



DEPARTMENT HEAD

## MSc Thesis Task Description

**István Péter**

candidate for MSc degree in Computer Engineering

# Decoding Neonatal Brain Development: Graph Neural Network Analysis of Multimodal MRI scans

Understanding the neonatal brain is a crucial step in unraveling the complexities of human development. Non-invasive brain imaging techniques (e.g. MRI, PET, EEG) have proven invaluable in this pursuit. One notable initiative pushing the boundaries of our knowledge is the Developing Human Connectome Project, which ambitiously collects MRI scans and additional medical information from neonates aged 20 to 44 weeks post-conception. The third release of this dataset encompasses diffusion MRI scans for structural connectivity and resting-state fMRI scans for functional connectivity. Exploiting this wealth of data, the candidate has to delve into the network-structured information, identifying patterns indicative of various phenotypic factors such as sex, age, and potential disorders.

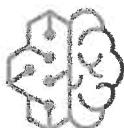
Tasks to be performed by the student include:

- **Conducting** an extensive review of scientific papers, focusing on neonatal brain connectomics, graph neural networks, and explainability methods.
- **Utilizing** algorithms to convert MRI data into graph format suitable for GNN analysis.
- **Implementing** baseline machine learning models using preprocessed data to establish performance benchmarks.
- **Design and implementation** of a GNN tailored for connectome data and **training** the GNN to generate high-quality latent representations.
- **Evaluation** of the GNN's performance in downstream prediction tasks and **comparing** with the baseline models.
- **Identifying** influential regions in the connectome network, using explainability algorithms suitable for GNNs.
- **Preparing** detailed documentation and open-source code on GitHub to enhance transparency, reproducibility, and collaboration within the scientific community.

**Supervisor at the department:** Bálint Gyires-Tóth PhD

**Co-supervisor:** Dániel Unyi

Budapest, 14 March 2024





Budapest University of Technology and Economics  
Faculty of Electrical Engineering and Informatics  
Department of Telecommunications and Artificial Intelligence

# Decoding Neonatal Brain Development: Graph Neural Network-based Analysis of Multimodal MRI Scans

MASTER'S THESIS

*Author*

István Péter

*Advisor*

Bálint Gyires-Tóth, PhD

Dániel Unyi

December 13, 2024

# Contents

<b>Kivonat</b>	i
<b>Abstract</b>	ii
<b>1 Introduction</b>	1
1.1 Motivation and scope . . . . .	1
1.2 The Developing Human Connectome Project . . . . .	2
1.3 Related Work . . . . .	3
<b>2 MRI and preprocessing for Machine Learning applications</b>	5
2.1 Terminology . . . . .	6
2.2 Physical background of the measurement . . . . .	6
2.3 DTI estimation: Fractional Anisotropy and Fiber Orientation Density Function . . . . .	8
2.4 Connectome generation . . . . .	8
2.5 Method . . . . .	9
2.6 Deployment . . . . .	15
2.7 Remarks and possibilities for further work . . . . .	16
<b>3 Problem Formulation and Methods</b>	19
3.1 Data representation . . . . .	19
3.2 Target variables, error functions and metrics . . . . .	21
3.3 Splits and validation . . . . .	21

<b>4 Graph Neural Network Solutions for Prediction of Neurodevelopmental Indicators in Infants</b>	<b>23</b>
4.1 Machine Learning baselines . . . . .	23
4.2 Connectome-based GNNs . . . . .	23
4.3 Model and dataset fusion . . . . .	27
4.3.1 Concepts and design . . . . .	27
4.3.2 Implementation . . . . .	28
4.3.3 Initial results . . . . .	30
4.3.4 Improvements . . . . .	32
4.3.5 Results on the birth age task . . . . .	34
4.3.6 Software and hardware resources . . . . .	35
4.3.7 Deployment . . . . .	35
4.4 Discussion . . . . .	36
<b>5 Explainability</b>	<b>37</b>
5.1 Overview . . . . .	37
5.2 GradCAM for convolutional and graph neural networks . . . . .	38
5.3 Results . . . . .	40
5.4 Conclusion . . . . .	42
<b>6 Summary and Future Work</b>	<b>44</b>
<b>Bibliography</b>	<b>46</b>
<b>Appendix</b>	<b>50</b>
A.1 Self-assessment for Human-Centered Artificial Intelligence Master’s (HCAIM) . . . . .	50
A.1.1 Overview . . . . .	50
A.1.2 Compliance with Human-Centered AI Principles . . . . .	51
A.1.3 Ethics Issues Checklist . . . . .	52
A.1.4 Conclusion . . . . .	52

## HALLGATÓI NYILATKOZAT

Alulírott Péter István, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettettem, egyértelműen, a forrás megadásával megjelöltettem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2024. december 12.

Péter István

Péter István

hallgató

# Kivonat

Az elmúlt fél évszázadban a különböző mágneses rezonanciás képalkotó (MRI) technikák megjelenése forradalmi előrelépést jelentett a nem-invazív anatómiai diagnosztika területén, különös tekintettel az agy szerkezetének és működésének vizsgálatára. A milliméteres felbontású voxelekkel történő 3D képalkotás támogatja a minél pontosabb körkép létrehozását, és felveti az intelligens adatfeldolgozás lehetőségét, betegségek és kóros minták automatikus felismerése céljából. A Developing Human Connectome projekt újszülöttek agyi felvételeit összesítő, különböző MRI-modalitásokat tartalmazó (sMRI, dMRI, fMRI) adathalmaz, amihez az egyes alanyok metaadatait is közölték anonimizált formában. Munkám során az adatokból kinyerhető különféle gráf-reprezentációkat vizsgálva, prediktív gráf neurális háló (GNN) modelleket építetek az alanyok magzati életkorának becslésére a születés, illetve a mérés pillanatában (az adatokban fellelhető, esetleges koraszülöttsgéhez köthető strukturális és aktivitásbeli elváltozások alapján).

A korábbi tanszéki eredményekből kiindulva, ahol csak a strukturális MRI-ből előállítható agyi felületi hálók alapján történt a modellezés, dMRI adatokat bevonva agyi régiók közti kapcsolati reprezentációt (konnektom, mint súlyozott gráf) hozok létre az adatokra szabott előfeldolgozási eljárással. Az így kapott új adatrepräsentációból és az ezekhez tervezett GNN architektúra segítségével vizsgálom az ezekből elérhető predikciós pontosságot, felmérve az új adatforrás információs értékét. Ezen felül vizsgálom az adatok és modellek fúziójának a lehetőségét, vagyis a multi-modális adat egyetlen neurális hálóban való felhasználását egy, erre a feladatra tervezett modell-családon keresztül.

Elemzem továbbá bizonyos modellek magyarázhatóságát is, figyelembe véve a transzparencia és a magyarázható mesterséges intelligencia (xAI) módszerek fontosságát, ha orvosi adatokból történő prediktív modellépítésről van szó.

# Abstract

The emergence of various types of magnetic resonance imaging (MRI) techniques in the last half century enhanced our ability to non-invasively study parts of the human anatomy, especially the brain's structure and activity. Spatial imaging with millimeter-resolution voxels helps the diagnostic process and introduces the possibility of intelligent automation with respect to recognising certain diseases and pathological patterns.

The Developing Human Connectome project gathers a large dataset of neonatal subjects, with scans of different MRI modalities (sMRI, dMRI, fMRI) along with extensive subject metadata. In my work, I explore the various graph-like representations that can be extracted from the data, building Graph Neural Network (GNN) models to predict subject age at birth and scan (based on supposed effects and patterns associated with term/preterm birth in brain structure and activity). I extend upon previous results at the department that use structural surface meshes of the subjects, while incorporating a new kind of data representation from the dMRI scans, along with a custom pre-processing pipeline to extract the connectome (connectivity matrix of brain regions). I explore various GNN models tailored to the resulting data representations, as well as introducing a family of data and model fusion methods to incorporate the strengths of different data modalities in a single neural network. Where relevant, I also assess the explainability of said models, considering the need for transparency and explainable AI (xAI) practices when building predictions on top of medical data.

# Chapter 1

## Introduction

### 1.1 Motivation and scope

The inner working of the human brain is one of the most fascinating phenomena in biology, with neurobiology providing continuous breakthroughs in understanding the structure, activity, and how intelligent behavior emerges from it. Using a recent dataset of detailed brain scans on newborns, the *Developing Human Connectome Project*, I chose to conduct deep learning-based modeling to determine the age of the infant based on the brain image data. I used Graph Neural Networks (GNN) for the task, a relatively recent and ever-emerging family of models, in order to exploit the predictive potential of network-structured data, as GNNs are shown to have inductive biases that help them in learning from non-regular, relational data.

During my work, I implemented a pipeline to create an approximation of the connectivity between functional brain regions, integrating new, recently released data into the process for better accuracy of the representation. I use these graphs as input to my models, evaluating their effectiveness, also showing ways to incorporate the functional connectivity matrices in other predictive models working on different domains (e.g. mesh data representing the brain surface) to improve the predictive power on the age regression task.

With transparency and explainability being crucial when applying machine learning models on medical data, I employ explainable AI (xAI) techniques to interpret the predictive process, assessing the impact of different input features on the model prediction.

## 1.2 The Developing Human Connectome Project

The Developing Human Connectome Project (dHCP) is the collaborative effort of several institutions and neuroscience professionals in creating the largest and most thoroughly validated dataset of perinatal subjects' MRI scans so far.<sup>1</sup> Their work goes beyond the simple acquisition of different types of MRI measurements, they do comprehensive quality control, validation, gathering metadata and following the subjects' neurodevelopmental outcomes in the months following their measurement (a good example is the publication of Bayley-III cognitive and motor scores, used as a proxy for the proper development of infants). Measuring and processing data coming from neonates comes with a set of challenges, with the team assessing their impact and designing custom error correction and noise reduction methodologies presented in several papers<sup>2</sup>. They publish the results of these pre-processing *pipelines* as structured datasets, which can be accessed after registration. The data was released iteratively, with the third data release being the basis of this work. At the time of writing, a fourth release has been made available, consisting of additional *in utero* scans, but which are outside the scope of this thesis.

The data most relevant to my current work is the *diffusional pipeline*, a set of processed dMRI measurements, the acquisition and method of post-processing being described in detail by the team in [1]. This data is the basis of the *connectome* generating process presented in my current work. Although the term *connectome* is used rather vaguely among the publications in the field, denoting some representation of the set of neural connections in the brain, when used in the context of my experiments, I use the term referring to the specific format of data I generate, as described in the later sections, an approximation of the neural connectivity between a set of functional brain regions as defined in brain *atlases*. They define anatomical parcellations of the human brain, but the boundaries of these regions change over the lifetime of a person, so it was particularly challenging for the dHCP team to apply these to neonatal subjects, whose brain structure is in fast development. Fitting this parcellation with the DrawEM method, using regions defined by the ALBERT atlas (containing 87 region labels), is described in [2], the paper accompanying the release of the *structural pipeline*. In my work, I use the brain segmentations defined in this dataset in order to have a more fitting representation of brain structure.

---

<sup>1</sup>Information about the project can be found online at <https://www.developingconnectome.org/> (accessed: 03.11.2024), regarding their methods, mission statement and a comprehensive list of scientific publications authored by the collaborators of the project.

<sup>2</sup>The list of publications by the collaborators can be found at <http://www.developingconnectome.org/dhcp-publications/> (accessed: 03.11.2024).

The target variables for my predictive models come from the metadata of the dHCP dataset. The subjects were scanned one or more times, at least once near the time of birth, at *gestational age* (GA), and once at term-equivalent age, around 40 weeks of *postmenstrual age* (PMA). When selecting the measurement for each task, I follow [6]: the *scan age* task consists of predicting the age of the infant when measured close to birth, while for the *birth age*, we use the term-equivalent measurement to predict GA, essentially relying on markers of premature birth on the term-equivalent state of the infant brain. When there is only one measurement per subject, it is used for both.

### 1.3 Related Work

Kawahara et. al. [3] works with similar data (a smaller set of 168 DTI scans consisting of very preterm infants, unrelated to the dHCP dataset), generating diffusion-based connectomes as input to a GNN, predicting several subject-level attributes, including gestational (GA) and postmenstrual age (PMA), but also later outcomes like Bayley-III cognitive and motor scores, using an architecture called BrainNetCNN (the name is not a typo, the authors refer to this model mostly as *convolutional*, rather than graph neural network, although they use components that later became commonplace in the emerging field of GNN research). Using message passing layers, they are "able to identify an infant's postmenstrual age to within about 2 weeks".

In my approach, after early experimentation with similar, general message passing layers, I opted for an architecture based on Dense Graph Convolutions instead, achieving better results with the latter in terms of convergence and overall accuracy.

The BrainNetCNN paper also explores the explainability of the predictions, assessing the importance of individual connections using partial derivatives of the model with respect to input edges. In my current work, I inspect the feature representations that emerge on a node-level, with the similar, but improved GradCAM method (described in [5]), that also takes into account activation values. My node-level feature importance interpretation complements [3]'s edge-level view of the connectome.

In [4], the authors use an earlier release of the dHCP dataset (524 subjects), predicting GA and PMA (among other neurodevelopmental outcomes) with classical Machine Learning methods like Random Forests and Deep Neural Networks, also using connectomes as input. They outline a similar connectome generating pipeline to mine, but use a different atlas for the parcellation of the brain, that defines 90 regions, while I integrate the ALBERT atlas into the process (consisting of 87 func-

tional brain regions), tailored specifically to neonates, allowing for better results. I also employ Random Forests as a baseline, to demonstrate the advantage of GNN models in tackling the age prediction task due to its inductive biases of prioritizing neighborhood information. The paper also discusses explainability, the authors infer the importance of input edges by iteratively zeroing them out and watching for the magnitude of change in the predicted value.

In [6], structural MRI measurements are used to create surface meshes of the neonatal brains. Their input data is structurally different to mine, and allows for greater fidelity in representing the underlying anatomical structure (as opposed to the probabilistic, reductive and coarse-grained approximation of the diffusional connectome), thus allowing state of the art results in GA and PMA regression, but I succeed in showing that the inclusion of connectomes alongside the mesh data and the design of a single *fused* model can improve their results somewhat. Inspecting the proper way to do the model fusion, and documenting the architectural choices that ensure good performance while avoiding overparametrization is my main contribution to their work. The authors analyze explainability via SHAP value estimation, providing visual representations of the influential regions on the surface mesh. Although conceptually different from my method, there are some brain regions that both their method and mine finds to be highly influential, strengthening my claim of relevant explanation through these congruences.

# Chapter 2

## MRI and preprocessing for Machine Learning applications

In the field of life sciences, magnetic resonance imaging (MRI) encompasses a family of spatial imaging measurements that are used, among else, for inspecting the brain tissue with high resolution, a priceless tool for medical diagnostics. High-level processing and inference based on these are mostly done manually, with automated diagnosis and prediction based on machine learning tools being an emerging field, but severely limited by the data available and the prohibitive cost of acquiring large sets of measurements.

This chapter discusses the theoretical background of the MRI imaging, focusing on the diffusional MRI (dMRI), followed by the description of generating the connectome. An overview of the measurement's background is relevant due to the need to build a custom version of the connectome-generating process. At the time of writing, there is no online or freely distributed tool to automatically generate the connectome matrix that is compatible with the dHCP's diffusional pipeline.<sup>1</sup>

Sections discussing the physical measurement process and the theoretical aspects of connectome generation are based on [8] and [9], the latter providing practical examples along with the steps (using programs from the MRtrix software package, introduced in [14]).

---

<sup>1</sup>The closest match is [7], with implementation to be found at <https://github.com/connectomicslab/connectomemapper3> (accessed: 03.11.2024), but this fails to generate the connectome when provided with the required measurement data, due to the dHCP dataset's folder and file structure being only partially BIDS-compatible.

## 2.1 Terminology

Like all MRI measurements, dMRI produces a three-dimensional contrast image (often visualized as grayscale slices of a three dimensional model) of the brain. The image is composed of voxels, with a scalar value in each voxel typically storing some magnetic excitation-response value. The two abbreviations frequently used in the literature are:

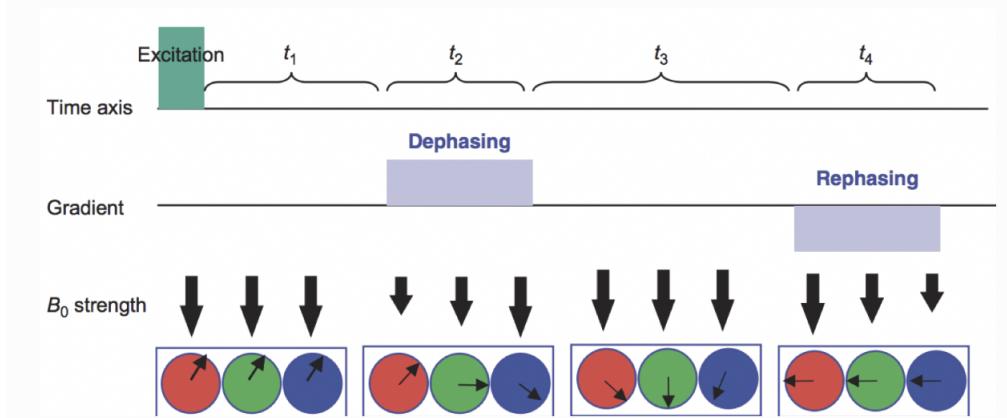
- **DWI**, or *diffusion-weighted imaging*, denotes the measurement process itself, as well as its result, usually containing one scalar value per voxel. The measurement is performed with several settings of the measuring apparatus (determined by different strengths and directions of the magnetic gradient), and these three-dimensional images are then stacked along a fourth dimension, time. Accompanying metadata includes magnetic field data and other parameters for each time slice.
- **DTI**, or *diffusion tensor imaging* is a DWI image-based processing method that aims to assign a tensor (in this case a spatial direction vector) to each voxel. The physical meaning of the vector is the average 'preferred' diffusion direction of water molecules (hydrogen atoms) in the given voxel. It is known that within the cortical white matter, these directions are parallel to the direction of the neural pathways (this is explained by the phenomenon of diffusional anisotropy: for example, in a cylindrical tank, the Brownian motion of particles placed in water has a greater displacement along the principal axis of the cylinder than in the direction perpendicular to the cylinder's mantle). The DTI result therefore no longer includes a time dimension, but instead provides a three-dimensional vector field, at the resolution of the DWI measurement (usually 1-2 millimeters). Each vector gives the point-to-point tangent of the nerve trajectories. It is possible to fit curves to these tangents, starting from pre-defined *seed* points, which can estimate neural pathway bundles in the brain (this estimate is coarse and reductive: a single neuron is way smaller than the spatial resolution of the MRI machines).

## 2.2 Physical background of the measurement

During the dMRI measurement, magnetic coils are used to generate two magnetic gradients in a given spatial direction, of equal strength but opposite sign, generated

in rapid succession. These are the *dephasing gradient* and *rephasing gradient*, while the direction is usually denoted as **bvec**.

The strength of the magnetic excitation is recorded in a *bval* value, considering the magnitude of the gradient and timing characteristics. The effect of the gradient is to produce an inhomogeneous magnetic field along the **bvec** axis, the field strength decreasing when following that direction. A schematic illustration can be seen in Figure 2.1.

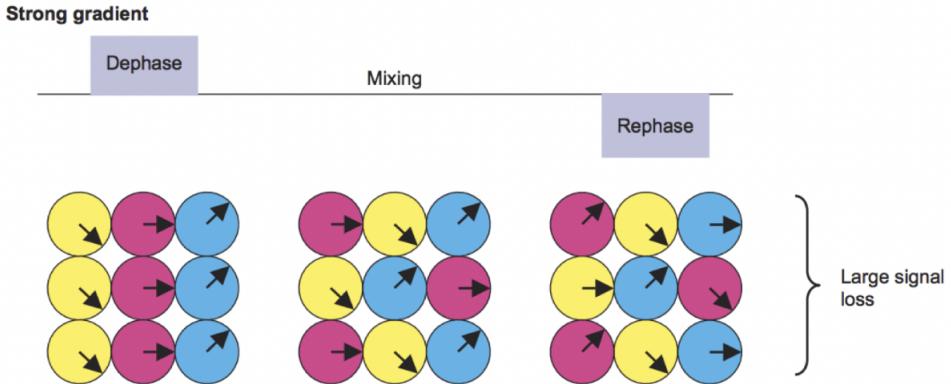


**Figure 2.1:** Variation of the magnetic field strength ( $B_0$ ) in the direction **bvec** during different phases of the dMRI measurement. The arrows indicate the phase of rotation. Source: [13].

This causes a change (proportional to the magnetic field strength) in the rotation frequency of the hydrogen atoms along the axis, leading to a phase difference (hence the name). The opposite-sign gradient has the opposite effect: if we assume that the hydrogen atoms remain stationary between the two steps, they will return to their original phase.

However, in the time between the two magnetic excitations (*mixing phase*), the hydrogen atoms are observed to diffuse and move. Compared to the baseline measurement  $bval = 0$  (without a gradient, equivalent to the T2 image of the structural MRI), if the diffusion was larger in the direction of **bvec**, the power loss of the signal measured in the voxel is also larger (the area is darker during visualization of the dMRI image). Thus, the intensity loss is proportional to the average displacement of the diffusion along the axis. The combined effect of diffusion and the two excitations is shown Figure 2.2.

Based on this, for a given **bvec** direction and a given voxel, we can see how much the measured response is reduced by gradually increasing *bval*: if it is a fast reduction, then the diffusion is larger. Varying the **bvec** as well, we can estimate the magnitude



**Figure 2.2:** The combined effect of the magnetic field and the diffusion on dMRI measurements. Source: [13].

and direction of the *diffusion tensor*. For the DWI, the time series for each voxel is the series of signal responses characterized by different (*bvec*, *bval*) pairs.

## 2.3 DTI estimation: Fractional Anisotropy and Fiber Orientation Density Function

There are several methods to estimate the diffusion tensors from DWI. The *Fractional Anisotropy* (FA) method gives a point estimate (meaning one single vector for each voxel), but a major drawback is that, if there are two axon bundles crossing in a voxel, we lose information about at least one of them, even if the original DWI shows multiple high diffusion directions.

Probabilistic methods help in this regard, with the approximation of the *Fiber Orientation Density Function* (FODf). In this case, each direction vector has a probability value (density function) associated to it. A specific method to estimate the FODf is the *Spherical Deconvolution*<sup>2</sup>, where each tissue type has a different basis function based on electro-magnetic properties.

## 2.4 Connectome generation

Given the FODf for each point (voxel), the connectome is generated by first segmenting the gray matter and white matter, determining the boundary surface between the two. Then, from seed points on this boundary, spatial curves are generated in-

---

<sup>2</sup>This step is implemented in the MRtrix package, introduced in [14]. I used this for creating the connectomes of the dhCP dataset.

side the white matter that correspond to high probability neural pathways in terms of FODf, meanwhile filtering anatomically unlikely results. The resulting proposals are the *streamlines*, the results of the *Anatomically Constrained Tractography*.

Based on the structural MRI images<sup>3</sup>, the brain volume needs to be parsed into *nodes*, relevant regions, using anatomical atlases. The dHCP dataset provides such parcellation with 87 nodes, for each subject. This is based on a custom atlas tailored to perinatal subjects, but in general, multiple choices are available depending on granularity, hierarchy etc.

Once the regions are defined, the streamlines are grouped by start and end points, so that the set of streamlines running between nodes  $i$ . and  $j$ . will define the weight  $c_{i,j}$ .

The only remaining design decision is to choose aggregation over the streamline groups to obtain a usable  $c_{i,j}$  metric. In the current work, 4 methods were explored, as suggested in [9] to be the most common: average streamline length within group (ln), average FA (as discussed previously) value along the paths (fa), weight sum along the streamlines (ws) and the same sum normalized by being divided with the volume of the nodes (nws). The weights being summed up come from a streamline refining algorithm implemented in MRtrix, that offsets biases arising from under- and over-representation of streamlines in certain regions.

## 2.5 Method

Generating connectomes from the dHCP data required the construction of a custom pipeline. Individual steps follow the general outline detailed in [9], with certain deviations adapted to the data set at hand. The implementation can be found in the related repository<sup>4</sup>, the rest of the section details the usage of MRtrix utilities in the code. Providing visual, flowchart-like explanations of the steps, I aim to accompany and explain the code. Details are presented as a report on the individual work done in the scope of this thesis, namely: understanding the steps from practical examples and related literature, as well as assembling the whole pipeline. This section can be seen as an expansion of the previous one by relating the steps that were outlined generally to the code itself.

The first step involved processing the dMRI data, starting from the DWI scans provided in the dHCP data release, the accompanying *bvec* and *bval* time series, as

---

<sup>3</sup>T1 images in the anat subfolder of the anatomical pipeline for dHCP data

<sup>4</sup><https://github.com/peter-i-istvan/msc-thesis-preprocess> (accessed: 13.12.2024)

well as the brain mask that indicated which voxels correspond to valid brain tissue at each time step. The outline of the process is seen in Figure 2.3, resulting in the creation of the estimated Fiber Orientation Density Function (FODf) for each space point (discretized voxel) inside the white matter.

Meanwhile, structural MRI data is also processed by conducting five tissue type (5TT) segmentation from the T1-weighted scans of the dHCP anatomical pipeline, as seen in Figure 2.4. The structural MRI is a less noisy representation of the brain's anatomy, as opposed to its diffusional counterpart, this being the reason for its inclusion as input to this step.

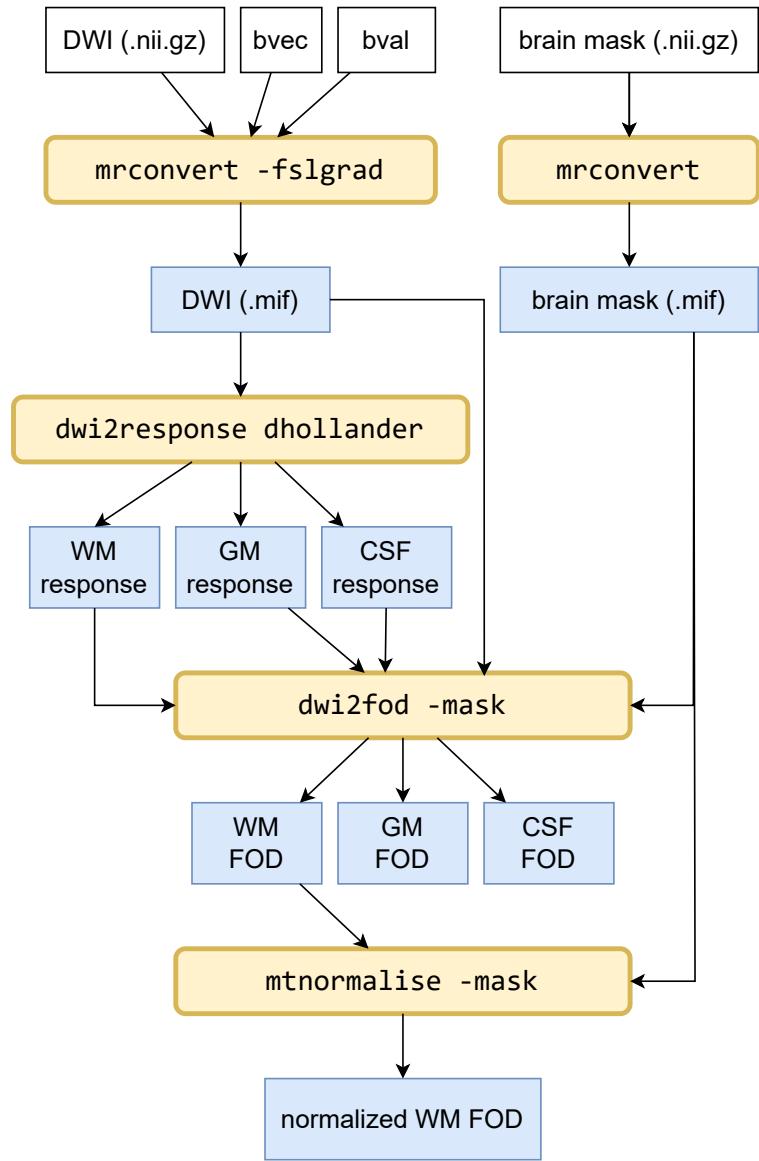
The two representations are consolidated in the co-registration step, where a spatial mapping is estimated between the sMRI and dMRI measurement spaces, transforming the 5TT segmentation into the dMRI domain, as seen in Figure 2.5. An additional dependency of this step is the FSL<sup>5</sup> software package.

Finally, the Fiber Orientation Density Function and the dMRI-aligned 5TT segmentation is combined in the connectome generation step (Figure 2.6), where Anatomically Constrained Tractography is conducted from seed points on the white matter-grey matter boundary, along with streamline weight adjustment and aggregation based on regions. The results after different aggregations can be seen in Figure 2.7.

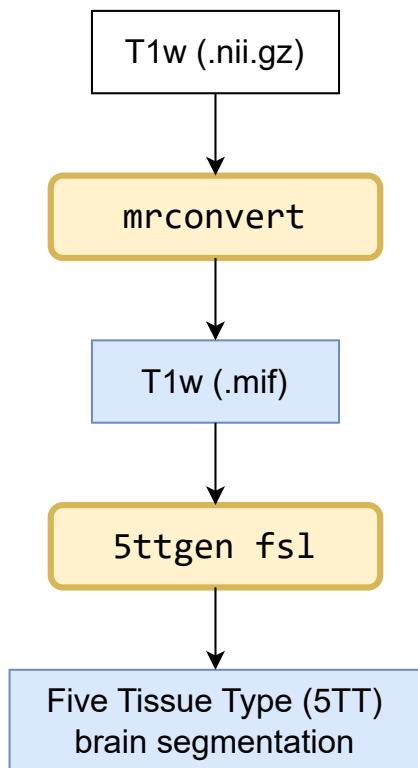
The code has additional options for visualizing intermediate results, to self-check between these stages, but these options are disabled by default to facilitate automation.

---

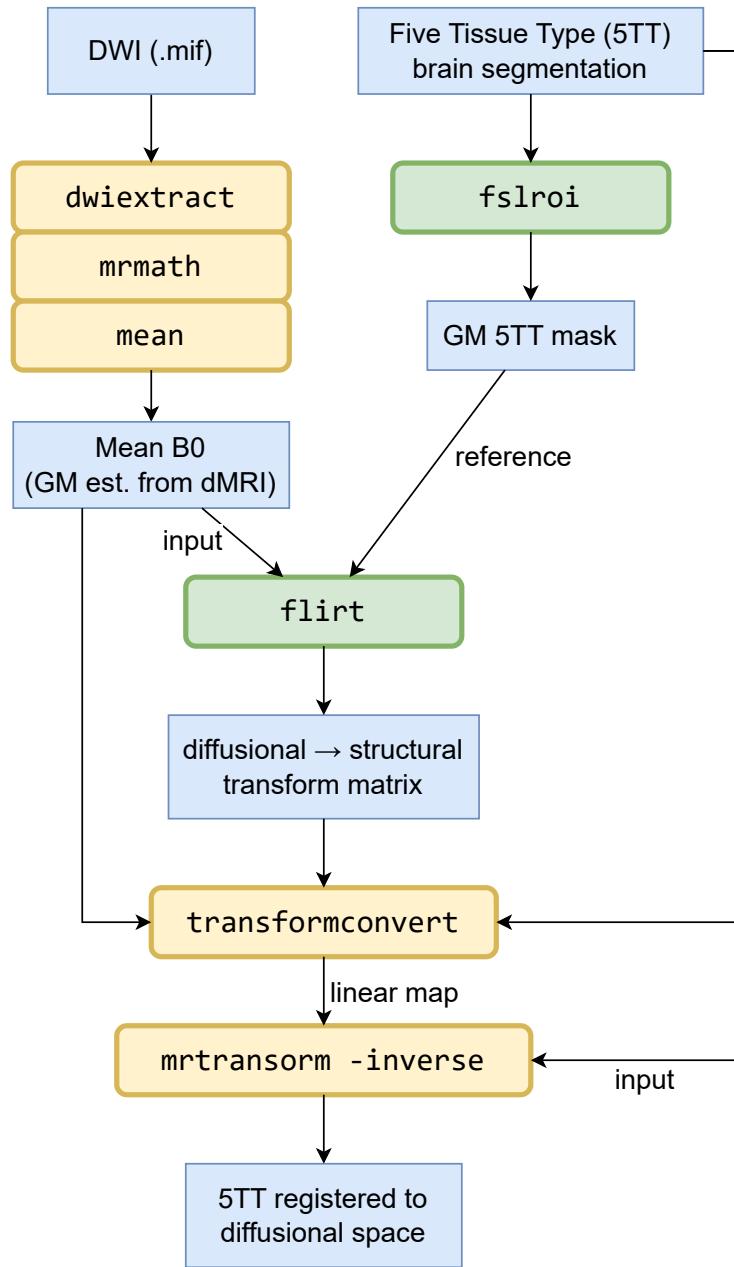
<sup>5</sup>For further information, see: <https://fsl.fmrib.ox.ac.uk/fsl/docs/#/> (accessed: 03.11.2024)



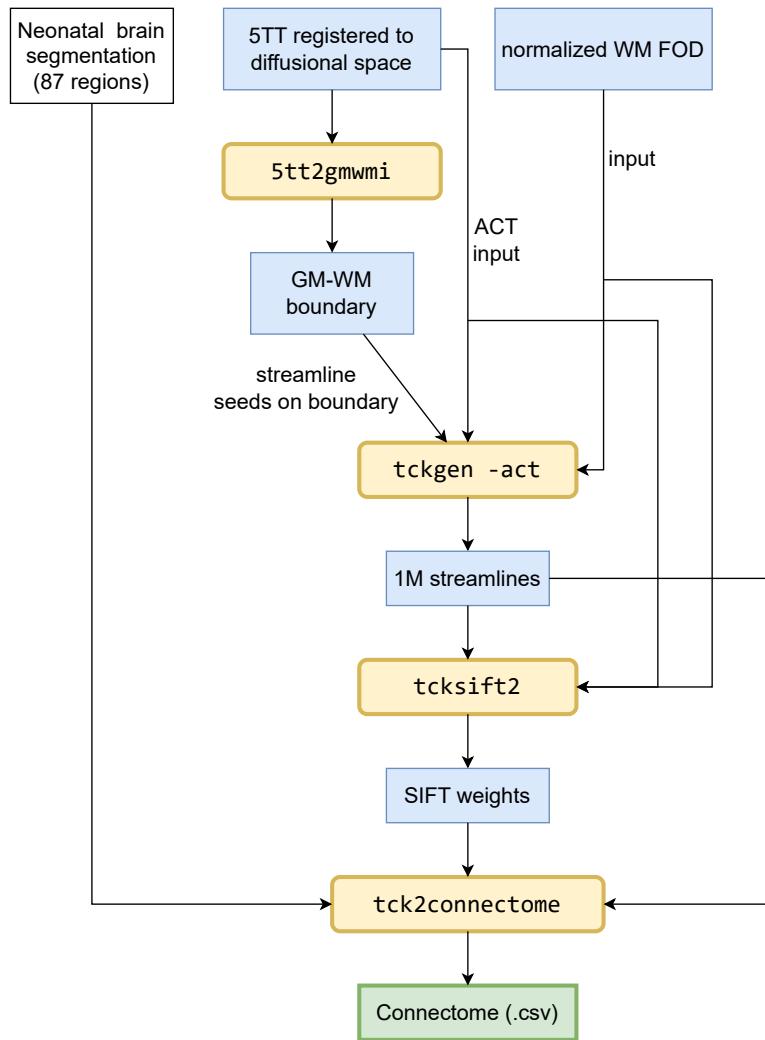
**Figure 2.3:** The implemented pre-processing pipeline’s DWI-related steps. The yellow elements denote the commands provided by the MRtrix package, while the blue ones show intermediate results saved to disk. The main goal of this step is the generation of the Fiber Orientation Density (FOD) function via Spherical Deconvolution (using the *Dhollander* algorithm for estimating the basis functions), as discussed in previous sections.



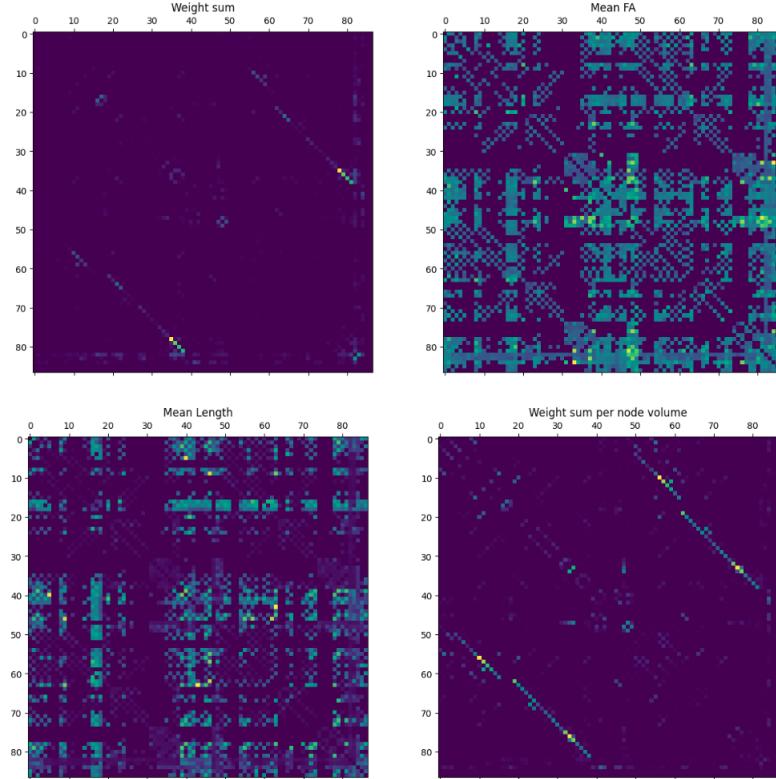
**Figure 2.4:** The implemented pre-processing pipeline's structural (anatomical) MRI-related steps. The main goal of this step is the five tissue type segmentation of the brain (grey matter, subcortical grey matter, white matter, cerebrospinal fluid and pathological tissue), based on the T1-weighted structural MRI scans, used later in the Anatomically Constrained Tractography (ACT) step as bounds to filter out biologically unlikely streamlines.



**Figure 2.5:** The implemented pre-processing pipeline’s co-registration step. The green elements are command line utilities from FSL, an optional dependency of the MRtrix package that must be installed separately. This step registers the 5TT image in the structural MRI space to the diffusional MRI space. This is done by estimating a transform based on the grey matter position as seen on the 5TT image, and on the B0 DWI image respectively, then applying it to the segmentation.



**Figure 2.6:** The final step of the connectome generation, using the 5TT image transformed in the co-registration step and the normalized white matter FOD from the DWI pre-processing, as well as the neonatal brain parcellation into 87 regions provided by the dHCP team. The tckgen utility uses Anatomically Constrained Tractography for streamline estimation (1 million streamlines), then filtering these streamlines (SIFT) to remove regional under- and over-representation biases, then aggregating to produce the connectome, as a connectivity matrix.



**Figure 2.7:** An example connectome extracted from a measurement of the Developing Human Connectome Project. During the generation, one of four aggregation methods can be chosen: ws, nws, ln, fa.

## 2.6 Deployment

The code in the aforementioned repository has several dependencies which are tricky to install, MRtrix requires an additional Anaconda environment (the recommended usage, according to the official documentation), as well as separate installation of the FSL utility. This necessitates a simple way to ship and deploy the implementation, thus using containerization is recommended for ease of use. I opted for Docker as a widespread and industry-standard solution.

Building the Docker image and running the container with the necessary options (e.g. bind mounts pointing to the dHCP data itself, as we assume the user has acquired the dHCP dataset on their own) is described in the *readme* section of the repository.

## 2.7 Remarks and possibilities for further work

Concluding this chapter, I will summarize by reflecting on the previously presented process, putting it into context and shedding light on the possible errors, caveats and ill-defined sub-problems inherent to this family of procedures. I also study the possibility of future work through integrating other MRI modalities, namely fMRI, but motivating its absence from the scope of this thesis.

In [8], the authors provide a thorough review of the literature about diffusional connectome generation. Rather than focusing on a specific choice of steps, it delineates an abstract process and its necessary stages, then presenting the set of specific ways known to literature on how to tackle these sub-problems. It also addresses the possible shortcomings of said sub-process, or the problem formulation itself.

One of the disadvantages of the connectome model is its non-completeness. According to the review, only 10% of neural pathways in the brain are *extrinsic* (white matter tracts going outside and between grey matter regions), the rest are *intrinsic* (located mostly inside the given grey matter region), meaning the model provides only a limited view of the neural connectivity landscape. The problem of regional segmentation (referred to as *node delineation*) is error-prone even when a proper anatomical atlas is used, mostly due to in-population variations in the volume and extent of certain functional regions of the subjects' brain. Regarding the Fiber Orientation Density function, one common disadvantage of the family of deconvolution methods (as used during my implementation) is the forced spatial (point-wise) symmetry of the supposed density function, although this probabilistic method is still preferable as opposed to the deterministic, but overly simplified single diffusional tensor estimation. The latter approximation breaks in the case of crossing fibres inside a voxel. In addition, the difference in size of the scan resolution (1-2 mm at best) and the neuron scale is considerable, so exact mapping from axon bundles to voxel-wise directional vectors (tensors) is reductive and coarse at best, ill-defined at worst. The biggest problem around the validation of the overall result (the weighted graph) itself is, however, the lack of a clear *ground truth* connectivity map, which makes the accuracy of different methods hard to compare.

The pitfalls and sources of noise are presented assuming adult subjects, but an added factor that can cause potential errors in the scope of the current work is the early age of the measured infants. Humans of perinatal age show exponential development of neural connectivity, but the lack of long-established bundles of neural connection (white matter) causes weaker signals of diffusion among the omnipresent noise in the surrounding tissue. For this reason, when using the connectomes in a machine

learning setting as a predictor of certain target variables, we must assume a large amount of observational noise is present in the input data.

Including functional MRI (fMRI) in the study of connectomics, and adding it as another input to a potential modeling process (including prediction of certain outcomes) can offset these issues by incorporating new sources of information, thus it is worth studying in general, but for the scope of the current thesis, fMRI was omitted due to the negative impact on sample size (very few publicized dHCP scans were properly pre-processed for such an application), and thus the significance of results.

For a potential study of the inclusion of fMRI in the task of perinatal brain modeling (and possibly age prediction), a good starting point is [10], where the authors present the processing framework used in the dHCP project for collection and publication of resting state fMRI data. Resting state refers to the fact that the subjects are not completing a specific task, the recorded brain activity is the spontaneous "firing" of neuron groups (which also map out the connectivity of the brain pretty well). The measured activity itself is a blood oxygen level dependent (BOLD) imaging. Each scan is a time series of voxel intensity values, and custom motion correction algorithms were developed by the team to fix warps due to the infant's head movement. By this process, a given voxel ends up to represent roughly the same volume segment, and its activity can be referenced against other regions' time series, with the goal of finding temporary correlations and thus hypothesizing a neural connection between the two regions. The only drawback is the aforementioned scarcity of available data: the process was conducted and results publicized on just a subset of 40 subjects, as the algorithm was computationally expensive to run, even on decent hardware.

Assuming properly motion corrected and otherwise good-quality data, connectome generation can be tackled in several ways: [11] describes a sophisticated case study based on the previously mentioned 40 subjects' corrected scans. The result is a weighted graph (connectivity matrix) similar to the previously seen diffusional connectomes. Windowed correlations were computed across the aggregated time series of regions of interest (ROIs), similar to the parcellation used for dMRI analysis. Correlations and other statistical descriptors get summarized in a single functional connectivity (FC) metric used as the weight function of the graph. The study does not use the data for prediction, but subjects it to statistical analysis and cross-references the supposed connectivity scores with anatomical and biological properties. Nevertheless, the small sample size hinders its application in proper machine learning use cases.

In contrast, [12] provides a more general overview of resting-state fMRI based functional connectomics. It introduces usage of partial correlations and Independent Component Analysis (ICA) for BOLD time series processing, along with a hierarchical decomposition of brain regions. It also presents a stronger graph theoretical and probabilistic overview of how we can infer causation (activity in region A causing activity in region B through neural impulse transmittance) from the observed data. This article, along with the previously mentioned ones, gives a good introduction and a strong basis for further work if and when large scale, properly motion corrected fMRI data gets released to accompany the other scan modalities in the Developing Human Connectome Project.

# Chapter 3

## Problem Formulation and Methods

### 3.1 Data representation

Graph neural networks (GNNs) are suitable for a prediction task if there exists a well-defined representation of the input variables in the form  $G = (V, E)$ . In addition to describing structure, node-level feature vectors ( $\mathbf{x}_u$ ), edge-level features ( $\mathbf{e}_{uv}$ ) and edge weights ( $c(u, v)$ ) can be defined and used in various GNN architectures. However, depending on the specific task, some elements may be missing, and in general the paradigm does not require that the modeled graphs have the same number of vertices or edges. Most of the mathematical operators or *layers* used in GNNs are invariant with respect to the order of neighbors of vertices, often performing aggregation from vertices towards edges or vice versa (these are the so-called *message passing* layers), and convolution and pooling operations can be generalized to graphs.

For the present problem and dataset, a natural representation of the input is a  $G = (V, E)$  graph with  $V = \{1, 2, \dots, 87\}$ , where the numbers identify brain regions as defined in [2].  $G$  is a complete graph for which a real-valued weight function  $c(u, v)$  is known. As indicated by [9], the default aggregation to be used in determining  $c$  is the normalized weight sum (nws) connectome (experiments with the GNN architectures explored also show that generally, this representation yields the best results of the 4). An additional restriction most common GNN operators assume is that  $c$  is strictly positive, which happens to be the case.

The connectome has no clear node feature representation  $\mathbf{x}_u$ . While this would not be a problem in itself, most published, popular and proven GNN layers have

a mandatory  $\mathbf{x}$  parameter, therefore such initial representation must be created. These representations are called *positional encodings*, with the following examples:

- **All ones.** A vector of  $F$  dimensions, initialized to all 1-s (with  $F$  to be set freely).
- **One-hot encoding.** The vertex  $i$ . is assigned the vector of the standard basis  $\mathbf{e}_i$ . The advantage is that the input features will all be unique, part of an orthonormal basis. The disadvantage is that we will have sparse vectors of 87 elements, which keeps the input dimensionality and the number of parameters of the first layer high, also not adjustable.
- **Spectral basis.** Eigenvectors of the Laplacian,  $\mathbf{L} = \mathbf{D} - \mathbf{A}$ , where  $\mathbf{D}$  is a diagonal matrix with the degrees of the vertices in its principal diagonal and  $\mathbf{A}$  is the adjacency matrix of the graph. We select the first  $K$  of the eigenvectors associated with eigenvalues  $0 = \lambda_1 \leq \lambda_2 \leq \dots$ , and assign them to the vertices in this order.
- **Local Degree Profile.** Assigns a descriptor to each vertex based on its degree and its neighbors according to the following formula ( $DN(i) := \{\deg(j) \mid j \in \mathcal{N}(i)\}$ ):

$$\mathbf{x}_i = \mathbf{x}_i \| (\deg(i), \min(DN(i)), \max(DN(i)), \text{mean}(DN(i)), \text{std}(DN(i)))$$

The connectome format has the advantage that the vertices have a well-defined identity (the cortical region), unlike other structures, like brain surface meshes. Thus the connectome obtained by permuting the vertices is not equivalent to the original. For many modeling tasks formulated on graphs (e.g. social nets, spatial point sets), the labeling of input vertices is sequence-invariant (in large social nets, point sets, etc., we cannot always distinguish between individuals, often it is only the structure of the neighborhood relations, the topology that is important: cliques, vertices with high degree, etc.). This is not the case here, useful in the sense that it allows us to give a fixed meaning to the vertices so we can *flatten* the matrix and interpret it as a vector input to a classical machine learning method, and have a well-defined problem. This shows that, in principle, traditional machine learning methods have a chance to succeed due to the well-defined input, although the properties of the graph nature (e.g.  $c(i, j)$  and  $c(j, k)$  meet at a common vertex, or they numerically characterize the relationship between  $i$  and  $k$ ) cannot be easily expressed in this flattened format (due to each model's *inductive biases*). Classical ML methods will be used as baselines, but this shortcoming should be kept in mind.

## 3.2 Target variables, error functions and metrics

As discussed previously, the target variables in the scope of the current work are the *scan age* (PMA<sup>1</sup>), from an MRI scan taken preferably as close to the time of birth as possible, and the *birth age* or gestational age (GA) based on the term-equivalent measurement. Other metadata is available as well, but in the scope of the current thesis, these two regression targets will be used.

For these tasks, I used mean squared error (MSE) during training, as it is the most widely used error function for regression. During validation I also use mean absolute error (MAE), correlation and the coefficient of determination ( $R^2$ ) as accompanying metrics.

## 3.3 Splits and validation

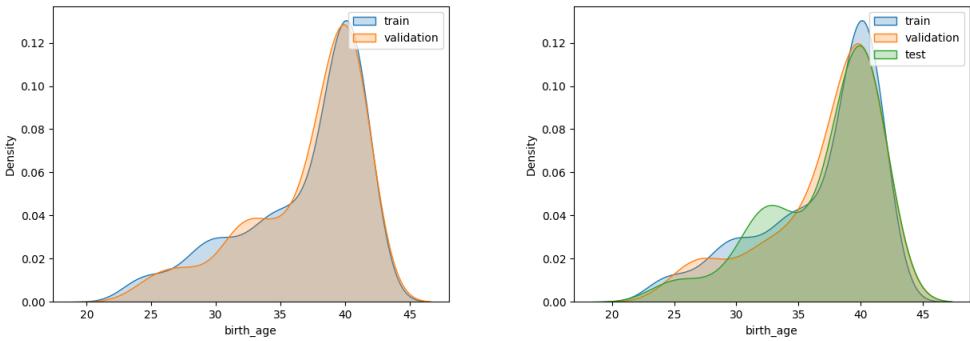
When working with only connectome data, the 674 subjects (with one single measurement being chosen for each, as defined by the target variable) were separated initially an approximately 70%-30% train-test split (471 and 203 samples respectively). Considering the relatively small dataset, two splits is an adequate way to provide both a reasonable amount of training data and a test/validation score that is significant enough (in the sense that is based on enough data). This split was used when hyperparameter-optimization or early stopping was omitted, as in both cases, validation scores are used for selecting best model candidate, which should be tested on a third, independent set as well to ensure no over-fitting. In such cases when a third, test set was needed, the smaller split was further separated into two subsets (101 and 102 samples, about 15% of the whole). The splits were inspected to ensure similar distributions across subsets, as seen in Figure 3.1 for the *birth age* target variable.

As mentioned previously, when using the train-validation-test split, *early stopping* was used on the validation MAE metric, when training was stopped after  $N$  consecutive epochs without improvement, meant to stop the over-fitting of the model.

When exploring the possibility of improving mesh-based models from [6], the brain surface mesh dataset provided by the authors is used as-is (each mesh being a graph with node features derived from the T1 and T2 MRI images from the dHCP structural pipeline), and the connectome data is split differently. As not all the measurements have both mesh and connectome data available, a subset of the [6]s

---

<sup>1</sup>Postmenstrual age, calculated from the start of the pregnancy.



**Figure 3.1:** Empirical density function of the *birth age* (GA) target for the train-test (left, 471 and 203 samples) and train-validation-test (right, 471, 101 and 102 samples) split.

train-validation-test split is used for which both are present. This meant working with a smaller *combined* dataset, as shown in Table 3.1, causing slightly worse baseline performance of the mesh-only models than in the original paper.

**Table 3.1:** Number of samples in each split depending on the task. *Combined* means that both structural and diffusional MRI data was available for that subject.

split	scan age		birth age	
	mesh	combined	mesh	combined
train	408	386	395	370
validation	53	52	51	51
test	54	51	52	49

# Chapter 4

## Graph Neural Network Solutions for Prediction of Neurodevelopmental Indicators in Infants

### 4.1 Machine Learning baselines

In [4], albeit with slightly different data (in terms of dataset size and node representation), the authors explore the usage of classical machine learning methods on the age regression task. Inspired by it, I trained similar models to set baselines for the GNN-based solutions, to assess the alleged advantage of GNNs for graph-structured data as opposed to models where no such connectivity property is exploited. For this reason, a *Lasso* regressor and a *Random Forest* model was fitted to the train set consisting of 70% of the total dataset. The Lasso regressor was tested with  $\alpha \in \{0.1, 1, 10, 100, 1000, 10000\}$ , while the Random Forest for  $n$  (the tree count) in  $\{1, 2, 3, 5, 10, 20, 40, 80\}$ . The best results for each task and best hyperparameter are shown in Table 4.1, to be used as baselines to substantiate the advantage of GNN models.

### 4.2 Connectome-based GNNs

After having experimented with several architectures, like message passing layers similar to [3], a Dense Graph Convolutional (DGC) layer-based architecture proved

**Table 4.1:** Performance of traditional machine learning methods with given parameters on a validation set corresponding to 30% of the total data set. Hyperparameters selected as best of a random search hyperopt. strategy.

Model	Target	MAE	Corr.	$R^2$
Lasso, $\alpha = 100$	birth age	2.0226	0.7086	0.4990
Random Forest, $n = 40$	birth age	<b>1.8119</b>	<b>0.7655</b>	<b>0.5608</b>
Lasso, $\alpha = 100$	scan age	<b>1.54</b>	<b>0.4871</b>	<b>0.2255</b>
Random Forest, $n = 5$	scan age	2.9385	0.4823	-1.5162

to be optimal for proper balance between model parameter count (that, in case of small datasets, is proportional to the risk of over-fitting the data) and accuracy. The DGC expects input as a weighted graph with node features, where  $e_{j,i}$  is the edge weight between nodes  $i$  and  $j$ , while  $x_i$  is the feature vector of node  $i$ . The operator implements the

$$\mathbf{x}'_i = \mathbf{W}_1 \mathbf{x}_i + \mathbf{W}_2 \sum_{j \in \mathcal{N}(i)} e_{j,i} \cdot \mathbf{x}_j$$

feature transformation, where  $\mathbf{W}_1, \mathbf{W}_2$  are the learnable parameters.

The proposed architecture for the model is composed of a *backend*, a feature extractor portion that keeps the graph structure, but implements transformations of the node feature vectors in multiple layers, with information transfer between the neighboring node features proportional to the edge weight that connects the two, and influenced by the learnable  $\mathbf{W}_2$  matrix. At the final feature extractor layer, the features are aggregated (concatenated in the order of the nodes). Initially, the *flatten* aggregation was chosen for its property of order-preserving among the vertices, keeping the nodes' identity related to their position, a choice that was assumed to help preserve information. Following the aggregation, the 87\*CONV\_OUT-sized, flattened feature vector passes through two fully connected (Linear) layers, as seen below:

```
BaselineGCN(
    (conv1): DenseGraphConv(FEAT, CONV_HIDDEN)
    (conv2): DenseGraphConv(CONV_HIDDEN, CONV_HIDDEN)
    (conv3): DenseGraphConv(CONV_HIDDEN, CONV_OUT)
    (flatten): Flatten()
    (lin1): Linear(in_features=FEAT*CONV_OUT, out_features=MLP_HIDDEN)
    (lin2): Linear(in_features=MLP_HIDDEN, out_features=1)
```

)

This latter *head* portion, akin to visual CNN models for image-related tasks like detection and classification, implements the regression, working with the higher-order feature representation learned by the backend. The parametric ReLU function was used in the model as activation after each layer (with each layer’s parameter being separately trainable), except the last one (although the [23, 43] target variable range would allow that, generality for the regression task was preserved). The usage of batch normalization worsened training convergence, therefore it was omitted.

The CONV\_OUT parameter can be set separately from the CONV\_HIDDEN feature dimension, to be used as a *bottleneck* parameter, because the output feature size of the backend portion strongly influences the parameter count of the first linear layer, thus it should generally be kept lower than CONV\_HIDDEN to avoid over-parametrization and early over-fitting.

The input channel size FEAT (the length of the node feature vectors) is influenced by the positional encoding. Early trials comparing the spectral basis positional embedding (using the *birth age* task as benchmark) with the 87-dimensional one-hot encoding for each node was not yet conclusive, showing better performance for the former embedding with smaller models (CONV\_HIDDEN=10, CONV\_OUT=3, MLP\_HIDDEN=5), but marginally worse for larger network sizes (CONV\_HIDDEN=20, CONV\_OUT=5, MLP\_HIDDEN=10), as seen in Table 4.2. These two basic model configurations were chosen as broadly representative (as found through experimentation) for the approximate lower and upper bounds to optimal convergence, avoiding both under- and overfitting.

**Table 4.2:** The performance of small vs. large models on the GA (*birth age*) regression task. Both models were trained for 200 epochs on the train-test split (70% - 30%), with a learning rate of 0.001 using the *Adam* optimizer. The best test set result is shown, indicating a large advantage for the *spectral basis* for smaller models, but a small disadvantage for the larger one.

Size	Positional Encoding	MAE	Correlation	$R^2$
small	<b>Spectral basis (k=10)</b>	<b>2.357</b>	<b>0.7831</b>	<b>0.525</b>
	One-hot	2.819	0.7515	0.4481
large	Spectral basis (k=10)	2.088	0.8323	0.6186
	<b>One-hot</b>	<b>2.024</b>	<b>0.8615</b>	<b>0.6366</b>

For the broader search for the best positional embedding, the train-validation-test split was used, as it is advised for hyperparameter search scenarios. The four embedding types were benchmarked on the *scan age* (PMA) task, as seen in Table 4.3.

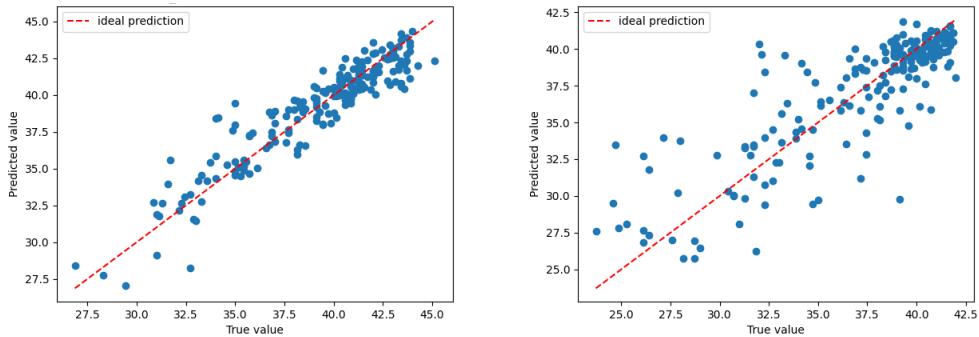
The experiments were being run on the larger, better performing model, with the one-hot embedding proving superior in MAE accuracy as well as correlation and  $R^2$  score on the test set and most of the validation set as well.

**Table 4.3:** The effect of positional encoding on model performance for *scan age* (PMA) prediction. V. and T. refer to the validation (15%) and test set (15%) respectively, LDP denotes the *Local Degree Profile*. Training was conducted on the family of larger models, for a maximum of 300 epochs, with a learning rate of 0.001 using the *Adam* optimizer. Early stopping with a patience of 70 epochs was used at each run. Based on the results, the one-hot encoding seems to be the optimal positional encoding.

Positional Encoding	V. MAE	T. MAE	V. Corr.	T. Corr.	V. $R^2$	T. $R^2$
<b>One-hot</b>	1.203	<b>0.9735</b>	<b>0.9215</b>	<b>0.9444</b>	0.7795	<b>0.8896</b>
LDP	1.246	1.01	0.9076	0.936	0.7516	0.8707
Ones (k=87)	<b>1.187</b>	1.023	0.9126	0.9371	<b>0.7894</b>	0.8641
Ones (k=10)	1.206	1.095	0.9151	0.9154	0.7839	0.8324
Spectral basis (k=10)	1.259	1.096	0.9159	0.9243	0.7626	0.8412

This result also clearly shows that on the *scan age* task, the GNN outperforms the previous baseline using traditional ML regressors, and by a large margin, warranting the usefulness of this family of models on the task of determining infant age based on the (re)constructed connectivity graph.

A visual regression plot can be seen in Figure 4.1, showing generally better performance on the *scan age* task, when plotting the predictions of the larger model (with one-hot input encoding) on the validation and test set (30% of the whole dataset) for both tasks.



**Figure 4.1:** Regression plot on task *scan age* (left), and *birth age* (right), based on the best models when predicted on the validation and test sets (203 samples in total).

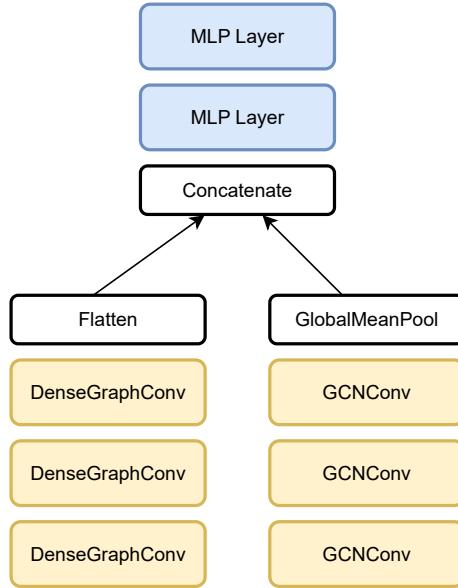
## 4.3 Model and dataset fusion

Assessing the performance of the connectome-based models, we can conclude that while achieving decent results, they fall short of the mesh-based models presented in [6]. The cause of this might be the stronger inherent noise present in the connectome as input data: it is a result of a long reconstruction process, with several assumptions made (and several known pitfalls, as detailed in [8]) and often relying on probabilistic methods when assessing the underlying structure of nerve bundles. Also, fidelity comes to question: the underlying ground truth (well-connectedness as a measure of density of the white matter tracts) can be described at a scale several orders of magnitude smaller than the base measurement itself (diffusional MRI, with millimeter-scale resolution), so any post-processing works with inherently inadequate (low resolution) information. In contrast, the ground truth of brain surface properties (e.g. curvature) can be accurately described on the millimeter scale of MRI measurements: any detail smaller than that can be considered noise when inspecting neonatal brain topology.

Despite the known limitations of the connectome as a model input, it still incorporates a kind of knowledge wildly different from the surface mesh information. This raises the idea of combining the two, and finding out how accurate (and how much better than the individual baselines) a well crafted model fusion might work.

### 4.3.1 Concepts and design

The fusion approach can be split into two interconnected areas of experimentation: the fusion of input data, and the fusion of the models. A combination of data (a pair of graphs, the mesh and the connectome) implies that the model should have two inputs, and after initial feature extraction along two separate "paths", these are mingled via some kind of aggregation, followed by some more feature extraction and prediction that exploits the merged information. Based on this, we should propose the "where" and "how" of this merge, which is a set of design choices that should be explored in the context of fusion of models. Other kinds of fusion strategies, while being theoretically possible along the location information of both mesh points and the region nodes of the connectome (thus exploiting proximity information), it is practically cumbersome and falls outside the scope of this work.



**Figure 4.2:** The proposed approach to model fusion: joining on latent space, with common information extracted through bottleneck layers.

### 4.3.2 Implementation

The implementation closely follows the schematic design for the proposed model fusion, that can be seen in Figure 4.2. The model structure consists of two convolutional backbones, one for the connectome and one for the mesh input, followed by an aggregation over the separate graphs, then an aggregation between the two, followed by a common head section that is meant to synthesize the merged information of the two input modalities and predict the target age.

A PyTorch summary-like outline of the proposed model, as well as its most important hyperparameters capitalized can be seen below:

```

FusionGNN(
    (mesh): MeshFeatureExtractor(
        (bn0): BatchNorm1d(MESH_FEAT)
        (fc1): GCNConvBlock(MESH_INPUT, MESH_HIDDEN)
        (fc2): GCNConvBlock(MESH_HIDDEN, MESH_HIDDEN)
        (fc3): GCNConvBlock(MESH_HIDDEN, MESH_HIDDEN)
        (fc4): GCNConvBlock(MESH_HIDDEN, MESH_HIDDEN)
        (act): ReLU()
    )
    (connectome): ConnectomeFeatureExtractor(
        (conv1): DenseGraphConv(CONNECTOME_FEAT, CONV_HIDDEN)
        (conv2): DenseGraphConv(CONV_HIDDEN, CONV_HIDDEN)
        (conv3): DenseGraphConv(CONV_HIDDEN, CONV_OUT)
        (act): PReLU()
    )
    (head): Sequential(
        (0): Linear(AGGREGATION_OUTPUT, HEAD_HIDDEN)
        (1): ReLU()
        (2): BatchNorm1d(HEAD_HIDDEN)
        (3): Linear(HEAD_HIDDEN, 1)
    )
)

```

Here, the `GCNConvBlock` inside `(mesh)` denotes a *convolution* → *activation* → *batch normalization* sequence using Pytorch Geometric's `GCNConv` implementation.

During the first experiments, the hyperparameters were set according to the best model configurations when these were trained separately. The hidden dimension of the mesh model has been lowered from 64 (the optimum) to 32 in order to fit VRAM constraints, with the `(head)` section having `HEAD_HIDDEN=32` too. Lowering hidden dimensions in the head for the mesh-only model proved to negatively impact accuracy.

Regarding the aggregation methods, I kept the global mean pooling (taking the average of every node's features) operation of the mesh model that resulted in output feature dimension of `MESH_HIDDEN`, while also keeping the flattening aggregation for the connectome that results in `87 * CONV_OUT` feature dimension. I aggregated these by concatenation, thus resulting in `AGGREGATION_OUTPUT` being the sum of these two.

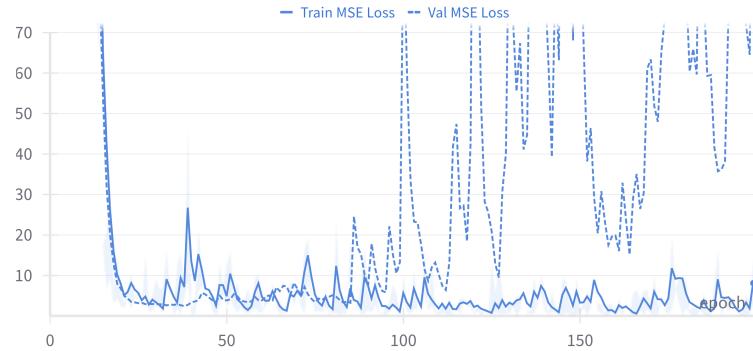
### 4.3.3 Initial results

The initial results on the *scan age* task were disappointing, as seen in Table 4.4. The connectome model under-performing the previous results was to be expected, considering the reduction in training samples (previously, there were 471 samples in the train split, when splitting the total of 674 connectomes, but when training is conducted only on those connectomes that are in the mesh train splits as well, this number decreases to 386 for the scan age and 370 for the birth age task). The more puzzling result came from the under-performance of the fusion model, as it could have been expected for the combination to work at least as well as the best of its constituents. This prompted a more thorough inspection of the training process.

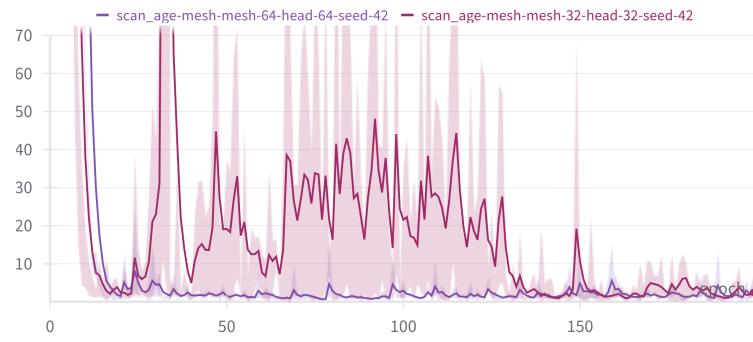
**Table 4.4:** Results for the fusion model training compared with the connectome and mesh models alone, on the *scan age* task. The best validation epoch's checkpoint (based on MAE) was saved and the tests were evaluated on this model. In the fusion model, the output of the connectome and mesh backbones were concatenated, with the connectome backbone's aggregation over its nodes being the *flatten* operation. In each case, I trained the model for 200 epochs with a learning rate of 0.001 using the *Adam* optimizer.

model	connectome	mesh	head	Val. MAE	Test		
	hidden	out	hidden		MAE	Corr.	R <sup>2</sup>
connectome	20	5	-	1.1900	1.309	0.5838	-0.850
mesh	-	-	32	0.6295	<b>0.6747</b>	<b>0.7618</b>	<b>0.227</b>
fusion	20	5	32	0.7040	0.794	0.7339	-0.029

Upon inspecting the connectome model's training, I found out that it overfits the data quite early (Figure 4.3). Previously, when training only on the connectomes, early stopping was used to avoid such outcomes, the experience being that usually 40-50 epochs were enough to reach best performance. In contrast, the mesh-only model training, while getting less stable from the point when it reaches validation MAE below 1 week, it does not produce a pronounced overfit, at least not in the long run, as seen in Figure 4.4. Longer training shows a slow trend of improvement (although with large variance) when using one-time learning rate scheduling (lowering the learning rate by a factor of 10 when first reaching validation MAE below 1 week). This shows that the mesh model, as opposed to the connectome-based one, benefits from longer training, periodically "stumbling into" marginally better configurations while not showing signs of the validation loss systematically increasing. This kind of training regime, thus the circumvention of early stopping, is required to reach validation MAEs of 0.65 and below.



**Figure 4.3:** The connectome-only training reaches optimum quite early, then overfits significantly (the validation loss starts to rise steeply) in the rest of the 200 epochs. While early stopping would stop the training, leaving us with the optimal checkpoint based on the validation score, the fusion model requires longer training to reach optimum from the mesh input's side, which places us in the overfit territory on the connectome data.



**Figure 4.4:** Training the mesh model with 32 (red) and 64 (purple) hidden parameter size. The filled patch is the region between the train and the validation loss (MSE) curve, while the line is the average of the two. The larger model produces more stable convergence, the smaller has a quite long period of high variance, but in the end, neither do overfit the data. Other validation metrics also reflect this trend.

These two conflicting training profiles cause the combined model to perform poorly, the inherent overfit of the connectome part acting as a liability to the whole.

Another liability is the high parameter count that comes as a consequence of integrating the connectome model. Its default aggregation consists of flattening the node feature vectors into a `87 * CONV_OUT` sized vector, that is significantly larger than the mesh model’s `MESH_HIDDEN`-sized feature extractor output. When combined, the input size of the first `Linear` layer in the *head* of the model grows considerably, and so does the parameter count of that layer. It can be observed in Table 4.5 that indeed, the base connectome model is quite *head-heavy*, with the majority of its parameters (more than 75%) being in the fully connected layers of the model head. The same is true for the fusion model. The results prove that this is not an optimal configuration, as the more balanced mesh-only model benefits from allocating a larger number of parameters to the graph convolutional backbone, having a structure better suited for generalization due to its shared parameters.

**Table 4.5:** Trainable parameter count of different model configurations, determined by Pytorch Lightning’s `ModelSummary(max_depth=2)` callback.

Name	Type	#Params.
Model	<code>ConnectomeGNN</code>	5.8 K
Feature Extractor	<code>ConnectomeFeatureExtractor</code>	1.4 K
Head	<code>Sequential</code>	4.4 K
Model	<code>MeshGNN</code>	5.1 K
Feature Extractor	<code>MeshFeatureExtractor</code>	3.9 K
Head	<code>Sequential</code>	1.2 K
Model	<code>FusionGNN</code>	20.4 K
Feature Extractor	<code>MeshFeatureExtractor</code>	3.9 K
Head	<code>ConnectomeFeatureExtractor</code>	1.4 K
	<code>Sequential</code>	15.1 K

#### 4.3.4 Improvements

Upon identifying a probable cause for suboptimal performance, I set out to improve the fusion model by reducing feature size on the connectome feature extractor (convolutional backbone) output. I implemented global mean pooling over the output feature vectors of the 87 nodes, reducing the output size of the connectome backbone to `CONV_OUT`. This hyperparameter already served as a bottleneck of information to regulate parameter size, but from now on, it could be increased more freely, without running the risk of drastically increasing parameter count.

An additional aggregation method *between* the two feature extractor outputs was also implemented, adding the possibility to aggregate by adding the two, provided the MESH\_HIDDEN and CONV\_OUT are equal. Although it had not proven to produce better test accuracy, it is a good option when conducting broader hyperparameter-optimization in the future.

These aggregations reduce the possibility of easily exploiting outstanding individual node features, by destroying the information about the identity of the feature vectors and promoting a more smoothed view of the graph, but the gain in generalization capability outweighed the cons.

Indeed, checking model parameter counts when using the global mean pooling aggregation, a more balanced distribution can be observed, as seen in Table 4.6. When considering this along the new results seen in Table 4.7, it shows that the improvement reduced initial trainable parameter count by a factor of seven, while managing to beat both baselines in all validation metrics, confirming the assessment about the cause of poor performance, and supporting the generalization benefits of a more reductive pooling.

**Table 4.6:** Trainable parameter count of the fusion model when using *global mean pooling* at the output of the connectome feature extractor. The first section presents a model that has identical hyperparameters to the previously shown fusion model (CONV\_HIDDEN=20, CONV\_OUT=5), but it has a third of the trainable parameters (6.7K from the original 20.4K, while the head parameter count dropped from 15.1K to 1.3K, a tenth of the original) due to the pooling. The second model shows that using this new pooling, increasing the hidden dimensions (CONV\_HIDDEN=CONV\_OUT=32) of the connectome feature extractor leads to only a modest increase in trainable parameters.

Name	Type	#Params.
Model	<b>FusionGNN</b>	6.7 K
Feature Extractor	MeshFeatureExtractor	3.9 K
	ConnectomeFeatureExtractor(20, 5)	1.4 K
Head	Sequential	1.3 K
Model	<b>FusionGNN</b>	10.9 K
Feature Extractor	MeshFeatureExtractor	3.9 K
	ConnectomeFeatureExtractor(32, 32)	4.8 K
Head	Sequential	2.2 K

Interestingly enough, training the connectome-only models with this new pooling does not improve upon the previous baseline of the same architecture. While this is only a quick assessment, it points to the fact that the new pooling is not the best

**Table 4.7:** New results on *scan age* when using global mean pooling at the connectome backbone’s output. Now the fusion model beats the others in all validation metrics, proving that the new aggregation does indeed help generalization and overall performance significantly through reducing parameter count (and therefore, the risk of overfit) while not reducing the effectiveness of the model.

model	connectome hidden	out	mesh hidden	head hidden	Val. MAE	Test MAE	Corr.	R <sup>2</sup>
connectome	20	5	-	10	1.9336	2.0297	0.727	0.474
mesh	-	-	32	32	0.6295	0.6747	0.7618	0.227
<b>fusion</b>	<b>20</b>	<b>5</b>	<b>32</b>	<b>32</b>	<b>0.6233</b>	<b>0.645</b>	<b>0.8561</b>	<b>0.631</b>

choice in all scenarios, but nevertheless it yielded considerable advantage when used as a building block of the model fusion method.

#### 4.3.5 Results on the birth age task

In the previous sections, the effectiveness of the new pooling method was demonstrated on the *scan age* task, as it is the easier of the two. After running experiments on the *birth age* as well, the results can be seen in Table 4.8. When considering MESH\_HIDDEN=32, the fusion model still outperforms the mesh-only counterpart, but the results show a higher variance for all model architectures in this task, indicating more noisy data in relation to the target variable.

**Table 4.8:** Experiment results on the *birth age* task. When using the smaller hidden layer size of 32 (the top two rows), the fusion model outperforms the mesh model. When raising the hidden layer size to 64 (bottom two rows), the results are more mixed, the MAE values on both the validation and test set are better for the mesh model, but the correlation and the coefficient of determination is still higher for the fusion model. Generally speaking, these results show higher variance than on *scan age*, indicating the difficulty of the task and the limited generalizability based on the 370 training samples. Furthermore, when using GPUs for training, even when the random number generators are seeded identically, the results vary wildly due to the randomized implementation of some GNN operators as CUDA kernels.

model	connectome hidden	out	mesh hidden	head hidden	Val. MAE	Test MAE	Corr.	R <sup>2</sup>
mesh	-	-	32	32	1.296	1.672	0.6173	-0.367
fusion	20	5	32	32	<b>1.214</b>	<b>1.236</b>	<b>0.6771</b>	<b>-0.115</b>
mesh	-	-	64	64	<b>1.197</b>	<b>1.269</b>	0.6463	-0.348
fusion	20	5	64	64	1.212	1.504	<b>0.7124</b>	<b>-0.284</b>

### 4.3.6 Software and hardware resources

I implemented the training code in Pytorch and Pytorch Geometric (PyG), a library for geometric and graph deep learning. Pytorch Lightning was used for reproducibility and minimizing boilerplate code. For experiment tracking, I used Weights and Biases.

The training experiments were run on both personal GPUs, as well as on the *Kommandor* supercomputer’s AI partition, through a project account. The code usually runs with about 8GB of VRAM for the larger fusion models.<sup>1</sup>

### 4.3.7 Deployment

As in the case of the connectome generation pipeline, the code base of these experiments has several dependencies, and was deployed on several machines, as explained in the previous section. These environments support slightly different execution environments, and for this reason, additional deployment solutions were used.

For running on consumer PC environments, or generally any platform that supports Docker, an accompanying *Dockerfile* was added to the repository to ensure reproducible dependencies and results. The *Kommandor* supercomputer supports a different, High Performance Computing-specific containerization technology called *Singularity*, so the Github repository includes build files for that platform as well. Singularity supports *bootstrapping* from certain trusted Docker images, to remove redundancy in the deployment code base.

Execution of containers in the aforementioned HPC environment required an additional runtime layer called SLURM, that uses job queue-based processes to ensure optimal utilization, fair and controlled access to the resources of the allocated partition, especially when experiencing high workload coming from multiple users. The repository contains SLURM batch scripts that specify resource allocation (preferred CPU cores, RAM and GPU) and container execution from Singularity image files (with .sif extension).

These configurations ensure easily reproducible experiments on a range of platforms, and optimal usage of the generous resources provided in a HPC environment.

---

<sup>1</sup>Code to reproduce the experiments can be found at <https://github.com/peter-i-istvan/msc-thesis> (accessed: 03.11.2024), from the `train_experiments.py` script.

## 4.4 Discussion

Based on the above, merging the previously discussed architectures does yield an improvement over both the mesh and the connectome baselines. While drawing conclusions, it is important to keep in mind the size of the training, validation and test dataset. Currently, depending on the task, training is done on about 400 samples, with cca. 50 samples held out each for validation and testing. This imposes upper limits on model size (avoiding over-parametrization), and also on generalization capability. It might be that we are already seeing diminishing returns, and improvements can hardly be deemed statistically significant.

# Chapter 5

## Explainability

This final chapter presents a short overview of applicable model explainability algorithms, followed by a specific technique that proves generally successful in interpreting *local* model behavior under small disturbance to the input features. Explainable AI (xAI) is crucial when dealing with medical data, or when incorporating any intelligent system in the medical decision support, thus there is the need for the application of such techniques on the GNNs presented previously, as a case study for other applications that use regression or classification based on similar input.

### 5.1 Overview

The explainability of neural network models, black-box by default, is a long standing problem in the field of machine learning. While other algorithms, like Decision Trees and other tree-based methods are interpretable by default, deep neural networks (DNNs) define a complex mapping from input feature space to intermediate representations and predicted target, where individual contributions of model parameters, as well as the importance of individual input features is hard to evaluate properly. In the case of Graph Neural Networks, explanation techniques emerged in tandem with the field itself, drawing on established research about DNNs.

In [20], the authors provide a recent overview of such methods, as well as an evaluation benchmark dataset (ShapeGGen) and a library of implemented algorithms (GraphXAI). It defines explanations as sub-graphs of the input that retain the information present in the original in terms of predictive power. The paper touches on some common pitfalls of explainability algorithms, like the lack of clear *ground truth* explanations in most cases. Such information is readily available for synthetic data (like the procedurally generated ShapeGGen), but might be available for other data

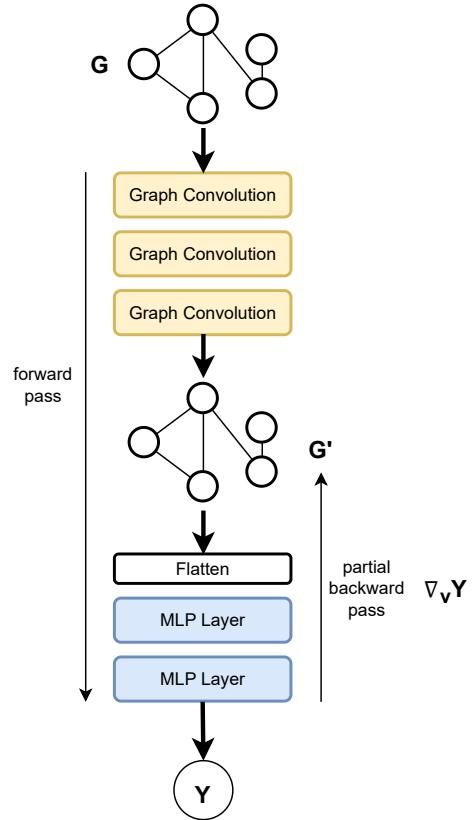
sources only through extensive expert insight, or not at all. The metrics reviewed by the authors (accuracy, faithfulness, stability) are applicable only in presence of clear ground truth sub-graphs, meaning that in the current application, for connectomes, such data cannot be retrieved, and thus evaluation of explainability in the current context can be done only qualitatively, by interpreting results and relating to common medical knowledge, and not quantitatively, with precise metrics.

During experimentation, I tried several methods across different paradigms. The Parameterized Graph Explainer (PGExplainer), classified as a perturbation-based algorithm [20], introduced in [19], proposes a learnable, probabilistic sub-graph generation that samples edges based on learned latent variables. The drawback with this approach, addressed in [18], is the out-of-distribution problem, where the sub-graphs come from a different data distribution than the original, so in fact the model would not encounter such samples normally, leading to irrelevant results in terms of the explanations. Due to the small number of samples available for explainer training, the method showed no meaningful convergence in terms of the objective function, thus being inadequate for the task.

Moving to simpler, *gradient-based* and *local* explainability methods provided more fruitful, with the GradCAM algorithm being able to show meaningful results and no need for separate training other than a single forward and backward pass of the model for each input sample.

## 5.2 GradCAM for convolutional and graph neural networks

The GradCAM explainability algorithm for Convolutional Neural Networks was introduced in [5] as a visual tool to assess the influence of image regions on the prediction of the final score (usually the class score for classification tasks, but can also be easily extended to regression output). It works by computing the gradients of the relevant score with respect to the last convolutional layer’s output feature maps, weighting these by the activation scores, and mapping the result to the image’s spatial domain for an intuitive explanation. Its advantages consist in its simplicity, not requiring additional training or constrained perturbation and masking of the input, which would run the risk of providing explanations out of distribution with respect to the original data. The results are easy to interpret, with heat map representations in the case of image classification being a human-friendly explanation medium, and also a way to diagnose possible causes of poor accuracy on certain data.



**Figure 5.1:** The GradCAM algorithm adapted to GNNs. Gradients are propagated backwards only up to the last graph neural layer’s output, multiplied by the activations of this layer to receive the explainability scores (impact on prediction) for each node.

Adaptation for Graph Neural Network architectures, namely for the Graph Convolution-based model presented previously, is straightforward, works by changing the spatial domain of GradCAM from the regular, grid-shaped input in the case of images to the irregular graph domain, with nodes as individual explainer entities. Resulting heat maps are thus overlaid to the set of graph vertices. A schematic illustration of the process can be seen in Figure 5.1.

Due to the nature of this method, we get an interpretation of the model’s local behavior (relative to the input vector) through the value of the gradient. This combines input sensibility with weighting the positive influences of the nodes through the activation scores. This explanation stemming from functional analysis does not cover the probabilistic aspects of the model and effectively treats node inputs as

independent, rather than being described by a general probability density function over the input feature vectors and target variables. This joint distribution can be accurately estimated for very large data sets via generative models, but for the case in point, proper performance of such a method is improbable due to sparsity and non-exhaustiveness of the input data. Nevertheless, approximate methods, exemplified by the use of GradCAM on the age prediction task, show promising performance.

## 5.3 Results

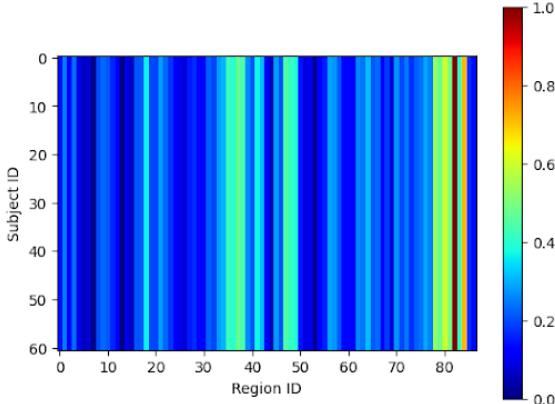
The GradCAM method was applied on a well-performing *scan age* predictor model, with accuracy and architecture similar to the one presented in the previous chapters. Evaluation was conducted on the test set of the task.

The resulting scaled importance scores for the nodes (cortical regions) can be seen in Figure 5.2. The importance map shows that a few regions have an outstandingly high positive impact on the predicted target variable. These regions show a strong correlation with higher values of the PMA target. After quantifying the normalized explainability scores, as seen in Table 5.1, the node features of the following regions are implied to be the most important, namely (in decreasing order of importance): the white matter region of the left parietal lobe, the cerebrospinal fluid, the extracranial background, and the gray and white matter of the right parietal lobe.

**Table 5.1:** Brain regions with the highest attributed importance, based on the GradCAM model applied to the *scan age* predictor model, scaled to the [0, 1] interval. WM and GM denote the white and grey matter tissue of the functional region. The region names are taken from the ALBERT brain atlas, as the region labels of the connectome extracted from the dHCP data correspond to this parcellation.

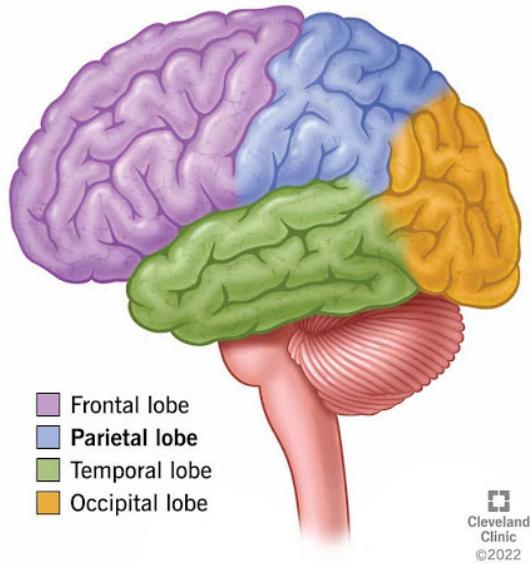
Region	Normalized Importance
Parietal lobe left WM	1.000
Cerebrospinal Fluid	0.769
Extracranial background	0.633
Parietal lobe right GM	0.516
Parietal lobe WM	0.489

Looking up medical and anatomical information regarding the function of the parietal lobe, it can be found that it serves as an important center for sensory integration, located as seen in Figure 5.3. This can indeed have an important role in differentiating brain patterns based on gestational age, as that region is in fast-paced development during the perinatal period of an infant’s life.



**Figure 5.2:** Feature importance of the brain regions for a connectome-based GNN model that predicts *scan age*. Each row of the bitmap represents the 87 regions of the given subject, the scalar value of the cell meaning the positive influence attributed to that node, scaled to the  $[0, 1]$  interval. The subjects are ordered by increasing age from top to bottom, taken from the test split of the *scan age* dataset. It can be seen that subjects associated with both high and low target values for the age have approximately the same node-importance distributions.

Cross-referencing these findings with explanations coming from other data sources, namely the structural MRI-based surface meshes used in [6], we find a significant overlap in the assessment of impactful regions. The authors inspect the average myelination in both term and preterm neonates, as well as the SHAP score explanations for the given regions, and find a pronounced concentration of meaningful features on the surface of the left parietal lobe (referred to in the paper as the somatosensory region). Considering the fact that these results stem from the interpretation of a different model architecture, significantly different input data and explainability algorithm, it is safe to assume that both results are close to a hypothetical *ground truth* explanation constructed from prior biological knowledge based on the relevant medical literature (for example, [21] indicates that the right, but mostly the left parietal lobe's development is correlated with later outcomes in cognitive and motor abilities, evaluated on infants and up to adulthood). This reinforces the claim that GradCAM is effective in interpreting the importance of input features to the proposed Graph Convolutional model, despite the scarcity of data and the approximate nature of the explainer algorithm.



**Figure 5.3:** Schematic figure of functional brain regions, including the Parietal lobe: the center for integration of different sensory stimuli.

## 5.4 Conclusion

Concluding this chapter, an effective explainability method was provided to assess the important nodes in the input graph when predicting birth and scan age with the models proposed in this thesis. During experimentation, it became apparent that gradient-based methods exploiting local model behavior are more viable as opposed to sub-graph generation-based or surrogate model-based methods in GNN explainability. The accuracy of the GradCAM algorithm is substantiated by comparison to other, independent methods with similar results, as well as common knowledge from medical literature, like [21]. This helps in strengthening transparency of otherwise black-box models, as human-centered, and most importantly, human health aspects in application of machine learning models on medical data are under considerable scrutiny, and rightly so. Proven and human-interpretable explanations are crucial when the responsibility of the decision lies on the healthcare practitioner who must use the output of such algorithms.

Birth age, as well as actual age is usually known to the medical professional when diagnosing an infant. However, related factors like preterm birth and its impact on brain structure and function might not be so obvious, and such methods, as presented in this chapter might help in drawing conclusions, or indicating possible brain regions for further inspection. Regression of age as target variable is provided

only as a case study, but training on other targets like neuro-developmental outcomes (resulting from cognitive and motor assessments, like Bayley scores, described in [22]), when and if such data is widely available, might help predict risk factors of developmental delays, and it is shown that premature birth is correlated with such early impairments. Thus, fitting accurate models on the birth age task can prove a strong foundation when pivoting to prediction of developmental outcomes.

Evaluation of explainability algorithms should follow a rigorous study including multiple models and interpretation methods, to minimize the effect of individual shortcomings. Results should also be checked against existing knowledge in medical literature. However, the uses of explainable AI methods, like the ones described here, can stretch beyond automating the application of current medical understanding. These methods have the potential to generate novel biological knowledge, when studying new phenomena, and fitting accurate predictors on input data, combined with a potent explainability algorithm. The data features suggested to be impactful and important, provided by such a procedure, can help to narrow the direction of further inquiry and experimentation, thus integrating human-centered AI applications in the workflow of medical research.

# Chapter 6

## Summary and Future Work

In this thesis, I presented a solution for neonatal age regression based on dMRI-derived connectomes (brain connectivity matrices). Raw measurement data from the Developing Human Connectome Project was used, encompassing a cohort of term and preterm neonates (born between 23 and 43 weeks of postmenstrual age). Creating the connectomes needed a custom preprocessing pipeline in lieu of any publicly available "works out-of-the-box" tool compatible with the data.

During my work, I implemented the preprocessing pipeline using the MRtrix neuroimaging library's utilities, based on recent literature regarding connectomics. I also proposed a Graph Neural Network (GNN) model that takes advantage of the inherent network-like structure of the input, successfully trained and evaluated it, also comparing with the baselines of more traditional machine learning models to substantiate the advantages of GNNs.

Based on prior work at the Department regarding the same age regression task from brain surface mesh data, I also proposed, implemented, trained, and evaluated a combined model that uses both mesh and connectome data to achieve better prediction accuracy.

Considering further requirements towards AI models, like interpretability and transparency, the last portion of the thesis presents the application of the GradCAM algorithm on the age predictor models, which shows those individual regions in the input that influenced the final prediction the most. The results show that the model prioritizes relevant brain regions, proving the effectiveness and usefulness of the method as another layer of oversight.

In the future, the method could be expanded to include functional MRI data as well, providing a multimodal fusion of knowledge that uses all measurement data publicized by the dHCP team.

# Acknowledgment

We acknowledge KIFÜ (Governmental Agency for IT Development, Hungary, <https://ror.org/01s0v4q65>) for awarding us access to the Komondor HPC facility based in Hungary. The models were trained on the AI partition of the system, which helped speed up prototyping and hyperparameter optimization.

# Bibliography

- [1] Matteo Bastiani, Jesper L.R. Andersson, Lucilio Cordero-Grande et. al. *Automated processing pipeline for neonatal diffusion MRI in the developing Human Connectome Project*, NeuroImage. 2019, volume 185, p. 750-763, <https://www.sciencedirect.com/science/article/pii/S1053811918304889> (accessed: 03.11.2024)
- [2] Antonios Makropoulos, Emma C. Robinson, Andreas Schuh et. al. *The developing human connectome project: A minimal processing pipeline for neonatal cortical surface reconstruction*, NeuroImage. 2018, volume 173, p. 88-112, <https://www.sciencedirect.com/science/article/pii/S1053811918300545> (accessed: 03.11.2024)
- [3] Jeremy Kawahara, Colin J. Brown, Steven P. Miller, Brian G. Booth, Vann Chau, Ruth E. Grunau, Jill G. Zwicker, Ghassan Hamarneh. *BrainNetCNN: Convolutional neural networks for brain networks; towards predicting neurodevelopment*, 2017, <https://www.sciencedirect.com/science/article/pii/S1053811916305237> (accessed: 03.11.2024).
- [4] Taoudi-Benbekroun Y, Christiaens D, Grigorescu I, Gale-Grant O, Schuh A, Pietsch M, Chew A, Harper N, Falconer S, Poppe T, Hughes E, Hutter J, Price AN, Tournier JD, Cordero-Grande L, Counsell SJ, Rueckert D, Arichi T, Hajnal JV, Edwards AD, Deprez M, Batalle D. *Predicting age and clinical risk from the neonatal connectome*. Neuroimage. 2022 Aug 15;257:119319. doi: 10.1016/j.neuroimage.2022.119319. Epub 2022 May 16. PMID: 35589001.
- [5] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, *Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization*, in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618-626, doi: 10.1109/ICCV.2017.74.

- [6] Dániel Unyi, Bálint Gyires-Tóth. *Neurodevelopmental Phenotype Prediction: A State-of-the-Art Deep Learning Model*, 2022, arXiv:2211.08831 [cs.CV], <https://arxiv.org/pdf/2211.08831.pdf> (access date: 2023.11.24).
- [7] Tourbier S., Queralt J. R., Glomb K. *Connectome Mapper 3: A Flexible and Open-Source Pipeline Software for Multiscale Multimodal Human Connectome Mapping* Journal of Open Source Software, volume 7 n. 74 p. 4248, doi: 10.21105/joss.04248 <https://doi.org/10.21105/joss.04248> (accessed: 03.11.2024)
- [8] Stamatis N Sotiropoulos, Andrew Zalesky. *Building connectomes using diffusion MRI: Why, how and but*, 2017, DOI:10.1002/nbm.3752, <https://core.ac.uk/works/8319787> (accessed: 03.11.2024)
- [9] Andrew Jahn. *Andy's Brain Book*, 2022, doi:10.5281/zenodo.5879293, [https://andysbrainbook.readthedocs.io/en/latest/MRtrix/MRtrix\\_Introduction.html](https://andysbrainbook.readthedocs.io/en/latest/MRtrix/MRtrix_Introduction.html). (accessed: 03.11.2024)
- [10] Sean P. Fitzgibbon, Samuel J. Harrison, Mark Jenkinson et. al. *The developing Human Connectome Project (dHCP) automated resting-state functional processing framework for newborn infants*, Neuroimage, 2020, vol. 223, doi:10.1016/j.neuroimage.2020.117303, <https://doi.org/10.1016/j.neuroimage.2020.117303>. (accessed: 03.11.2024)
- [11] Ziyi Huang, Qi Wang, Senyu Zhou et. al. *Exploring functional brain activity in neonates: A resting-state fMRI study*, Developmental Cognitive Neuroscience, 2020, vol. 45, doi:10.1016/j.dcn.2020.100850, <https://doi.org/10.1016/j.dcn.2020.100850>. (accessed: 03.11.2024)
- [12] Smith SM, Vidaurre D, Beckmann CF, Glasser MF, Jenkinson M, Miller KL, Nichols TE, Robinson EC, Salimi-Khorshidi G, Woolrich MW, Barch DM, Uğurbil K, Van Essen DC. *Functional connectomics from resting-state fMRI*. Trends Cogn Sci. 2013 Dec;17(12):666-82. doi: 10.1016/j.tics.2013.09.016. Epub 2013 Nov 12. PMID: 24238796; PMCID: PMC4004765. <https://pmc.ncbi.nlm.nih.gov/articles/PMC4004765/> (accessed: 03.11.2024)
- [13] Alexander AL, Lee JE, Lazar M, Field AS. *Diffusion tensor imaging of the brain*. Neurotherapeutics. 2007 Jul;4(3):316-29. doi: 10.1016/j.nurt.2007.05.011. PMID: 17599699; PMCID: PMC2041910.
- [14] Tournier J-D., Smith R., Raffelt D. et. al. *MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation*, NeuroImage,

2019, volume 202, p. 116137, doi: 10.1016/j.neuroimage.2019.116137, <https://www.sciencedirect.com/science/article/pii/S1053811919307281> (accessed: 03.11.2024)

- [15] Daigavane, Ameya and Ravindran, Balaraman and Aggarwal, Gaurav. *Understanding Convolutions on Graphs*, 2021, Distill, <https://distill.pub/2021/understanding-gnns> (accessed: 03.11.2024), doi: 10.23915/distill.00032
- [16] Škoch, A., Rehák Bučková, B., Mareš, J. et al. *Human brain structural connectivity matrices-ready for modelling.*, 2022, Sci Data 9, 486., <https://www.nature.com/articles/s41597-022-01596-9> (accessed: 03.11.2024), doi: <https://doi.org/10.1038/s41597-022-01596-9>
- [17] Sanchez-Lengeling, Benjamin and Reif, Emily and Pearce, Adam and Wiltschko, Alexander B. *A Gentle Introduction to Graph Neural Networks*, 2021, Distill, <https://distill.pub/2021/gnn-intro> (accessed: 03.11.2024), doi: 10.23915/distill.00033
- [18] Kenza Amara, Mennatallah El-Assady, Rex Ying. *GInX-Eval: Towards In-Distribution Evaluation of Graph Neural Network Explanations*, arXiv:2309.16223 [cs.AI], <https://doi.org/10.48550/arXiv.2309.16223> (accessed: 03.11.2024)
- [19] Luo, Dongsheng and Cheng, Wei and Xu, Dongkuan and Yu, Wenchao and Zong, Bo and Chen, Haifeng and Zhang, Xiang. *Parameterized explainer for graph neural network*, 2020, Proceedings of the 34th International Conference on Neural Information Processing Systems, art. 1646, p. 12, NIPS '20. <https://dl.acm.org/doi/10.5555/3495724.3497370> (accessed: 03.11.2024)
- [20] Agarwal, C., Queen, O., Lakkaraju, H. et al. Evaluating explainability for graph neural networks. Sci Data 10, 144 (2023). <https://doi.org/10.1038/s41597-023-01974-x> (accessed: 03.11.2024)
- [21] Paterson SJ, Heim S, Friedman JT, Choudhury N, Benasich AA. *Development of structure and function in the infant brain: implications for cognition, language and social behaviour*. Neurosci Biobehav Rev. 2006;30(8):1087-105. doi: 10.1016/j.neubiorev.2006.05.001. Epub 2006 Aug 4. PMID: 16890291; PMCID: PMC1933387. <https://pmc.ncbi.nlm.nih.gov/articles/PMC1933387/> (accessed: 03.11.2024)
- [22] Balasundaram P, Avulakunta ID. *Bayley Scales Of Infant and Toddler Development*. [Updated 2022 Nov 21]. In: StatPearls [Internet]. Treasure Island

(FL): StatPearls Publishing; 2024 Jan-. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK567715/> (accessed: 03.11.2024)

# Appendix

## A.1 Self-assessment for Human-Centered Artificial Intelligence Master's (HCAIM)

### A.1.1 Overview

This thesis explores the use of Graph Neural Networks to predict infant birth age and scan age based on diffusional connectomes (brain connectivity matrices). Additionally, it incorporates GradCAM-based explainability techniques to provide interpretability for the model's predictions. The medical data used as input and the implications of the explanations provided by GradCAM about the connectivity and state of brain regions places this work in the realm of human-centered AI applications.

Based on the Human-Centered Artificial Intelligence Master's (HCAIM) program's Diploma Thesis Guidelines<sup>1</sup>, this work adheres to the requirements for admissibility as a HCAIM diploma thesis, namely: it presents a machine learning-based solution to a given problem, and contains discussions about *Explainable AI aspects* (the entirety of the *Explainability* chapter), present in at least 10% of the scope of the thesis, closely related to the problem being solved.

In the following sections, I will present key parts of the self-assessment in accordance with principles laid out in the HCAIM program.

---

<sup>1</sup><https://hcaim.bme.hu/hu/msc-bme/msc-bme-phase2/> (accessed: 2024.12.11)

## A.1.2 Compliance with Human-Centered AI Principles

In this section, compliance with human-centered AI principles is evaluated through the list of factors described by *The Assessment List for Trustworthy Artificial Intelligence* (ALTAI), provided by the European AI Alliance<sup>2</sup>.

**Human Agency and Oversight** — The proposed AI system can act as a decision support tool for clinicians. Although the prediction target itself is not a diagnosis, but a biological attribute that is usually known to the clinician, the employed explainable AI (xAI) method proposes results about functional brain regions, which can narrow down further diagnostic inquiry when assessing possible conditions, like effects of premature birth. Furthermore, as stated in the Summary section of the Explainability chapter, based on the promising results of the age prediction, the training of the AI system can easily be extended (upon arrival of further subject metadata) to predict future motor and cognitive scores based on current brain state, which is a more direct diagnosis-related outcome and a source of new information for the health care practitioner analyzing the results. In this process, human oversight is key, and the decision always rests at the clinician (*human in the loop*), thus this solution should largely avoid problems that may arise when fully automated processes make decisions that might affect humans negatively.

**Privacy and Data Governance** — All data is publicly provided by the Developing Human Connectome Project (dHCP), who have taken extensive measures to anonymize infant subject data, minimize provided metadata to exclude all personally identifiable information (PII) and mitigate bias by sampling a large, diverse and representative cohort of neonates.

**Transparency** — The thesis tackles issues of explainability, with extensively documented design process. The GradCAM method, and its results are presented in a way for readers to understand the key features influencing the prediction outcome. The variety of possible explainability algorithms, as well as the strengths and eventual limitations of the chosen method are explained in a separate section of the relevant chapter. Transparent discussion is similarly presented about the input data and its acquisition, as well as limitations based on sample size.

**Fairness, Diversity, and Non-Discrimination** — The authors of the Developing Human Connectome Project have taken measures to ensure a diverse neonatal dataset in terms of socio-economic status, race, sex etc., which ensure that the AI system can be used on subjects with different backgrounds. Protected class

---

<sup>2</sup><https://futurium.ec.europa.eu/en/european-ai-alliance/pages/welcome-altai-portal> (accessed: 2024.12.11)

attributes are not released among subject metadata, so the functional equity or non-discrimination cannot be directly assessed, but it is highly probable that protected attributes and negative outcomes related to them are not included directly or through proxy variables in the training data, and thus there is no systematic, unfair bias present in the predictions.

**Societal and Environmental Well-Being** — The AI system and its transparency-related components, as well as its variants could be part of a medical diagnostic software in the future, providing positive impact in the field of neonatal healthcare, with minimal negative environmental impact (negligible training resource consumption).

**Accountability** — The whole scope of the thesis project includes publicly accessible open source code provided as an individual contribution, as well as heavy reliance on free and open source tools as dependencies, also providing extensible documentation of the design and implementation process. Any real-life application is only possible through *human-in-the-loop* procedures, with features and limitations made clear.

### A.1.3 Ethics Issues Checklist

Table A.1 presents a checklist of potential ethical issues that may arise in the case of an AI system and information about the relevance to the current work where needed. The review follows the 'ethics by design' paradigm to ensure compliance to EU law and directives is compatible with the conceptional design of the thesis.

### A.1.4 Conclusion

The thesis demonstrates adherence to the principles of Human-Centered AI by prioritizing transparency, fair use of data and accountability through open-source implementation as well as meaningful integration and discussion of xAI techniques, described in a separate chapter, with relevant results included to ensure human oversight into the developed AI system.

**Table A.1:** Ethics Issues Checklist for the current thesis.

Activity	Y/N	Information provided in the thesis	Documents to be provided as appendix
Does this activity involve the development, deployment and/or use of Artificial Intelligence-based systems?	Yes	Measures to avoid bias in data and to ensure proper, informed and ethical data acquisition is handled by the dHCP. Potential for ethical risk is low.	Risk assessment currently not needed.
Could the AI based system/technique potentially stigmatise or discriminate against people	No	-	-
Does the AI system/technique interact, replace or influence human decision-making processes	Limited	Human in the loop is default when used for medical decision support, as detailed in the chapter <i>Explainability</i> .	-
Does the AI system/technique have the potential to lead to negative social (e.g. on democracy, media, labour market, freedoms, educational choices, mass surveillance) and/or environmental impacts either through intended applications or plausible alternative uses?	No	-	-
Does this activity involve the use of AI in a weapon system?	No	-	-
Does the AI to be developed/used in the project raise any other ethical issues not covered by the questions above (e.g., subliminal, covert or deceptive AI, AI that is used to stimulate addictive behaviours, lifelike humanoid robots, etc.)?	No	-	-