A simulation-driven supervised learning framework to estimate brain microstructure using diffusion MRI

Chengran Fang[a,b], Zheyi Yang[a], Demian Wassermann[b], Jing-Rebecca Li[a,*]

[a]*INRIA Saclay, Equipe IDEFIX, UMA, ENSTA Paris, 828, Boulevard des Maréchaux, 91762 Palaiseau, France*

[b]*INRIA Saclay, Equipe MIND, 1 Rue Honoré d'Estienne d'Orves, 91120 Palaiseau, France*

Abstract

We propose a framework to train supervised learning models on synthetic data to estimate brain microstructure parameters using diffusion magnetic resonance imaging (dMRI). Although further validation is necessary, the proposed framework aims to seamlessly incorporate realistic simulations into dMRI microstructure estimation.

Synthetic data were generated from over 1,000 neuron meshes converted from digital neuronal reconstructions and linked to their neuroanatomical parameters (such as soma volume and neurite length) using an optimized diffusion MRI simulator that produces intracellular dMRI signals from the solution of the Bloch-Torrey partial differential equation. By combining random subsets of simulated neuron signals with a free diffusion compartment signal, we constructed a synthetic dataset containing dMRI signals and 40 tissue microstructure parameters of 1.45 million artificial brain voxels.

To implement supervised learning models we chose multilayer perceptrons (MLPs) and trained them ona subset of the synthetic dataset to estimate some microstructure parameters, namely, the volume fractions of soma, neurites, and the free diffusion compartment, as well as the area fractions of soma and neurites. The trained MLPs perform satisfactorily on the synthetic test sets and give promising *in-vivo* parameter maps on the *MGH Connectome Diffusion Microstructure Dataset* (CDMD). Most importantly, the estimated volume fractions showed low dependence on the diffusion time, the diffusion time independence of the estimated parameters being a desired property of quantitative microstructure imaging.

The synthetic dataset we generated will be valuable for the validation of models that map between the dMRI signals and microstructure parameters. The surface meshes and microstructures parameters of the aforementioned neurons have been made publically available.

Keywords: Bloch-Torrey equation, diffusion magnetic resonance imaging, dMRI, finite elements, matrix formalism, machine learning, supervised learning, neuron modeling.

[*] Corresponding author
Email address: jingrebecca.li@inria.fr

1. Introduction

MRI scanners can encode the diffusion of water protons into dMRI signals by applying a time-varying magnetic field gradient to brain voxels at the resolution of around 1 mm^3 [1–3]. Because the movement of water protons is restricted by, for example, cell membranes [4], brain voxels with varying microstructures generate different signals [5, 6]. Due to the sensitivity of dMRI to structures below the spatial resolution, non-invasive estimation of brain microstructure parameters, such as axon radii [7–9] and neurite density [10, 11], is an increasingly important area of research in the community. However, *in-vivo* estimation is challenging due to the complex microstructure composition [12–14] and the intricate signal generation mechanism of dMRI [15–18].

Brain microstructure imaging often relies on "inverting" a forward model explaining the dMRI signal generation. Therefore, the accuracy of the forward model is of essential importance. The predominant forward models, i.e., biophysical models, typically subdivide a brain voxel into compartments described by simplified geometries such as cylinders with zero radii (sticks) [11, 19] and spheres (balls) [20]. Together with some additional assumptions, especially the Gaussian phase assumption (GPA), a biophysical model allows deriving an analytical signal expression as a function of the model parameters related to several microstructure parameters [21].

One often fits the signal expression to experimental data to estimate the model parameters. However, the indeterminacy inherent in some biophysical models makes the parameter estimation unstable [22]. Moreover, an accurate fit does not necessarily justify the underlying biophysical model, and the estimated model parameters might be biophysically meaningless [23, 24]. Subtle effects like neurite undulation are excluded from biophysical models because of mathematical complications [11, 20, 25]. In addition to the error brought by the simplified geometric models, the validity of some assumptions, such as the GPA, remains unknown [23, 26]. Besides, the validity regimes of several signal expressions depend on microstructure length scales [27]. A voxel may exhibit multiple length scales (e.g., various soma radii) so that different validity regimes may co-exist or emerge progressively [24], making comprehensive model validation difficult.

To address the above shortcomings and achieve a more accurate forward model, we aim to replace the simplified geometries with realistic neuron meshes and the analytical intracellular signal expressions with diffusion MRI simulations. Indeed, the numerical dMRI simulation methods, including both algorithms based on solving the Bloch-Torrey partial differential equation (BT equation) [28–34] and Monte-Carlo methods [35–42], are the "ground-truth" forward model for dMRI signal generation[43]. With realistic neuron meshes, numerical simulation can seamlessly incorporate effects arising from, for instance, neurite undulation or water exchange between soma and neurites.

Although various simulation methods have been proposed for a long time, their use is still confined to selectively verifying the validity of some biophysical models [26, 44, 45]. Three main factors limit the widespread use of dMRI simulation. First, dMRI simulators were not efficient enough to perform large number of computations on realistic neuron meshes [46]. Second, automatically building a variety of numerical phantoms is challenging. Few open-source modeling software is available to the community [46–49]. Third, dMRI simulators mostly behave like black boxes due to the lack of insight into the dMRI signal generation mechanism, i.e., the Bloch-Torrey equation [16–18]. The relationships between the numerical phantoms and the simulated dMRI signals are implicit, non-parametric, and possibly high-dimensional [46], making "inverting" the ground-truth forward model challenging. Therefore, the biophysical models with explicit signal expressions, which give plausible results at the compromise of accuracy, dominate the research about brain microstructure estimation using diffusion MRI.

In contrast, we proposed a new framework that uses an ultra-fast dMRI simulator and an automatic neuron mesh generator to create a large synthetic dataset to feed into machine learning (ML) algorithms to overcome the above limitations and directly leverage dMRI simulation as the forward model.

For the dMRI simulations, we adopt the numerical matrix formalism (NMF) method [50–53] based on a finite-element (FE) discretization and implemented in the SpinDoctor Toolbox [34, 46]. Integrating matrix formalism with a finite element method (FEM) brings significant advantages in terms of computational efficiency. We optimized the NMF method by speeding up the eigendecomposition algorithm and leveraging GPU computations. A ten-fold speedup is achieved, making large-scale dMRI simulation practical.

To create realistic neurons, we implemented a neuron mesh generator that constructs computational meshes based on digital neuronal reconstructions archived in NeuroMorpho.Org [54]. By harnessing three well-established methods in computer graphics [55–60], the generator is automatic and robust and produces high-quality neuron meshes while keeping the meshes reasonably sized. We obtained 1213 realistic neuron meshes and randomly combine them with a free diffusion compartment to mimic an artificial brain voxel. In addition, the neuron meshes allow us to simulate intracellular signals and measure neuroanatomical parameters on them. A variety of microstructure parameters computed from the neuroanatomical measurements could characterize artificial brain voxels. However, the mappings from dMRI signals to microstructure parameters are implicit and possibly high-dimensional.

Finally, to "invert" the ground-truth forward model, we leverage machine learning techniques. The adoption of ML algorithms in dMRI dates back to the last century [61] and has seen a recent resurgence [62–66]. Artificial neural networks are believed to be superior for function approximation [67–69], especially in high dimensions [70, 71]. In this paper we leverage multilayer perceptrons (MLPs) to approximate the underlying mappings. Specifically, we build a synthetic dataset containing both the dMRI signals and more than 40 microstructure parameters of over 1.45 million artificial brain voxels. MLPs are trained on a subset of the synthetic dataset in a supervised way. We demonstrate eight exemplary MLPs for volume and area fraction estimation on synthetic test datasets and the MGH Connectome Diffusion Microstructure Dataset (CDMD)[72]. Finally, the MLPs are compared with the state-of-the-art impermeable biophysical model, SANDI [20].

2. Theory

Suppose one would like to simulate the diffusion MRI signal due to spin-bearing particles inside a medium. A domain $\Omega = \cup_{i=1}^{N} \Omega_i$ describes the medium that comprises N non-overlapping compartments. Ω_i denotes the i-th compartment which is a connected subset of $\mathbb{R}^n$ (n is 2 or 3), and $\partial\Omega_i$ its boundary. In this paper, we will neglect the exchange between compartments. A domain Ω represents a 3-dimensional artificial brain voxel of $N-1$ intracellular compartments and a free diffusion compartment.

2.1. Bloch-Torrey partial differential equation

In diffusion MRI, a time-varying magnetic field gradient is applied to encode the diffusive motion of water protons. Denoting the effective diffusion-encoding magnetic field gradient by $\boldsymbol{G}(t)$, the complex transverse water proton magnetization in the rotating frame satisfies the Bloch-Torrey partial differential equation (BT equation) [43]:

$$\frac{\partial}{\partial t} M_i(\boldsymbol{x}, t) = \left(D_i \nabla \cdot \nabla - \underline{\jmath} \gamma \boldsymbol{x} \cdot \boldsymbol{G}(t) - \frac{1}{T_i} \right) M_i(\boldsymbol{x}, t), \ \boldsymbol{x} \in \Omega_i, \ t \in [0, TE], \tag{1}$$

where D_i and T_i are respectively the self-diffusion coefficient and transverse relaxation time of water protons in the compartment Ω_i ($i \in \{1, ..., N\}$), $\gamma = 0.26752 \ rad/(\mu s \cdot mT)$ is the gyromagnetic ratio of the water proton [73], TE is the echo time, and $\underline{\jmath}$ is the imaginary unit. Magnetization, the density of proton transverse magnetic moment, is a function of position $\boldsymbol{x}$ and time t. M_i denotes the magnetization in Ω_i.

Some commonly used magnetic field gradient sequences (diffusion-encoding sequences) include the pulsed-gradient spin echo (PGSE) sequences [2], the double pulsed-gradient spin echo (double-PGSE) sequences [74, 75], and the oscillating gradient spin echo (OGSE) sequences [76, 77]. This study utilizes the PGSE sequence whose magnetic field gradient is the product of a constant gradient $\boldsymbol{g} \in \mathbb{R}^3$ and a time profile $f : \mathbb{R} \to [-1, 1]$. The parameters of the PGSE sequence are the pulse duration δ, the inter-pulse duration Δ, the gradient intensity $g = \|\boldsymbol{g}\|$, and the direction of the magnetic field gradient $\boldsymbol{u}_g = \boldsymbol{g}/g$.

The initial magnetization is assumed to be equilibrial in the brain voxel:

$$M_i(\boldsymbol{x}, 0) = \rho, \ \boldsymbol{x} \in \Omega_i, \ i \in \{1, ..., N\}, \tag{2}$$

where ρ is the initial magnetization.

The outer boundaries are subject to impermeable boundary conditions:

$$D_i \nabla M_i(\boldsymbol{x}, t) \cdot \boldsymbol{n}_i(\boldsymbol{x}) = 0, \ \boldsymbol{x} \in \Gamma_i, \ i \in \{1, ..., N\}, \tag{3}$$

where $\boldsymbol{n}_i$ is the unit outward pointing normal vector of $\partial \Omega_i$.

Equations (1) to (3) govern the time evolution of magnetization in all compartments. Once we solved the BT equation, the dMRI signal S from the voxel Ω measured at TE is the spatial integral of the magnetization:

$$S(g, \boldsymbol{u}_g, \delta, \Delta) = \sum_{i=1}^{N} \int_{\boldsymbol{x} \in \Omega_i} M_i(\boldsymbol{x}, TE) d\boldsymbol{x}. \tag{4}$$

In practice, the dMRI signal is normalized by the signal in the absence of gradient to get the signal attenuation

$$E(g, \boldsymbol{u}_g, \delta, \Delta) = \frac{S(g, \boldsymbol{u}_g, \delta, \Delta)}{S(g = 0)}. \tag{5}$$

We denote $S(g = 0)$ by S_0. The normalization mainly leaves the attenuation due to diffusion and eliminates the effects of transverse relaxation [6, p. 6].

One usually interprets the signal attenuation E against two quantities: diffusion time t_d and b-value (or simply b). Even though diffusion time is an important parameter, it doesn't have a unique definition. Various diffusion time definitions exist in literature for PGSE sequence, such as $\Delta - \delta/3$ [78], Δ [6, 79], and $\Delta + \delta$ [80]. These definitions converge to Δ in the limit of $\frac{\delta}{\Delta} \to 0$, but it is still not clear if there is a well-defined diffusion time for long gradient pulses (i.e., $\delta \sim \Delta$) [15]. We adopt the widely used definition $t_d = \Delta - \delta/3$ and only take it as a rough estimation of the diffusion time.

For PGSE sequence, the b-value is defined as $b = \gamma^2 g^2 \delta^2 (\Delta - \delta/3)$. In free diffusion or the case when the GPA is applicable, the signal attenuation is or can be approximated by $e^{-D_e \cdot b}$ where D_e is an effective diffusion coefficient or apparent diffusion coefficient (ADC).

Assumed that the transverse relaxation time is independent of the spatial position, denoted by T. Finally, the signal attenuation of a voxel is

$$E(g, \boldsymbol{u}_g, \delta, \Delta) = \frac{\sum \int_{\Omega_i} \rho e^{-\frac{TE}{T}} \varphi_i(\boldsymbol{x}, TE) d\boldsymbol{x}}{\sum \rho e^{-\frac{TE}{T}} V_i} = \frac{\sum V_i E_i}{\sum V_i} = \sum f_i E_i, \tag{6}$$

where $f_i = V_i / \sum V_j$ is the volume fraction of $\Omega_i$and $\varphi_i(\boldsymbol{x}, t)$ is the unrelaxed magnetization subject to unit initial conditions.

Given the two assumptions, eq. (6) reveals that the voxel signal attenuation is the volume fraction weighted sum of the compartment signal attenuations. Since compartments are decoupled, we can now focus on solving eqs. (1) to (3) for a single compartment. We numerically solve the Bloch-Torrey equation system by using the matrix formalism representation in a finite element (FE) basis. For details, we refer the reader to Appendix Section 7.1.

3. Method

This section describes the proposed supervised learning framework for estimating brain microstructure using diffusion MRI. Specifically, we aim to approximate the underlying mappings (if they exist) from the dMRI signals to some target brain microstructure parameters, such as soma and neurite volume fractions, using machine learning models such as multilayer perceptrons (MLPs) [81, 82].

3.1. The Neuron Data Set

A large amount of high-quality data is crucial for the performance of supervised learning models. Since real-world ground-truth brain microstructure measurements with the corresponding dMRI signals are scarce, we aim to construct a synthetic dataset generated by diffusion MRI simulations on realistic neuron meshes.

First, we built a neuron mesh database comprising 1,163 neurons and 50 glia via an automatic and robust neuron mesh generator developed in house, where we made use of the digitally reconstructed real human neurons collected in NeuroMorpho.Org [54]. The details about this pipeline will appear in a separate publication. Note that the mesh database also includes 50 non-neuronal cells. For ease of notation, we refer to both neurons and glia as neurons in this paper because they share structural similarities.In the end, we obtained 1213 realistic neuronal meshes. Each mesh corresponds to a human cell extracted from a particular brain region. Figure 1 shows the regional distribution of the cells included in the mesh database. These cells come from a wide range of brain areas.

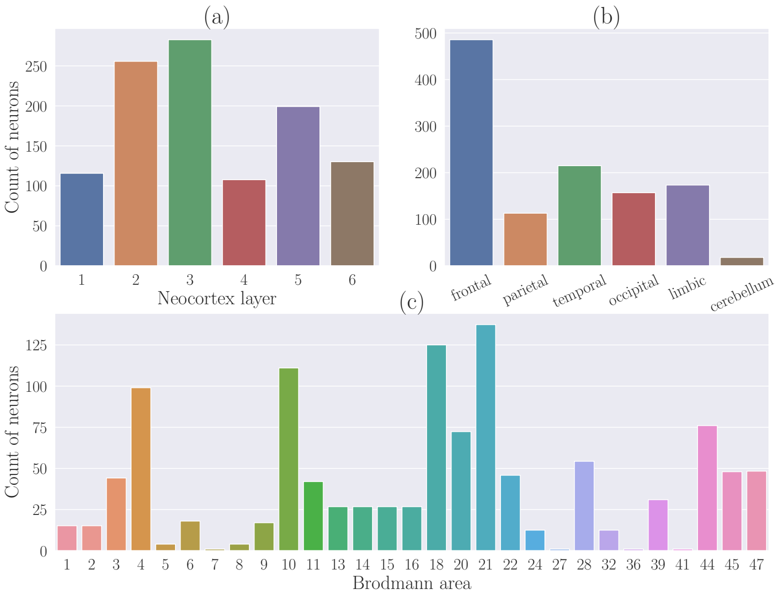
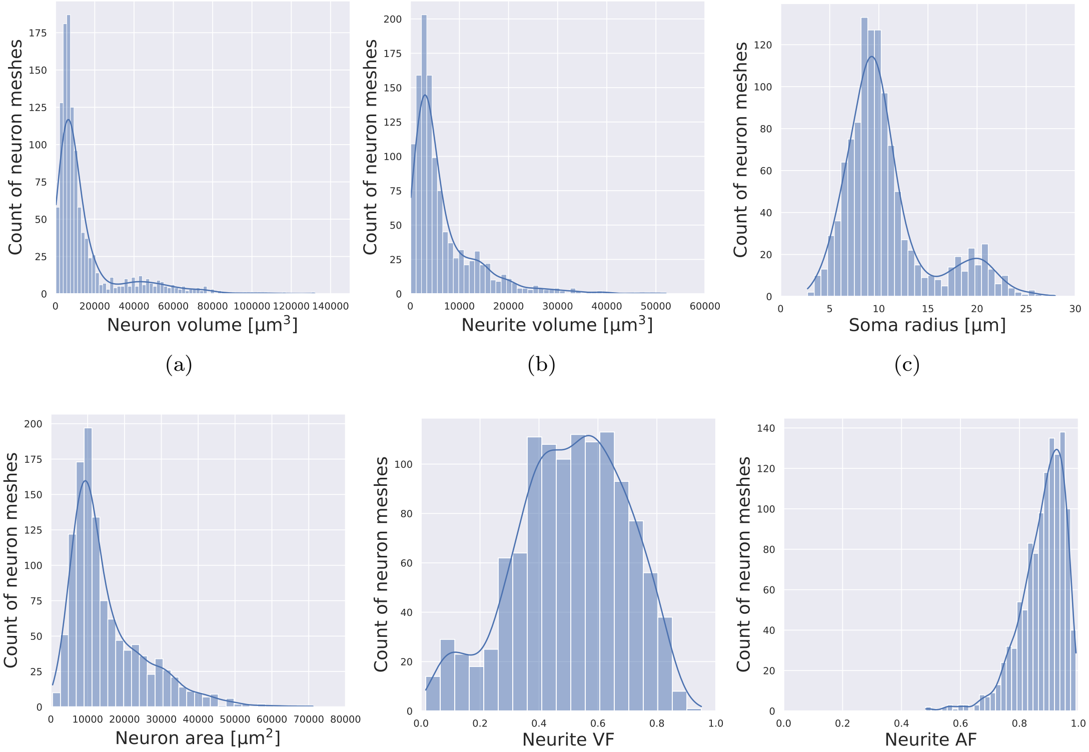


Figure 1: Cell distribution in brain regions. (a) distribution in neocortex layers. (b) distribution in brain lobes. (c) distribution in Brodmann areas.

The cell morphological parameters are another critical part of the synthetic dataset because they are the ground truth for model training. The neuron meshes and skeletons enable the measurements of various neuroanatomical parameters. L-measure[1] is widely used to measure neuron morphology based on skeletons [83]. However, L-measure cannot accurately measure the neurite area and volume because it treats neurite segments as cylinders. For example, the neurite area given by L-measure is the sum of the lateral area of many cylinders. This simplified treatment omits the spatial variations in the neurite radius and the complex structure in the bifurcation regions. We remedy the defect by measuring the surface areas and volumes of the neuron meshes. We got over 40 neuroanatomical parameters of biophysical interest based on the neuron meshes and skeletons. With a watertight neuron surface mesh, the area A_{neuron} and volume V_{neuron} can be efficiently computed [84]. Besides, the soma area is $A_{\text{soma}} \simeq 4\pi r^2$ and volume is $V_{\text{soma}} \simeq \frac{4}{3}\pi r^3$. The area of neurites is $A_{\text{neurite}} = A_{\text{neuron}} - A_{\text{soma}}$ and volume is $V_{\text{neurite}} = V_{\text{neuron}} - V_{\text{soma}}$. Figure 2 shows the distributions of six different morphological parameters of the 1213 neurons in the Neuron Data Set.

In addition to being realistic, all neuron surface meshes are watertight and simulation-ready. Furthermore, the mesh quality, a commonly disregarded factor affecting the simulation precision, is worth considering. We refer to a triangle as low-quality or “bad” if its aspect ratio (two times the ratio between the inradius and circumradius) is inferior to 1/3. For each triangulated surface mesh, we quantify the mesh quality using the proportion of bad triangles, i.e., the number of bad triangles divided by the total number of triangles. Figure 3 shows distributions of surface meshes information of the Neuron Data Set. Seventy-five percent of neuron meshes have a

[1] http://cng.gmu.edu:8080/Lm/



(a)

(b)

(c)

(d)

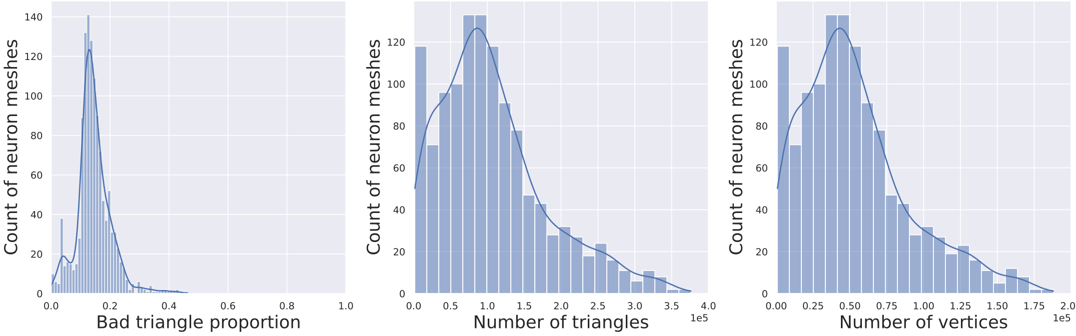
(e)

(f)

Figure 2: Distributions of morphological parameters of the Neuron Data Set which contains 1213 cells.

bad triangle proportion below 0.2, meaning that most meshes generated by our pipeline are high-quality. While maintaining the mesh quality, we reduce the mesh size as much as possible. The maximum number of vertices is 200,000, which is within the capability of our dMRI simulator. The surface meshes and microstructures parameters of 1,163 neurons and 50 glia are available at `https://github.com/SpinDoctorMRI`.

To perform diffusion MRI simulations, we used Tetgen [85], an external finite elements mesh generator, to robustly convert watertight surface meshes to tetrahedral volume meshes. Simulating dMRI signals on a realistic neuron mesh is a computational challenge [46]. This study required simulations on 1213 neuron meshes, which on average has around $80,000$ vertices and $150,000$ faces. To meet such a huge computing demand, we leverage the numerical matrix formalism based on finite element discretization implemented in the SpinDoctor toolbox [34]. The theory of this method is explained in section 2, and the convergence properties have been studied by Li *et al.* [52, 53]. Nonetheless, the previous implementation is not fast enough. The numerical matrix formalism has two time-consuming operations, namely, matrix eigendecomposition and matrix exponential (eq. (26)). For this study, we accelerated the matrix eigen-decomposition 20-fold by implementing a shift-and-invert transformation [86]. We also accelerated the matrix exponential and the matrix-vector product in Equation (26) 10-fold by implementing the algorithm in Al-Mohy *et al.*



(a) (b) (c)

Figure 3: Distributions of surface meshes information of the Neuron Data Set which contains 1213 cells.

[87] which computes the action of a matrix exponential on a vector without explicitly evaluating the matrix exponential on GPUs. The generation of 1213 neuron meshes and the dMRI simulations took less than one month in total. It is worth emphasizing that the mesh generation and the eigendecomposition only need to be performed once. If we want to change the magnetic field gradient $\boldsymbol{G}$ or the diffusivity, the eigenfunctions can be re-used.

In the Appendix Section 7.2 we compare the numerical matrix formalism with the direct finite element method (FEM) solution of the Bloch-Torrey equation to validate the choice of two simulation parameters: the spatial discretization and the Laplace eigenvalue spectrum truncation, then we show the computational efficiency of the new numerical matrix formalism implementation on GPUs.

3.2. The Synthetic dataset

We aim to construct a synthetic dataset comprising dMRI signals and microstructure parameters of artificial brain voxels. In practice, a gray matter voxel of 1 mm^3 is a medium comprising tens of thousands of cell bodies, millions of neurites, blood vessels, extracellular space (ECS), etc. [14]. To model such a complex tissue, we make various simplifications to the braingray matter voxels. First, we ignore compartments like blood vessels because cells and the ECS occupy most of the volume. Ideally, we could wrap the neuron meshes with another mesh to mimic ECS. However, neurons densely intertwine in the brain. One should carefully pack the neuron meshes to obtain a reasonable ECS volume fraction ($< 20\%$) [12]. Neuron packing is an extremely challenging problem that is far from being solved. Therefore, in this study, we do not include a ECS compartment in the dMRI signal model. We simply add a free diffusion compartment signal to the intra-cellular signal. Second, we assume that the averaged signals from hundreds of archetypical neurons from the above described Neuron Data Set are enough to represent the intracellular signals of an actual brain voxel.

We start with the simulation on a single neuron mesh using a similar dMRI protocol as MGH CDMD:

- two PGSE sequences with $\delta/\Delta = 8/19$ or $8/49$ ms;

- 64 gradient intensities linearly space between 0 and 290 mT/m;
- 32 diffusion encoding directions uniformly distributed on a hemisphere (equivalent to 64 directions on a sphere because simulated signals are antipodally symmetric).

The diffusivity inside neuron is fixed to be $3 \times 10^{-3}\,\mathrm{mm}^2/\mathrm{s}$, the water self-diffusion coefficient at 37 °C [88]. In addition, the numerical matrix formalism requires truncation of the Laplace eigenvalue spectrum. We retain the eigenvalues less than 4.39 $\mu\mathrm{m}^{-2}$, which correspond to characteristic time scales greater than 76 μs and characteristic length scales greater than 1.5 μm. The diffusion MRI signals are normalized by S_0 to get the signal attenuation. The above simulations are performed for each of the 1213 neurons in the Neuron Data Set. The direction-averaged simulated signals for the neurons in the Neuron Data Set are available at `https://github.com/SpinDoctorMRI`.

3.2.1. Voxel signal model

Our gray matter voxel model consists of numerous neurons and a free diffusion compartment. Once we have obtained the signals from every neuron mesh, we can compute the signal attenuation from a brain voxel by adding the neurons signals and the free diffusion compartment signals.

Suppose an artificial brain voxel contains M neurons and a free diffusion compartment whose volume fraction is f_{free}. According to eq. (6), the signal attenuation arising from the brain voxel is

$$E(g, \boldsymbol{u}_g, \delta, \Delta) = (1 - f_{\text{free}}) \times \frac{\sum_{m=1}^{M} V_m \cdot E_m}{\sum_{m=1}^{M} V_m} + f_{\text{free}} \times e^{-Db}, \tag{7}$$

where the subscription m indicates the m-th cell, V_m is the neuronal volume, E_m is the signal attenuation of m-th neuron, and the diffusivity is $D = 3 \times 10^{-3}\,\mathrm{mm}^2/\mathrm{s}$.

We randomly pick M neurons from the Neuron Data Set to comprise the artificial brain voxels. The number M ranges from 1 to 500. Each combination of M cells is then paired with ten different free diffusion compartment signals whose volume fractions follow a Gaussian distribution $\mathcal{N}(\mu = 0.5, \sigma^2 = 0.25^2)$. The choice of the Gaussian distribution is empirical. In this study, the combinations of cells and f_{free} produce the diffusion MRI signals from 1.45 million distinct artificial brain voxels. These signals form a part of the Synthetic dataset.

The signal attenuation obtained from eq. (7) depends on gradient intensity, gradient direction, pulse duration, and inter-pulse duration. We average the signal attenuations over all measured directions. The direction-averaged signal is

$$\overline{E}(g, \delta, \Delta) = \frac{1}{N_{\text{dir}}} \sum_{i=1}^{N_{\text{dir}}} E(g, \boldsymbol{u}_g^i, \delta, \Delta), \tag{8}$$

where N_{dir} is the number of gradient directions and $\boldsymbol{u}_g^i$ is the i-th gradient direction for a given gradient intensity g. The average over directions is a common practice to reduce the signal dimensionality [89]. Moreover, it is also helpful in denoising the experimental data. But the directional average removes all the orientation-dependent information. So we focus only on rotationally-invariant microstructure parameters in this study.

3.2.2. Features extraction

The direction-averaged signals at the 64 gradient intensities can directly serve as the first set of features denoted by $\boldsymbol{F}_1$. We follow the terminology of the machine learning community, referring to an input variable of an ML model as a feature.

We also propose a second set of features that are less obvious than the signal values. Inspired by Veraart *et al.* [9, 45], our previous work [46] shows that the inflection point of a direction-averaged signal curve helps define markers that are useful for the soma size estimation. Here, we adopt the same idea and use five markers as the second set of features $\boldsymbol{F}_2$. By plotting the direction-averaged signals against $\beta = 1/\sqrt{b}$, we obtain an S-shaped curve on which we can define five potential features (see fig. 4). Due to the change of concavity, there is an inflection point. If there are several inflection points, we take the rightmost one (with the smallest b-value). We then draw the tangent line through the inflection point. The x and y coordinates of the inflection point, the slope and intercept of the tangent line, and the ADC are the five markers forming the second set of features. The ADC can be estimated using the signal attenuation with the smallest b-value.

To summarize, we have two sets of features. One includes the direction-averaged signals, and another is composed of the five markers. We can get both sets from the simulated signals with ease. However, computing the two sets of features from experimental data requires an interpolation method.

Real-world acquisition of dMRI signals has many fewer gradient intensities than simulation. Besides, the way of sampling varies across practitioners. However, we train MLPs on the direction-averaged signals at 64 fixed g's or the five markers whose computation requires signals at numerous gradient intensities. The application of MLPs to *in-vivo* data requires the interpolation of the measured signals. We acknowledge that the imaging data required for the MLPs described in this paper are not actually feasible for most practioners as the required gradient strengths are only available on Siemens Connectom scanners (which are very rare). This paper is about a proof of concept using the MGH CDMD.

We adopt the fourth-order B-spline interpolation implemented in Scipy [90]. A vanilla cubic spline suffers a large fluctuation. To moderate the fluctuation, we leverage the GPA when b-values are smaller than b_1. The GPA allows us to approximate $\overline{E}$ by $e^{D_e \cdot b} = e^{D_e/\beta^2}$ for $b \leq b_1$. Furthermore, the GPA provides two boundary conditions which are the continuity of the first and second derivatives at β_1:

$$\overline{E}'(\beta_1) = 2D_e \cdot e^{-D_e/\beta_1^2}/\beta_1^3, \tag{9}$$

$$\overline{E}''(\beta_1) = \left(4D_e^2 - 6D_e\beta_1^2\right) e^{-D_e/\beta_1^2}/\beta_1^6. \tag{10}$$

At the high b-value end (small β), we adopt the "natural" boundary condition [91]

$$\overline{E}''(\beta_8) = 0. \tag{11}$$

The boundary conditions help moderate the fluctuation of the interpolation and allow us to sample g's within the maximum gradient intensity and find the inflection point. Figure 4 demonstrates the measured and interpolated signals and the tangent line passing through the estimated inflection point. It is worth noting that one should not extrapolate the direction-averaged signals beyond the maximum gradient intensity.

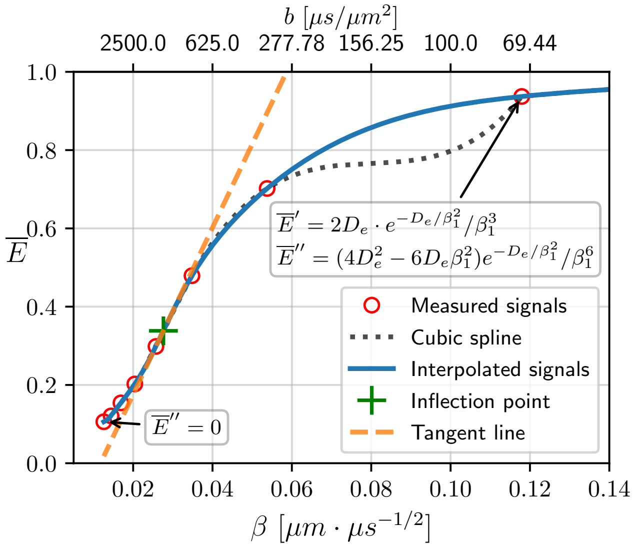


Figure 4: Fourth-order B-spline interpolation of direction-averaged signals. Red circles represent the direction-averaged signals at eight non-zero b-values measured from a white matter voxel of the first subject (`sub_001`) in MGH CDMD. The voxel index is (19, 25, 73). A vanilla cubic spline interpolation represented by the dotted black line suffers a large fluctuation. By incorporating the three boundary conditions annotated in the boxes, the fourth-order B-spline method interpolates the eight measured signals giving the solid blue line. The rightmost root of the second derivative of the interpolated signals gives the inflection point shown as the green cross. The dashed orange line is the tangent line passing through the inflection point.

3.2.3. Voxel microstructure parameters

Another part of the Synthetic dataset is the microstructure parameters of artificial brain voxels. Given the neuroanatomical parameters, we can compute the microstructure parameters of an artificial brain voxel consisting of M neuron meshes and a free compartment whose volume fraction is f_{free}. Some microstructure parameters (where the superscript indicated the m-th neuron) are the soma volume fraction:

$$f_{\text{soma}} = (1 - f_{\text{free}})\frac{\sum_{m=1}^{M} V_{\text{soma}}^m}{\sum_{m=1}^{M} V_{\text{neuron}}^m}, \tag{12}$$

the neurite volume fraction:

$$f_{\text{neurite}} = (1 - f_{\text{free}})\frac{\sum_{m=1}^{M} V_{\text{neurite}}^m}{\sum_{m=1}^{M} V_{\text{neuron}}^m}, \tag{13}$$

the soma area fraction:

$$a_{\text{soma}} = \frac{\sum_{m=1}^{M} A_{\text{soma}}^m}{\sum_{m=1}^{M} A_{\text{neuron}}^m}, \tag{14}$$

the neurite area fraction:

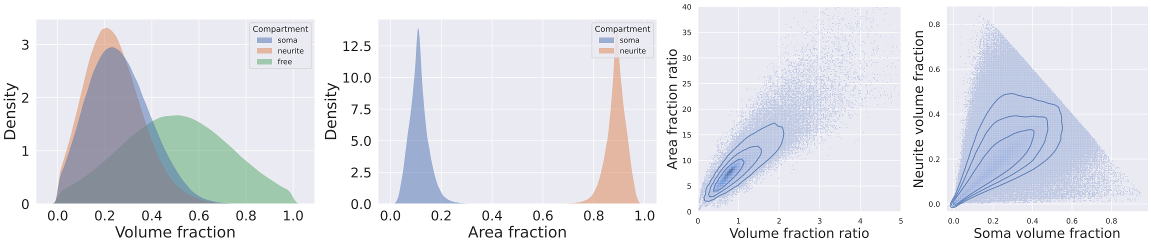
$$a_{\text{neurite}} = \frac{\sum_{m=1}^{M} A_{\text{neurite}}^m}{\sum_{m=1}^{M} A_{\text{neuron}}^m}, \tag{15}$$

the volume averaged soma radius:

$$\overline{r}_{\text{soma}} = \frac{\sum_{m=1}^{M} V_{\text{soma}}^{m} r_{\text{soma}}^{m}}{\sum_{m=1}^{M} V_{\text{soma}}^{m}}. \tag{16}$$

The list is not exhaustive. More neuroanatomical and microstructural parameters can be defined. We emphasize that not all microstructure parameters measured on neuron meshes can be inferred using diffusion MRI. Whether we can estimate a parameter mainly depends on the dynamics of MR physics. This paper focuses on volume fractions, area fractions, and the soma radius.

Figure 5 presents the distributions of the volume fractions and area fractions in the Synthetic dataset. We note that the ratio of the volume fractions of neurites and soma is somewhat correlated with their area fraction ratio. On the other hand, the soma volume fraction and the neurite volume fraction are not very correlated.



(a) Volume fractions

(b) Area fractions

(c) Neurite over soma ratios

(d) Volume fractions

Figure 5: The distributions of the volume fractions and area fractions in the Synthetic dataset. (a) The distributions of f_{soma}, f_{neurite}, f_{free}. (b) The distributions of a_{soma} and a_{neurite}. c) The joint distributions of $f_{\text{neurite}}/f_{\text{soma}}$ and $a_{\text{neurite}}/a_{\text{soma}}$. d) The joint distributions of f_{soma} and f_{neurite}. The Synthetic dataset contains 1.45 million artificial voxels, where f_{free} follows a Gaussian distribution $\mathcal{N}(\mu = 0.5,\ \sigma^2 = 0.25^2)$, f_{soma} and f_{neurite} are derived from realistic neuron meshes. The contour lines in the joint distributions contain 50%, 75%, and 90% of the data points.

3.3. Model training

We use MLPs to infer some microstructure parameters of interest. Consider a set of tuples that is extracted from the Synthetic dataset, $\mathcal{T} = \{(\boldsymbol{X}_i, \boldsymbol{Y}_i), i \in \{1, ..., N_{\text{voxel}}\}\}$ where N_{voxel} ($= 1.45$ million) is the number of the artificial brain voxels. We refer to a tuple $(\boldsymbol{X}, \boldsymbol{Y})$ as a data point. The input of a MLP is denoted by $\boldsymbol{X}$, which can be the direction-averaged signals $\boldsymbol{F}_1$ or the five markers $\boldsymbol{F}_2$ arising from an artificial brain voxel. The desired output (some microstructure parameters) is denoted by $\boldsymbol{Y}$. We focus only on volume fractions and area fractions in this paper. The four combinations of input and output of MLPs are

1. $\boldsymbol{X} \in [0, 1]^n$ is the direction-averaged signals at n gradient intensities linearly spaced between 0 and 290 mT/m with Δ being 19 or 49 ms, $\boldsymbol{Y} = [f_{\text{soma}}, f_{\text{neurite}}, f_{\text{free}}]^T \in [0, 1]^3$ represents the volume fractions;
2. $\boldsymbol{X} \in [0, 1]^n$ is same as above, $\boldsymbol{Y} = [a_{\text{soma}}, a_{\text{neurite}}]^T \in [0, 1]^2$ represents the area fractions;

3. $\boldsymbol{X} \in \mathbb{R}^5$ is the five markers with Δ being 19 or 49 ms, $\boldsymbol{Y} = [f_{\text{soma}}, f_{\text{neurite}}, f_{\text{free}}]^T \in [0,1]^3$ represents the volume fractions;

4. $\boldsymbol{X} \in \mathbb{R}^5$ is same as above, $\boldsymbol{Y} = [a_{\text{soma}}, a_{\text{neurite}}]^T \in [0,1]^2$ represents the area fractions.

To improve the robustness of MLPs to noise, we apply Rician noise to the simulated signals in items 1 and 2. The Rician noise was applied to the direction-averaged data, however, we acknowledge that ideally such noise should be applied before directional averaging. We keep the same SNR as the MGH CDMD ($S_0/\sigma_{\text{R}} = 21$), where σ_{R} is the Rician scaling parameter.

It is worth noting that the measurements at two diffusion times are analyzed separately throughout this paper. Moreover, the L1-norm of the output must be unity for the four above cases ($\|\boldsymbol{Y}\|_1 = 1$), which is an additional constraint that should be accounted for when training. We randomly select one million artificial voxels to form the training set $\mathcal{T}_{\text{train}}$; the rest (450,000 artificial voxels) makes up the test set $\mathcal{T}_{\text{test}}$ which is held out and not used for model training. The test set allows us to assess the generalization of a trained MLP [92].

A MLP is a nonlinear function h parameterized by its weights $\boldsymbol{\theta}$ [92]. The model training is to find optimal weights $\boldsymbol{\theta}^*$ that minimize the distance between the MLP's output and the desired output

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \frac{1}{\#\mathcal{T}_{\text{train}}} \sum_{i=1}^{\#\mathcal{T}_{\text{train}}} \|\boldsymbol{Y}_i - h(\boldsymbol{X}_i; \boldsymbol{\theta})\|_2^2. \tag{17}$$

Here, we use the mean squared error (MSE) as the loss function. The minimization is possible if an underlying function ζ mapping $\boldsymbol{X}_i$ to $\boldsymbol{Y}_i$ exists. Once the optimization converged, the trained MLP could be a good approximation of the underlying function, i.e., $h(\cdot; \boldsymbol{\theta}^*) \simeq \zeta$ in the sense of minimizing L2 distance in the training set. Nonetheless, such an underlying function may not exist, and convergence is not guaranteed. The generalization of the trained MLP to unseen data also needs to be assessed.

The function ζ varies with the choice of the target microstructure parameters, and the MR physics determines its existence. We must be careful about the activation function, initial weights, and optimization algorithm to reach the convergence [93]. We employ the Gaussian error linear unit (GELU) [94], a ReLU-like activation function that incorporates the properties of stochastic regularizers such as dropout [95]. The weights $\boldsymbol{\theta}$ are initialized using Kaiming initialization [96] because we employ ReLU-like activation functions. The optimization is performed with a variant of the Adam optimizer that has a long-term memory of past gradients to enhance the convergence [97, 98]. The initial learning rate is 0.01, the batch size is 10,000, and the maximum number of epochs is 500. The two parameters (betas) of the Adam optimizer for computing running averages of gradient and its square are 0.9 and 0.999.

The architecture of an artificial neural network can also significantly affect its performance. Finding a suitable network architecture for brain microstructure estimation is a subject worth investigating in the future. In this paper, we chose four-layer MLPs. To guarantee the outputs are all positive and sum to unity, we append a softmax function [99] to the output layer. The implementation and training of MLPs are performed with PyTorch [100].

Finally, we assess the performance of a trained MLP in the holdout test set $\mathcal{T}_{\text{test}}$. We use the

L1-norm to evaluate the test loss

$$l_{\text{test}} = \frac{1}{dim(\boldsymbol{Y})\#\mathcal{T}_{\text{test}}} \sum_{i=1}^{\#\mathcal{T}_{\text{test}}} \|\boldsymbol{Y}_i - h(\boldsymbol{X}_i;\boldsymbol{\theta}^*)\|_1,\ (\boldsymbol{X}_i,\boldsymbol{Y}_i) \in \mathcal{T}_{\text{test}}. \tag{18}$$

We employ four-layer MLPs in this paper. Even though the MLP structure is simple, they still have several hyperparameters, namely, the size of the input layer n, the first hidden layer n_1, and the second hidden layer n_2. There are two sets of features: direction-averaged signals $\boldsymbol{F}_1$ and the five markers $\boldsymbol{F}_2$. When the input is $\boldsymbol{F}_1$, the hyperparameters are the sizes of 1) the input layer, 2) the first hidden layer, and 3) the second hidden layer. We denote them by (n, n_1, n_2). With $\boldsymbol{F}_2$ as input, the size of the input layer is 5. In this case, the hyperparameters are n_1 and n_2. We have two sets of output, i.e., volume fractions and area fractions. Moreover, the two diffusion times are analyzed separately. In total, we trained forty-eight MLPs [2] to find optimal hyperparameters. Details can be found in the Appendix (Section 7.3). The selected hyperparameters are $(64, 128, 64)$ for direction-averaged signals and $(30, 30)$ for the five markers. In the next section, we assess the final performance of each MLP in the holdout test set $\mathcal{T}_{\text{test}}$.

4. Results

In this section we demonstrate eight exemplary MLPs for volume fractions and area fractions estimation in the synthetic test set and the MGH CDMD [72]. The MLPs are compared with the state-of-the-art impermeable biophysical model, SANDI [20].

4.1. Experimental data

The MGH CDMD [72] is an open-access diffusion MRI database providing preprocessed *in-vivo* human brain scans for 26 healthy subjects, seven of which are scanned twice. The diffusion MRI data were acquired on the 3T Connectome MRI scanner (Magnetom CONNECTOM, Siemens Healthineers), and a 64-channel phased array head coil [101] was used for signal reception. The maximum slew rate is 62.5 mT/m/ms. The diffusion encoding sequence is PGSE, whose parameters are:

- the pulse duration $\delta = 8$ ms, two inter-pulse durations $\Delta = 19,\ 49$ ms;
- eight gradient intensities (31, 68, 105, 142, 179, 216, 253, 290) mT/m, corresponding to b-values (72, 346, 825, 1509, 2400, 3491, 4789, and 6292) s/mm^{2}for $\Delta = 19$ ms and (204, 981, 2340, 4279, 6800, 9902, 13,584, 17,848) s/mm^{2}for $\Delta = 49$ ms.
- 32 diffusion encoding directions uniformly distributed on a sphere for $b < 2400$ s/mm^{2}and 64 uniform directions for $b \geq 2400$ s/mm^2.

Other imaging parameters are as follows: the echo time $TE = 77$ms, repetition time $TR = 3800$ms, field of view (FOV) $= 216 \times 216$ mm, slice thickness = 2mm, voxel size $= 2 \times 2 \times 2$ mm^3. The diffusion MRI data were preprocessed to correct gradient nonlinearity, eddy currents, and susceptibility-induced distortions. The estimated median signal-to-noise ratio (SNR) is 21 [72, 102]. MGH

CDMD provides the real part of the signal for some subjects. In this paper, we only use the signal magnitude. More details about the data acquisition and processing can be found in the work of Tian *et al.* [72].

4.2. Model training and validation

We leverage the L1-loss to quantify the test loss (see eq. (18)). For clarity, we assign each MLP a name and list the network structures, the input, the output, the final test losses, and the R^2 scores in table 1. The average test loss is about 0.01, while the volume fraction or area fraction is of the order of 0.1. The R^2 scores for volume fraction estimation are around 0.9, and about 0.6 for area fraction estimation. Note that four MLPs, `mlp_F1_vol_19`, `mlp_F1_vol_49`, `mlp_F1_area_19`, and `mlp_F1_area_49`, are tested under a noise condition similar to MGH CDMD (SNR = 21). The rest of the MLPs, which take $\boldsymbol{F}_2$ as input, are assessed on noise-free test sets.

MLP name	MLP structure	diffusion time δ/Δ [ms]	input	output	test loss	R^2 scores
`mlp_F1_vol_19`	64, 128, 64, 3	8/19	64 signals	$f_{\text{soma}}/f_{\text{neurite}}/f_{\text{free}}$	0.0223	0.94/0.95/0.98
`mlp_F1_vol_49`	64, 128, 64, 3	8/49	64 signals	$f_{\text{soma}}/f_{\text{neurite}}/f_{\text{free}}$	0.0187	0.95/0.94/0.99
`mlp_F2_vol_19`	5, 30, 30, 3	8/19	5 markers	$f_{\text{soma}}/f_{\text{neurite}}/f_{\text{free}}$	0.0171	0.95/0.94/0.99
`mlp_F2_vol_49`	5, 30, 30, 3	8/49	5 markers	$f_{\text{soma}}/f_{\text{neurite}}/f_{\text{free}}$	0.0218	0.93/0.90/0.99
`mlp_F1_area_19`	64, 128, 64, 2	8/19	64 signals	$a_{\text{soma}}/a_{\text{neurite}}$	0.0157	0.66
`mlp_F1_area_49`	64, 128, 64, 2	8/49	64 signals	$a_{\text{soma}}/a_{\text{neurite}}$	0.0174	0.62
`mlp_F2_area_19`	5, 30, 30, 2	8/19	5 markers	$a_{\text{soma}}/a_{\text{neurite}}$	0.0153	0.66
`mlp_F2_area_49`	5, 30, 30, 2	8/49	5 markers	$a_{\text{soma}}/a_{\text{neurite}}$	0.0196	0.51

Table 1: Summary of the eight MLPs. The structure of a MLP is represented by four numbers “a, b, c, d”, i.e., the input layer size “a”, the first hidden layer size “b”, the second hidden layer size “c”, and the output layer size “d”. The inputs of the MLPs are either the direction-averaged signals at 64 gradient intensities or the five markers. The outputs are volume or area fractions. Four MLPs, `mlp_F1_vol_19`, `mlp_F1_vol_49`, `mlp_F1_area_19`, and `mlp_F1_area_49`, are tested under the noise condition similar to MGH CDMD (SNR = 21). The rest of the MLPs, which take the five markers as input, are assessed on noise-free test sets. We list the final test losses of MLPs (after 500 epochs). The three R^2 scores of the first four MLPs are for soma, neurite, and free diffusion compartment respectively. For example, the R^2 scores of soma, neurite, and free diffusion compartment volume fraction estimation using `mlp_F1_vol_19` are 0.94, 0.95, and 0.98, respectively. Because the sum of soma and neurite area fractions is unity, the soma area fraction estimation has the same R^2 score as neurite.

4.3. Comparison with SANDI

The state-of-the-art impermeable biophysical model for soma and neurite density imaging, SANDI [20], has similar assumptions to our brain voxel model. Both assume that the soma and neurite membranes are impermeable, somas are spherical, and there is a free diffusion compartment. The differences in modeling are:

1. SANDI considers the neurites as a set of randomly oriented sticks (long cylinders with zero radii), whereas our neurites have realistic radii and length, faithful undulation, and real dispersion;

2. SANDI utilizes disconnected soma and neurites because the water exchange between them is believed to be negligible when $t_d \leq 20$ms. In contrast, our neurites are connected to the soma, forming a continuous cell space.

Due to the impermeability assumption and item 2, the validity regime of SANDI is $t_d \leq 20$ms. So the short diffusion time $\delta/\Delta = 8/19$ms is in the regime, while the long diffusion time is not.

The significant difference is in the dMRI signal generation. The direction-averaged signal of SANDI has an explicit expression

$$\overline{E}_{\text{SANDI}} = f_{\text{soma}} e^{-D_s b} + f_{\text{neurite}} \sqrt{\frac{\pi}{4bD_{\text{in}}}} \text{erf}\left(\sqrt{D_{\text{in}} b}\right) + f_{\text{free}} e^{-D_{\text{free}} b}, \tag{19}$$

where D_{in} is the longitudinal diffusion coefficient in the sticks; D_{free} is the scalar effective diffusion constant in the free diffusion compartment, f_{soma}, f_{neurite}, and f_{free} are the volume fractions for the soma, neurite and free diffusion compartments, respectively. The soma signal term $e^{-D_s b}$ is derived, under the GPA, by Murday and Cotts on a spherical liquid particle with radius r_s and liquid self-diffusion coefficient D_s [103]. The quantity D_s, which is a function of δ, Δ, r_s, and D_s, has an explicit formulation which can be found in [103] and [27]. We refer to the soma signal term as the "MC equation" following the terminology adopted by Balinov *et al.* [27]. For eq. (19) to hold, it is necessary to add at least three additional assumptions:

(a) the Gaussian phase assumption and the MC equation hold under the experimental condition;

(b) the signal from a spherical soma with an "apparent" radius r_s can approximate the volume-weighted average signal from a group of somas;

(c) the stick power law, which gives the neurite signal term $\sqrt{\frac{\pi}{4bD_{\text{in}}}}\text{erf}(\sqrt{D_{\text{in}}b})$, is valid.

With these assumptions, the direction-averaged signal of SANDI ($\overline{E}_{\text{SANDI}}$) is an explicit function whose variables are: δ, Δ, b, f_{soma}, f_{neurite}, f_{free}, r_s, D_s, D_{in}, and D_{free}. Among them, δ, Δ, and b are known experimental parameters. In [20], D_s is fixed to be $3\times10^{-3}\,\text{mm}^2/\text{s}$, and the remaining six variables are the microstructure parameters to be estimated. SANDI estimates the six parameters of biophysical interest by fitting eq. (19) to the direction-averaged signals. In contrast to our method, the SANDI model has an analytical signal expression, which allows one to recompute the direction-averaged signals by substituting SANDI's estimations into the signal formula. Since our neuron models are more realistic than SANDI's and the signal simulation requires fewer biophysical assumptions, it is worth comparing SANDI with our simulation framework.

To avoid the commonly encountered numerical instabilities due to the difficulty of finding the global minimum in optimization procedures, we chose to perform SANDI fitting using an exhaustive search approach on a saved signal library. Exhaustive search should be more numerically stable than a fast optimization method such as AMICO[104]. To make the library search computationally feasible, we further simplified the problem by fixing the cylinder diffusion coefficient to $D_{\text{in}} = D_s = 3 \times 10^{-3}\,\text{mm}^2/\text{s}$, as well as fixing the free diffusion coefficient to be $D_{\text{free}} = 3 \times 10^{-3}\,\text{mm}^2/\text{s}$.

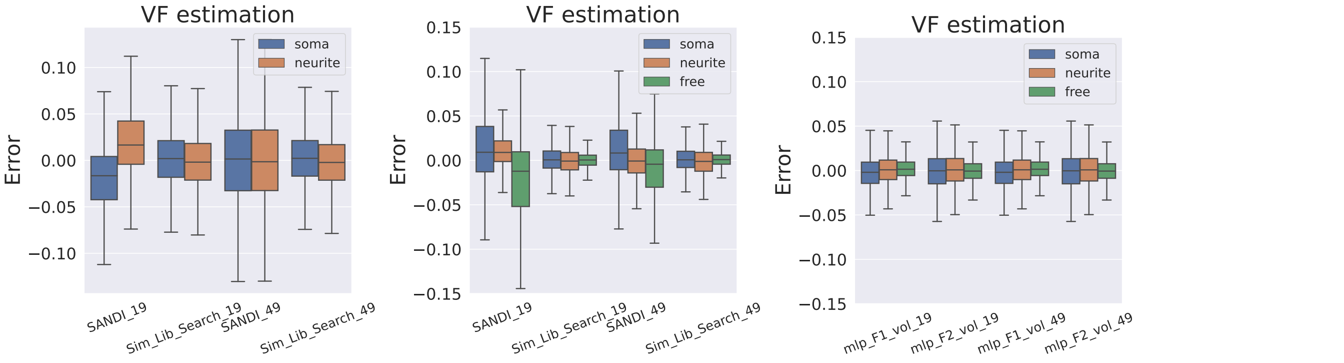
Thus, the signals library for SANDI is populated along three dimensions: the soma radius discretized on the interval $[0, 35]\mu$m in 0.35μm increments, the f_{soma} and f_{neurite} discretized on the interval $[0, 1]$ in 0.01 (1 percent) increments. The set of three parameters $\{\overline{r}_{soma}, f_{\text{soma}}, f_{\text{neurite}}\}$ that gives the smallest L_2 difference at the 64 direction-averaged signal values will be called the SANDI fitted parameters for that voxel.

In addition to comparing the previously described MLPs to SANDI, we additionally compare SANDI to an exhaustive search approach within our simulation framework, where the signal libary is a subset of the Synthetic dataset. This makes it possible to compare the two signal models only, without having to account for errors that come from the optimization procedures. In the case of exhaustive search within our simulation framework, because multiple combinations of neurons and f_{free} can give similar direction-averaged signals, we averaged the parameters from the elements of the signal library that gave the 10 lowest L_2 differences at the 64 direction-averaged signal values. Thus, our simulation framework fitted parameters $\{\overline{r}_{soma}, f_{\text{soma}}, f_{\text{neurite}}\}$ are the averages of 10 values from the simulation framework signal library. We note here as well that even though the SANDI parameters are exactly $\{\overline{r}_{soma}, f_{\text{soma}}, f_{\text{neurite}}\}$ due to the signal model being an explict expression of those variables, we can obtain additional morphological parameters from the simulation framework library, such as the soma area fraction.

First, we compare SANDI and the simulation framework in terms of their signal models for the neurons. We remind the reader that the Synthetic dataset includes 145,000 combinations of neurons, where in essence, $f_{\text{free}} = 0$ for these artificial voxels. We randomly select 75,000 out of the 145,000 artificial voxels as the Synthetic neurons-only test set. We remind the reader that we have the ground truth morphological parameters for the test set. We will use the remaining 70,000 artificial voxels as the simulation framework signals library. We compare the performance of exhaustive search using the simulation framework signals library (70,000 elements) with using the SANDI signals library (10,000 elements, incremented in $\overline{r}_{\text{soma}}$ and f_{soma}, setting $f_{\text{free}} = 0$). In fig. 6a we show the errors between the SANDI fitted volume fractions and the ground truth, as well as the errors between the simulation signals library fitted volume fractions and the ground truth. We see a bias in f_{soma} and in f_{neurite} using the SANDI signals library at diffusion time 19ms, there is no bias in the SANDI fit at diffusion time 49ms. There is no bias in the simulation signals library fit at either diffusion time. The errors are also smaller for the simulation framework than SANDI.

Now we turn to the Synthetic dataset's full 1.45 millions artificial voxels containing neurons and a free diffusion compartment. We randomly picked 450,000 artificial voxels in the Synthetic dataset as the Synthetic test set. We used the remaining 1 million artificial voxels as the simulation framework signals library. We compare the performance of exhaustive search using the simulation framework signals library (1 million elements) with using the SANDI signals library (1 million elements, incremented in $\overline{r}_{\text{soma}}$, f_{soma} and f_{free}). In fig. 6b we see a larger bias in f_{soma} and in f_{free} in the SANDI library fit than the simulation library fit. The errors are also smaller for the simulation library search than the SANDI library search. In fig. 6c we see the fitted volume fractions are also unbiased for the MLPs, with the `mlp_mk` having slightly larger errors than the `mlp_sig`. The MLPs have slightly larger errors than the simulation framework library search, but not as big as the SANDI library search.

It is not possible to obtain area fractions information from SANDI, but it is possible to obtain this information from the simulation framework. In fig. 7a we see the fitted a_{soma}, a_{neurite} have small errors for `mlp_F1` and `mlp_F2`, as well as for the simulation framework library search. Because we did not train MLPs to obtain $\overline{r}_{\text{soma}}$ (we only trained MLPs for the volume and area fractions), we now compare the estimated $\overline{r}_{\text{soma}}$ between simulation-framework library search and SANDI library search. In fig. 7b, the fittted $\overline{r}_{\text{soma}}$ errors are much smaller for the simulation library search than



(a) Synthetic neurons-only test set

(b) Synthetic test set

(c) Synthetic test set

Figure 6: The box plots summarizing the distributions of the absolute errors of f_{soma}, f_{neurite}, f_{free}. a) Comparing SANDI library search versus simulation framework library search on the Synthetic neurons-only test set. b) Comparing SANDI library search versus simulation framework library search on the Synthetic test set. c) Comparing Signal MLPs and Biomarkers MLPs on the Synthetic test set. The Synthetic neurons-only dataset has 145,000 artificial voxels containing neurons only. The Synthetic neurons-only test set has 75,000 voxels from the Synthetic neurons-only dataset. The simulation framework neurons-only library has the remaining 70,000 voxels from the Synthetic neurons-only dataset. The Synthetic dataset has 1.45 millions artificial voxels containing neurons and a free diffusion compartment. The Synthetic test set has 450,000 voxels from the Synthetic dataset. The simulation framework signals library has the remaining 1 million voxels from the Synthetic dataset. A box plot denotes the median, interquartile range, and 1.5 times the interquartile range by the center line, hinges, and whiskers.

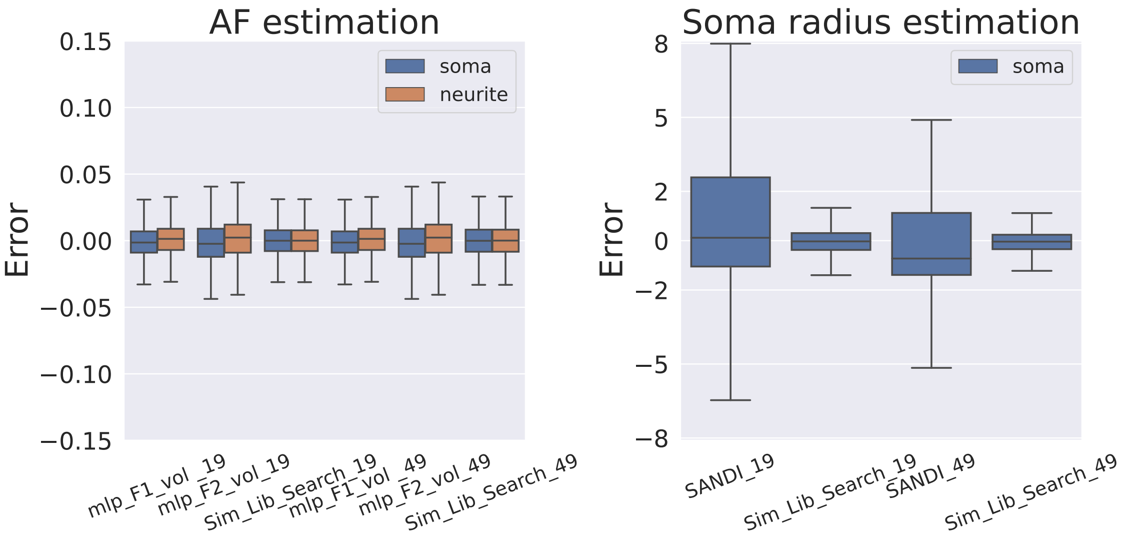
SANDI library search.

4.4. Fitting brain white matter voxels

To be able to fit voxels that contain primarily axons, we created 20,000 random combinations of cylinders with radius from the interval $[0.2, 5]\mu$m and a free diffusion compartment, and added them to the simulation framework library. Henceforth, the full simulation framework library has 1 million artificial voxels from the Synthetic dataset plus 20,000 voxels that contain cylinders and a free diffusion compartment only. Then we generated a Synthetic Cylinders-only test set that has 10,000 voxels containing random combinations of cylinders. We fitted f_{soma}, f_{neurite}, f_{free} using SANDI library search and the simulation framework library search on the Synthetic Cylinders-only test set. In table 2, we see that with SANDI the mean of the fitted values are $f_{\text{soma}} = 0.14$, f_{neurite}=0.86, f_{free}=0, whereas with the simulation framework library search, $f_{\text{soma}} = 0$, f_{neurite}=1, f_{free}=0. This means the simulation framework library search will be able to fit axons-only voxels.

4.5. In vivo parameters estimation

We now apply the trained MLPs to the MGH CDMD. Specifically, the eight direction-averaged signals from a brain voxel are interpolated to get features, namely the 64 signals or the five markers. A trained MLP takes the features to predict the desired microstructure parameters. We obtain a parameter map by applying a MLP to every brain voxel of a subject.



(a) a_{soma}, a_{neurite}

(b) $\overline{r}_{\mathrm{soma}}$

Figure 7: The box plots summarizing the distributions of the absolute errors of a_{soma}, a_{neurite}, $\overline{r}_{\mathrm{soma}}$. a) Comparing `mlp_sig`, `mlp_mk`, and simulation framework library search on the Synthetic test set. b) Comparing SANDI library search and simulation framework library search on the Synthetic test set. The Synthetic dataset has 1.45 millions artificial voxels containing neurons and a free diffusion compartment. The Synthetic test set has 450,000 voxels from the Synthetic dataset. The simulation framework library has the remaining 1 million voxels from the Synthetic dataset. A box plot denotes the median, interquartile range, and 1.5 times the interquartile range by the center line, hinges, and whiskers.

4.5.1. Validation of the interpolation method

The dMRI signals in MGH CDMD are measured at eight gradient intensities, with the highest intensity being 290 mT/m. However, we train the eight MLPs on the five markers or 64 direction-averaged signals. Before applying the MLPs to the experimental data, we need the signal interpolation.

The interpolation inevitably brings in errors. We assess the interpolation error using data from the MGH CDMD. The direction-averaged signals at eight gradient intensities from a brain voxel are split into two subsets. The first set contains signals at N_1 ($4 \leq N_1 < 8$) gradient intensities, including the lowest and the highest gradients. The second set includes the remaining N_2 signals. Following the method described previously, we obtain the fourth-order B-spline polynomial using the first set. We then predict the signals stored in the second set using the polynomial. The measured and predicted signals are denoted by $\overline{E}_{\mathrm{m}}$ and $\overline{E}_{\mathrm{p}}$, respectively. The interpolation error is assessed by the relative error $(\overline{E}_{\mathrm{p}} - \overline{E}_{\mathrm{m}})/\overline{E}_{\mathrm{m}} \times 100\%$.

We adopt two splitting strategies:

1. the first set contains four direction-averaged signals whose gradient intensities are 31, 105, 179, and 290 mT/m; the second set includes four signals at 68, 142, 216, and 253 mT/m; i.e., $N_1 = 4$ and $N_2 = 4$;

	f_{soma}		f_{neurite}		f_{free}	
	mean	std	mean	std	mean	std
SANDI_19	0.14	0.0076	0.86	0.0076	0	0
Sim_Lib_Search_19	0	0	1	0.00024	0	0.00024
SANDI_49	0.14	0.0076	0.86	0.0076	0	0
Sim_Lib_Search_49	0	0	1	0.00024	0	0.00024

Table 2: Fitted f_{soma}, f_{neurite}, f_{free} using SANDI library search and the simulation framework library search on the Synthetic Cylinders-only test set. The Synthetic Cylinders-only test set has 10,000 voxels containing random combinations of cylinders. The simulation framework library has 1 million artificial voxels from the Synthetic dataset plus 20,000 voxels that contain cylinders and a free diffusion compartment only. The SANDI library has 1 million elements, incremented in $\overline{r}_{\text{soma}}$, f_{soma} and f_{free}.

2. the first set contains six direction-averaged signals whose gradient intensities are 31, 68, 105, 179, 253, and 290 mT/m; the second set includes signals at 142 and 216 mT/m; i.e., $N_1 = 6$ and $N_2 = 2$;

The relative errors are computed for all brain voxels at the N_2 gradient intensities. For example, the first subject in MGH CDMD has 142,201 brain voxels. We can obtain $142,201 \times 4$ relative errors using the first splitting strategy. The box plots in fig. 8 summarize the distribution of the relative errors for the first four subjects in MGH CDMD. We refer to the first strategy as "4/4" and the second as "6/2".

We notice that the first strategy can adequately interpolate the direction-averaged signals when $\delta/\Delta = 8/19\ ms$. However, the interpolation with four signals becomes biased for the long diffusion time. Using more measured signals can help reduce interpolation errors. The second strategy is satisfactory in both cases. More than 50% predicted signals have a relative error inferior to 5%. Almost all relative errors are below 15%. We believe the second strategy is adequate for signal interpolation. We actually interpolate with eight direction-averaged signals. We can expect the actual interpolation error is even lower.

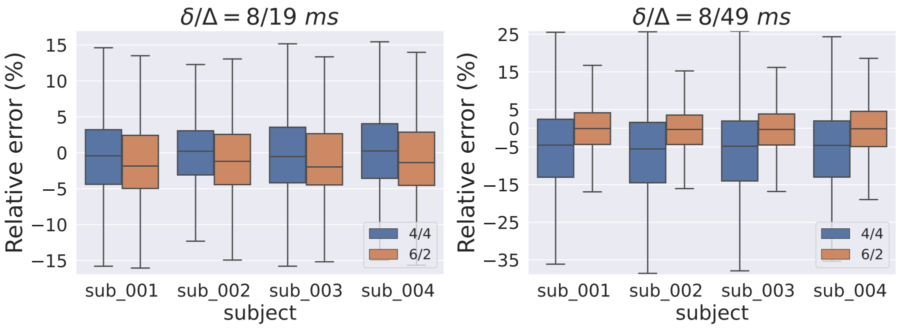
4.5.2. Parameter maps during training

During the model training, the test error of a MLP decreases, meaning that the performance improves on the Synthetic test set. However, the MLP may not generalize well on experimental data. We plot several parameter maps in different training stages in fig. 9 to show that the performance also improves on actual measurements. As the test error decreases, the contrast of, for example, the cerebellar white matter becomes more pronounced. Besides, the test loss becomes stable after 400 epochs. As mentioned in section 3.3, we stop the training at 500 epochs.

4.5.3. In vivo parameter maps

We obtain *in-vivo* parameter maps by applying the trained MLPs to every brain voxel of a subject. The second subject in MGH CDMD (`sub_002`) serves as an exemplar. Figure 10 shows the volume fractions estimation by applying the first four MLPs to the scanned data of `sub_002`.

We also include the SANDI library search parameter maps as well as the simulation framework library search parameter maps in fig. 11. We emphasize that the SANDI model is supposed to be



(a)

(b)

Figure 8: The box plots summarizing the distribution of the interpolation errors for the first four subjects in MGH CDMD. A box plot presents the median, interquartile range, and 1.5 times the interquartile range by the center line, hinges, and whiskers. Outliers are ignored for clarity. The interpolation error is assessed by the relative error $(\overline{E}_{\mathrm{p}} - \overline{E}_{\mathrm{m}})/\overline{E}_{\mathrm{m}} \times 100\%$. We refer to the first splitting strategy as "4/4" and the second as "6/2". It turns out that the second strategy is satisfactory for signal interpolation. More than 50% predicted signals have a relative error inferior to 5%. Almost all relative errors are below 15%.

invalid at the long diffusion time ($\delta/\Delta = 8/49\ ms$) because the cellular membrane permeability and the water exchange between soma and neurites may cause considerable effects. However, it seems that SANDI still gives reasonable estimations at the long diffusion time because the parameter maps are similar at the two diffusion times. We quantitatively study the dependence on diffusion time in the next section.

In addition to the volume fraction maps, the area fraction maps are shown in fig. 12. Because the sum of soma and neurite area fractions is unity, we only present the parameter maps for soma.

4.5.4. Independence of diffusion time

We present the voxelwise joint distributions for the estimated parameters. All brain white and gray matter (WM and GM) voxels of `sub_002` are included. In fig. 13, we show the voxelwise joint distributions of f_{soma}, f_{neurite}, f_{free} for both the simulation library search and the SANDI library search. The estimated volume fractions of both SANDI and simulation framework library searches lie on the identity line. The simulation framework library search produces lower f_{neurite} and higher f_{free} in the white matter than SANDI library search.

In fig. 14, we show the voxelwise joint distributions of f_{soma}, f_{neurite}, f_{free} for the `mlp_F1` and the `mlp_F2`. For f_{soma} and f_{free}, the spread of the distributions is wider than for the simulation framework library search.

In fig. 15a, we show that the distributions of the estimated a_{soma} for the simulation framework library search. In fig. 15b, we show the diffusion time dependence in the estimation of $\overline{r}_{\mathrm{soma}}$ for the simulation framework library search. In fig. 15c, we show the diffusion time dependence in the estimation of $\overline{r}_{\mathrm{soma}}$ for the SANDI library search. At the lower diffusion time, $\overline{r}_{\mathrm{soma}}$ ranges from $[8, 13]\mu$m for both library searches, whereas at the longer diffusion time, $\overline{r}_{\mathrm{soma}}$ ranges from $[10 - 18]\mu$m for the simulation library search and from $[10 - 20]\mu$m for SANDI library search.

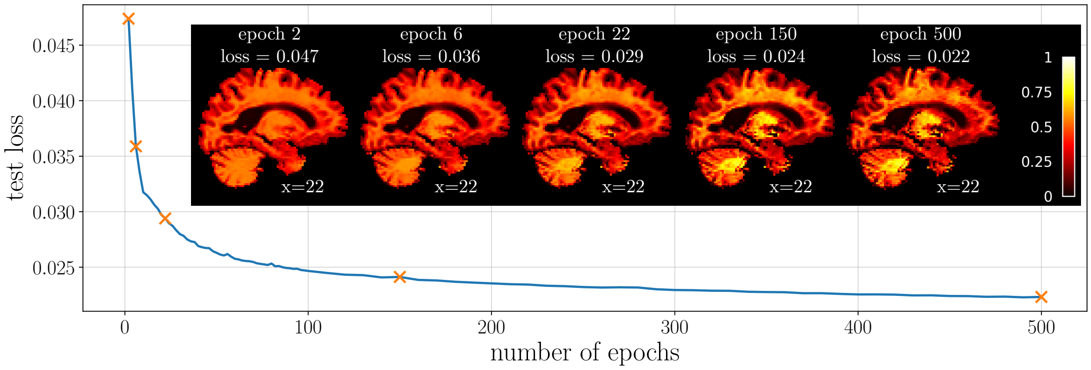


Figure 9: The estimation improvement on the test set and the experimental data as the test loss decreases. We plot the test losses during the training of `mlp_sig_vol_19`. The MLP at five distinct training stages is picked to infer the neurite volume fraction on experimental data. Orange crosses mark the selected epochs in the error curve. We present the evolution of a parameter map for the first subject (`sub_001`) in MGH CDMD. As the test error decreases, more details appear.

5. Discussion

This study proposes a novel framework that employs realistic neuron modeling and diffusion MRI simulations to replace some simplified biophysical models and analytical intracellular signal expressions. Effects arising from, for instance, neurite undulation or water exchange between soma and neurites, which are hard to include in biophysical models, are seamlessly incorporated into our simulations. Consequently, our framework achieves higher modeling accuracy while requiring many fewer biophysical assumptions. Moreover, the microstructure parameters do not need to be explicitly expressed in the signal, allowing us to explore new contrasts, such as area fractions. Using realistic neurons is a way to inject good quality prior information for inference, or reject unlikely solutions. In an explicit mathematical signal model, without good quality prior information on the joint distributions of model parameters, one can get solutions that are not physically realistic, such as combinations of effective diffusion coefficients that are inconsistent with each other, with the radius estimations, or with the combination of multiple diffusion times. These model parameters solutions need to be rejected. Using realistic neurons may mean that these inconsistent combinations do not get produced in the first place.

The proposed framework has two cornerstones. The first one is the numerical matrix formalism implemented in the SpinDoctor toolbox [34, 50]. To achieve the results shown in this paper, we obtained a further ten-fold speedup by optimizing the eigendecomposition algorithm and leveraging GPU computations. It now takes 10 minutes to simulate 1,000 dMRI signals from a neuron mesh of around $80,000$ vertices and $150,000$ faces. Besides, the simulation accuracy has been validated in the Appendix (section 7.2.1).

The second cornerstone is the Neuron Data Set. We developed an automatic pipeline that can robustly convert the neuron skeletons to simulation-ready meshes by exploiting three well-established computer graphics algorithms. The neuron mesh generator can faithfully reproduce the neuron morphology, allowing precise neuroanatomical measurements. In addition, we guarantee mesh quality by maintaining the percentage of high-quality triangles. The meshing pipeline enables us to convert

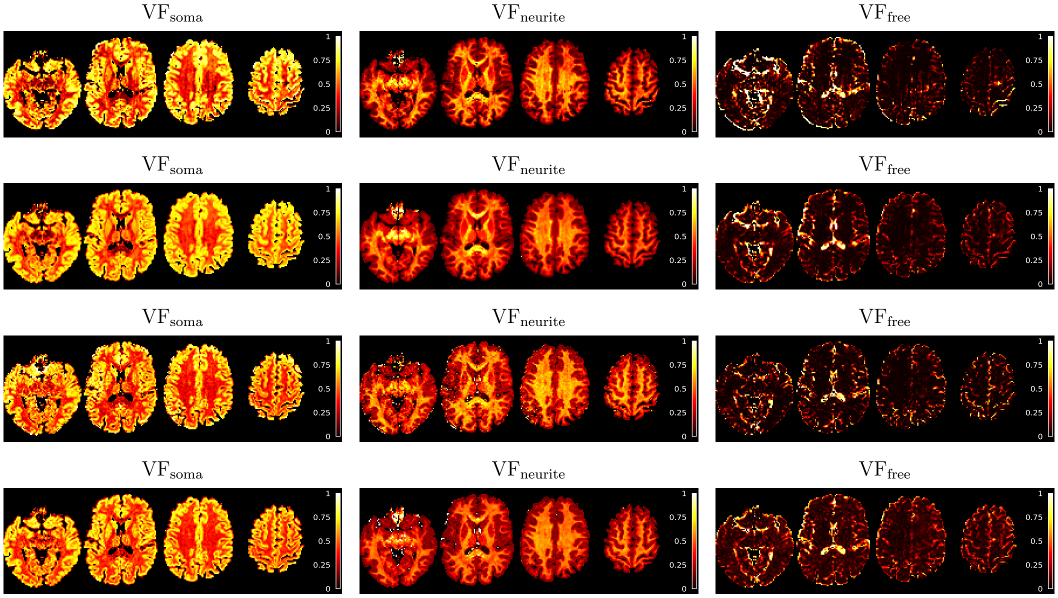

$\mathrm{VF}_{\mathrm{soma}}$
1
0.75
0.5
0.25
0
$\mathrm{VF}_{\mathrm{neurite}}$
1
0.75
0.5
0.25
0
$\mathrm{VF}_{\mathrm{free}}$
1
0.75
0.5
0.25
0
$\mathrm{VF}_{\mathrm{soma}}$
1
0.75
0.5
0.25
0
$\mathrm{VF}_{\mathrm{neurite}}$
1
0.75
0.5
0.25
0
$\mathrm{VF}_{\mathrm{free}}$
1
0.75
0.5
0.25
0
$\mathrm{VF}_{\mathrm{soma}}$
1
0.75
0.5
0.25
0
$\mathrm{VF}_{\mathrm{neurite}}$
1
0.75
0.5
0.25
0
$\mathrm{VF}_{\mathrm{free}}$
1
0.75
0.5
0.25
0
$\mathrm{VF}_{\mathrm{soma}}$
1
0.75
0.5
0.25
0
$\mathrm{VF}_{\mathrm{neurite}}$
1
0.75
0.5
0.25
0
$\mathrm{VF}_{\mathrm{free}}$
1
0.75
0.5
0.25
0

Figure 10: Parameter maps for MGH CDMD `sub_002`. The first column is f_{soma}, the second column is f_{neurite}, the third column is f_{free}. First row is `mlp_F1_vol_19` at the short diffusion time ($\delta/\Delta = 8/19$ms), the second row is `mlp_F1_vol_49` at the long diffusion time ($\delta/\Delta = 8/49$ms), the third row is `mlp_F2_vol_19` at the short diffusion time ($\delta/\Delta = 8/19$ms), the fourth row is `mlp_F2_vol_49` at the long diffusion time ($\delta/\Delta = 8/49$ms).

1213 human neuron skeletons stored in NeuroMorpho.Org to high-quality meshes, of which less than ten percent required manual post-cleaning.

Nonetheless, neuron meshes are merely the building blocks of an artificial brain voxel. To construct a voxel phantom, we would need to densely pack the neurons so that the ECS has a reasonable volume fraction and the neurons must not intersect each other. However, neuron packing is still a highly challenging open question. Thus, we circumvented this problem by adding a free diffusion compartment to the neuron signals and assuming impermeable cell membranes. This enabled us to compute the signals from a voxel without explicitly constructing the numerical phantom.

The two cornerstones bring advanced modeling capabilities allowing us to build a Synthetic dataset for supervised learning or exhaustive search. Our Synthetic dataset consists of 1.45 million artificial brain voxels with the associated simulated signals. Each voxel corresponds to thousands of directional dMRI signals and over forty microstructure parameters. The Synthetic dataset contains rich information that helps investigate the relationships between dMRI signals and tissue microstructure. Besides, the dataset is also a good reference for verifying biophysical models.

In addition to the size, the dataset quality can significantly affect the final performance of ML models[3], especially for supervised learning. We maintain the modeling quality and the simulation accuracy to ensure the dMRI signals and the ground-truth microstructure parameters are accurate.

[3] Garbage in, garbage out.

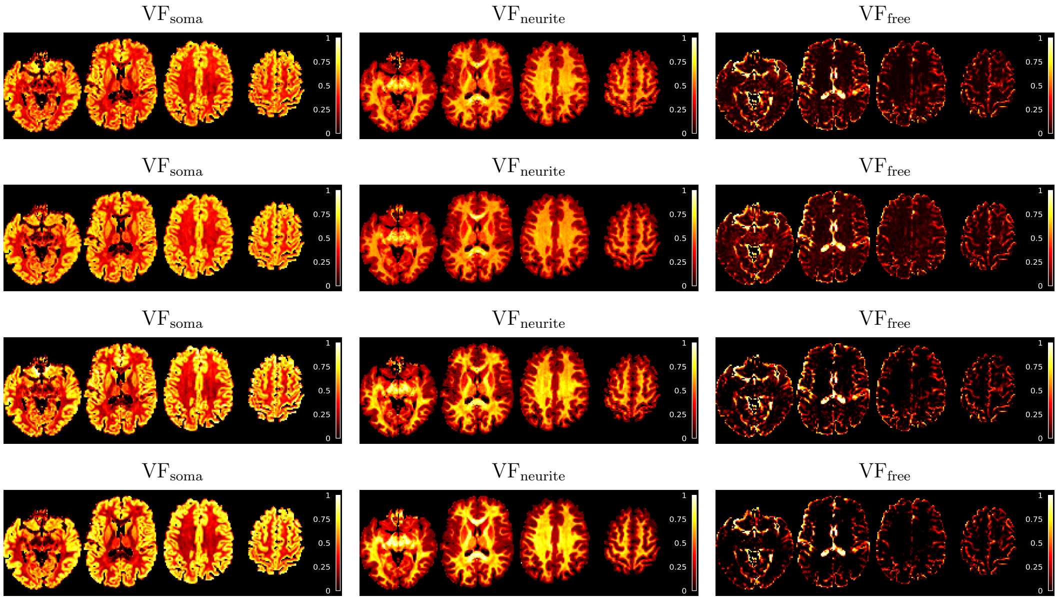
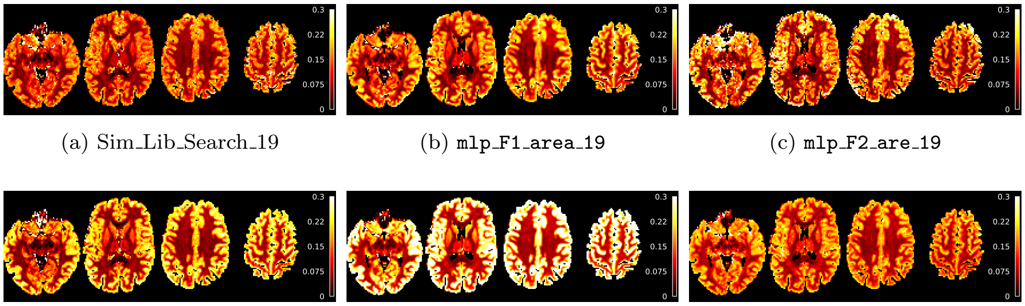


Figure 11: Parameter maps for MGH CDMD `sub_002`. The first column is f_{soma}, the second column is f_{neurite}, the third column is f_{free}. First row is simulation framework library search at the short diffusion time ($\delta/\Delta = 8/19$ms), the second row is simulation framework library search at the long diffusion time ($\delta/\Delta = 8/49$ms), the third row is SANDI library search at the short diffusion time ($\delta/\Delta = 8/19$ms), the fourth row is SANDI library search at the long diffusion time ($\delta/\Delta = 8/49$ms).

Moreover, the training data distribution should be relevant to the distribution we encounter in real-world applications due to the no-free-lunch theorem [92, 105]. This requirement has also been recognized by the diffusion MRI community [106]. Palombo *et al.* [107] provide some reference distributions for microstructure modeling based on over 3500 neuron skeletons from NeuroMorpho.Org, among which over 1000 are human cells. Because we also use the cells stored in NeuroMorpho.Org, our neuron meshes naturally follow all reference distributions.

To adapt the trained MLPs to various experimental acquisition settings, we proposed an interpolation method by imposing three boundary conditions on fourth-order B-spline interpolators. As long as enough measurements are given, the interpolation method can effectively mitigate the fluctuation caused by vanilla splines and give satisfactory approximations to the measured signals. One can freely sample signals within maximum gradient intensity or compute signal features using the proposed interpolation method.

In the Appendix (section 7.4) we show the analogous parameter maps and joint distributions for the SANDI library search and the simulation framework library search, where we used directly the 8 experimental direction-averaged signals from the MGH CDMD instead of interpolating to 64 signals. The simulation library search using the 8 experimental direction-averaged signals yields the same results as using 64 interpolated signals for both diffusion times. The results from SANDI using the 8 experimental signals are similar to using 64 interpolated signals at the shorter diffusion time, however, at the longer diffusion time, using 8 experimental signals gives a higher f_{free} compared to using 64 interpolated signals.



(a) Sim_Lib_Search_19 (b) `mlp_F1_area_19` (c) `mlp_F2_are_19`

(d) Sim_Lib_Search_49 (e) `mlp_F1_area_49` (f) `mlp_F2_area_49`

Figure 12: Soma area fraction a_{soma} parameter maps for MGH CDMD `sub_002`. First row is the short diffusion time ($\delta/\Delta = 8/19$ms), the second row is the long diffusion time ($\delta/\Delta = 8/49$ms).

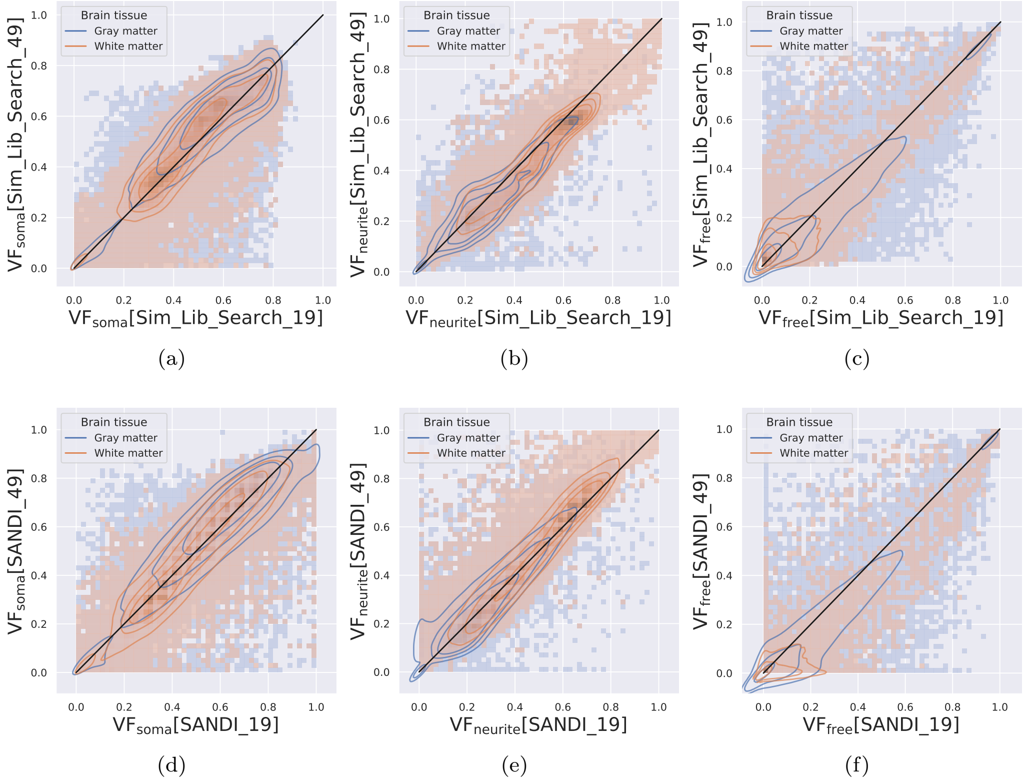
Next, we focus on approximating the mappings from the dMRI signals to the volume fractions or area fractions using MLPs. Our preliminary work suggests that such mappings exist and may be high-dimensional [46]. The MLP hyperparameters are determined on validation sets. Two sets of features are studied: the 64 direction-averaged signals and the five markers. We note the most salient points of the trained MLPs here:

1. it is possible to predict the volume fractions or area fractions by solely feeding direction-averaged signals to MLPs trained in the proposed framework;
2. the five markers form a set of concise features that can effectively predict the volume and area fractions;
3. the dependence of the volume fraction estimation on diffusion time is minimal in the MGH CDMD;
4. it is possible to investigate new microstructure parameters, such as area fractions, using the proposed framework;

5.1. Synthetic data experiments

We conducted a comprehensive performance evaluation of the eight MLPs on the Synthetic test set by reporting the test losses (L1-norm), R^2 scores, and the absolute and relative errors. The average L1-loss between predictions and ground truth for the volume fraction estimation is around 0.01. The absolute errors of more than fifty percent of predictions are less than 0.025, and most relative errors are inferior to 10%. Among the three compartments, the free diffusion volume fraction estimation is better than that of soma and neurite. Most importantly, MLPs' volume fraction predictions are almost unbiased, and their R^2 scores for the three compartments are greater than 0.9. For area fraction estimation, the R^2 scores of the four corresponding MLPs are around 0.6. The soma area fraction estimation suffers large relative errors. Nonetheless, the predictions are still unbiased. These results indicate that the MLPs are good estimators in the Synthetic dataset.

Figure 6b shows that significant errors occur in SANDI's estimation of soma and free diffusion compartment volume fractions. This is not surprising because the soma term $f_{\mathrm{soma}}e^{-D_s b}$ in eq. (19)



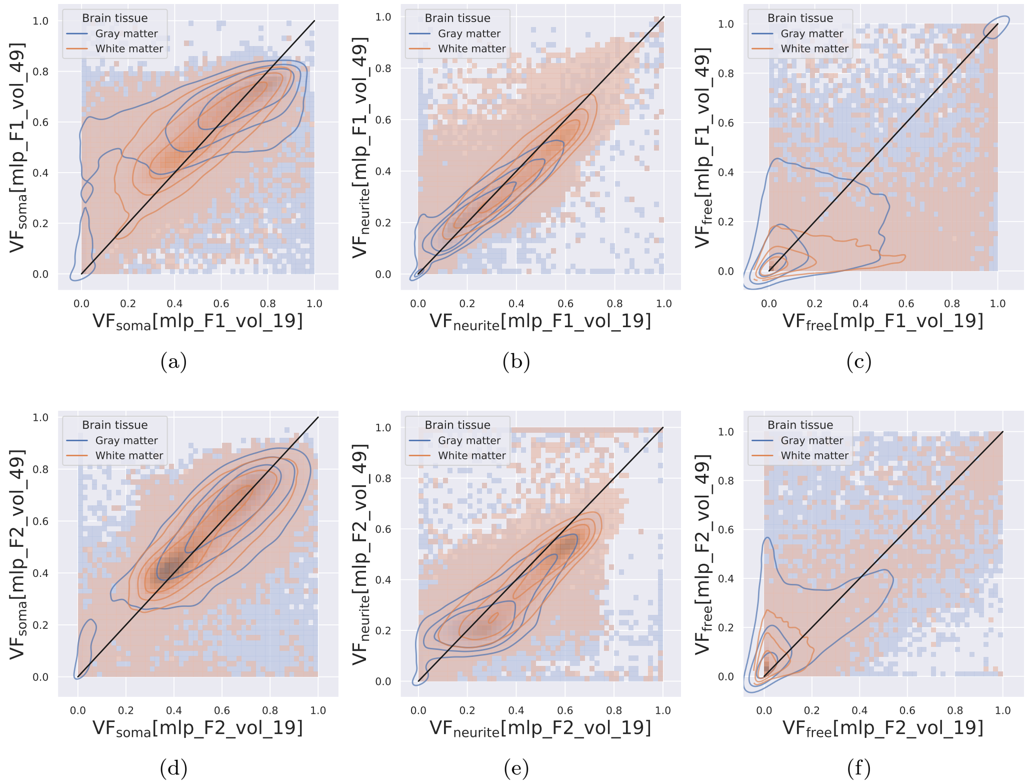
(a) (b) (c)

(d) (e) (f)

Figure 13: The voxelwise joint distribution of f_{soma}, f_{neurite}, f_{free} at two diffusion times. Top row: simulation framework library search. Bottom row: SANDI library search. All brain white and gray matter voxels of sub_002 are included. The x- and y-axes represent the estimated fractions at ($\delta/\Delta = 8/19$ms) and ($\delta/\Delta = 8/49$ms), respectively. The black lines are the identity lines. The contour lines represents 50%, 75%, and 90% of the data.

has the same form as the free diffusion term $f_{\text{free}}e^{-D_{\text{free}}b}$, causing an indeterminancy problem. Based on the sum of the two exponentials alone ($f_1e^{-D_1b} + f_2e^{-D_2b}$), there is no way to tell which exponential belongs to soma and which belongs to the free diffusion compartment. The MLPs' performance in the Synthetic test set suggests they do not suffer from such a problem. We note that in contrast to SANDI, the NODDI model [11] has two diffusion compartments, one is an ECS compartment with a low diffusion coefficient, another is a compartment labeled free water or CSF that has a much higher diffusion coefficient. It seems likely that if either of these compartments has a diffusion coefficient that is close in magnitude to the exponent of the signal term from the spheres, then an indeterminancy can happen.

Finally, concerning the estimation of the soma radius, fig. 7b shows that significant errors occur in SANDI's estimation of $\overline{r}_{\text{soma}}$, compared to the much smaller errors from the simulation framework library search in the Synthetic test set.



(a) (b) (c)

(d) (e) (f)

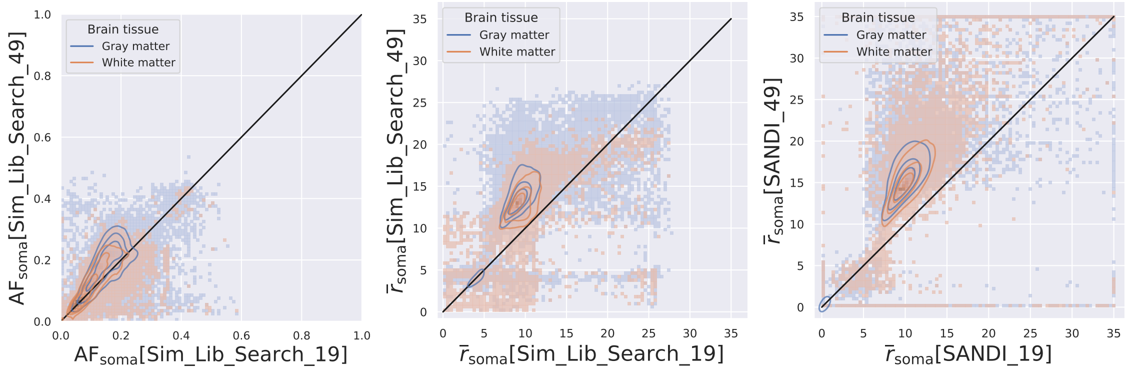
Figure 14: The voxelwise joint distribution of f_{soma}, f_{neurite}, f_{free} at two diffusion times. Top row: `mlp_F1`. Bottom row: `mlp_F2`. All brain white and gray matter voxels of `sub_002` are included. The x- and y-axes represent the estimated fractions at ($\delta/\Delta = 8/19$ms) and ($\delta/\Delta = 8/49$ms), respectively. The black lines are the identity lines. The contour lines represents 50%, 75%, and 90% of the data.

5.2. In-vivo parameter maps

We obtained the parameter maps by applying the MLPs to every brain voxel of a subject in the MGH CDMD. Figure 9 demonstrate the evolution of a parameter map during the model training. The improvement in estimation ability indicates the generalization of the MLPs to both the Synthetic test set and the experimental data.

Figure 10 presents the estimated volume fractions. There are no significant differences between the two sets of features. However, the `mlp_F2`'s parameter maps are less clean than those from `mlp_F1`. This is because the MLPs, `mlp_F2_vol_19` and `mlp_F2_vol_49`, that take the five markers as input are trained with noiseless data. In addition, the computation of the five markers, especially the coordinates of the inflection point, is sensitive to noise. Nonetheless, the five markers still manage to give reasonable volume fraction estimations.

The maps of f_{soma} properly highlight the gray matter and the cerebral nuclei. In contrast, the maps of f_{neurite} are prominent in the white matter, especially the WM tracts located at the corpus callosum, the corona radiata, and the brain stem. The parameter maps of f_{neurite} are similar



(a) a_{soma}, simulation library search. (b) $\bar{r}_{\text{soma}}$, simulation library search. (c) $\bar{r}_{\text{soma}}$, SANDI library search.

Figure 15: The voxelwise joint distribution of a_{soma} and $\bar{r}_{\text{soma}}$ at two diffusion times. Left: simulation library search. Middle: simulation library search. Right: SANDI library search. All brain white and gray matter voxels of sub_002 are included. The x- and y-axes represent the estimated fractions at $(\delta/\Delta = 8/19\text{ms})$ and $(\delta/\Delta = 8/49\text{ms})$, respectively. The black lines are the identity lines. The contour lines represents 50%, 75%, and 90% of the data.

between SANDI library search and simulation framework library search, except SANDI values are slightly higher than the simulation framework.

Compared to SANDI, our new framework can be used to estimate other physically meaningful morphological parameters. The soma area fraction is a new contrast obtained using the proposed framework. The maps of a_{soma} can also properly highlight the gray matter and the cerebral nuclei. The two MLPs that take the 64 signals as input produce cleaner parameter maps. However, the area fraction estimation is inconsistent between the two diffusion times. For example, the high diffusion time map has higher fraction estimations in the gray matter than the low diffusion time map. The moderate performance is also reflected by the low R^2 scores in the Synthetic test set. But the area fraction maps are satisfactory as a proof of concept to demonstrate the potential of the proposed simulation framework for investigating new microstructure parameters.

The above results qualitatively demonstrate that the MLPs trained in the proposed simulation framework can yield encouraging estimations. We further validate the parameter maps by investigating the consistency across diffusion times.

5.3. Independence of diffusion time

Due to the lack of real-world ground truth, validating parameter maps remains largely qualitative. Given this limitation, the community has begun to seek consistency across acquisition parameters, sequences, and scanners [108–110], instead of qualitative visual assessment. In our case, we focus on the dependence of the volume fractions on the two diffusion times. Indeed, microstructure imaging aims to infer the objective tissue properties based on dMRI signals. If the estimated tissue properties largely depend on the acquisition parameters, the estimation interpretation becomes non-trivial.

Figure 10 demonstrates that the parameter maps given by the MLPs are mostly consistent between the short and long diffusion times. For a more quantitative comparison, we plot the voxelwise joint distribution of the estimated soma and neurite volume fractions at the two diffusion times in fig. 13 and fig. 14. If the estimation is consistent, the scatter points should lie around the identity line. It turns out that the estimated f_{neurite} is mainly invariant to the change of diffusion time. The soma volume fraction f_{soma} locates fairly near the identity line with a relatively broad variance. We stress that the cell membrane permeability may not be ignored when $t_d \geq 20\ ms$ [111]. Nonetheless, the MLPs can still give reasonable estimations at the long diffusion time, which suggests that the exchange effect on volume fraction estimation is still minor when $t_d \sim 46\ ms$.

Interestingly, the SANDI model can also give consistent volume fractions estimations even though it is believed to be inapplicable when $t_d \geq 20\ ms$. The consistency agrees with the performance of SANDI in the Synthetic test set regarding volume fractions estimation. In addition, the distributions of the volume fractions are quite similar between the SANDI library search and the simulation framework library search (see fig. 13).

Recently, evidence from experiments and simulations shows that the stick power law is valid in WM and GM [9, 26, 45, 89]. In gray matter, however, the aggregation of neurites and somas modifies the concavity of the direction-averaged signals. Even though the stick power law is not observable in the GM, the direction-averaged intra-neurite signals still follow the power law well, as demonstrated in our previous work [46]. The consistency of the neurite volume fraction estimation further supports the stick power law.

The advantage of the proposed supervised learning framework manifests itself by giving consistent estimations for both neurite and soma volume fractions. It takes years for the community to reach a consensus about the stick power law, while it only takes hundreds of epochs for a MLP to give a similar neurite fraction estimation.

Finally, fig. 15b and fig. 15c show the diffusion time dependence of the simulation framework library search and the SANDI library search in estimating $\overline{r}_{\text{soma}}$. This means that the estimation of soma radius needs to be improved in the future.

5.4. Limitations

The main limitation of the proposed method is the geometrical modeling capacity. To circumvent the difficulty of neuron packing, we used a simplified brain voxel model by assuming that cell membranes are impermeable and there is a free diffusion compartment. Therefore, the ECS model is unrealistic, and the synthetic brain voxel signals may not be accurate at long diffusion time.

In addition, we did not include enough glia in the mesh database. Considering that there are the same numbers of neurons and glia in the human brain [13], adding glia helps achieve a more realistic brain voxel modeling.

Contrary to the biophysical models that could apply to a range of diffusion times, a MLP is for a particular time profile only. If one wants to employ a new diffusion time, the dMRI simulations must be rerun on all neurons. The simulations could be highly time-consuming if one adopts a simulation method other than the numerical matrix formalism. Because the eigendecomposition has already been obtained, the time overhead of computing new signals by the numerical matrix formalism is minimal.

5.5. Future perspectives

The proposed framework can readily help in many aspects of diffusion MRI. First, the Synthetic dataset can be used to verify biophysical models. Second, we have over forty rotationally-invariant microstructure parameters. This paper focuses only on the volume fractions, the area fractions, and the soma radius. It is worth investigating other parameters. Third, we can explore ML models other than MLPs. Fourth, the direction-averaged signals obtained with several acquisition protocols (e.g., various diffusion times or different sequences) can be fed into a MLP together to achieve joint estimation. This could help reduce indeterminacy. The MLP training is the same for diffusion-encoding sequences other than PGSE. However, some biophysical models' signal expressions are derived only for PGSE sequences. If one adopts another type of sequence, it is necessary to re-derive or revalidate the signal expressions. Fifth, the Synthetic dataset can help find other concise signal features less sensitive to noise.

Four extensions to the framework are foreseeable. First of all, we could remove the impermeability assumption. Agdestein *et al.* [53] have extended the numerical matrix formalism to include permeable compartments. We can solve the complete BT equation system, with permeable membranes using numerical matrix formalism. Note that the computational optimization made in this paper also applies to the permeable case. The main challenge is the ECS mesh generation. Second, we could remove the assumption about homogeneous transverse relaxation and employ different compartmental T_2 values. The simulation with transverse relaxation is straightforward because T_2 relaxation just introduces some exponential multipliers to the computation. Third, we can generate more cellular meshes for humans or other species based on myriad neurons stored in NeuroMorpho.Org. Finally, the neuron meshes also contain orientation information. Estimating orientation-dependent microstructure using the Synthetic dataset can also be expected. A further potential application is diffusion-weighted MR spectroscopy, which estimates the diffusion of metabolites. This has the advantage over regular diffusion-weighted MRI because these metabolites are predominantly intra-cellular so one can safely ignore the ECS.

6. Conclusion

We proposed a novel framework leveraging a highly efficient simulator, modern computer graphics algorithms, and supervised learning methods to infer the brain microstructure *in-vivo* using diffusion MRI. The fundamental tools of the framework have been made publicly available. We demonstrated that the framework helps approximate the underlying mappings from diffusion MRI signals to several microstructure parameters. As proof of concept, we present how to estimate volume fractions and area fractions using the direction-averaged signals or the five markers via training MLPs on a synthetic dataset generated by the framework. Qualitatively, the MLPs can give promising parametric maps. Quantitatively, the estimated volume fractions and area fractions are quite robustly independent of the diffusion time. Although the obtained parameter maps still require further validation, we believe the proposed framework can substantially help achieve quantitative microstructure imaging and promote a broader adoption of diffusion MRI simulation.

Acknowledgment

We would like to thank the French National Institute for Research in Computer Science and Automation (Inria) for providing the PhD scholarship of Chengran Fang and all necessary support. We thank Wei Li from Realsee, Beijing, China, for his constructive advice on the neuron mesh generator. We thank Syver Døving Agdestein from Centrum Wiskunde & Informatica, Amsterdam, Noord-Holland, Nederland, for refactoring the SpinDoctor toolbox and suggesting the *expmv* function. Zheyi Yang is financially supported by a PhD scholarship from the China Scholarship Council (CSC).

7. Appendix

7.1. Matrix formalism representation in FE basis

In this section, we briefly describe the matrix formalism (MF) representation of the magnetization in a finite element (FE) basis for solving eq. (1).

7.1.1. Matrix formalism

Consider the eigenvalue problem for the Laplace operator $\nabla \cdot \nabla$ in the compartment Ω_i ($i \in \{1, ..., N\}$) with zero Neumann boundary condition:

$$-\nabla \cdot \nabla \psi(\boldsymbol{x}) = \lambda \psi(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega_i, \tag{20}$$

$$\nabla \psi(\boldsymbol{x}) \cdot \boldsymbol{n}_i(\boldsymbol{x}) = 0, \qquad \boldsymbol{x} \in \partial\Omega_i, \tag{21}$$

where λ is an eigenvalue of the Laplacian and ψ the corresponding eigenfunctions.

Let $\{(\lambda_j, \psi_j)\}_{j \in \mathbb{N}^*}$ be a set of solutions of eqs. (20) and (21). We sort the eigenvalues in non-decreasing order:

$$0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \dots$$

Since Ω_i is a connected domain and subject to the zero Neumann boundary condition, λ_1 is zero and ψ_1 is a constant function [112]. Besides, we only keep the J smallest eigenvalues for numerical purposes.

We decompose the nonrelaxed magnetization φ_i (subject to unit initial conditions) in the truncated Laplacian eigenbasis $\{\psi_j\}_{j \in \{1,...,J\}}$ to get

$$\varphi_i(\boldsymbol{x}, t) \simeq \sum_{j=1}^{J} c_j(t) \psi_j(\boldsymbol{x}) = \boldsymbol{C}(t)^T \boldsymbol{\Psi}(\boldsymbol{x}), \ \boldsymbol{x} \in \Omega_i, \ t \in [0, TE], \tag{22}$$

where $\{c_j\}_{j \in \{1,...,J\}}$ are complex-valued time-dependent coefficients, $\boldsymbol{C}(t) = [c_1(t), ..., c_J(t)]^T$ is the projection of the magnetization to the Laplacian eigenbasis $\boldsymbol{\Psi}(\boldsymbol{x}) = [\psi_1(\boldsymbol{x}), ..., \psi_J(\boldsymbol{x})]^T$.

The magnetization projection satisfies an ordinary differential equation [52, 53]

$$\frac{d}{dt}\boldsymbol{C}(t) = -\left(D_i \boldsymbol{\Lambda} + \underline{\jmath}\gamma f(t) \boldsymbol{A}(\boldsymbol{g})\right) \boldsymbol{C}(t), \ t \in [0, TE]. \tag{23}$$

The matrix $\boldsymbol{\Lambda}$ comprises the eigenvalues in its diagonal, i.e., $\boldsymbol{\Lambda} = diag(\lambda_1, ..., \lambda_J)$. $\boldsymbol{A}(\boldsymbol{g})$ is a $J \times J$ matrix that depends on the magnetic field gradient $\boldsymbol{g}$. The elements in $\boldsymbol{A}(\boldsymbol{g})$ are defined as

$$[\boldsymbol{A}(\boldsymbol{g})]_{mk} = \int_{\Omega_i} \boldsymbol{x} \cdot \boldsymbol{g} \psi_m(\boldsymbol{x}) \psi_k(\boldsymbol{x}) d\boldsymbol{x},\ (m, k) \in \{1, ..., J\}^2. \tag{24}$$

The initial condition related to eq. (23) is

$$\boldsymbol{C}(0) = \int_{\Omega_i} \boldsymbol{\Psi}(\boldsymbol{x}) d\boldsymbol{x} = [\sqrt{V_i}, 0, ..., 0]^T. \tag{25}$$

In general, $\boldsymbol{\Lambda}$ and $\boldsymbol{A}(\boldsymbol{g})$ do not commute, so one cannot integrate eq. (23) unless $f(t)$ is a piecewise constant function. One can approximate an arbitrary time profile by a series of piecewise constant functions [50]. In our case, the time profile of PGSE is a piecewise constant function, so solving eq. (23) does not require discretization in time. For the ease of notation, we define $\boldsymbol{\mathcal{D}} = D_i \boldsymbol{\Lambda}$ and $\boldsymbol{\mathcal{G}} = \gamma \boldsymbol{A}(\boldsymbol{g})$. The projection at time TE is

$$\boldsymbol{C}(TE) = e^{-(\boldsymbol{\mathcal{D}} - \underline{\jmath} \boldsymbol{\mathcal{G}})\delta} e^{-\boldsymbol{\mathcal{D}}(\Delta - \delta)} e^{-(\boldsymbol{\mathcal{D}} + \underline{\jmath} \boldsymbol{\mathcal{G}})\delta} \boldsymbol{C}(0). \tag{26}$$

Substituting eq. (26) into eq. (22), we find $\varphi_i(\boldsymbol{x}, TE)$, thus the solution of eq. (1) as well as the signal attenuation E_i (eq. (6)).

7.1.2. Eigenfunctions in FE basis

The above matrix formalism of the magnetization depends on the known Laplacian eigenstates. However, only simple geometries, such as disks and spheres, have explicit eigenfunctions. For complex geometries, it is nearly impossible to find their analytical eigenfunctions. Li *et al.* [52, 53] proposed a way to numerically compute the eigenstates for complex geometries to make the matrix formalism useful for practical simulation. The idea is to discretize the complex geometries and calculate the eigenstates in the FE basis.

It is worth stressing that, with the zero Neumann boundary conditions (eqs. (20) and (21)), the Laplacian eigenstates depend only on geometries. For a tetrahedral volume mesh, the eigendecomposition only needs to be done once. When the eigenstates are known, one can freely change the diffusion coefficient and the magnetic field gradient (sequence, direction, and intensity) to compute dMRI signals with minimal computational overhead.

7.2. Diffusion MRI simulation with numerical matrix formalism

The numerical matrix formalism is studied by Li *et al.* [52, 53] and implemented in the SpinDoctor Toolbox [34]. This paper focuses on speeding it up and applying the method to neuron meshes to get the dMRI signals. In this section, we first compare the numerical matrix formalism with the finite element method (FEM) to validate the choice of two simulation parameters: spatial discretization and spectrum truncation. Then we show the computational efficiency of the numerical matrix formalism.

7.2.1. Simulation accuracy

The precision of numerical matrix formalism depends on spatial discretization and spectrum truncation. The SpinDoctor Toolbox calls Tetgen [85] to tetrahedralize a surface mesh. A discretization parameter H controls the maximum volume of the tetrahedra. If one sets H to -1, the default discretization routine[4] of Tetgen is triggered. In this study, we use $H = -1$ for the numerical matrix formalism.

The truncation of the Laplacian spectrum depends on the characteristic time scales (see in [52, 53]) of eigenvalues. We only keep the eigenstates that contribute significantly to the signals at the echo time, i.e., $\tau/t_d \gg 0$. We retain the eigenvalues whose time scales are greater than 76 μs.

To validate the choice about H and τ, we compare the numerical matrix formalism with a FEM simulator [34]. We made the following simulation choices for the FEM simulator: $H = 0.5\ \mu m^3$, the relative error tolerance $\varepsilon_{\text{rel}} = 10^{-5}$, and the absolute error tolerance $\varepsilon_{\text{abs}} = 10^{-7}$. We refine the neuron meshes by setting H to 0.5 μm^3. A comparison between $H = -1$ and $H = 0.5\ \mu m^2$ is given in fig. 16. We conduct the FEM simulation on the refined meshes to produce the reference solution.

As for diffusion MRI protocol, the diffusion-encoding sequences are PGSE with $\delta/\Delta = 8/19$ and $8/49\ ms$. The gradient intensity is fixed to the maximum value used in MGH CDMD (290 mT/m) because strong gradient typically suffers large numerical errors [46, 53]. Nine gradient directions are evenly distributed in a semicircle and parameterized by an angle χ. We denote by S_{MF} and S_{FEM} the dMRI signals simulated by the numerical matrix formalism and the finite element method, respectively. Figure 16c gives the relative error in percent ($|S_{\text{MF}} - S_{\text{FEM}}|/|S_{\text{FEM}}| \times 100\%$) at nine directions for three randomly picked cells[5]. The relative errors are below 4%, which validates the choices regarding H and τ for the numerical matrix formalism and the simulation accuracy.

7.2.2. Simulation efficiency

The computational efficiency is crucial for simulating over 1000 neuron meshes. To the best of our knowledge, no other dMRI simulator can finish such a large number of calculations within a reasonable time.

We optimized the eigendecomposition by introducing a shift-and-invert transformation to the matrix eigenvalue problem. To compare the computational efficiency before and after the optimization, we conduct the eigendecomposition for three spheres to find the first 1000 eigenvalues using the *eigs* function [86, 113] implemented in Matlab R2021b. The computation is performed on a computer with 20 physical cores (2 Intel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz), 256GB RAM, running CentOS Stream release 8.

Table 3 lists the computation times before and after the optimization. The optimization achieved a twenty-fold speedup. Moreover, the speedup does not degrade the computational accuracy. We got the same eigenvalues and eigenvectors. A typical neuron mesh has around 80,000 FE nodes and needs about 3,000 eigenvalues. After the optimization, the eigendecomposition for each neuron takes less than 10 minutes.

[4] `https://wias-berlin.de/software/tetgen/1.5/doc/manual/manual005.html#cmd-q`

[5] The IDs of the three cells in NeuroMorpho.Org are NMO_01042, NMO_85592, and NMO_85632.

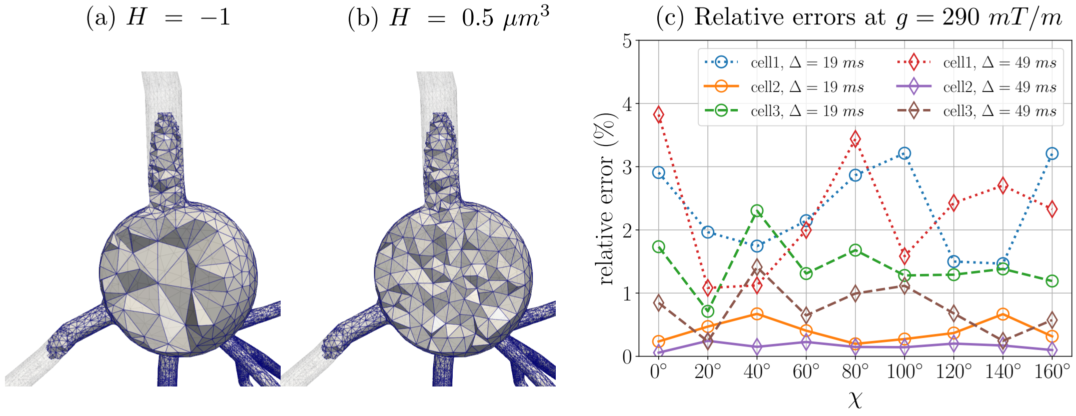


Figure 16: Simulation accuracy of the numerical matrix formalism. (a) a tetrahedral mesh for the numerical matrix formalism. The default discretization routine of Tetgen generates the tetrahedra. Some parts of the mesh are made transparent to show the internal meshing. (b) the refined tetrahedral mesh for the FEM simulator. The maximum volume of tetrahedra is 0.5 μm^3. Some parts of the mesh are made transparent to show the mesh refinement. (c) the relative errors of the numerical matrix formalism for three randomly picked cells with two diffusion times. The gradient intensity is fixed to 290 mT/m. In NeuroMorpho.Org, the IDs of the three cells are NMO_01042 (cell1), NMO_85592 (cell2), and NMO_85632 (cell3). The meshes in (a) and (b) correspond to cell2. When $H = -1$, the numbers of FE nodes of the three cells are 32294, 48551, and 79992, respectively. When $H = 0.5\ \mu m^3$, the numbers of FE nodes are 109660, 80940, and 163905, respectively. The FEM simulations are conducted on the refined meshes to give the reference solution S_{FEM}. The relative errors in percent are $|S_{\mathrm{MF}} - S_{\mathrm{FEM}}|/|S_{\mathrm{FEM}}| \times 100\%$.

sphere radius	5 μm	10 μm	20 μm
number of FE nodes	5222	18981	80191
before optimization	51.48 s	665.49 s	12944.36 s
after optimization	10.93 s	47.86 s	431.97 s
speedup	4.71	13.90	29.97

Table 3: Computation times of eigendecomposition for three spheres. The optimization achieved a twenty-fold speedup.

Another time-consuming step is to calculate the magnetization projection, i.e., $\boldsymbol{C}(TE)$ in eq. (26). The previous implementation utilizes Matlab's built-in function *expm* running on CPUs. We replace *expm* by a GPU-version of *expmv* [87]. We compare the computational efficiency of these two implementations by conducting simulations on ten neuron meshes. The numbers of FE nodes of the meshes range from 30,000 to 150,000. For each cell, we compute the magnetization projection using PGSE sequences with two diffusion times, fifty gradient intensities ranging from 0.5 to 1000 mT/m, and ten gradient directions.

We run the previous implementation ("CPU+expm") on a computer with 64 physical cores (AMD EPYC 7742 64-Core Processor), 512GB RAM, running CentOS Stream release 8. The new implementation ("GPU+expmv") is performed on a computer with 20 physical cores (Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz), 384GB RAM, one Nvidia A40 48G graphics card, using CUDA 11.7, running CentOS Stream release 8.

Figure 17 shows the overall computation time for evaluating eq. (26) for each neuron mesh as a function of its FE node number. The combination of the *expmv* and the GPU computation brings a ten-fold speedup. Additionally, the new implementation does not reduce the computational accuracy.

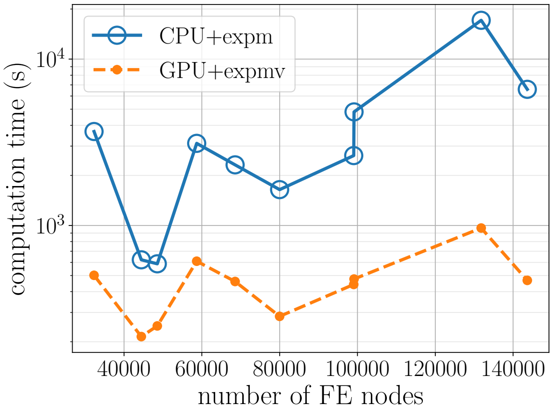


Figure 17: Overall computation time for evaluating eq. (26) with two diffusion times, fifty gradient intensities ranging from 0.5 to 1000 mT/m, and ten gradient directions. The x-axis represents the number of FE nodes. The y-axis in the logarithmic scale shows the computation times. The blue line with circular markers corresponds to the previous implementation that runs Matlab's built-in function *expm* on CPUs. The orange dashed line with dot markers corresponds to the new implementation with a GPU-version of *expmv* function. The combination of the *expmv* and the GPU computation brings a ten-fold speedup.

The generation of 1213 neuron meshes and the dMRI simulations took less than one month in total. It is worth emphasizing that the mesh generation and the eigendecomposition only need to be performed once. Next time if we want to change the magnetic field gradient $\boldsymbol{G}$ or the diffusivity, at least half of the time can be saved. The optimization we explained previously helps accelerate the numerical matrix formalism, making large-scale simulation practical.

7.3. MLP Hyperparameter tuning

We employ four-layer MLPs in this paper. Even though the MLP structure is simple, they still have several hyperparameters, namely, the size of the input layer n, the first hidden layer n_1, and the second hidden layer n_2. To determine the hyperparameters, we split out 20% of the training set $\mathcal{T}_{\text{train}}$ as the validation set $\mathcal{T}'_{\text{v}}$. The remaining eighty percent constitute the new training set $\mathcal{T}'_{\text{train}}$. We compute the validation error using L1-loss on $\mathcal{T}'_{\text{v}}$, similarly to eq. (18).

There are two sets of features: direction-averaged signals $\boldsymbol{F}_1$ and the five markers $\boldsymbol{F}_2$. When the input is $\boldsymbol{F}_1$, the hyperparameters are the size of the input layer, the first hidden layer, and the second hidden layer. We denote them by (n, n_1, n_2). With $\boldsymbol{F}_2$ as input, the size of the input layer is 5. In this case, the hyperparameters are n_1 and n_2. We test the following hyperparameters

1. $\boldsymbol{F}_1$ as input: $(n, n_1, n_2) = (16, 16, 8)$, $(16, 32, 16)$, $(32, 32, 16)$, $(32, 64, 32)$, $(64, 64, 32)$, or $(64, 128, 64)$;

2. $\boldsymbol{F}_2$ as input: $(n_1, n_2) = (10, 5)$, $(10, 10)$, $(20, 10)$, $(20, 20)$, $(30, 15)$, or $(30, 30)$.

We have two sets of output, i.e., volume fractions and area fractions. The four combinations of the input and output (I/O combinations) are listed in section 3.3. Moreover, the two diffusion times are analyzed separately. In total, we trained forty-eight MLPs[6] to find optimal hyperparameters. Figure 18 demonstrates the final validation errors of the MLPs. We see that a more complex network structure usually has better performance. Hence, the selected hyperparameters are $(64, 128, 64)$ for direction-averaged signals and $(30, 30)$ for the five markers.

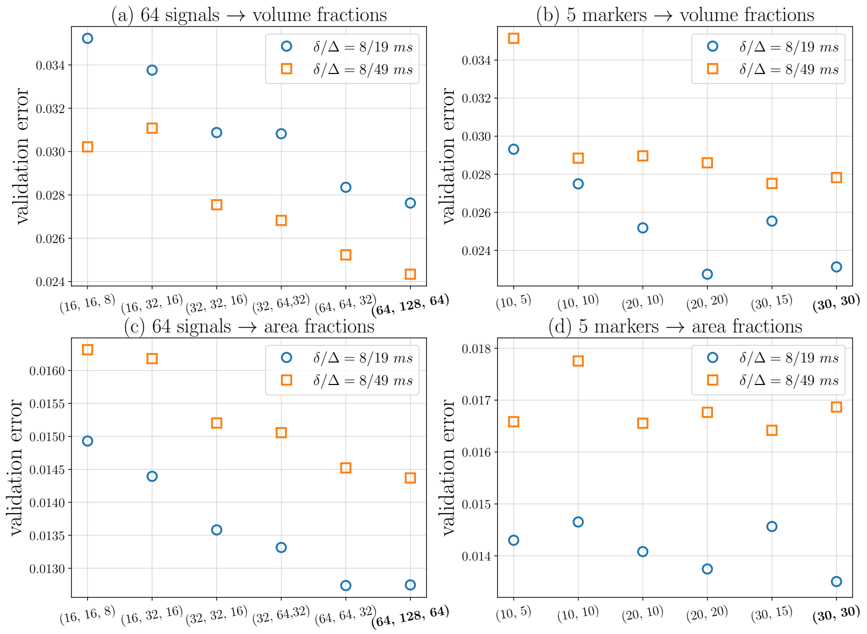


Figure 18: The validation errors for tuning hyperparameters. A blue circle corresponds to a MLP for the short diffusion time. An orange square is for the long diffusion time. The labels of the x-axis are the hyperparameters. The selected hyperparameters are in bold. (a) the validation errors of twelve MLPs whose hyperparameters are shown on the x-axis. The MLPs predict the volume fractions using the direction-averaged signals. (b) the validation errors of MLPs for predicting volume fractions using the five markers. (c) the validation errors of MLPs for predicting area fractions using the direction-averaged signals. (d) the validation errors of MLPs for area fraction estimation using the five markers.

With the chosen hyperparameters, we can now re-train MLPs on the entire training set $\mathcal{T}_{\text{train}}$ following the procedure described in section 3.3. In the next section, we assess the final performance of each MLP in the holdout test set $\mathcal{T}_{\text{test}}$.

7.4. Results of in-vivo parameters estimation using 8 direction-averaged signals

In figures 19-22 we show the analogous parameter maps and joint distributions for the SANDI library search and the simulation framework library search, where we used directly the 8 experimental direction-averaged signals from the MGH CDMD instead of interpolating to 64 signals. The

[6] 2 diffusion times × 4 combinations of input and output × 6 combinations of hyperparameters

simulation framework library search using the 8 experimental direction-averaged signals yields the same results as using 64 interpolated signals for both diffusion times. The results from SANDI using the 8 experimental signals are similar to using 64 interpolated signals at the shorter diffusion time, however, at the longer diffusion time, using 8 experimental signals gives a higher f_{free} compared to using 64 interpolated signals.

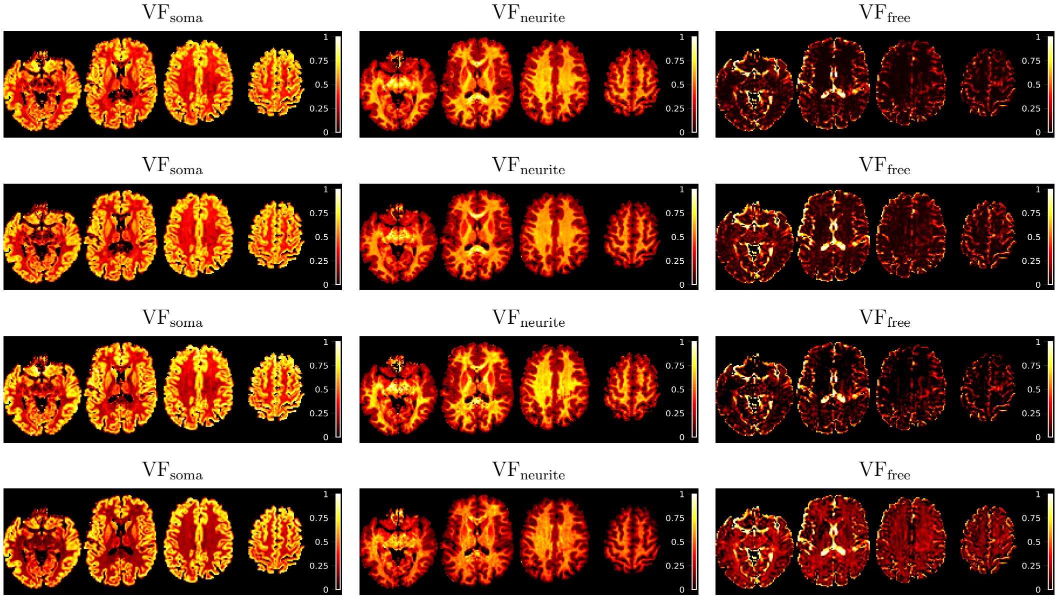

VF_{soma}
$\text{VF}_{\text{neurite}}$
VF_{free}
1
0.75
0.5
0.25
0

Figure 19: The comparison of volume fractions. The first column is for soma volume fraction f_{soma} , the second for neurite, and the third for free. Two rows, (a) and (c), are for the short diffusion time ($\delta/\Delta = 8/19$ ms). (b) and (d) are for the long diffusion time ($\delta/\Delta = 8/49$ ms). (a) (b), (c), and (d), are obtained by respectively applying `Sim_Lib_Search_19`, `Sim_Lib_Search_49`, `SANDI_19`, and `SANDI_49`, to the experimental data from `sub_002`. Obtained from using directly the 8 experimental direction-averaged signals from the MGH CDMD instead of interpolating to 64 direction-averaged signals.

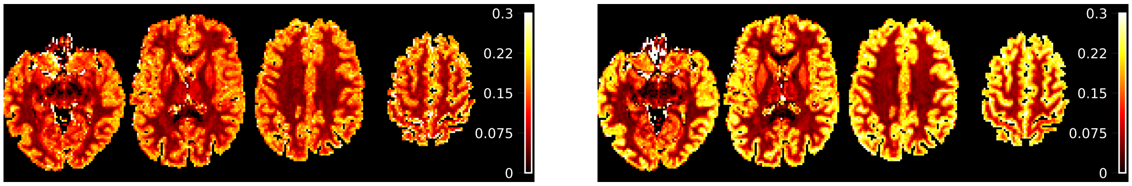
References

[1] E. L. Hahn, Spin echoes, Phys. Rev. 80 (1950) 580–594.

[2] E. O. Stejskal, J. E. Tanner, Spin diffusion measurements: Spin echoes in the presence of a time-dependent field gradient, The Journal of Chemical Physics 42 (1) (1965) 288–292. `doi:10.1063/1.1695690`.

[3] D. L. Bihan, E. Breton, D. Lallemand, P. Grenier, E. Cabanis, M. Laval-Jeantet, MR imaging of intravoxel incoherent motions: application to diffusion and perfusion in neurologic disorders., Radiology 161 (2) (1986) 401–407, pMID: 3763909.

[4] C. Beaulieu, The basis of anisotropic water diffusion in the nervous system–a technical review, NMR in Biomedicine: An International Journal Devoted to the Development and Application of Magnetic Resonance In Vivo 15 (7-8) (2002) 435–455.



(a) `Sim_Lib_Search_19`

(b) `Sim_Lib_Search_49`

Figure 20: The maps of soma area fraction by applying `Sim_Lib_Search_19`, `Sim_Lib_Search_49` to the experimental data from `sub_002`. (a) maps of soma area fraction for the short diffusion time. (b) maps of soma area fraction for the long diffusion time. Obtained from using directly the 8 experimental direction-averaged signals from the MGH CDMD instead of interpolating to 64 direction-averaged signals.

[5] A. T. Van, C. Granziera, R. Bammer, An introduction to model-independent diffusion mri, Topics in magnetic resonance imaging: TMRI 21 (6) (2010) 339.

[6] H. Johansen-Berg, T. E. Behrens, Diffusion MRI: from quantitative measurement to in vivo neuroanatomy, Academic Press, 2013.

[7] D. C. Alexander, P. L. Hubbard, M. G. Hall, E. A. Moore, M. Ptito, G. J. Parker, T. B. Dyrby, Orientationally invariant indices of axon diameter and density from diffusion mri, NeuroImage 52 (4) (2010) 1374–1389.
URL `http://www.sciencedirect.com/science/article/pii/S1053811910007755`

[8] H. Zhang, P. L. Hubbard, G. J. Parker, D. C. Alexander, Axon diameter mapping in the presence of orientation dispersion with diffusion mri, NeuroImage 56 (3) (2011) 1301–1315.
URL `http://www.sciencedirect.com/science/article/pii/S1053811911001376`

[9] J. Veraart, D. Nunes, U. Rudrapatna, E. Fieremans, D. K. Jones, D. S. Novikov, N. Shemesh, Noninvasive quantification of axon radii using diffusion mri, Elife 9 (2020) e49855.

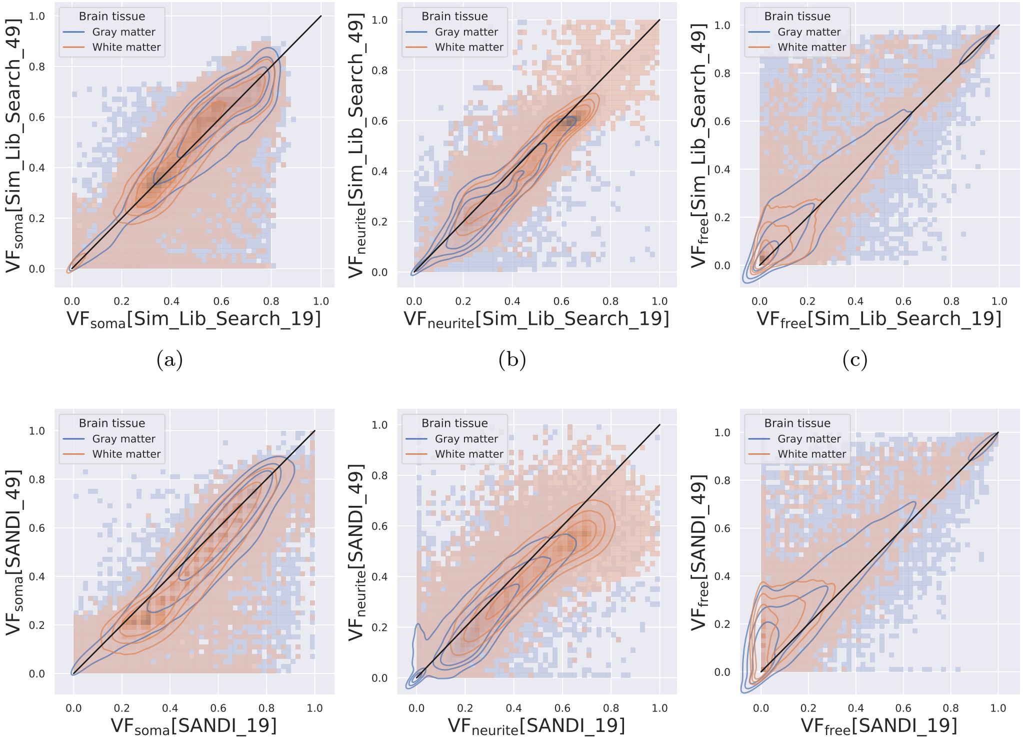
[10] E. Panagiotaki, T. Schneider, B. Siow, M. G. Hall, M. F. Lythgoe, D. C. Alexander, Compartment models of the diffusion MR signal in brain white matter: A taxonomy and comparison, NeuroImage 59 (3) (2012) 2241–2254.
URL `http://www.sciencedirect.com/science/article/pii/S1053811911011566`

[11] H. Zhang, T. Schneider, C. A. Wheeler-Kingshott, D. C. Alexander, Noddi: practical in vivo neurite orientation dispersion and density imaging of the human brain, Neuroimage 61 (4) (2012) 1000–1016.

[12] E. Syková, C. Nicholson, Diffusion in brain extracellular space, Physiological reviews 88 (4) (2008) 1277–1340.

[13] C. S. Von Bartheld, J. Bahney, S. Herculano-Houzel, The search for true numbers of neurons and glial cells in the human brain: A review of 150 years of cell counting, Journal of Comparative Neurology 524 (18) (2016) 3865–3895.

[14] A. Shapson-Coe, M. Januszewski, D. R. Berger, A. Pope, Y. Wu, T. Blakely, R. L. Schalek, P. H. Li, S. Wang, J. Maitin-Shepard, et al., A connectomic study of a petascale fragment of human cerebral cortex, BioRxiv (2021).



(a) (b) (c)

(d) (e) (f)

Figure 21: The voxelwise joint distribution of estimated fractions at two diffusion times. All brain white and gray matter voxels of sub_002 are included. The x- and y-axes represent the estimated fractions at $\delta/\Delta = 8/19\ ms$ and $\delta/\Delta = 8/49\ ms$, respectively. The black lines are the identity lines. From the inner to outer, the contour lines represents 50%, 75% and 90% of the probability mass will lie below. Obtained from using directly the 8 experimental direction-averaged signals from the MGH CDMD instead of interpolating to 64 direction-averaged signals.

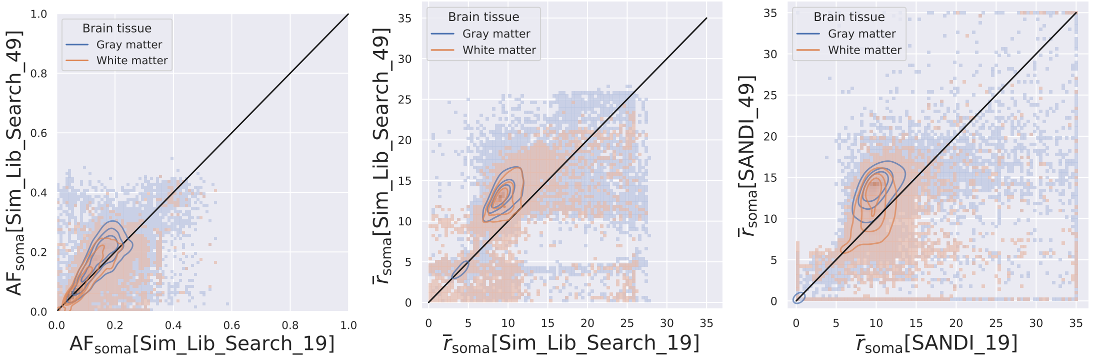
[15] V. G. Kiselev, Fundamentals of diffusion mri physics, NMR in Biomedicine 30 (3) (2017) e3602.

[16] D. S. Grebenkov, Diffusion mri/nmr at high gradients: challenges and perspectives, Microporous and Mesoporous Materials 269 (2018) 79–82.

[17] N. Moutal, K. Demberg, D. S. Grebenkov, T. A. Kuder, Localization regime in diffusion nmr: theory and experiments, Journal of Magnetic Resonance 305 (2019) 162–174.

[18] N. Moutal, Study of the bloch-torrey equation associated to diffusion magnetic resonance imaging, Ph.D. thesis, Institut polytechnique de Paris (2020).

[19] M. Zucchelli, S. Deslauriers-Gauthier, R. Deriche, A computational framework for generating rotation invariant features and its application in diffusion mri, Medical image analysis 60 (2020) 101597.



(a) (b) (c)

Figure 22: The voxelwise joint distribution of estimated effective soma radius at two diffusion times. All brain white and gray matter voxels of sub_002 are included. The x- and y-axes represent the estimated effective soma radius at $\delta/\Delta = 8/19\ ms$ and $\delta/\Delta = 8/49\ ms$, respectively. The black lines are the identity lines. From the inner to outer, the contour lines represents 50%, 75% and 90% of the probability mass will lie below. (a) the distribution of soma area fraction estimated by Sim_Lib_Search_19 and Sim_Lib_Search_49. (b) the distribution of the effective soma radius estimated by Sim_Lib_Search_19 and Sim_Lib_Search_49. (c)the distribution of the effective soma radius estimated by the SANDI model at the two diffusion. Obtained from using directly the 8 experimental direction-averaged signals from the MGH CDMD instead of interpolating to 64 direction-averaged signals.

[20] M. Palombo, A. Ianus, M. Guerreri, D. Nunes, D. C. Alexander, N. Shemesh, H. Zhang, Sandi: a compartment-based model for non-invasive apparent soma and neurite imaging by diffusion mri, NeuroImage (2020) 116835.

[21] D. S. Novikov, E. Fieremans, S. N. Jespersen, V. G. Kiselev, Quantifying brain microstructure with diffusion MRI: Theory and parameter estimation, NMR in Biomedicine 32 (4) (2019) e3998. doi:10.1002/nbm.3998.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nbm.3998

[22] D. S. Novikov, J. Veraart, I. O. Jelescu, E. Fieremans, Rotationally-invariant mapping of scalar and orientational metrics of neuronal microstructure with diffusion mri, NeuroImage 174 (2018) 518–538.

[23] D. S. Grebenkov, Use, misuse, and abuse of apparent diffusion coefficients, Concepts Magn. Reson. 36A (1) (2010) 24–35. doi:10.1002/cmr.a.20152.

[24] N. Moutal, D. S. Grebenkov, The localization regime in a nutshell, Journal of Magnetic Resonance 320 (2020) 106836.

[25] I. O. Jelescu, A. de Skowronski, F. Geffroy, M. Palombo, D. S. Novikov, Neurite exchange imaging (nexi): A minimal model of diffusion in gray matter with inter-compartment water exchange, NeuroImage 256 (2022) 119277.

[26] J. L. Olesen, L. Østergaard, N. Shemesh, S. N. Jespersen, Diffusion time dependence, power-law scaling, and exchange in gray matter, NeuroImage 251 (2022) 118976.

[27] B. Balinov, B. Jonsson, P. Linse, O. Soderman, The nmr self-diffusion method applied to restricted diffusion. simulation of echo attenuation from molecules in spheres and between planes, Journal of Magnetic Resonance, Series A 104 (1) (1993) 17–25.

[28] J. Xu, M. Does, J. Gore, Numerical study of water diffusion in biological tissues using an improved finite difference method, Physics in medicine and biology 52 (7) (Apr. 2007).
URL http://view.ncbi.nlm.nih.gov/pubmed/17374905

[29] J.-R. Li, D. Calhoun, C. Poupon, D. L. Bihan, Numerical simulation of diffusion mri signals using an adaptive time-stepping method, Physics in Medicine and Biology 59 (2) (2014) 441.
URL http://stacks.iop.org/0031-9155/59/i=2/a=441

[30] D. V. Nguyen, J.-R. Li, D. Grebenkov, D. Le Bihan, A finite elements method to solve the Bloch-Torrey equation applied to diffusion magnetic resonance imaging, Journal of Computational Physics 263 (0) (2014) 283–302.
URL http://www.sciencedirect.com/science/article/pii/S0021999114000308

[31] L. Beltrachini, Z. A. Taylor, A. F. Frangi, A parametric finite element solution of the generalised bloch–torrey equation for arbitrary domains, Journal of Magnetic Resonance 259 (2015) 126 – 134. doi:https://doi.org/10.1016/j.jmr.2015.08.008.
URL http://www.sciencedirect.com/science/article/pii/S1090780715001743

[32] G. Russell, K. D. Harkins, T. W. Secomb, J.-P. Galons, T. P. Trouard, A finite difference method with periodic boundary conditions for simulations of diffusion-weighted magnetic resonance experiments in tissue, Physics in Medicine and Biology 57 (4) (2012) N35.
URL http://stacks.iop.org/0031-9155/57/i=4/a=N35

[33] V.-D. Nguyen, J. Jansson, H. T. A. Tran, J. Hoffman, J.-R. Li, Diffusion mri simulation in thin-layer and thin-tube media using a discretization on manifolds, Journal of Magnetic Resonance 299 (2019) 176 – 187. doi:https://doi.org/10.1016/j.jmr.2019.01.002.
URL http://www.sciencedirect.com/science/article/pii/S1090780719300023

[34] J.-R. Li, V.-D. Nguyen, T. N. Tran, J. Valdman, C.-B. Trang, K. V. Nguyen, D. T. S. Vu, H. A. Tran, H. T. A. Tran, T. M. P. Nguyen, Spindoctor: A matlab toolbox for diffusion mri simulation, NeuroImage 202 (2019) 116120. doi:https://doi.org/10.1016/j.neuroimage.2019.116120.
URL http://www.sciencedirect.com/science/article/pii/S1053811919307116

[35] B. D. Hughes, Random walks and random environments, Clarendon Press Oxford ; New York, 1995.

[36] P. A. Cook, Y. Bai, M. G. Hall, S. Nedjati-Gilani, K. K. Seunarine, D. C. Alexander, Camino: Diffusion mri reconstruction and processing (2005).

[37] C.-H. Yeh, B. Schmitt, D. Le Bihan, J.-R. Li-Schlittgen, C.-P. Lin, C. Poupon, Diffusion microscopist simulator: A general monte carlo simulation system for diffusion magnetic resonance imaging, PLoS ONE 8 (10) (2013) e76626. doi:10.1371/journal.pone.0076626.

[38] M. Hall, D. Alexander, Convergence and parameter choice for monte-carlo simulations of diffusion mri, Medical Imaging, IEEE Transactions on 28 (9) (2009) 1354 –1364. doi:10.1109/TMI.2009.2015756.

[39] G. T. Balls, L. R. Frank, A simulation environment for diffusion weighted mr experiments in complex media, Magn. Reson. Med. 62 (3) (2009) 771–778.
URL http://dx.doi.org/10.1002/mrm.22033

[40] K. V. Nguyen, E. H. Garzon, J. Valette, Efficient gpu-based monte-carlo simulation of diffusion in real astrocytes reconstructed from confocal microscopy, Journal of Magnetic Resonance (2018). doi:10.1016/j.jmr.2018.09.013.
URL http://www.sciencedirect.com/science/article/pii/S1090780718302386

[41] C. A. Waudby, J. Christodoulou, Gpu accelerated monte carlo simulation of pulsed-field gradient nmr experiments, Journal of Magnetic Resonance 211 (1) (2011) 67 – 73. doi:https://doi.org/10.1016/j.jmr.2011.04.004.
URL http://www.sciencedirect.com/science/article/pii/S1090780711001376

[42] L. Kerkelä, F. Nery, M. G. Hall, C. A. Clark, Disimpy: A massively parallel monte carlo simulator for generating diffusion-weighted mri data in python, Journal of Open Source Software 5 (52) (2020) 2527.

[43] H. Torrey, Bloch equations with diffusion terms, Physical Review Online Archive (Prola) 104 (3) (1956) 563–565. doi:10.1103/PhysRev.104.563.

[44] S. N. Jespersen, J. L. Olesen, A. Ianuş, N. Shemesh, Effects of nongaussian diffusion on "isotropic diffusion" measurements: An ex-vivo microimaging and simulation study, Journal of Magnetic Resonance 300 (2019) 84–94.
URL http://www.sciencedirect.com/science/article/pii/S1090780719300072

[45] J. Veraart, E. Fieremans, D. S. Novikov, On the scaling behavior of water diffusion in human brain white matter, NeuroImage 185 (2019) 379–387.
URL http://www.sciencedirect.com/science/article/pii/S1053811918319475

[46] C. Fang, V.-D. Nguyen, D. Wassermann, J.-R. Li, Diffusion mri simulation of realistic neurons with spindoctor and the neuron module, NeuroImage 222 (2020) 117198.

[47] J. P. Brito, S. Mata, S. Bayona, L. Pastor, J. DeFelipe, R. Benavides-Piccione, Neuronize: a tool for building realistic neuronal cell morphologies, Frontiers in neuroanatomy 7 (2013) 15.

[48] M. Palombo, D. C. Alexander, H. Zhang, A generative model of realistic brain cells with application to numerical simulation of the diffusion-weighted mr signal, NeuroImage 188 (2019) 391 – 402. doi:https://doi.org/10.1016/j.neuroimage.2018.12.025.
URL http://www.sciencedirect.com/science/article/pii/S1053811918321694

[49] K. Ginsburger, F. Matuschke, F. Poupon, J.-F. Mangin, M. Axer, C. Poupon, Medusa: A gpu-based tool to create realistic phantoms of the brain microstructure using tiny spheres, NeuroImage 193 (2019) 10 – 24. doi:https://doi.org/10.1016/j.neuroimage.2019.02.055.
URL http://www.sciencedirect.com/science/article/pii/S105381191930151X

[50] D. Grebenkov, Laplacian eigenfunctions in nmr. i. a numerical tool, Concepts in Magnetic Resonance Part A 32A (4) (2008) 277–301. doi:10.1002/cmr.a.20117.

[51] D. S. Grebenkov, Laplacian eigenfunctions in nmr. ii. theoretical advances, Concepts Magn. Reson. 34A (5) (2009) 264–296. doi:10.1002/cmr.a.20145.

[52] J.-R. Li, T. N. Tran, V.-D. Nguyen, Practical computation of the diffusion mri signal of realistic neurons based on laplace eigenfunctions, NMR in Biomedicine 33 (10) (2020) e4353.

[53] S. D. Agdestein, T. N. Tran, J.-r. Li, Practical computation of the diffusion mri signal based on laplace eigenfunctions: permeable interfaces, NMR in Biomedicine 35 (3) (2022) e4646.

[54] G. A. Ascoli, D. E. Donohue, M. Halavi, Neuromorpho.org: A central resource for neuronal morphologies, Journal of Neuroscience 27 (35) (2007) 9247–9251. arXiv:http://www.jneurosci.org/content/27/35/9247.full.pdf, doi:10.1523/JNEUROSCI.2055-07.2007. URL http://www.jneurosci.org/content/27/35/9247

[55] M. Kazhdan, M. Bolitho, H. Hoppe, Poisson surface reconstruction, in: Proceedings of the fourth Eurographics symposium on Geometry processing, Vol. 7, 2006, pp. 61–70.

[56] M. Kazhdan, H. Hoppe, Screened poisson surface reconstruction, ACM Transactions on Graphics (ToG) 32 (3) (2013) 1–13.

[57] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle, Mesh optimization, in: Proceedings of the 20th annual conference on Computer graphics and interactive techniques, 1993, pp. 19–26.

[58] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, G. Ranzuglia, et al., Meshlab: an open-source mesh processing tool., in: Eurographics Italian chapter conference, Vol. 2008, Salerno, Italy, 2008, pp. 129–136.

[59] N. Pietroni, M. Tarini, P. Cignoni, Almost isometric mesh parameterization through abstract domains, IEEE Transactions on Visualization and Computer Graphics 16 (4) (2009) 621–635.

[60] M. Garland, P. S. Heckbert, Surface simplification using quadric error metrics, in: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, 1997, pp. 209–216.

[61] P. W. Kuchel, C. J. Durrant, Permeability coefficients from nmr q-space data: models with unevenly spaced semi-permeable parallel membranes, Journal of Magnetic Resonance 139 (2) (1999) 258–272.

[62] S. Merlet, E. Caruyer, A. Ghosh, R. Deriche, A computational diffusion mri and parametric dictionary learning framework for modeling the diffusion signal and its features, Medical Image Analysis 17 (7) (2013) 830–843.

[63] G. Rensonnet, B. Scherrer, G. Girard, A. Jankovski, S. K. Warfield, B. Macq, J.-P. Thiran, M. Taquet, Towards microstructure fingerprinting: Estimation of tissue properties from a dictionary of monte carlo diffusion mri simulations, NeuroImage 184 (2019) 964–980.
URL http://www.sciencedirect.com/science/article/pii/S1053811918319487

[64] D. Karimi, L. Vasung, C. Jaimes, F. Machado-Rivas, S. Khan, S. K. Warfield, A. Gholipour, A machine learning-based method for estimating the number and orientations of major fascicles in diffusion-weighted magnetic resonance imaging, Medical Image Analysis 72 (2021) 102129.

[65] I. Hill, M. Palombo, M. Santin, F. Branzoli, A.-C. Philippe, D. Wassermann, M.-S. Aigrot, B. Stankoff, A. Baron-van Evercooren, M. Felfli, et al., Machine learning based white matter models with permeability: An experimental study in cuprizone treated in-vivo mouse model of axonal demyelination, NeuroImage 224 (2021) 117425.

[66] M. Jallais, P. L. Rodrigues, A. Gramfort, D. Wassermann, Cytoarchitecture measurements in brain gray matter using likelihood-free inference, in: International Conference on Information Processing in Medical Imaging, Springer, 2021, pp. 191–202.

[67] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural networks 2 (5) (1989) 359–366.

[68] A. R. Barron, Approximation and estimation bounds for artificial neural networks, Machine learning 14 (1) (1994) 115–133.

[69] Z. Lu, H. Pu, F. Wang, Z. Hu, L. Wang, The expressive power of neural networks: A view from the width, Advances in neural information processing systems 30 (2017).

[70] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, Q. Liao, Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review, International Journal of Automation and Computing 14 (5) (2017) 503–519.

[71] D. L. Donoho, et al., High-dimensional data analysis: The curses and blessings of dimensionality, AMS math challenges lecture 1 (2000) (2000) 32.

[72] Q. Tian, Q. Fan, T. Witzel, M. N. Polackal, N. A. Ohringer, C. Ngamsombat, A. W. Russo, N. Machado, K. Brewer, F. Wang, et al., Comprehensive diffusion mri dataset for in vivo human brain microstructure mapping using 300 mt/m gradients, Scientific Data 9 (1) (2022) 1–11.

[73] V. Y. Shifrin, P. G. Park, V. N. Khorev, C. H. Choi, C. S. Kim, A new low-field determination of the proton gyromagnetic ratio in water, IEEE Transactions on Instrumentation and Measurement 47 (3) (1998) 638–643.

[74] N. Shemesh, E. Özarslan, P. J. Basser, Y. Cohen, Detecting diffusion-diffraction patterns in size distribution phantoms using double-pulsed field gradient nmr: theory and experiments, The Journal of chemical physics 132 (3) (2010) 034703.

[75] N. Shemesh, C.-F. Westin, Y. Cohen, Magnetic resonance imaging by synergistic diffusion-diffraction patterns, Physical review letters 108 (5) (2012) 058103.

[76] P. T. Callaghan, J. Stepianik, Frequency-domain analysis of spin motion using modulated-gradient NMR, Journal of Magnetic Resonance, Series A 117 (1) (1995) 118–122.
URL `http://www.sciencedirect.com/science/article/pii/S1064185885799597`

[77] M. D. Does, E. C. Parsons, J. C. Gore, Oscillating gradient measurements of water diffusion in normal and globally ischemic rat brain, Magn. Reson. Med. 49 (2) (2003) 206–215. `doi:10.1002/mrm.10385`.

[78] M. H. Blees, The effect of finite duration of gradient pulses on the pulsed-field-gradient nmr method for studying restricted diffusion, Journal of Magnetic Resonance, Series A 109 (2) (1994) 203–209.
URL `http://www.sciencedirect.com/science/article/pii/S1064185884711569`

[79] J. Kärger, Nmr self-diffusion studies in heterogeneous systems, Advances in Colloid and Interface Science 23 (1985) 129–148.
URL `http://www.sciencedirect.com/science/article/pii/000186868580018X`

[80] N. F. Lori, T. E. Conturo, D. Le Bihan, Definition of displacement probability and diffusion time in q-space magnetic resonance measurements that use finite-duration diffusion-encoding gradients, Journal of Magnetic Resonance 165 (2) (2003) 185–195.

[81] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, nature 323 (6088) (1986) 533–536.

[82] G. E. Hinton, Connectionist learning procedures, in: Machine learning, Elsevier, 1990, pp. 555–610.

[83] R. Scorcioni, S. Polavaram, G. A. Ascoli, L-measure: a web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies, Nature protocols 3 (5) (2008) 866–876.

[84] C. Zhang, T. Chen, Efficient feature extraction for 2d/3d objects in mesh representation, in: Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205), Vol. 3, IEEE, 2001, pp. 935–938.

[85] H. Si, Tetgen, a delaunay-based quality tetrahedral mesh generator, ACM Trans. Math. Softw. 41 (2) (2015) 11:1–11:36. `doi:10.1145/2629697`.
URL `http://doi.acm.org/10.1145/2629697`

[86] R. B. Lehoucq, D. C. Sorensen, C. Yang, ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods, SIAM, 1998.

[87] A. H. Al-Mohy, N. J. Higham, Computing the action of the matrix exponential, with an application to exponential integrators, SIAM journal on scientific computing 33 (2) (2011) 488–511.

[88] M. Holz, S. R. Heil, A. Sacco, Temperature-dependent self-diffusion coefficients of water and six selected molecular liquids for calibration in accurate 1h nmr pfg measurements, Physical Chemistry Chemical Physics 2 (20) (2000) 4740–4742.

[89] E. T. McKinnon, J. H. Jensen, G. R. Glenn, J. A. Helpern, Dependence on b-value of the direction-averaged diffusion-weighted imaging signal in brain, Magnetic resonance imaging 36 (2017) 121–127.

[90] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al., Scipy 1.0: fundamental algorithms for scientific computing in python, Nature methods 17 (3) (2020) 261–272.

[91] C. De Boor, C. De Boor, A practical guide to splines, Vol. 27, springer-verlag New York, 1978.

[92] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT press, 2016.

[93] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[94] D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus), arXiv preprint arXiv:1606.08415 (2016).

[95] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (1) (2014) 1929–1958.

[96] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.

[97] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[98] S. J. Reddi, S. Kale, S. Kumar, On the convergence of adam and beyond, arXiv preprint arXiv:1904.09237 (2019).

[99] J. Bridle, Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters, Advances in neural information processing systems 2 (1989).

[100] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).

[101] B. Keil, J. N. Blau, S. Biber, P. Hoecht, V. Tountcheva, K. Setsompop, C. Triantafyllou, L. L. Wald, A 64-channel 3t array coil for accelerated brain mri, Magnetic resonance in medicine 70 (1) (2013) 248–258.

[102] A. F. Howard, M. Cottaar, M. Drakesmith, Q. Fan, S. Y. Huang, D. K. Jones, F. J. Lange, J. Mollink, S. U. Rudrapatna, Q. Tian, et al., Estimating axial diffusivity in the noddi model, bioRxiv (2022) 2020–10.

[103] J. Murday, R. M. Cotts, Self-diffusion coefficient of liquid lithium, The Journal of Chemical Physics 48 (11) (1968) 4938–4945.

[104] A. Daducci, E. J. Canales-Rodríguez, H. Zhang, T. B. Dyrby, D. C. Alexander, J.-P. Thiran, Accelerated microstructure imaging via convex optimization (amico) from diffusion mri data, Neuroimage 105 (2015) 32–44.

[105] D. H. Wolpert, The lack of a priori distinctions between learning algorithms, Neural computation 8 (7) (1996) 1341–1390.

[106] N. G. Gyori, M. Palombo, C. A. Clark, H. Zhang, D. C. Alexander, Training data distribution significantly impacts the estimation of tissue microstructure with machine learning, Magnetic resonance in medicine 87 (2) (2022) 932–947.

[107] M. Palombo, D. Alexander, H. Zhang, Large-scale analysis of brain cell morphometry informs microstructure modelling of gray matter, in: Proc. Intl. Soc. Mag. Reson. Med, Vol. 29, 2021.

[108] U. Ferizi, B. Scherrer, T. Schneider, M. Alipoor, O. Eufracio, R. H. Fick, R. Deriche, M. Nilsson, A. K. Loya-Olivas, M. Rivera, et al., Diffusion mri microstructure models with in vivo human brain connectome data: results from a multi-group comparison, NMR in Biomedicine 30 (9) (2017) e3734.

[109] A. De Luca, A. Ianus, A. Leemans, M. Palombo, N. Shemesh, H. Zhang, D. C. Alexander, M. Nilsson, M. Froeling, G.-J. Biessels, et al., On the generalizability of diffusion mri signal representations across acquisition parameters, sequences and tissue types: chronicles of the memento challenge, NeuroImage 240 (2021) 118367.

[110] S. Coelho, S. H. Baete, G. Lemberskiy, B. Ades-Aron, G. Barrol, J. Veraart, D. S. Novikov, E. Fieremans, Reproducibility of the standard model of diffusion in white matter on clinical mri systems, NeuroImage 257 (2022) 119290.

[111] D. M. Yang, J. E. Huettner, G. L. Bretthorst, J. J. Neil, J. R. Garbow, J. J. Ackerman, Intracellular water preexchange lifetime in neurons and astrocytes, Magnetic resonance in medicine 79 (3) (2018) 1616–1627.

[112] D. S. Grebenkov, B.-T. Nguyen, Geometrical structure of laplacian eigenfunctions, siam REVIEW 55 (4) (2013) 601–667.

[113] G. W. Stewart, A krylov–schur algorithm for large eigenproblems, SIAM Journal on Matrix Analysis and Applications 23 (3) (2002) 601–614.