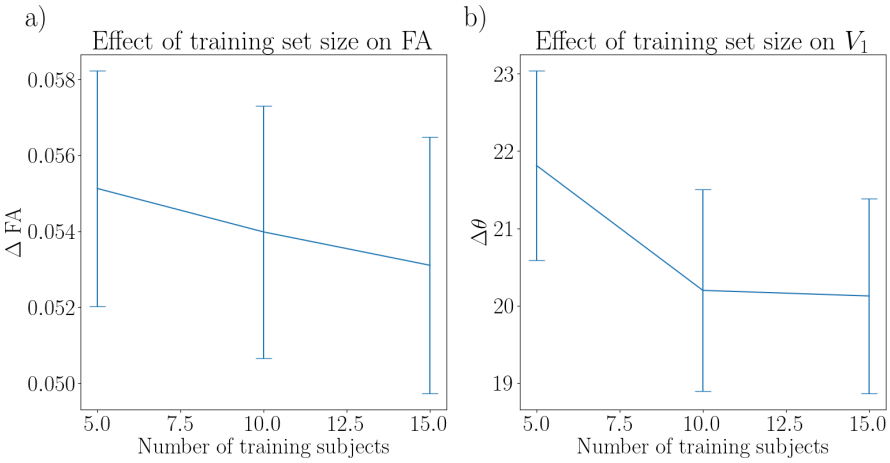

Figure 5: This figure shows the effect of training sample size on FA and V_1 when training with 5, 10, 15 subjects. Each point shows the mean MAD value where the mean is computed over subjects, the error bars show the standard deviation.

inspired by DeepDTI [21] and uses a similar residual architecture [15], but offers certain enhancements. A particularly difficult problem to tackle with convolutions on the dMRI signal is the non-euclidean manifold underlying the data, since the signal is acquired on a sphere. Additionally, given the antipodal symmetry in Equation 19 the manifold of interest is the real projective plane, $\mathbb{R}P^2$, which is also non-orientable [10]. Our approach completely alleviates this problem, while providing additional benefits. We are not limited by input data acquired in particular directions as DeepDTI, but are able to process any six directions provided they are evenly spaced on the sphere. This is useful for acquisitions where the scan stops early and only a limited number of directions are acquired, we have simulated this by using the first six directions in the HCP acquisition sequence. Further we are able to reach similar performance to DeepDTI but with only 15 subjects compared to 40 subjects. This training efficiency arise as a result of the rotation and reflection of convolution filters, since new rotated and reflected versions of existing filters do not have to be learned during the training. Note that although the final MAD value for the primary vector, V_1 is lower with DeepDTI [21], the initial raw (no network enhancement) V_1 MAD value is also lower due to the special choice of six diffusion gradient directions. When we compare the subject shown in Fig. 5 of [21] we see a 46% reduction in MAD compared to our Figure 4, where we see a 49% reduction in MAD. Our approach is also general, as the gauge equivariant layers can be added to any architecture that aims to solve a task that pertains to dMRI, for example, kurtosis, tractography, neurite orientation dispersion and density imaging (NODDI) [26], etc.

5 Conclusion

In this work we have introduced gCNN layers for the real projective plane, $\mathbb{R}P^2$, which are applied to the problem of denoising DTI metrics as derived from using only six diffusion gradient directions. By modelling $\mathbb{R}P^2$ as the top half of an icosahedron and using the full dihedral group (including reflections) in the group convolutions, we are able to generalize the denoising architecture of [21] to use any six diffusion gradient directions while maintaining the same level of denoising. Further, since rotated and reflected filters are given as priors and are not learned by the network, the

model is data efficient and requires significantly fewer subjects to train. Finally, our gCNN layers can be utilized in any architecture involving the dMRI signal, and are straightforward to implement given our openly available code.

6 Acknowledgements

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. During the course of this study UH was supported by a postdoctoral fellowship award from the Canadian Open Neuroscience Platform (CONP) and AK was supported by the Canada Research Chairs program #950-231964, NSERC Discovery Grant #6639, and Canada Foundation for Innovation (CFI) John R. Evans Leaders Fund Project #37427, the Canada First Research Excellence Fund, and Brain Canada. This research was enabled in part by the support provided by Compute Ontario (www.computeontario.ca) and Compute Canada (www.computecanada.ca)

7 Code Availability

The open source code for this can be found at `https://github.com/akhanf/gcnn_dmri`

8 Declaration of Competing Interest

The authors do not have any conflicts of interest (financial or otherwise) to disclose.

References

[1] Andrew L Alexander et al. "Diffusion tensor imaging of the brain". In: *Neurotherapeutics* 4.3 (2007), pp. 316–329.

[2] Emmanuel Caruyer et al. "Design of multishell sampling schemes with uniform coverage in diffusion MRI". In: *Magnetic Resonance in Medicine* 69.6 (2013), pp. 1534–1540.

[3] Hu Cheng et al. "Denoising diffusion weighted imaging data using convolutional neural networks". In: *bioRxiv* (2022).

[4] Taco Cohen and Max Welling. "Group equivariant convolutional networks". In: *International conference on machine learning*. PMLR. 2016, pp. 2990–2999.

[5] Taco Cohen et al. "Gauge equivariant convolutional networks and the icosahedral CNN". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 1321–1330.

[6] Taco S Cohen and Max Welling. "Steerable cnns". In: *arXiv preprint arXiv:1612.08498* (2016).

[7] Anders Dale, Bruce Fischl, and Martin I. Sereno. "Cortical Surface-Based Analysis: I. Segmentation and Surface Reconstruction". In: *NeuroImage* 9.2 (1999), pp. 179–194.

[8] Bruce Fischl. "FreeSurfer". In: *Neuroimage* 62.2 (2012), pp. 774–781.

[9] Bruce Fischl, Martin I. Sereno, and Anders Dale. "Cortical Surface-Based Analysis: II: Inflation, Flattening, and a Surface-Based Coordinate System". In: *NeuroImage* 9.2 (1999), pp. 195–207.

[10] Theodore Frankel. *The geometry of physics: an introduction*. Cambridge university press, 2011.

[11] Vladimir Golkov et al. "Q-space deep learning: twelve-fold shorter and model-free diffusion MRI scans". In: *IEEE transactions on medical imaging* 35.5 (2016), pp. 1344–1351.

[12] Emiel Hoogeboom et al. "Hexaconv". In: *arXiv preprint arXiv:1803.02108* (2018).

[13] Mark Jenkinson et al. "Fsl". In: *Neuroimage* 62.2 (2012), pp. 782–790.

[14] Davood Karimi and Ali Gholipour. "Diffusion tensor estimation with transformer neural networks". In: *Artificial Intelligence in Medicine* (2022), p. 102330.

[15] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. "Accurate image super-resolution using very deep convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1646–1654.

[16] Catherine Lebel, Sarah Treit, and Christian Beaulieu. "A review of diffusion MRI of typical white matter development from early childhood to young adulthood". In: *NMR in Biomedicine* 32.4 (2019), e3778.

[17] John M Lee. *Introduction to Riemannian manifolds*. Springer, 2018.

[18] Philip Müller et al. "Rotation-equivariant deep learning for diffusion MRI". In: *arXiv preprint arXiv:2102.06942* (2021).

[19] Lauren J O'Donnell and Carl-Fredrik Westin. "An introduction to diffusion tensor image analysis". In: *Neurosurgery Clinics* 22.2 (2011), pp. 185–196.

[20] José M Soares et al. "A hitchhiker's guide to diffusion tensor imaging". In: *Frontiers in neuroscience* 7 (2013), p. 31.

[21] Qiyuan Tian et al. "DeepDTI: High-fidelity six-direction diffusion tensor imaging using deep learning". In: *NeuroImage* 219 (2020), p. 117017.

[22] Wu-Ki Tung. *Group theory in physics*. Vol. 1. World Scientific, 1985.

[23] David C Van Essen et al. "The Human Connectome Project: a data acquisition perspective". In: *Neuroimage* 62.4 (2012), pp. 2222–2231.

[24] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[25] Maurice Weiler, Fred A Hamprecht, and Martin Storath. "Learning steerable filters for rotation equivariant cnns". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 849–858.

[26] Hui Zhang et al. "NODDI: practical in vivo neurite orientation dispersion and density imaging of the human brain". In: *Neuroimage* 61.4 (2012), pp. 1000–1016.