Applications of Spatio-Temporal Graph Neural Network Models for Brain Connectivity Analysis

DISSERTATION

ZUR ERLANGUNG DES DOKTORGRADES
DER NATURWISSENSCHAFTEN (DR. RER. NAT.)
DER FAKULTÄT FÜR PHYSIK DER UNIVERSITÄT REGENSBURG

VORGELEGT VON

SIMON WEIN

AUS NÜRNBERG

2022

DAS PROMOTIONSGESUCH WURDE EINGEREICHT AM: 26.01.2022

DIE ARBEIT WURDE ANGELEITET VON:

Prof. Dr. Elmar W. Lang	(doctoral advisor)
Prof. Dr. Mark W. Greenlee	(co-supervisor)

PRÜFUNGSAUSSCHUSS:

Vorsitzender:	Prof. Dr. Tilo Wettig
Erstgutachter:	Prof. Dr. Elmar W. Lang
Zweitgutachter:	Prof. Dr. Mark W. Greenlee
Weiterer Prüfer:	Prof. Dr. Christian Schüller

Contents

List of Figures

List of Abbreviations

BOLD	**B**lood **O**xygen **L**evel **D**ependent
CE	**C**onnectome **E**mbedding
$CMRO_2$	**C**erebral **M**etabolic **R**ate of **O**xygen
CNN	**C**onvolutional **N**eural **N**etwork
DCM	**D**ynamic **C**ausal **M**odeling
DCRNN	**D**iffusion **C**onvolution **R**ecurrent **N**eural **N**etwork
DTI	**D**iffusion **T**ensor **I**maging
DWI	**D**iffusion **W**eigthed **I**maging
FC	**F**unctional **C**onnectivity
FID	**F**ree **I**nduction **D**ecay
fMRI	functional **M**agnetic **R**esonance **I**maging
GC	**G**ranger **C**ausality
GNN	**G**raph **N**eural **N**etwork
GWN	**G**raph **W**ave **N**etwork
GRU	**G**ated **R**ecurrent **U**nit
ICA	**I**ndependent **C**omponent **A**nalysis
LSTM	**L**ong **S**hort **T**erm **M**emory
MAE	**M**ean **A**bsolute **E**rror
MRI	**M**agnetic **R**esonance **I**maging
MSE	**M**ean **S**quared **E**rror
RNN	**R**ecurrent **N**eural **N**etwork
ROI	**R**egion **O**f **I**nterest
SC	**S**tructural **C**onnectivity
STGNN	**S**patio-**T**emporal **G**raph **N**eural **N**etwork
TCN	**T**emporal **C**onvolution **N**etwork
TE	**E**cho **T**ime
TR	**R**epetition **T**ime
VAR	**V**ector **A**uto **R**egression

Chapter 1

Preliminary Concepts

1.1 Introduction

Research in the notion of brain connectivity aims to improve our understanding on how neural populations, organized in complex networks, communicate and exchange their information in the human brain. Until now, different concepts of brain connectivity could provide us with distinct, but complementary aspects of the information processing in brain networks [128, 78]. On one hand, non-invasive neuroimaging techniques like functional magnetic resonance imaging (fMRI) allow us to temporally resolve dynamic neural activity distributions in distinct locations in the brain, when a subject is stimulated, performs a task or simply rests. Statistical approaches that describe the coherency of activity profiles in separate units of the nervous system were developed in the field of functional connectivity (FC) [113]. Such measures of coherency are highly time-dependent, and thereby characterize dynamic aspects of the communication in brain networks. On the other hand, neuroimaging modalities like diffusion weighted imaging (DWI) provide us a possibility to resolve aspects of structural organization of the brain [61]. By measuring the diffusion of water, this modality allows us to reconstruct tracks of white matter bundles, which form the structural substrate for the information exchange in brain networks. This type of connectivity between brain areas is usually denoted as the anatomical or structural connectivity (SC) and is considered as the static counterpart of functional connectivity. In comparison to rapid fluctuations in FC, SC is usually associated with alterations on considerably longer time-scales. Such fundamental changes in the brain structure are mainly related to the natural development of the brain, aging or disease [10, 62]. While correlation-based FC and SC constitute undirected measures of connectivity, a third category of brain connectivity was introduced, which deals with directed and potentially causal relationships between brain regions [113]. Such directed dependencies in brain networks are typically inferred from Granger causality or dynamic causal modeling [47]. Based on these different concepts of functional and structural brain connectivity, a central question in neuroscience is how the structure of the brain is related to its functions. Between regions with strong SC we can usually observe also a pronounced FC, but the inverse observation can not always be made [69]. Accordingly, the

structure-function relation in brain networks is apparently more complex and is still a current topic in brain connectivity research.

First, comprehending this interplay between different areas in brain networks can supplement our understanding of how information is distributed and processed in the human brain. On shorter timescales, changes in the functional organization of such brain networks can be related to different cognitive states of a subject, while on longer timescales, alterations of the functional and structural networks can be caused naturally by aging or learning [36]. Moreover, concepts of brain connectivity find initial applications in clinical research topics. For example changes in the structural and functional connectivity can serve as biomarkers for the diagnosis of multiple sclerosis [43, 84]. In Parkinson's disease, functional and structural connectivity profiles can be used as predictors for the outcome of the treatment with deep brain stimulation [70]. By this means, brain connectivity analysis has established itself as a useful research method for various applications in neuroimaging studies.

Recently, the analysis and processing of data with graph-like structures has also received increasingly attention in the field of machine learning [133, 22]. Artificial neural network based models were developed in the notion of graph neural networks (GNNs), which are able to effectively account for the non-Euclidean geometry in graph-structured data [23, 40]. This makes GNNs also interesting for applications in brain connectivity research [99, 126, 77, 8, 81, 75, 128], where the dynamic states of a brain network can be associated with graph-like signal distributions. On one hand, neural activity in different regions in the brain network would represent the temporally varying signal on the graph structure. On the other hand, edges in this graphical model of a the brain would then reflect the strength of interactions between segregated neural populations. For such dynamic graph signals a specific type of GNN has been developed, denoted as spatio-temporal graph neural network (STGNN). The idea of STGNNs is to simultaneously model spatial and temporal dependencies in graph-structured signals. As such they allow us to combine information on temporal neural dynamics, as observed in fMRI, with structural spatial information, derived from DWI. In this manner they are able to provide us a new possibility to study spatio-temporal dynamics in brain networks from a multi-modal perspective. The concept of this graphical representation of a brain state is further illustrated in figure 1.1.

As of date, in machine learning research different STGNN architectures have been developed, proposing different strategies to model the information propagation across time and space in graph-structured signals [133]. To capture spatial dependencies in graph-like signal distributions, driven by their success in computer vision, convolutional neural networks (CNNs) have been recently extended for data with graphical structures [40]. In a subsequent step, these graph convolutional networks have been combined with recurrent neural networks (RNNs) [100], which enable us to account for temporal

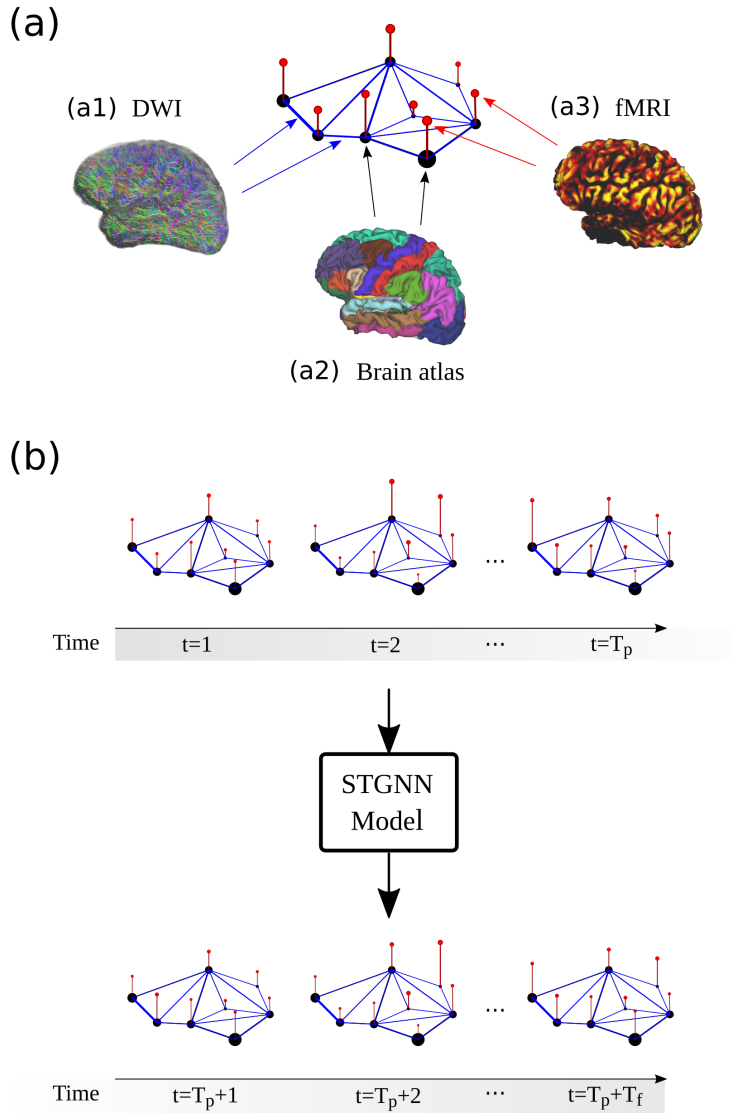

FIGURE 1.1: An illustration of a spatio-temporal graph signal is provided in (a). Based on the definition of a brain atlas (a2), the human brain can be segregated into functionally distinct regions, which are represented by the nodes in the graphical representation of a brain network. The neural signal strengths in these different regions or nodes of the network can be observed with neuroimaging techniques like functional magnetic resonance imaging (fMRI). An example of a volumetric fMRI image is shown in (a3). Dynamic functional interactions between different regions are spatially constrained by the structural layout of the brain. These structural white matter tracks can be reconstructed from diffusion weighted imaging (DWI), which would characterize the edge strengths in the brain graph (a1). These spatial and temporal dependencies in dynamic brain networks can be modeled with spatio-temporal graph neural networks (STGNNs), as illustrated in (b). The objective of a STGNN model is to predict from a sequence of T_p past brain network states a sequence of T_f future network states. Thereby the STGNN can learn to detect temporal and spatial dependencies in the imaging data, what allows us to study spatio-temporal characteristics of neural dynamics in brain networks.

relations in graph-structured signals. This variant of STGNN, denoted as diffusion convolution recurrent neural network (DCRNN) [82], will be the first architecture studied in its task of spatio-temporal modeling of neural dynamics in brain networks. As an alternative to this RNN based approach, one-dimensional convolutions were implemented in the so-called graph WaveNet (GWN) architecture to capture temporal dependencies in the graph signals [134]. In addition to these different temporal models, different strategies will be compared, to model the neural signal propagation between different brain regions in the network. Based on the idea that white matter tracks establish the physical substrate for the propagation of neural signals, the structural connectivity as observed in DWI will be incorporated as the substrate for the information exchange between brain regions. In a recent study, Rosenthal et al. [99] have shown that so-called connectome embeddings (CEs) of structural connectivity can inherently capture higher order topological relations between nodes in the structural brain graph. Therefore, as an alternative to the original structural connectivity, these CEs will be used to account for higher order transitions of information between regions in the structural network. Finally these scenarios are compared to the case when there is no predefined spatial layout integrated into STGNN models, thereby trying to learn all spatial relations between brain areas during the training of the STGNN model. Based on these comparisons the objective is first to identify the most efficient STGNN architectures to model spatial and temporal dynamics observed in complex brain networks.

In a next step these STGNN based approaches are then compared to the currently most popular data-driven approach for directed connectivity analysis in brain networks. The analysis with Granger causality follows the idea that if one event A would cause another event B, then event A should precede B, and the occurrence of event A should contain information about the occurrence of event B [9]. In the context of neuroimaging the idea of Granger causality is implemented in a predictive framework, by testing if adding information on the activity in a certain brain region A improves the prediction of activity in another region B. Until now the underlying predictive model in Granger causality is most often based on a vector auto regression (VAR) for multivariate timeseries forecasting [47, 9]. But in a brain network with N regions, the number of parameters which determine the coupling between all individual regions grow in a VAR with order N^2, so for larger brain networks it can be challenging to accurately fit the model if only limited data are available. This limitation is especially problematic in fMRI, where the temporal sampling rate is relatively low, while its high spatial resolution would allow for a detailed network analysis including a high number of regions N. Therefore it would be advantageous for fMRI studies to have a predictive model that learns interactions between all N brain areas of interest, and in addition naturally scales to larger brain networks. For this purpose the STGNN

approaches will be compared to a classical VAR model in this thesis. To account for different scenarios in their applications, the test accuracy of these models will be studied on a variety of network sizes and dataset sizes. The results will show that by learning localized functional interactions based on the structural network, STGNN models are able to accurately forecast functional neural dynamics, even when the brain network of interest becomes very complex and only few data are available to fit the model. This demonstrates that the STGNN approaches perform reliably among a large variety of fMRI study scenarios, and can also be utilized for investigations of smaller subject cohorts, like in studies of patients with rare neurological diseases.

In a subsequent step a concrete application of STGNN for directed connectivity analysis will be presented. In this context, a method is introduced to make spatial dependencies learned by STGNN models explainable, which allows us then to reconstruct interactions between brain regions captured in these models. By combining information inferred from fMRI and DWI data, these STGNN models can provide us a new multi-modal perspective on directed relations between individual areas in brain networks. Unlike the majority of current approaches that investigate the structure-function relation in the brain, which mostly try to infer only the overall functional connectivity from the structural graph [69, 39, 88, 41, 1, 12, 4], the STGNN models are able to directly replicate the observed neural activity distributions and their dynamic interactions on the structural substrate. Thus, this concept of simultaneously modeling spatio-temporal dynamics allows us to study the structure-function relationship in the human brain from a novel perspective.

Usually more complex machine learning models require a larger amount of data to achieve a very good performance, but often it is not economical feasible in MRI to conduct studies with very large sample sizes. To address such issues, a model training strategy will be presented that can improve the accuracy of STGNN models in small MRI studies. So-called transfer learning can enhance the performance of machine learning models by pre-training the model on a large dataset and transferring the acquired knowledge to the new target domain [94]. Based on this idea the DCRNN will be pre-trained on a publicly available large-scale dataset of 100 resting-state fMRI sessions provided by the Human Connectome Project [122]. It will be shown that this pre-training strategy can considerably improve the model accuracy on a smaller dataset of 10 MRI sessions collected at the Brain Imaging Center of the University of Regensburg. This demonstrates that the DCRNN is able to generalize across MRI scanner types and acquisition protocols, which grants us the possibility to apply transfer learning in this context of brain connectivity analysis.

Before discussing these possible applications of STGNNs in MRI, the thesis will at first in section 1.2 treat the general theory on artificial neural networks, which will provide a foundation for introducing the STGNN models. Further the relevant neuroimaging techniques will be outlined in section 1.3,

in addition with the currently established concepts of brain connectivity analysis in section 1.4. The second part of the thesis will then discuss in chapter 2 the concepts and results of STGNN applications in neuroimaging in more detail.

1.2 Neural Networks

The following section will introduce the basic concepts of *artificial neural networks*, and then focus on several neural network variants, which are relevant for applications in the field of brain connectivity. The derivations rely on the books of Goodfellow et al. [55] and Bishop [19], together with the publications of the original research. Research in artificial neural networks and research in neuroscience have a rich history of inspiring and influencing each other. On one hand early neural network models were designed to achieve a better understanding on how information is processed in biological brain networks [87, 98, 48]. On the other hand neural network algorithms nowadays have successfully contributed to our ability to process and interpret complex large-scale data, which are often available in neuroimaging studies [35, 127]. This thesis presents an application of artificial neural networks for analyzing and interpreting network structures observed in the human brain.

For different types and structures of data, specialized variants of artificial neural networks have been developed. *Feedforward neural networks* constitute the most elementary form of artificial neural networks, which will be introduced in section 1.2.1. The subsequent section 1.2.2 elaborates on current optimization techniques for successfully training neural network models in order to find relevant patterns in the data. In the subsequent parts some more specialized neural network architectures will be described. Section 1.2.3 starts to outline *convolutional neural networks* (CNNs), which are specifically designed for data with a grid-like structures. These principles are then generalized to graph-like structures in section 1.2.4 and subsequently in section 1.2.5 network architectures for sequential data structures are introduced. These geometric variants are then combined in the notion of *spatio-temporal graph neural networks* (STGNNs), which are designed to model dynamic signals on graph-like structures. In this manner, the theory on artificial neural networks will provide a foundation to understand their applications in brain connectivity analysis presented in the second part of this thesis in chapter 2.

1.2.1 Feedforward Neural Networks

Feedforward neural networks, also denoted as *multi-layer perceptrons* in literature, were the first and simplest form of artificial neural networks, and represent the elementary components of recent neural network architectures. In general a neural network model can be considered as an universal approximator for a function f^*. As an example we can consider a classification problem, where an input $\mathbf{x}$ is related to a category y via the function $y = f^*(\mathbf{x})$. Then a feedforward neural network can be used to learn a mapping $y = f(\mathbf{x}, \boldsymbol{\theta})$ by optimizing its parameters $\boldsymbol{\theta}$ such that it best approximates the underlying relation between observed variables y and $\mathbf{x}$. In a *feedforward* network model the information is propagated in one way from the input $\mathbf{x}$ through the computations performed by f to its output y, without any feedback or cyclic connections in the network. Such models $f(\mathbf{x})$ are typically composed of multiple functions $f^{(l)}$, which are then concatenated in a chain-like structure, like for example $f(\mathbf{x}) = f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x})))$. In this context the individual functions $f^{(l)}$ are referred to as *layers* of the neural network, and the number of layers define the *depth* of the network. Models with multiple layers are usually referred to as *deep* neural networks. They are loosely inspired by biological neural networks in the sense that the layers $f^{(l)}$ are often vector-valued functions, and each element of the vector could be associated with a neuron in the brain. Each such unit (neuron) receives its input from multiple other units (neurons), which then determine its output value (activity level). Therefore research in neural networks is often guided by insights drawn from neuroscience and biology, but in practical applications the main goal is still to acquire good statistical generalization, instead of representing an exact replication of biological intelligence.

The layer-wise structure for neural networks can be motivated by at first considering a single-layer network. Assume we would like to learn a relation f^* between some vector-valued P-dimensional data points $\mathbf{x} \in \mathbb{R}^P$ and its corresponding Q-dimensional output values $\mathbf{y} \in \mathbb{R}^Q$ by using a single-layer neural network denoted as $f^{(1)}(\mathbf{x}; \boldsymbol{\theta})$. With its parameters $\boldsymbol{\theta}$ consisting of $\mathbf{W}$ and $\mathbf{b}$, a single-layer neural network $f^{(1)}(\mathbf{x}; \mathbf{W}, \mathbf{b})$ can be defined as:

$$\mathbf{y} = \Phi(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{1.1}$$

The individual entries x_p of an observed data sample $\mathbf{x} \in \mathbb{R}^P$ are called the *features* of the dataset. The multiplication with a parameter matrix $\mathbf{W} \in \mathbb{R}^{Q \times P}$ yields then a weighted sum of the input features $\mathbf{x}$ and the parameters $\mathbf{W}$ are therefore referred to as *weights* of the neural network. The parameters in the second term $\mathbf{b} \in \mathbb{R}^Q$ add an additional offset to the input values and are denoted as *biases*. Further $\Phi(\nu)$ denotes the (element-wise applied) *activation function*, what is usually defined to be a non-linear function of its argument $\nu = \mathbf{W}\mathbf{x} + \mathbf{b}$. The structure of this single-layer network can be illustrated like in figure 1.2 (a). Typical functions to model the activation level $\Phi(\nu)$ include

the sigmoid function $\Phi(\nu) = \frac{1}{1+e^{-\nu}}$, the signum function $\Phi(\nu) = \text{sgn}(\nu)$, the hyperbolic tangent $\Phi(\nu) = \tanh(\nu)$, or the rectified linear unit (ReLU) function $\Phi(\nu) = \max(0, \nu)$. These commonly used activation functions $\Phi(\nu)$ are shown in figure 1.2 (c).

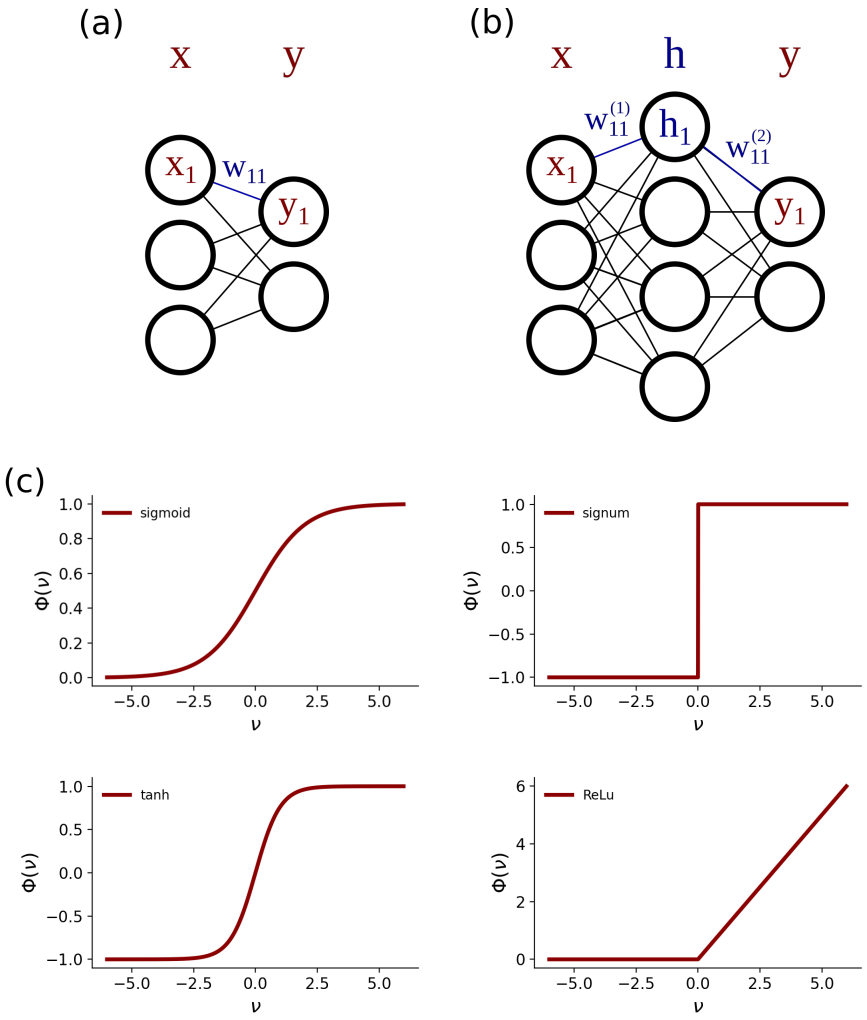

FIGURE 1.2: Figure (a) shows an example of a single-layer network. In this example an element x_1 of the input $\mathbf{x}$ is directly connected to an output neuron y_1 via a scalar weight w_{11}, as defined in equation 1.1. In a multi-layer network (b) an input x_1 is first mapped to an intermediate representation h_1 via a weight in the first layer $w_{11}^{(1)}$. Then this *hidden* state is connected to an output neuron y trough a weight in the second layer $w_{11}^{(2)}$. Figure (c) shows functions that are commonly used to define the activation level $\Phi(\nu)$ in neural networks.

A multi-layer network structure can then be obtained by using the output of single layer $\mathbf{h} = f^{(1)}(\mathbf{x})$ as an input for a subsequent layer $f^{(2)}(\mathbf{h})$. This intermediate representation $\mathbf{h}$ is seen only by the network itself and is therefore called the *hidden state*. The comparison between the structure of a single-layer network and a multi-layer network is illustrated in figure 1.2 (a) and (b). It is necessary to choose $\Phi(\nu)$ as a non-linear function, in order

to avoid that the concatenation of two functions $f(x) = f^{(2)}(f^{(1)}(\mathbf{x}))$ results again in a linear model. For example if we neglect the bias terms and define $f^{(1)} = \mathbf{W}^{(1)}\mathbf{x}$ and $f^{(2)} = \mathbf{W}^{(2)}\mathbf{x}$, then we would obtain the trivial representation $f(\mathbf{x}) = \mathbf{W}^{(2)}\mathbf{W}^{(1)}\mathbf{x} = \hat{\mathbf{W}}\mathbf{x}$. This shows that a non-linear activation function $\Phi(\nu)$ is required in order to also learn non-linear relations between the input variables $\mathbf{x}$. Concatenating multiple functions $f^{(l)}$ allows us to represent more complex patterns in the data, and enables us to also solve non-linear tasks. An illustration of the effect of using such hidden layers is given in figure 1.3. On the other hand using more layers $f^{(l)}$ increases the number of parameters and complexity of the neural network model, and makes it considerably more challenging to find the optimal parameters $\boldsymbol{\theta}$ to achieve a model with good statistical generalization. In the following section some techniques will be introduced, which were developed to account for such complex optimization problems.

1.2.2 Neural Network Training

The goal of training a neural network $f(\mathbf{x}, \boldsymbol{\theta})$ is to approximate some empirical relation $y = f^*(\mathbf{x})$ as accurately as possible by optimizing its model parameters $\boldsymbol{\theta}$. An essential aspect is therefore the definition of a suitable *cost function*[1] for this optimization problem. In most cases such functions can be derived from the maximum likelihood principle. If we define the probability distribution of our model outputs as $p_{model}(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$, we can obtain a cost function as the negative log-likelihood of the distribution:

$$J(\boldsymbol{\theta}) = -\mathbb{E}_{\mathbf{x},\mathbf{y}} \log p_{model}(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) \tag{1.2}$$

Here $\mathbb{E}_{\mathbf{x},\mathbf{y}}$ denotes the expectation value over samples $\mathbf{x}$ and respective targets $\mathbf{y}$. If we assume that the distribution of our model follows a Gaussian distribution $p_{model}(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{\mathbf{y}-f(\mathbf{x};\boldsymbol{\theta})}{\sigma})^2}$ we would then recover the *mean squared error* (MSE) cost function:

$$J(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_{\mathbf{x},\mathbf{y}} \|\mathbf{y} - f(\mathbf{x};\boldsymbol{\theta})\|_2^2 + const \tag{1.3}$$

with a scaling factor of $\frac{1}{2}$ and a constant term, which is independent of the models parameters $\boldsymbol{\theta}$. Another cost function can be derived from the family of exponential distributions by replacing the squared L^2 norm in the argument of the Gaussian distribution by a L^1 norm. This yields then the *mean absolute error* (MAE):

$$J(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x},\mathbf{y}} \|\mathbf{y} - f(\mathbf{x};\boldsymbol{\theta})\|_1 \tag{1.4}$$

Based on the learning task, a suitable cost function can be designed, and besides those already introduced above, also entropy-like functions and other

[1] In machine learning literature the cost function of neural network optimization is also often denoted as *objective function* or *error function*.

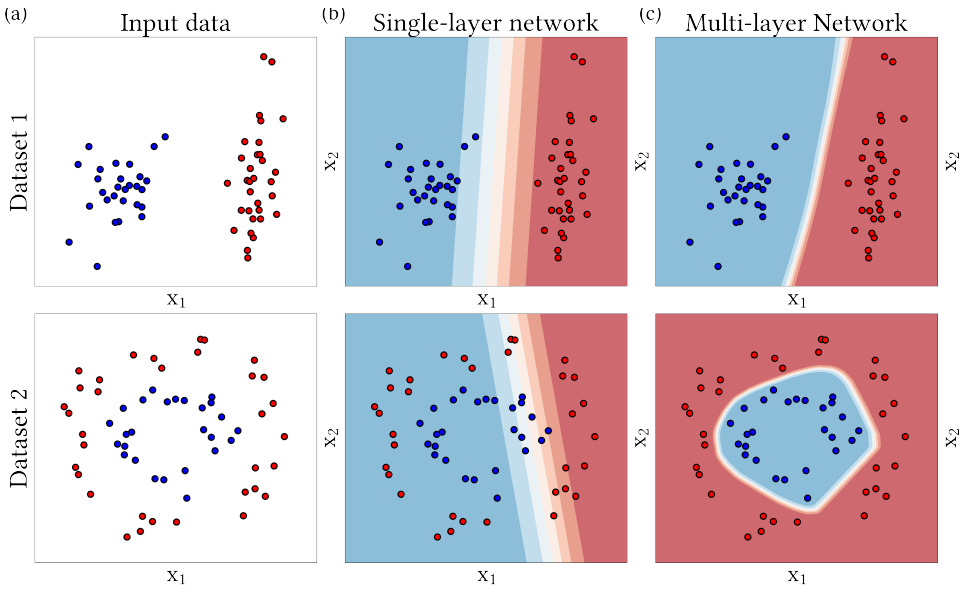

FIGURE 1.3: This figure illustrates the differences between a single and multi-layer neural network in their capabilities of learning structures in data. In this example the data samples $\mathbf{x} \in \mathbb{R}^2$ are points in a two-dimensional space, as shown in (a). The target values $\mathbf{y} \in \mathbb{R}^1$ are either 0 (depicted in blue) or 1 (depicted in red). In dataset 1, in the top row, the two categories of data points can be separated linearly, while in dataset 2, in the bottom row, there exists a more complex relationship between the features x_1 and x_2. The predictions of a single-layer network are illustrated in the second column in (b), after fitting the neural network parameters to the observed data. The predicted values of the neural network are illustrated by the blue and red contours across the feature space, whereby blue contours illustrate the areas where the network predicts a 0 and red areas where its output is 1. It is apparent that a single-layer network can successfully solve the linearly separable problem, but it is not able to learn the non-linear relationship between x_1 and x_2 in dataset 2. In contrast thereto, the last column (c) shows the predictions of a multi-layer network, which consists of an input and output layer, and additionally a hidden layer $\mathbf{h} \in \mathbb{R}^{100}$. By learning an intermediate non-linear representation $\mathbf{h}$, such a multi-layer network architecture can also solve the non-linear problem given in dataset 2.

specialized cost functions have been developed [55]. Because they were not used for the applications in this thesis, they will here not be outlined in more detail.

Using a nonlinear neural network model for $y = f(\mathbf{x}, \boldsymbol{\theta})$ causes the cost functions $J(\boldsymbol{\theta})$ to become non-convex, which makes it considerably more challenging to find optimal parameters $\boldsymbol{\theta}$ than in linear models. The problem can be visualized by considering a geometric view of the cost function landscape, like that shown in an example in figure 1.4. Because it is not possible to find analytic solutions for such optimization problems in most cases, specialized algorithms have been developed for the training of neural networks, which mainly emerged from gradient-based optimization.

Suppose we would like to minimize the cost function with respect to the

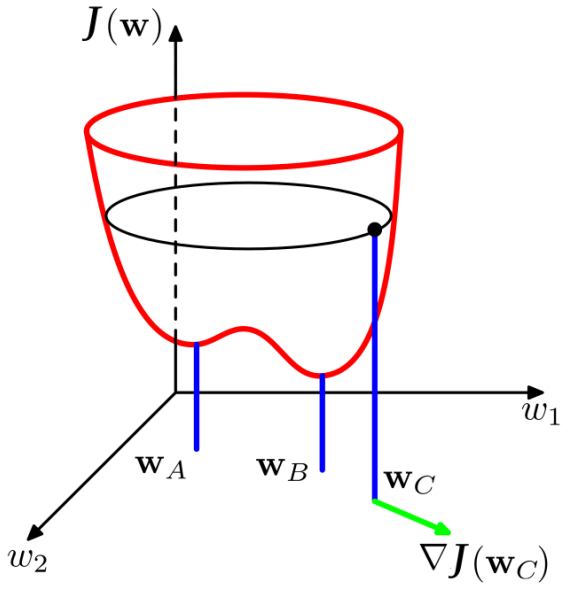

FIGURE 1.4: An example of a cost function is illustrated in this figure. If we would like to find optimal values for some network weights $\mathbf{w}$, then the values of the non-convex cost function $J(\mathbf{w})$ can be viewed as a surface on the model weight space $\mathbf{w}$, as illustrated in red. A local optimum is located in point $\mathbf{w}_A$ and the global optimum can be found in point $\mathbf{w}_B$. The local gradient of the cost function in any point $\mathbf{w}_C$ can be represented by a vector $\nabla J(\mathbf{w}_C)$, shown here in green, which points into the direction of steepest increase. Adapted from [19].

parameters $\boldsymbol{\theta}$, then the gradient $\mathbf{g} = \nabla_\theta J(\boldsymbol{\theta})$ would point into the direction of greatest increase of the cost function $J(\boldsymbol{\theta})$ with respect of $\boldsymbol{\theta}$, like that shown in figure 1.4. Accordingly, taking a small step into the direction of $-\nabla_\theta J(\boldsymbol{\theta})$ would reduce the cost. To obtain a scalar value for the cost function $J(\boldsymbol{\theta})$ during learning, the initial information in form of inputs $\mathbf{x}$ is propagated through all hidden layers in the network to obtain the predictions $\mathbf{y}$ of the neural network, which is denoted as *forward propagation*. In a second step, to derive the gradients of the cost function, the information is passed backwards through the network. This algorithm to compute the gradients is therefore called *back-propagation*, and is described in more detail in Appendix A.1. Using these gradients $\mathbf{g} = \nabla_\theta J(\boldsymbol{\theta})$ allows us then to deduce different learning rules for optimizing the models parameters θ. Based on this idea, a simple iterative update rule denoted as *gradient descent* can be defined as:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \tag{1.5}$$

The parameter η in this equation determines the step size of the parameter update and is referred to as *learning rate* of the algorithm. The original cost functions $J(\boldsymbol{\theta})$ include the expectation value $\mathbb{E}_{\mathbf{x},\mathbf{y}}$ across the complete set of data samples and labels $\mathbf{x}, \mathbf{y}$ (as defined in equations 1.3 and 1.4), but computing the exact gradient can be become computationally very expensive for

large datasets. Therefore most often the gradient is evaluated on a subset of data samples denoted as *minibatch*. Additionally using small batches of training samples has a regularization effect on the optimization and can be beneficial for the generalization performance of neural network models [131]. By drawing a random subset of samples $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(S)}\}$ an unbiased estimate of the gradient $\hat{\mathbf{g}}$ can be computed as:

$$\hat{\mathbf{g}} = \frac{1}{S} \nabla_{\boldsymbol{\theta}} \sum_{s=1}^{S} J(f(\mathbf{x}^{(s)}, \boldsymbol{\theta}), \mathbf{y}^{(s)}) \tag{1.6}$$

And the model parameters $\boldsymbol{\theta}$ can be updated respectively with:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \hat{\mathbf{g}} \tag{1.7}$$

Such gradient-based techniques do not guarantee that we will find the global minima for such optimizations problems, but for many applications local minima often have a sufficiently low value to achieve a satisfactory model performance [38, 55].

In areas of flat spots in the cost function landscape the gradient of the cost function $\nabla_{\theta} J(\boldsymbol{\theta})$ can become very small, and using an update rule like in equation 1.7 would slow down the learning process of the model. To address this problem, the concept of *momentum* learning was introduced. In analogy to physics the gradient of the algorithm can be interpreted as force which accelerates the learning according to the following update rule:

$$\mathbf{v} \leftarrow \alpha \mathbf{v} - \eta \frac{1}{S} \nabla_{\boldsymbol{\theta}} \sum_{s=1}^{S} J(f(\mathbf{x}^{(s)}, \boldsymbol{\theta}), \mathbf{y}^{(s)}) \tag{1.8}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{v} \tag{1.9}$$

with velocity $\mathbf{v}$, a parameter $\alpha \in [0, 1)$ that describes the weight decay of the subsequent update step, and a parameter η that determines the contribution of the new gradient. Increasing the value of α would lead to an update which points more strongly into the direction of the previous gradient, so if the successive gradients are aligned to each other, such update would accelerate the learning into the prevailing direction. This approach helps us to mitigate the problem of finding the proper learning rate, but comes with the cost of introducing an extra hyperparameter, which can affect the optimization process. In addition, individual parameters might require different learning rates in order to optimally converge.

This gave rise to the idea of using separate learning rates for each parameter individually, and automatically adapting those during the optimization process. Based on this concept various algorithms have been developed [102] and one of the most prominent is the *adaptive moments* algorithm, denoted as the *Adam* optimizer [76]. Similar to the concept of momentum learning introduced above, this algorithm keeps an exponentially decaying average of gradients $\mathbf{g}$ in the past. At first the Adam optimizer computes the decaying

averages of the gradient and the squared gradient as described in the following:

$$\mathbf{s} \leftarrow \rho_1 \mathbf{s} + (1 - \rho_1)\mathbf{g} \tag{1.10}$$

$$\mathbf{r} \leftarrow \rho_2 \mathbf{r} + (1 - \rho_2)\mathbf{g} \odot \mathbf{g} \tag{1.11}$$

were $\mathbf{s}$ and $\mathbf{r}$ represent the estimates for the first moment (mean) and second moment (uncentered variance) of the gradients $\mathbf{g}$. In equation 1.11 the symbol $\odot$ denotes the elementwise (Hadamar) product. The parameters $\rho_1, \rho_2 \in [0, 1)$ determine the decay rate for the moment estimates. Initializing $\mathbf{s}$ and $\mathbf{r}$ with 0 can lead to a bias in the early training towards 0, which motivates us to introduce the following bias correction in dependence of training time step t:

$$\hat{\mathbf{s}} \leftarrow \frac{\mathbf{s}}{1 - \rho_1^t} \tag{1.12}$$

$$\hat{\mathbf{r}} \leftarrow \frac{\mathbf{r}}{1 - \rho_2^t} \tag{1.13}$$

With these bias corrected moments the model parameters $\boldsymbol{\theta}$ can then be updated with:

$$\Delta\boldsymbol{\theta} = -\eta \frac{\hat{\mathbf{s}}}{\sqrt{\hat{\mathbf{r}}} + \delta} \tag{1.14}$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta} \tag{1.15}$$

where the constant δ is chosen to be a small number to stabilize the division in 1.14, usually set to 10^{-6} and η represents the global learning rate. Such algorithms with adaptive learning rates have been shown to perform well on a large variety of tasks [102], but another crucial aspect that can determine the result of the optimization is the initialization strategy of the model parameters.

The learning strategies introduced above are all based on iterative update rules, and therefore require the user to predefine initial values for the parameters $\boldsymbol{\theta}$. This initial point can have a strong influence on how quickly an optimization algorithm converges or even, if it converges at all. One crucial property, which the weights w of a neural network should have, is that activations in the forward pass and the gradients in the backward pass should not explode nor vanish. A popular initialization strategy based on this idea was proposed by Glorot and Bengio [52], who recommended to initialize weights w of a layer with Q_{in} input neurons and Q_{out} output neurons as the following:

$$w \sim p_U\left(\frac{\sqrt{6}}{Q_{in} + Q_{out}}\right) \tag{1.16}$$

Here $p_U(a)$ denotes a uniform distribution in the interval from $-a$ to a. Glorot and Bengio [52] have showed in their study that initializing w by sampling from this distribution allows us to preserve the variance in the forward and in

the backward pass at the beginning of model training. Originally the initialization strategy in equation 1.16, was derived for neural networks containing only linear activation functions, but this principle has shown to be useful for also non-linear models [55]. However for highly non-linear functions like the ReLU activation function (as illustrated figure 1.2 (c)), an improvement was proposed by He et al. [65]. They could demonstrate that the variance in the forward and backward pass in networks with ReLU activations can be preserved by initializing the weights w with:

$$w \sim p_G\left(\sqrt{\frac{2}{Q_{in}}}\right) \tag{1.17}$$

Whereby $p_G(a)$ denotes a zero-mean Gaussian distribution with variance a. This strategy could improve the convergence in large multi-layer neural networks with ReLU activation functions and is typically denoted as *Kaiming* or *He* initialization [65].

A sometimes even more efficient strategy used to initialize the neural network parameters is offered by the so-called *transfer learning*. This learning strategy uses the knowledge gained from training in one domain and transfer this knowledge to a second domain, with the goal to improve the performance in the new learning task. The intuition behind this can be illustrated by an example in computer vision, if one for instance has the goal to detect foxes in an image. If for the learning task only few example images of foxes are available, one could at first train the model on a large dataset of cat images, and use this prior information contained in the pretrained model weights to learn to recognize foxes in the new task. More formally speaking we can define a source domain dataset as $D_S \in \{(x_S^{(1)}, y_S^{(1)}), \ldots, (x_S^{(N)}, y_S^{(N)})\}$, where $x_S^{(n)}$ represent data samples and $y_S^{(n)}$ the corresponding labels. We can first learn a function $f_S(x_S; \theta_S)$ which predicts the labels $y_S^{(n)}$ from data samples $x_S^{(n)}$ in the source domain. If we have our target domain data $D_T \in \{(x_T^{(1)}, y_T^{(1)}), \ldots, (x_T^{(N)}, y_T^{(N)})\}$ we can then try to improve the generalizability of our target function $f_T(x_T; \theta_T)$ by using prior knowledge obtained from our learning problem $f_S(x_S; \theta_S)$ in D_S. This can be achieved by using the parameters θ_S of our model in the source domain, and initialize with these parameters the training of our target function $f_T(x_T; \theta_T)$. The hope is that some factors which determine the relation between $y_T^{(n)}$ and $x_T^{(n)}$ can also be found in the relation between $y_S^{(n)}$ and $x_S^{(n)}$. For example some basic visual shapes which are relevant to recognize a cat might be similar to the patterns which can be used to detect a fox.

1.2.3 Convolutional Neural Networks

For data with a fixed, grid-like geometry, specialized neural network architectures were developed in the notion of *convolutional neural networks* (CNNs) [48, 80]. Typical examples of data with such a geometry are images, which can be considered as a 2-dimensional grids of pixel values, or time-series data, which could be thought of as 1-dimensional grids of points regularly sampled in time. As their name already suggests, they are based on the mathematical convolution operation, which can be defined as the product of two functions $x(t)$ and $\theta(t)$ after the function $\theta(t)$ is flipped and shifted:

$$(x * \theta)(t) = \int x(\tau)\theta(t-\tau)\partial\tau \tag{1.18}$$

In the context of neural networks the first argument x is here referred to as the *input* to the convolution and the second argument θ as the *kernel*. The *flipping* of the kernel in a convolution is based on the property that if τ increases, the index of the input $x(\tau)$ increases, while the index of the kernel $\theta(t-\tau)$ decreases. In practice the input is most often data sampled from discrete measurements, so therefore we can correspondingly use the discrete convolution defined as:

$$(x * \theta)(t) = \sum_{\tau=-\infty}^{\infty} x(\tau)\theta(t-\tau) \tag{1.19}$$

In our context, the input is a multidimensional array of data, and the kernel represents a multidimensional array of parameters, which are adapted during the training of the neural network model. This operation can then further be generalized to higher-dimensional data structures, for example to a convolution of a two-dimensional image X with a two-dimensional kernel Θ:

$$(X * \Theta)(i,j) = \sum_n \sum_m X(m,n)\Theta(i-m, j-n) \tag{1.20}$$

An example of such a convolution operation with a two-dimensional kernel is visualized in figure 1.5.

The motivation to employ convolution operations in neural networks has multiple facets. First it enables us to gain *sparse interactions* between input and outputs units. As introduced in section 1.2.1, in a fully connected network layer with P input and Q output units the number of weight parameters would sum up to $P \cdot Q$. But often it is sufficient for a unit to detect only small local features, for example in an image it would be possible to detect contrast edges by considering only a small number of pixels. Therefore one can choose the size of a kernel K usually to be much smaller than the input size $K \ll P$, which significantly reduces the number of parameters and the computational cost of the model.

The second principle entailed by convolutional neural networks is *parameter sharing*. By using convolutions each parameter of the kernel is applied at

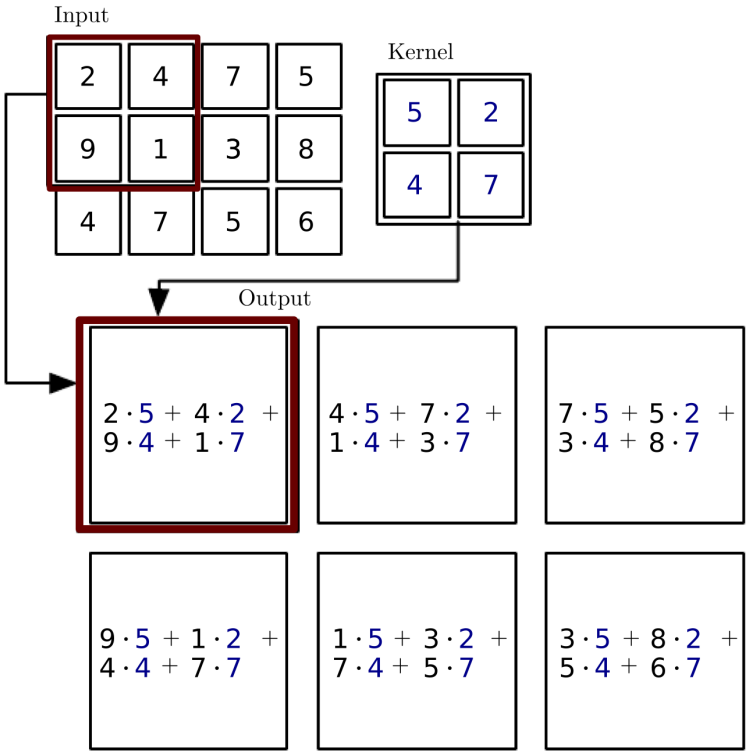

FIGURE 1.5: This figure shows an example of a two-dimensional convolution operation. The input array with size 3×4 is convolved with a 2×2 kernel, so shifting the kernel to every position of the input results in a 2×3 output array. The elements of the kernel are adapted by the CNN during the training and by evaluating the similarity of the kernel with the input at every position, the CNN can learn to detect patterns in all different locations of the input array. Note that in violation to the formal definition of a convolution, the flipping of the kernel is here omitted, as the values of the kernel are ultimately adapted by the algorithm during training. The operation without flipping of the kernel is denoted as *cross-correlation*, but in the context of machine learning it is usually equivalently referred to as *convolution*. Adapted from [55].

every position of the input in a layer[2]. This principle of parameter sharing is illustrated in figure 1.6. By learning only one set of parameters, instead of, as in fully connected networks, learning a separate parameters for every location, the storage and memory requirements of the model parameters can be reduced considerably.

This form of parameter sharing leads to another characteristic of CNNs referred to as *equivariance* to translation. If we characterize the convolution operation by a function f and define a second function g that translates its input, then we would obtain the equivariance property $f(g(x)) = g(f(x))$. This implies that if we shift our input, this would shift the output of the convolution layer the same way. This property is useful for instance when detecting

[2]Note that the kernel can be also applied at the boundary of a data array by adding proper boundary conditions, for example by padding an image with zeros.

edges in an image, which can appear at multiple locations, what motivates to share the parameters of the kernel across the whole input space.

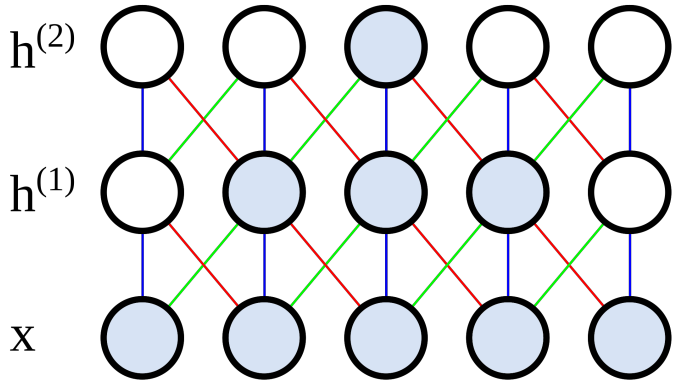

FIGURE 1.6: This figure illustrates the principle of parameter sharing like that used in CNNs. In this example each unit in the first hidden layer $\mathbf{h}^{(1)}$ is connected with 3 weights to the input $\mathbf{x}$. These weights are shared across the units in layer $\mathbf{h}^{(1)}$ and in this illustration shared connections are depicted in the same color. This example resembles a convolution with a kernel of size of $K = 3$. Multiple of such convolution layers can be stacked in neural network architectures to obtain more complex representations of the input. Even though each of the single units has only sparse connections, units in higher layers can be indirectly connected to a larger portion of the input. The region of the input that affects a neuron is called the *receptive field*, which grows for neurons in higher layers. The receptive field of a unit in the layer $\mathbf{h}^{(2)}$ is highlighted in blue in this figure.

In practice we do not only apply a single convolution, but multiple convolution operations taking place in parallel as introduced in equations 1.19 and 1.20. Using a number of such convolutions simultaneously, with different parameterized convolution kernels $\theta^{(q)}$, allows us to detect distinct features in the data. The outputs of such a convolution operation are usually referred to as *feature maps*. By applying Q convolution operations simultaneously, we would obtain Q different feature maps, whereby each feature map can capture different characteristics of the data. For example if our convolution kernels were trained to simply detect local contrast edges, one feature map could detect horizontal edges in an image and another feature map the vertical edges.

Another technique which is often used in CNN architectures is the *pooling* operation. The idea of incorporating pooling is to create a summary statistic of nearby neurons in a CNN layer. This can be achieved by for example *max pooling*, which only selects the neuron with highest activation within a certain neighborhood. The principle of max pooling is illustrated in figure 1.7 (a). An alternative thereto is *average pooling*, where the average value of a neighborhood is computed. The effect of incorporating pooling is also that learned representations of the data become relatively invariant to small translations. This is convenient in computer vision tasks for example, where it is more import if a feature is present in an image, but the exact location is less

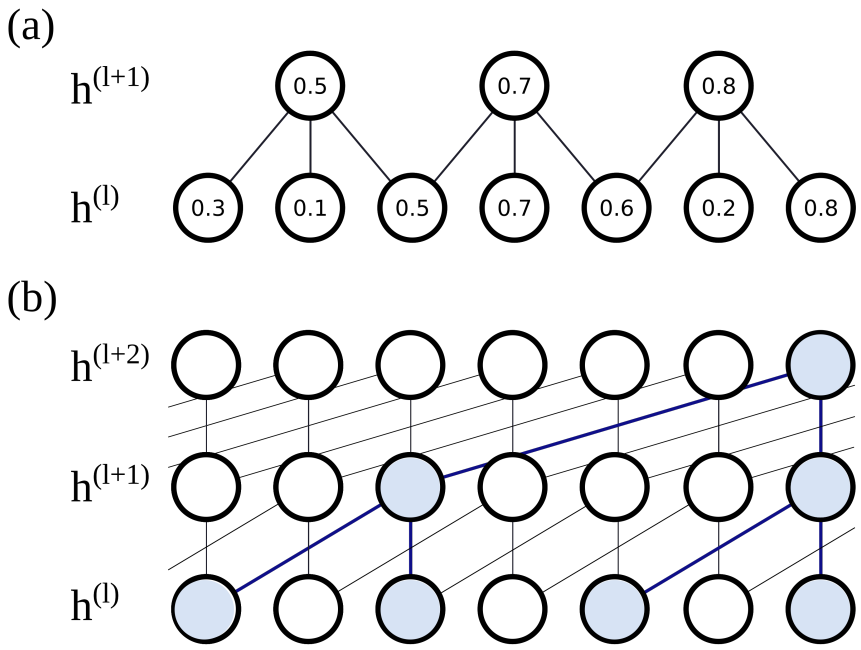

FIGURE 1.7: In figure (a) the principle of max pooling is shown. Max pooling aggregates the activity of layer $\mathbf{h}^{(l)}$ by taking its maximum value within a pre-defined neighborhood of 3 in this example. The distance between the pooling regions here is $d = 2$, which reduces the number of neurons from 7 to 3 in the subsequent layer $\mathbf{h}^{(l+1)}$. Another possibility to process the input on a coarser scale is to utilize dilated convolutions, as illustrated in (b). In this example a dilation factor of 2 is used for the convolution between layer $\mathbf{h}^{(l)}$ and $\mathbf{h}^{(l+1)}$ and a factor of 4 between $\mathbf{h}^{(l+1)}$ and $\mathbf{h}^{(l+2)}$. This allows the receptive field to grow exponentially fast in higher network layers, highlighted here in blue, and additionally helps to preserve the order structure of the inputs.

relevant. Creating such summaries of neurons we can then effectively reduce the number of parameters in a model, if we choose the distance between two pooling regions as d, we have roughly d times fewer neurons to process in the subsequent layer, as shown in figure 1.7 (a). On the contrary in some applications it is crucial to preserve the exact ordering of the data input, for example in time series analysis, where certain events in the past can be identified as the potential cause for some events in the future. For processing temporally structured data with CNNs, the *WaveNet* architecture was proposed by van de Oord et al. [121]. The main idea of this CNN architecture is to implement so-called *dilated causal convolutions*, which are depicted in figure 1.7 (b). Such convolutions skip input values with a certain step, what allows the receptive field then to grow exponentially fast in higher layers of the neural network. The causal convolution operation $*_C$ can be derived from equation 1.19 by additionally introducing a dilation factor d:

$$(x *_C \theta)(t) = \sum_{\tau} x(\tau)\theta(t - d \cdot \tau) \tag{1.21}$$

Using a dilation factor of $d = 1$ would resemble again a standard convolution. Figure 1.7 (b) illustrates the convolution when considering a dilation factor of 2 between layer $\mathbf{h}^{(l)}$ and layer $\mathbf{h}^{(l+1)}$, and a dilation factor of 4 between layer $\mathbf{h}^{(l+1)}$ and layer $\mathbf{h}^{(l+2)}$. Similar to pooling this principle enables the neural network to process the data on a coarser scale, thereby using only a few hidden layers and also preserving the causal structure of the data.

Several aspects of CNNs are inspired by findings in neuroscience research on the human visual system [48]. Some of those principles implemented by CNNs can be roughly compared to the processing in the primary visual cortex, also denoted as V1. In a simplified view, V1 can be considered as the area in the brain that performs the first significant processing of the visual input. For instance V1 has a two-dimensional structure, which mirrors the mapping of a an image in the retina. Similar to this characteristic, also CNNs for image processing have arranged their features in two-dimensional maps, as illustrated in figure 1.5. Further by moving to higher layers in an artificial neural network, we can find neurons that encode higher level concepts of the network input from a composition of simple representations in the lower layers. This concept is illustrated in figure 1.6, and can be compared to hierarchical processing also appearing in the ventral stream. These principles have helped CNNs to become very successful in image processing problems, what encouraged to transfer these ideas also to other geometric domains. While convolution operations can be well defined on Euclidean data with grid-like structures like images, it is less intuitive to define such convolution operations in the irregular domain of graphs. Only relatively recently practical concepts have been developed to generalize CNNs for data on a graph-like geometries, which will be the focus in the following section.

1.2.4 Graph Neural Networks

Inspired by their numerous applications of CNNs in computer vision and language processing, recently the interest emerged to modify CNNs to also operate in the non-Euclidean domain of graphs. Data with graph-like structure can be found in various domains, for instance in citation networks with papers interconnected via citationships, biochemistry where the structure of molecules can be represented as graphs, or in neuroscience where neural interactions can be described by functional brain networks. Such architectures which are designed to deal with graphical representations of data are developed in the notion *graph neural networks* (GNNs) [133]. But unlike data on a regular grid, graphs can have an irregular structure with a varying number of unordered nodes, where each node can have a different number of neighboring nodes. This renders it considerably more challenging to define useful operations like convolutions in the irregular domain of graphs. A comparison between a regular grid structure and an irregular graph structure is depicted in figure 1.8.

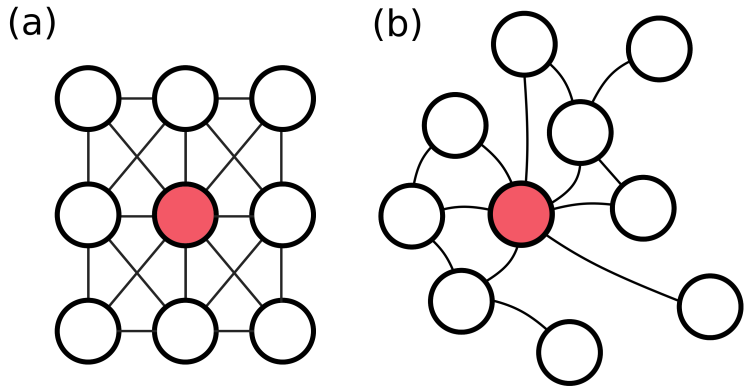

FIGURE 1.8: Comparison of a regular grid with a graph-like structure. Data structures like images can be interpreted as regular grids (a), thereby representing a special case of a graph structure. In this case each node (marked in red in this example) has a regular neighborhood. In contrast thereto, in an irregular graph structure, as illustrated in (b), each node can have a varying number of neighbors.

In graph signal processing we can define a graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, with $\mathcal{V}$ denoting a set of $|\mathcal{V}| = N$ nodes (or vertices), $\mathcal{E}$ representing a set of corresponding edges, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ denoting the *weighted adjacency matrix*. One entry $w_{nn'}$ of the adjacency matrix $\mathbf{A}$ would indicate the connection strength between node n and node n' of a graph $\mathcal{G}$. Then a feature or signal $\mathbf{x} : \mathcal{V} \rightarrow \mathbb{R}$ on the nodes of the graphs can be defined as a vector $\mathbf{x} \in \mathbb{R}^N$ where one entry x_n describes the signal strength in node n. As we cannot simply define a meaningful translation operation in the non-Euclidian geometry of graphs, the graph convolution operation $*_\mathcal{G}$ is defined by exploiting the *graph Fourier transform* of the signal. This transformation can be defined using the combinatorial *graph Laplacian* operator [33]:

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \tag{1.22}$$

where $\mathbf{D} \in \mathbb{R}^{N \times N}$ represents the diagonal degree matrix with its entries obtained as $d_{nn} = \sum_n W_{nn'}$. The normalized graph Laplacian can then be defined as:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \tag{1.23}$$

with $\mathbf{I} \in \mathbb{R}^{N \times N}$ representing the identity matrix. The graph Laplacian $\mathbf{L}$ is defined as a real symmetric positive semidefinite matrix, and therewith a complete set of orthonormal eigenvectors $\mathbf{u}_n \in \mathbb{R}^N$ can be associated with $\mathbf{L}$. The eigenvectors $\mathbf{u}_n$ are denoted as *graph Fourier modes*, and its corresponding eigenvalues λ_n are called the *frequencies* of the graph. The Laplacian can be diagonalized by the Fourier basis $\mathbf{U} = [\mathbf{u}_0, \ldots, \mathbf{u}_{N-1}] \in \mathbb{R}^{N \times N}$ with:

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \tag{1.24}$$

where $\mathbf{\Lambda} = \text{diag}([\lambda_0, \dots, \lambda_{N-1}]) \in \mathbb{R}^{N \times N}$ denotes the eigenvalue matrix. Incorporating this basis allows us to define the graph Fourier transform of a signal $\mathbf{x}_t$ as:

$$\mathbf{x}_w = \mathbf{U}^T \mathbf{x}_t \tag{1.25}$$

and its inverse:

$$\mathbf{x}_t = \mathbf{U}\mathbf{x}_w \tag{1.26}$$

This enables to define a graph convolution operator $*_G$ in the Fourier domain [23], obtained as:

$$\begin{aligned} \mathbf{y}_t &= \mathbf{x}_t *_G \mathbf{f}_\theta && (1.27)\\ &= \mathbf{U}((\mathbf{U}^T \mathbf{f}_\theta) \odot (\mathbf{U}^T \mathbf{x}_t)) && (1.28)\\ &= \mathbf{U}(\boldsymbol{\theta}_w \odot \mathbf{x}_w) && (1.29) \end{aligned}$$

where $\mathbf{f}_\theta$ represents a graph filter parameterized by θ and $\odot$ the Hadamar product in the conjugate domain. The parameterized filters θ_n are captured in the conjugate vertex domain by the vector $\mathbf{U}^T \mathbf{f}_\theta \equiv \boldsymbol{\theta}_w = [\theta_1(\omega), \dots, \theta_N(\omega)]^T$. If it is replaced by a diagonal matrix of free parameters, which can be learned by the model $\boldsymbol{\theta}_w \rightarrow \mathbf{\Theta}_w = \text{diag}(\theta_1(\omega) \dots \theta_N(\omega))$, it resembles a convolution kernel and we obtain for the filtered signal:

$$\mathbf{y}_t = \mathbf{U}\mathbf{\Theta}_w \mathbf{U}^T \mathbf{x}_t \tag{1.30}$$

Learning such filters depends on the number of nodes N in the graph, which can be computationally expensive for large graph structures. Therefore it would be desirable to have geometric properties similar to CNNs, as introduced in the previous section 1.2.3. To reduce the learning complexity and to obtain filters, which are strictly localized in space, Defferrard et al. [40] proposed to approximate the filter kernel by an orthogonal basis of Chebyshev polynomials. The Chebyshev polynomial $C_k(x)$ of order k can be computed by the recurrence relation $C_k(x) = 2xC_{k-1}(x) - C_{k-2}(x)$ with $C_0 = 1$ and $C_1 = x$. Accordingly, the graph filters can then be approximated be the following truncated expansion of polynomials:

$$\mathbf{\Theta}_w = \sum_{k=0}^{K} \theta_k C_k(\tilde{\mathbf{\Lambda}}) \tag{1.31}$$

of order K. Here the parameters θ_k represent the Chebyshev coefficients and matrix $C_k(\tilde{\mathbf{\Lambda}}) \in \mathbb{R}^{N \times N}$ is the Chebyshev polynomial of order k, with $\tilde{\mathbf{\Lambda}} = 2\mathbf{\Lambda}/\lambda_{max} - \mathbf{I}$ denoting the eigenvalues re-scaled between $[-1, 1]$. Using this polynomial filter defined in equation 1.31 and applying the transformation of the graph signal in equation 1.30 leads to the following relation:

$$\mathbf{y}_t = \sum_{k=0}^{K} \theta_k \mathbf{U} C_k(\tilde{\mathbf{\Lambda}}) \mathbf{U}^T \mathbf{x}_t \tag{1.32}$$

$$= \sum_{k=0}^{K} \theta_k C_k(\tilde{\mathbf{L}}) \mathbf{x}_t \tag{1.33}$$

with $\tilde{\mathbf{L}} = 2\mathbf{L}/\lambda_{max} - \mathbf{I}$ representing the scaled Laplacian. In equation 1.33 we obtain a spectral formulation of graph filtering, which is strictly localized in space, i.e. restricted to K steps from the central vertex. This also effectively reduces the learning complexity, because often only the local neighborhood is most relevant for extracting features for a node, and K can be chosen with an order $K \ll N$. In analogy to classical CNNs, multiple graph convolutions q can be incorporated to learn different feature representations on the graph structure, and by including a suitable non-linear transformation $\sigma(\cdot)$, the hidden state in a graph neural network can be obtained as follows:

$$\mathbf{h}_t^{(q)} = \sigma\left(\mathbf{y}_t^{(q)}\right) = \sigma\left(\sum_{k=0}^{K} \theta_k^{(q)} C_k(\tilde{\mathbf{L}}) \mathbf{x}_t\right) \tag{1.34}$$

The spectral graph convolution defined in 1.33 is similar to a K-step diffusion process on the graph. If we define a state transition matrix of a diffusion process as $\mathbf{T} = \mathbf{D}^{-1}\mathbf{A}$ we can define a *diffusion convolution* layer as:

$$\mathbf{h}_t^{(q)} = \sigma\left(\sum_{k=0}^{K} \theta_k^{(q)} (\mathbf{D}^{-1}\mathbf{A})^k \mathbf{x}_t\right) \tag{1.35}$$

More precisely it can be shown that the diffusion convolution operation in equation 1.35 is equivalent to the spectral formulation in 1.34 up to a similarity transform [82, 126]. Based on this principle, the diffusion convolution operation follows the spatial interpretation of the K-step truncation of a diffusion process, where to each step k is assigned a trainable weight $\theta_k^{(q)}$.

This derivation of a graph convolution operations presupposes a knowledge about the spatial structure of the underlying graph, represented in its adjacency matrix. Still there may exist hidden spatial relations of signals in the network that are not represented in the original adjacency matrix used to construct the Laplacian operator. For this purpose we can introduce an additional self-adaptive, normalized adjacency matrix $\mathbf{A}_{Adap} \in \mathbb{R}^{N \times N}$ [134]. The latter is defined as a matrix of trainable weights $\mathbf{V}_{Adap} \in \mathbb{R}^{N \times N}$, which can be trained via gradient descent based optimization [76]. Inspired by the study of Wu et al. [134], a normalized self-adaptive adjacency matrix can be defined as [128]:

$$\mathbf{A}_{Adap} = \frac{\sigma_{Adap}\left(\mathbf{V}_{Adap}\right)}{N} \tag{1.36}$$

The function $\sigma_{Adapt}(\cdot) \equiv \tanh(\cdot)$ confines the weights within the range $[-1, 1]$,

which are normalized by the number of nodes N of the graph structure. This self-adaptive adjacency matrix can help to uncover any hidden unknown dependencies between nodes of a given graph structure. By including this adaptive adjacency structure $\mathbf{A}_{Adap}$ into equation 1.35, we can extend a graph diffusion convolution layer to yield its output as:

$$\mathbf{h}_t^{(q)} = \sigma\left(\sum_{k=0}^{K}\left(\theta_k^{(q)}\mathbf{T}^k + \beta_k^{(q)}\left(\mathbf{A}_{Adap}\right)^k\right)\mathbf{x}_t\right) \tag{1.37}$$

In this equation $\mathbf{T}$ represents the transition operator, which was defined for a diffusion convolution as $\mathbf{T} = \mathbf{D}^{-1}\mathbf{A}$ and $\theta_k^{(q)}$ denote the parameterized filter kernels on the graph. The normalized self-adaptive adjacency matrix $\mathbf{A}_{Adap}$ may be considered as an additional transition operator here, with its respective filter parameters $\beta_k^{(q)}$. If no prior knowledge about the graph is available, the first term within parentheses can be skipped and the self-adaptive adjacency matrix may possibly identify the underlying graph structure from the data alone. This convolution operation can in that case be formulated as:

$$\mathbf{h}_t^{(q)} = \sigma\left(\sum_{k=0}^{K}\beta_k^{(q)}\left(\mathbf{A}_{Adap}\right)^k\mathbf{x}_t\right) \tag{1.38}$$

Higher order relations between nodes have been characterized in graph convolutions by filter parameters $\theta_k^{(q)}, \beta_k^{(q)}$ which determine the influence of k-hop transitions on the graph. Still the learning complexity of the GNN model grows linearly with k when accounting for higher order transitions.

An alternative possibility for inherently capturing higher order relations in a graph structure is provided by so-called node embedding algorithms. The goal of such an embedding is to represent each node $v \in \mathcal{V}$ in the graph $\mathcal{G}$ by a Q-dimensional vector, and thereby preserving the neighborhood role of the node within network in this Q-dimensional embedding subspace [59]. One efficient variant is the *node2vec* model [59], which follows the idea of the *word2vec* model, originally proposed by Mikolov et al. [89]. The word2vec model learns vector-valued embeddings of words that can inherently capture their semantic context within a sentence. This idea can be transferred to graph-like structures, by replacing sequences of words by sequences of nodes, obtained from a biased random walk along neighboring nodes in a graph [59]. To gain an intuition how node embeddings can be learned, the *Skip-gram* model will first be introduced. In the context of graph signal processing, the learning objective of this model is to predict from an input node denoted as v_I the C surrounding context nodes $v_{O,c}$ with $c = 1, \ldots, C$. Therefore the Skip-gram model has the objective to maximize the following average log-probability [89, 97] of observing some context nodes $v_{O,c}$ within the neighborhood of an input node v_I:

$$J(v_I) = \sum_{c=1}^{C}\log p(v_{O,c} \mid v_I) \tag{1.39}$$

where C determines the size of the training context around an input node v_I. The Skip-gram model is composed of an input layer $\mathbf{x} \in \mathbb{R}^J$, one hidden layer $\mathbf{h} \in \mathbb{R}^Q$ and C output panels, consisting of C vectors denoted as $\mathbf{y}_c \in \mathbb{R}^J$. In the input and output layers nodes are represented using a one-hot encoding, meaning that each element of the vector is 0 except for one element $x_k = 1$. An overview of the Skip-gram model is provided in figure 1.9. For the hidden layer the model uses a linear activation function, so the hidden state $\mathbf{h}$ can be computed as:

$$\mathbf{h} = \mathbf{W}^{(1)}\mathbf{x} = \mathbf{W}^{(1)}_{:,k} := \mathbf{w}^{(1)}_{v_I} \tag{1.40}$$

where $\mathbf{W}^{(1)} \in \mathbb{R}^{Q \times J}$ denotes the weight matrix of the first layer. Because $\mathbf{x}$ is only non-zero at the k-th entry, this equation can be interpreted in a way that $\mathbf{x}$ selects the k-th column of $\mathbf{W}^{(1)}$ and simply copies it to the hidden layer $\mathbf{h}$. Each column of the matrix $\mathbf{W}^{(1)}$ is accordingly selected by one specific node in the graph, which motivates us to use the column $\mathbf{w}^{(1)}_{v_I} \in \mathbb{R}^Q$ of $\mathbf{W}^{(1)}$ as a vector representation of the respective input node v_I. In a second step, one shared weight matrix $\mathbf{W}^{(2)} \in \mathbb{R}^{J \times Q}$ is used to generate predictions for the C context nodes, as illustrated in figure 1.9. With the hidden state $\mathbf{h}$, a single unit j of the output layer $\mathbf{u}_c = \mathbf{W}^{(2)}\mathbf{h}$ of the c-th output panel can be obtained by:

$$u_{c,j} = \mathbf{W}^{(2)}_{j,:}\mathbf{h} = \mathbf{W}^{(2)}_{j,:}\mathbf{w}^{(1)}_{v_I} := \mathbf{w}^{(2)T}_{v_j}\mathbf{w}^{(1)}_{v_I} \tag{1.41}$$

Here $\mathbf{w}^{(2)T}_{v_j}$ denotes the j-th row of weight matrix $\mathbf{W}^{(2)}$, which determines the activation corresponding to the j-th output node. To obtain a probability to select a specific output node v_j for a given input node v_I, a softmax function is finally applied:

$$p(v_{c,j} \mid v_I) = \frac{\exp(u_j)}{\sum_{j'=1}^{J}\exp(u_{j'})} = \frac{\exp(\mathbf{w}^{(2)T}_{v_j}\mathbf{w}^{(1)}_{v_I})}{\sum_{j'=1}^{J}\exp(\mathbf{w}^{(2)T}_{v_{j'}}\mathbf{w}^{(1)}_{v_I})} \tag{1.42}$$

Because of the weight sharing between the c panels, like those illustrated in figure 1.9, the output probabilities $p(v_{c,j} \mid v_I)$ are identical for the c output panels. With the output probability defined in equation 1.42 the objective function in equation 1.39 can be maximized using stochastic gradient descend based optimization techniques [89].

If two nodes appear frequently within a similar context c, then the output probabilities generated by the Skip-gram model are optimized to become similar for these two nodes during training. Equation 1.42 shows that the output probabilities are derived from the dot products between $\mathbf{w}^{(2)}_{v_j}$ and $\mathbf{w}^{(1)}_{v_I}$. Therefore, in order to generate a similar response in the output, two input node representations that appear frequently in a similar context are thereby also optimized to have a high (cosine) similarity to each other. This can give us a first intuition why representations of nodes $\mathbf{w}^{(1)}_{v_I}$ preserve meaningful

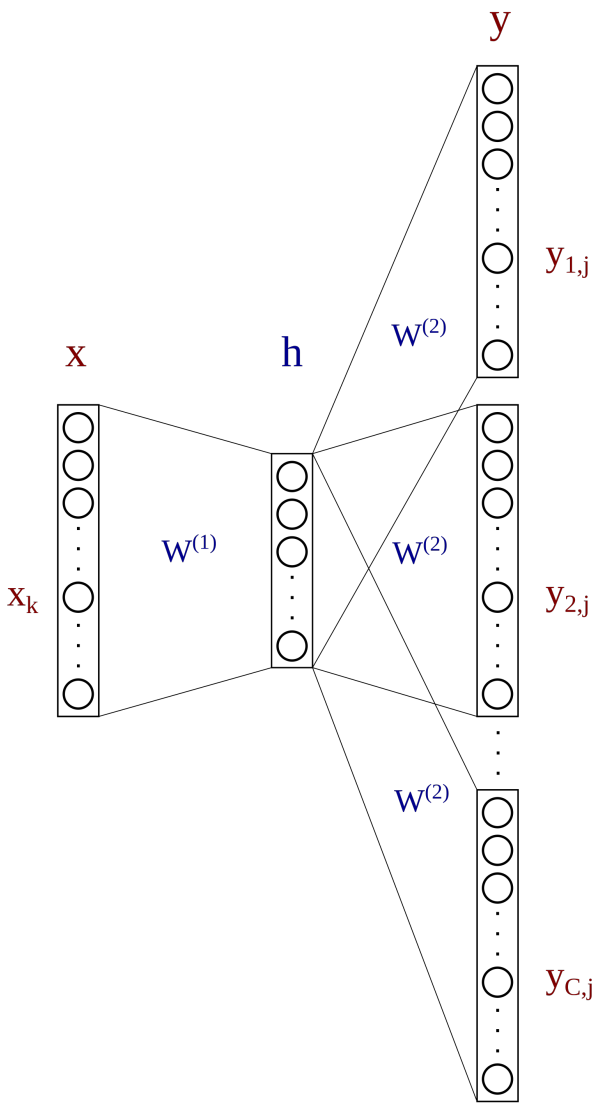

FIGURE 1.9: This figure shows an illustration of the Skip-gram model. The input $\mathbf{x}$ is a one-hot encoded node, which is first linearly projected onto its hidden state $\mathbf{h}$ by a weight matrix $\mathbf{W}^{(1)}$. The prediction of the C context nodes is obtained by projecting $\mathbf{h}$ onto C one-hot encoded vectors $\mathbf{y}_c$, which represent the output nodes. The weight matrix $\mathbf{W}^{(2)}$ is thereby shared between all output panels $\mathbf{y}_c$.

relations to neighboring nodes in their Q-dimensional subspace.

To learn the node representations, we have to additionally find a proper definition of the context. In language processing the context of a word within a text can be simply generated by sliding windows of neighboring words across a text. But it is less straightforward for graphs to find an appropriate definition of the context of a node $v \in \mathcal{V}$ within a graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The node2vec algorithm proposed one efficient procedure for sampling such sequences of nodes that capture higher order topological relations between

nodes [59]. In general different node sampling strategies have to find a trade-off between local walks on the graph, e.g. visiting only nodes within the intermediate neighborhood of source node v, or a more global exploration by sampling nodes with increasing distances from the source node v [59]. Implementing the idea of a biased random walk to generate a node sequence $c_1, c_2, \ldots, c_C$, the node2vec algorithm defines the probability of a transition between node v and v' as:

$$p(c_i = v' \mid c_{i-1} = v) = \begin{cases} \frac{\pi_{vv'}}{Z} & \text{if } (v, v') \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{1.43}$$

with $\pi_{vv'}$ representing the unnormalized transition probability between node v and v', and Z denoting a normalization constant. The transition probability is computed as $\pi_{vv'} = \alpha_{pq}(t, v') \cdot w_{vv'}$ whereby $w_{vv'}$ denotes the edge strength between node v and v' derived from the weighted adjacency matrix $\mathbf{A}$. The parameter $\alpha_{pq}(t, v')$ determines how quickly the walk will leave a previously visited neighborhood. If t denotes the previously visited node, then the parameter is computed as:

$$\alpha_{pq}(t, v') = \begin{cases} \frac{1}{p} & \text{if } d_{tv'} = 0 \\ 1 & \text{if } d_{tv'} = 1 \\ \frac{1}{q} & \text{if } d_{tv'} = 2 \end{cases} \tag{1.44}$$

whereby $d_{tv'}$ represents the shortest path distance between two nodes t and v'. In this formulation $d_{tv'} = 0$ would represent the case returning back to node t, $d_{tv'} = 1$ visiting a node within the neighborhood of t, and $d_{tv'} = 2$ leaving the neighborhood of the previous node t. The different parameter settings of $\alpha_{pq}(t, v')$ are additionally illustrated in figure 1.10.

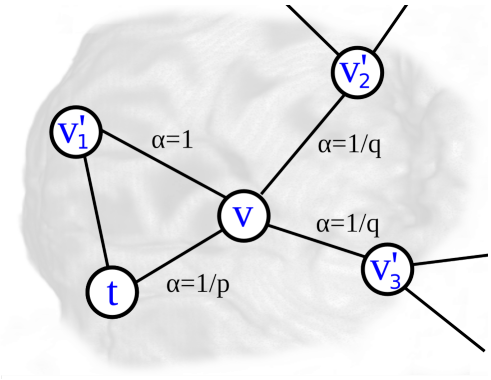

FIGURE 1.10: The figure illustrates the values of parameter $\alpha_{pq}(t, v')$ for the different pairs of nodes. The transition probability between node v and the previously visited node t is weighted with $\alpha = \frac{1}{p}$. For visiting a node v'_1 within the intermediate neighborhood of t it will take a value of $\alpha = 1$. And for transitions to nodes v'_2 and v'_3 in a new neighborhood it is obtained with $\alpha = \frac{1}{q}$.

Accordingly, the parameters p and q would characterize the ratio of local and global walks on the network. Setting p to a high value would lower the probability of returning to an already visited node and support a more global exploration. Setting q to a high value would bias the random walk to visit more nodes within the neighborhood of the previous node t, favoring a rather local exploration [59]. Using such a sampling strategy allows us to obtain a sequence of nodes which captures the local and global neighborhood of nodes within the graph, and helps us to define a meaningful context for a target node in the node2vec model.

1.2.5 Recurrent Neural Networks

Similar to CNNs, which were developed to process data on grid-like structures, or GNNs, which are designed to deal with graphical representations, so-called *recurrent neural networks* (RNNs) were established for analyzing data with sequential structures [100]. Such RNN architectures were based on the idea to iteratively process a sequence of data samples and to share parameters across different processing steps in the recurrent model. This parameter sharing across a sequence is useful in language processing applications for example. If we consider two sentences "I visited Regensburg in 2019" and "In 2019 I visited Regensburg" and we would like to ask when the narrator has visited Regensburg, the relevant information appears at different positions in the two sentences [55]. Sharing the parameters across the sequence therefore avoids to learn patterns for every possible position in the sequence separately and allows us to detect features independently of their positions. If we assume the data is represented by a sequence of vectors $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(T)}$ with length T, then we can in generally describe the principle of a RNN with the following equation:

$$\mathbf{h}^{(t)} = f(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}; \boldsymbol{\theta}) \tag{1.45}$$

In RNNs $\mathbf{h}^{(t)}$ is referred to as the state, $\mathbf{x}^{(t)}$ is the data input to the network at step t, and $\boldsymbol{\theta}$ summarizes the parameters of the model. The RNN model therewith combines the information of the input $\mathbf{x}^{(t)}$ at step t and its previous state $\mathbf{h}^{(t-1)}$ to recursively generate the subsequent state $\mathbf{h}^{(t)}$. For example in a prediction task the state $\mathbf{h}^{(t)}$ could represent a condensed summary of relevant information in the past up to timestep t. With this idea in mind, a very basic form of a RNN can be defined as:

$$\mathbf{h}^{(t)} = \Phi_h(\mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)} + \mathbf{b}) \tag{1.46}$$

$$\mathbf{y}^{(t)} = \Phi_y(\mathbf{V}\mathbf{h}^{(t)} + \mathbf{c}) \tag{1.47}$$

In this notation the matrices $\mathbf{W}$,$\mathbf{U}$ and $\mathbf{V}$ contain the trainable weights of the model, and $\mathbf{b}$ and $\mathbf{c}$ summarize the bias terms. Further Φ_h and Φ_y represent

some activation functions as introduced in section 1.2.1. The schematic layout of a RNN is further illustrated in figure 1.11. Analogous to section 1.2.1 we can define a cost function which describes the distance between the model outputs $y^{(t)}$ and the desired target outputs $\hat{y}^{(t)}$ and minimize the cost by using optimization algorithms as introduced in section 1.2.2. Computing the gradient of the cost function with respect to the model parameters $\boldsymbol{\theta}$ requires us to backpropagate the error through the individual steps t of the sequence, which is referred to as *backpropagation through time* (BPTT) [130]. The principle of the BPTT algorithm is described in more detail in appendix A.2.

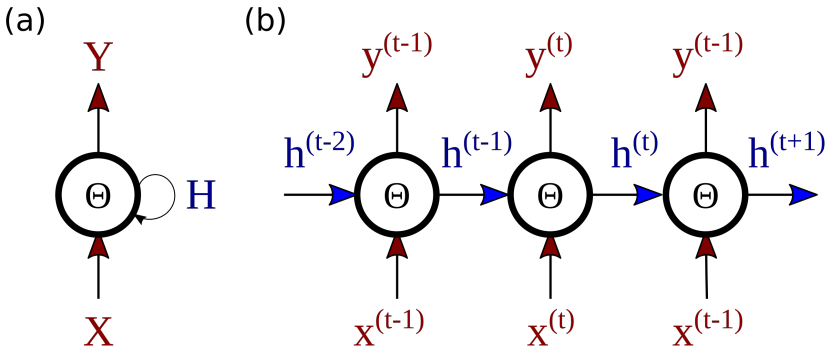

FIGURE 1.11: The structure of a RNN can be illustrated in two different ways. In a compact representation, as shown in (a), the RNN model receives an input $\mathbf{X} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(T)}]$ containing a sequence of states $\mathbf{x}^{(t)}$. The model parameters $\boldsymbol{\theta}$ are reused for every input $\mathbf{x}^{(t)}$ in order to generate a sequence of hidden states $\mathbf{H} = [\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \ldots, \mathbf{h}^{(T)}]$. The RNN model repeatedly uses its hidden states $\mathbf{h}^{(t)}$ to generate an output sequence of target values $\mathbf{Y} = [\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(T)}]$. The individual processing steps t can be explicitly displayed by unfolding the circuit shown in (a) to the stepwise computations represented in (b). At each step the input $\mathbf{x}^{(t)}$ is combined with its previous hidden state $\mathbf{h}^{(t-1)}$ to generate the subsequent state $\mathbf{h}^{(t)}$ and a target value $\mathbf{y}^{(t)}$. The parameters of the RNN summarized in $\boldsymbol{\theta}$ are shared across the whole sequence.

One drawback of the recursive formulation in equation 1.46 is that by accumulating the gradient across very long sequences, the gradient tends to become very small (vanishing) or extremely large (exploding) [67]. To address this issue so-called *gated* RNNs were proposed. They are based on the idea to introduce paths through time where the gradient values are more stable. The first very popular neural network architecture which implements such mechanisms is the *long short-term memory* (LSTM) network proposed by Hochreiter and Schmidhuber [67]. A LSTM network includes different gating mechanisms, which allow us to dynamically control the temporal scale of integration of information. The weight of the self-loops is first controlled by a so-called *forget gate*, which is computed as the following:

$$\mathbf{f}^{(t)} = \sigma(\mathbf{W}^{(f)}\mathbf{h}^{(t-1)} + \mathbf{U}^{(f)}\mathbf{x}^{(t)} + \mathbf{b}^{(f)}) \tag{1.48}$$

where $\mathbf{x}^{(t)}$ denotes the input, and $\mathbf{h}^{(t-1)}$ the hidden state vector. The weights and biases of the forget gate are summarized in $\mathbf{W}^{(f)}$, $\mathbf{U}^{(f)}$ and $\mathbf{b}^{(f)}$. The activity of the forget gate is scaled between 0 and 1 by a sigmoid activation function $\sigma(\cdot)$. The crucial component of the LSTM is its *internal state* $\mathbf{s}^{(t)}$ which is updated as follows:

$$\mathbf{s}^{(t)} = \mathbf{f}^{(t)} \odot \mathbf{s}^{(t-1)} + \mathbf{g}^{(t)} \odot \tanh(\mathbf{W}^{(i)}\mathbf{h}^{(t-1)} + \mathbf{U}^{(i)}\mathbf{x}^{(t)} + \mathbf{b}^{(i)}) \tag{1.49}$$

with $\mathbf{W}^{(i)}$, $\mathbf{U}^{(i)}$ respectively denote the recurrent and input weights for the LSTM and $\mathbf{b}^{(i)}$ the bias term. In equation 1.49 the forget gate $\mathbf{f}^{(t)}$ controls how much information from the previous state $\mathbf{s}^{(t-1)}$ is kept and the *input gate* $\mathbf{g}^{(t)}$ how much new information is added from its current data input $\mathbf{x}^{(t)}$ and previous hidden state $\mathbf{h}^{(t-1)}$. The input gate is obtained as the following:

$$\mathbf{g}^{(t)} = \sigma(\mathbf{W}^{(g)}\mathbf{h}^{(t-1)} + \mathbf{U}^{(g)}\mathbf{x}^{(t)} + \mathbf{b}^{(g)}) \tag{1.50}$$

Finally the output of the LSTM is controlled by an *output gate* $\mathbf{o}^{(t)}$:

$$\mathbf{o}^{(t)} = \sigma(\mathbf{W}^{(o)}\mathbf{h}^{(t-1)} + \mathbf{U}^{(o)}\mathbf{x}^{(t)} + \mathbf{b}^{(o)}) \tag{1.51}$$

$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \odot \tanh(\mathbf{s}^{(t)}) \tag{1.52}$$

The schematic layout of a LSTM cell is illustrated in figure 1.12. This architecture has shown to efficiently capture long-term dependencies in data, and made it possible to make reliable predictions for large horizons into the future [67]. More complex representations of the sequence data can be learned by including multiple concatenated LSTM cells at each time step.

A simplification of the gating introduced above was put forward in the notion of *gated recurrent units* (GRUs) [32], which implement the idea of using a single gating mechanism to determine the forgetting factor and the update of the hidden state. They introduce an *update gate* $\mathbf{z}^{(t)}$ and a *reset gate* $\mathbf{r}^{(t)}$ as depicted in the following:

$$\mathbf{z}^{(t)} = \sigma(\mathbf{W}^{(z)}\mathbf{h}^{(t-1)} + \mathbf{U}^{(z)}\mathbf{x}^{(t)} + \mathbf{b}^{(z)}) \tag{1.53}$$

$$\mathbf{r}^{(t)} = \sigma(\mathbf{W}^{(r)}\mathbf{h}^{(t-1)} + \mathbf{U}^{(r)}\mathbf{x}^{(t)} + \mathbf{b}^{(r)}) \tag{1.54}$$

The reset gate $\mathbf{r}^{(t)}$ is used to control the amount of information which is transferred to the so-called *candidate state* $\mathbf{c}^{(t)}$:

$$\mathbf{c}^{(t)} = \tanh(\mathbf{W}^{(c)}(\mathbf{r}^{(t)} \odot \mathbf{h}^{(t-1)}) + \mathbf{U}^{(c)}\mathbf{x}^{(t)} + \mathbf{b}^{(c)}) \tag{1.55}$$

The update gate $\mathbf{u}^{(t)}$ can take values between 0 and 1 and therewith determines the proportion of the candidate $\mathbf{c}^{(t)}$ which is passed to the new state in the following way:

$$\mathbf{h}^{(t)} = (1 - \mathbf{z}^{(t)}) \odot \mathbf{h}^{(t-1)} + \mathbf{z}^{(t)} \odot \mathbf{c}^{(t)} \tag{1.56}$$

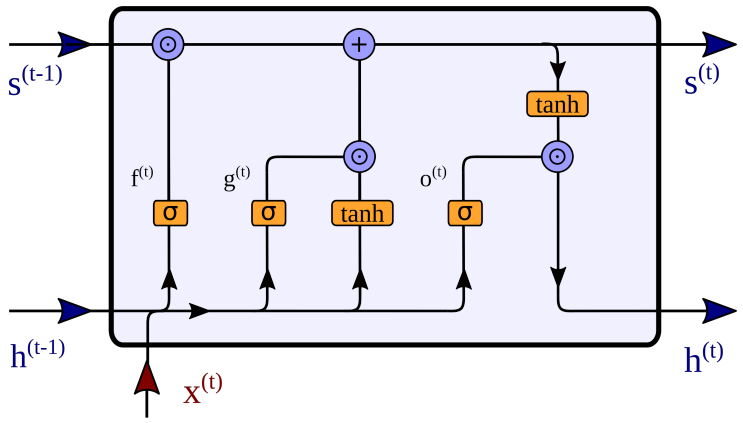

FIGURE 1.12: This figure shows an overview of the different gating mechanisms in a single LSTM cell. The data input $\mathbf{x}^{(t)}$ and the previous hidden state $\mathbf{h}^{(t-1)}$ are used to determine the activation of the forget gate $\mathbf{f}^{(t)}$ which controls the amount of information which is kept from the state $\mathbf{s}^{(t-1)}$. In a next step the proportion of new information is regulated by the input gate $\mathbf{g}^{(t)}$ to compute the new cell state $\mathbf{s}^{(t)}$. Finally the output gate $\mathbf{o}^{(t)}$ is used to control the information flow from the cell state $\mathbf{s}^{(t)}$ to the new hidden state $\mathbf{h}^{(t)}$. The hidden state $\mathbf{h}^{(t)}$ and cell state $\mathbf{s}^{(t)}$ can be passed to the subsequent LSTM cell to recurrently update the model.

The schematic layout of a GRU cell is further depicted in figure 1.13. Besides the LSTM and GRU other variants of gated RNNs were proposed, but these two RNN architectures still proved to be the most reliable across a wide variety of tasks [57].

In various applications like speech recognition, machine translation or time series forecasting, it is required to map a sequence of input values to a sequence of generated outputs. For example in timeseries forecasting we would like to infer from a sequence of historical observation a sequence of future values. The so-called *sequence-to-sequence* learning [114, 32] provides one possibility to process such sequentially structured data. The structure of sequence-to-sequence architecture is illustrated in figure 1.14. A sequence-to-sequence model is composed of two components, the first is an *encoder* RNN, which recursively processes an input sequence of T_p past observations $[\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T_p)}]$. The encoder RNN summarizes the information in the input sequence in its final hidden state $\mathbf{h}^{(T_p)}$, which is also called the *context* state. In a next step the context state $\mathbf{h}^{(T_p)}$ is passed to an *decoder* RNN which uses the information in $\mathbf{h}^{(T_p)}$ to recursively generate a corresponding output sequence of T_f future values $[\mathbf{x}^{(T_p+1)}, \ldots, \mathbf{x}^{(T_p+T_f)}]$. Typically LSTMs or GRUs, as introduced above, are used as encoding and decoding models. The advantage of this architecture is that by using such an encoder-decoder framework it can also be applied to sequences with arbitrary input lengths T_p and output lengths T_f.

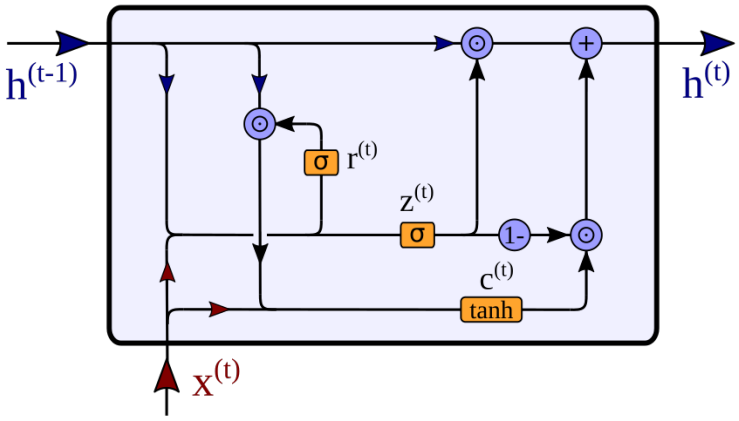

FIGURE 1.13: This figure illustrates the gates implemented in a GRU cell. At first the data input $\mathbf{x}^{(t)}$ and the previous hidden state $\mathbf{h}^{(t-1)}$ are used to compute the activation of the reset gate $\mathbf{r}^{(t)}$, which determines the amount of information which is preserved from the previous hidden state $\mathbf{h}^{(t-1)}$ to compute a new candidate state $\hat{\mathbf{h}}^{(t)}$. The update gate $\mathbf{z}^{(t)}$ then regulates which proportion of the previous hidden state $\mathbf{h}^{(t-1)}$ and the candidate state $\hat{\mathbf{h}}^{(t)}$ enter the new hidden state $\mathbf{h}^{(t)}$.

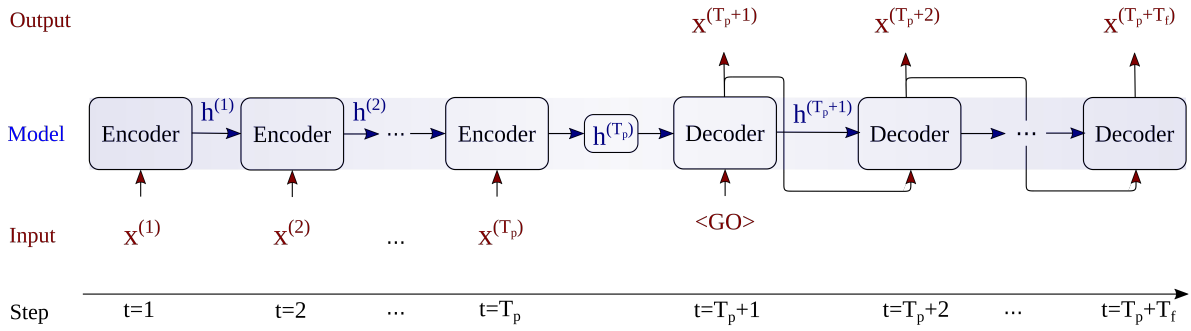

FIGURE 1.14: The figure shows an overview of the architecture used in sequence-to-sequence learning. The encoder receives an input sequence $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T_p)}$ and iteratively updates its hidden state $\mathbf{h}^{(t)}$. When the encoder has seen the complete input sequence, it passes its finals state $\mathbf{h}^{(T_p)}$ to the decoder, which generates the corresponding output sequence $\mathbf{x}^{(T_p+1)}, \ldots, \mathbf{x}^{(T_p+T_f)}$. As an input the decoder uses its own predictions made in the previous step. The first input (<GO> label) of the decoder can be simply be defined as a vector of zeros.

1.2.6 Spatio-Temporal Graph Neural Networks

In section 1.2.4 we introduced GNNs, which allowed us to model spatial dependencies in graph-like data structures. But in many real-world problems we have to additionally consider the temporal dynamics of graph structured signals. One typical example for such an application would be the task of traffic prediction, where the traffic speed is measured over time by different sensors, which are connected to each other by a network of streets [82]. Dynamic graph-signals also show up in the context of neuroimaging, where we temporally resolve the neural activity in different areas of the brain, which are anatomically interconnected by a network of white matter tracks. Figure 1.15 illustrates such a time-varying signal with a graph-like structure.

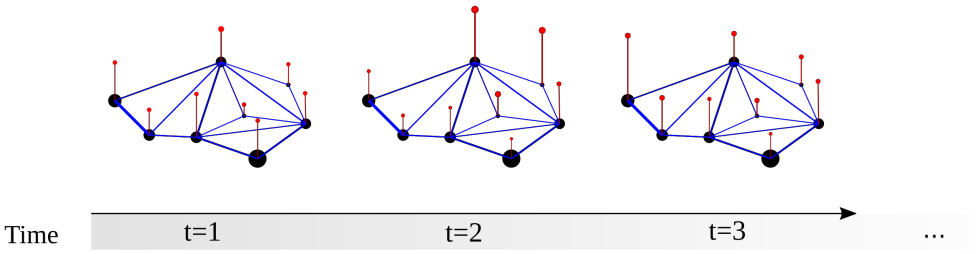

FIGURE 1.15: The figure shows an example of a dynamic signal with graph-like structure. The nodes in the graph are depicted in black, and for each node n a signal $x_n^{(t)}$ that varies over time $t = 1, 2, \ldots$ can be associated, as illustrated in red. The spatial dependencies between the individual signals are characterized by the edges of a graph, which are marked here in blue.

To deal with this specific type of structure in timeseries data, so-called *spatio-temporal graph neural networks* (STGNN) were developed. To capture the temporal patterns of the dynamic signal, they typically employ techniques for sequential data structures, like RNNs (as described in section 1.2.5) or one-dimensional CNNs (section 1.2.3). In addition, to model spatial interdependencies between multivariate timeseries data, they make use of the graphical structure of the signal by invoking graph convolution operations, as introduced in section 1.2.4. We can formally describe the task of graph signal prediction by first considering a signal $x_n^{(t)}$ in node n, sampled at timestep t. We can collect these $n = 1, \ldots, N$ signals sampled in a time interval $t = 1, \ldots, T$ into a matrix $X \in \mathbb{R}^{N \times T}$. Then the spatial relationship between the nodes can be defined by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, including a set of nodes (vertices) $\mathcal{V}$, with $|\mathcal{V}| = N$, and edges $\mathcal{E}$. The graph structure is captured by the weighted adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, where each entry $w_{nn'}$ describes the edge weight between node n and n'. Accordingly, the goal in graph signal prediction is to learn a function $f(\cdot)$ which maps T_p past states of $\mathbf{x}^{(t)}$, to T_f future states:

$$[\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T_p)}; \mathcal{G}] \xrightarrow{f(\cdot)} [\mathbf{x}^{(T_p+1)}, \ldots, \mathbf{x}^{(T_p+T_f)}] \tag{1.57}$$

Besides the information in the past states of the system $\mathbf{x}^{(t)}$ with $t = 1, \ldots, T_p$ the characteristic of STGNN is to additionally incorporate information on the graph structure $\mathcal{G}$ to predict the future states $t = T_p + 1, \ldots, T_p + T_f$. For this kind of task numerous variants of STGNN have been developed [133], and in the following two established STGNN architectures will be introduced.

One of the first STGNN architectures was proposed by Li et al. [82], who introduced the so-called *diffusion convolution recurrent neural network* (DCRNN) for the application of traffic forecasting. The DCRNN model is based on a RNN, using a sequence-to-sequence architecture as described in section 1.2.5. In this RNN based variant, the idea is to capture the information of past nodes states $t = 1, \ldots, T_p$ in the encoding part of the sequence-to-sequence model, and use the encoding (or context) state $\mathbf{H}^{(T_p)} \in \mathbb{R}^{N \times Q}$ to predict the future states of the nodes $t = T_p + 1, \ldots, T_p + T_f$ in the decoding part. As an encoder and decoder the DCRNN model employs GRU cells (introduced in section 1.2.5), which are additionally modified to process multivariate graph-structured signals. The idea is to replace the multiplications with weight matrices in GRUs by the diffusion convolution operations introduced in section 1.2.4:

$$\mathbf{y}^{(q)} = \sum_{k=0}^{K} \theta_k^{(q)} (\mathbf{D}^{-1}\mathbf{A})^k \mathbf{x} \tag{1.58}$$

Here $\mathbf{A} \in \mathbb{R}^{N \times N}$ represents the adjacency matrix of our graph $\mathcal{G}$, and $\mathbf{D}$ the diagonal node degree matrix. The graph filters $\theta_k^{(q)}$ characterize the influence of walks of order k on the graph, and are learned by the model during the training. Usually the number of walks k which have to be considered are relatively small with $K \ll N$, because the relevant information comes mainly from the neighboring nodes, which effectively reduces the number of model parameters. If we denote the diffusion convolution operation on the graph with $*_G$, we can rewrite gating mechanisms in a GRU cell as:

$$\mathbf{r}^{(t)} = \sigma\left(\mathbf{\Theta}^{(r)} *_G \left[\mathbf{x}^{(t)}, \mathbf{H}^{(t-1)}\right] + \mathbf{b}^{(r)}\right) \tag{1.59}$$

$$\mathbf{z}^{(t)} = \sigma\left(\mathbf{\Theta}^{(z)} *_G \left[\mathbf{x}^{(t)}, \mathbf{H}^{(t-1)}\right] + \mathbf{b}^{(z)}\right) \tag{1.60}$$

$$\mathbf{c}^{(t)} = \tanh\left(\mathbf{\Theta}^{(c)} *_G \left[\mathbf{x}^{(t)}, (\mathbf{r}^{(t)} \odot \mathbf{H}^{(t-1)})\right] + \mathbf{b}^{(c)}\right) \tag{1.61}$$

$$\mathbf{H}^{(t)} = \mathbf{z}^{(t)} \odot \mathbf{H}^{(t-1)} + (1 - \mathbf{z}^{(t)}) \odot \mathbf{c}^{(t)} \tag{1.62}$$

where $\mathbf{x}^{(t)} \in \mathbb{R}^N$ denotes the graph signal in the N nodes at a timestep t. Furthermore $\mathbf{H}^{(t)} \in \mathbb{R}^{N \times Q}$ is the hidden state of the GRU cell and $[\mathbf{x}^{(t)}, \mathbf{H}^{(t-1)}]$ denotes their concatenation. In these equations $\mathbf{r}^{(t)}, \mathbf{z}^{(t)}, \mathbf{c}^{(t)}$ represent the reset and update gates, and the candidate state at a time step t, and $\mathbf{b}^{(r)}, \mathbf{b}^{(z)}, \mathbf{b}^{(c)}$, respectively denote the corresponding bias terms. In addition the parameters $\mathbf{\Theta}^{(r)}, \mathbf{\Theta}^{(z)}, \mathbf{\Theta}^{(c)}$ denote a set of the corresponding graph filters. An illustration of the gating mechanisms in this so-called *diffusion convolution gated recurrent*

unit (DCGRU) cell is provided in figure 1.16. Incorporating these DCGRU cells in a sequence-to-sequence architecture then allows us to simultaneously account for temporal and spatial dependencies in the graph signal data.

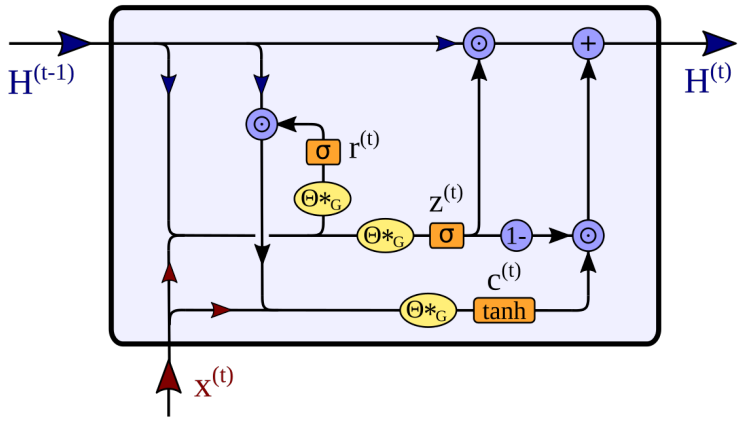

FIGURE 1.16: The figure provides an overview on the individual processing steps in a DCGRU cell. The input data $\mathbf{x}^{(t)}$ together with the previous hidden state $\mathbf{H}^{(t-1)}$ are concatenated and enter the reset gate $\mathbf{r}^{(t)}$ and the update gate $\mathbf{z}^{(t)}$. The reset gate $\mathbf{r}^{(t)}$ determines the proportion of the information in $\mathbf{H}^{(t-1)}$ and $\mathbf{x}^{(t)}$ which enters the candidate state $\mathbf{c}^{(t)}$. Finally the old hidden state $\mathbf{H}^{(t-1)}$ is updated by $\mathbf{c}^{(t)}$, whereby the proportion of new information is controlled by the update gate $\mathbf{z}^{(t)}$.

As an alternative to this RNN based architecture, also one-dimensional convolutions can be incorporated to detect temporal relations in the multivariate timeseries data. Following this idea, the *graph WaveNet* (GWN) combines dilated causal convolutions, as introduced in section 1.2.3, with the graph convolution operations, described in section 1.2.4, to simultaneously capture the temporal and spatial features of the signal [134]. Replacing the recurrent computations of the DCRNN with temporal convolutions avoids the iterative computation of the gradient and can help us to prevent vanishing or exploding gradients. The characteristic components of the WaveNet, the dilated causal convolutions, were defined as the following:

$$(\mathbf{x} *_C \boldsymbol{\theta})(t) = \sum_{\tau} \mathbf{x}(\tau)\boldsymbol{\theta}(t - d \cdot \tau) \tag{1.63}$$

whereby d denotes the dilation factor and $\boldsymbol{\theta}$ represents the filter kernel. This dilated convolution can be implemented by sliding over the input sequence $\mathbf{x}(t)$ while skipping input values by increasing the step size $d \cdot \tau$ from layer to layer. This leads to an exponential growth of the receptive field with increasing layer depth, as illustrated in figure 1.7. The WaveNet architecture is organized in blocks of layers, whereby the dilation factor d is doubled in every subsequent layer within a block as $d = 1, 2, 4, \ldots$ up to a certain limit. This dilation scheme is repeated in the same manner in the next block of layers, until a fixed output size is reached [121]. In addition a gating mechanism

is introduced to control the flow of information in the temporal convolution layers:

$$\mathbf{H} = \tanh\left(\mathbf{\Theta}_1 *_C \mathbf{X} + \mathbf{b}_1\right) \odot \sigma\left(\mathbf{\Theta}_2 *_C \mathbf{X} + \mathbf{b}_2\right) \tag{1.64}$$

In this equation the input of the temporal convolution layer is referred to as $\mathbf{X} \in \mathbb{R}^{N \times P \times T^{(in)}}$, with N representing the number of nodes, P the number of input features and $T^{(in)}$ the temporal dimension of the input. The activation function for the output is a $\tanh(\cdot)$ function, $*_C$ describes the causal convolution operation, and $\mathbf{\Theta}_1, \mathbf{\Theta}_2$ and $\mathbf{b}_1, \mathbf{b}_2$ represent filter parameters and biases respectively. Further $\odot$ denotes the Hadamard product and $\sigma(\cdot)$ is a logistic function, which controls the information passed to the subsequent layer. The output is denoted as $\mathbf{H} \in \mathbb{R}^{N \times Q \times T^{(out)}}$. By applying Q temporal convolutions, the input $\mathbf{X}$ is projected onto a Q-dimensional feature map with a temporal dimension of $T^{(out)}$. Using such a gated *temporal convolution network* (TCN) layer, the GWN architecture is able to detect patterns in the temporal domain of the signal.

To additionally account for the spatial dependencies in the graph signal, the GWN model applies after each gated TCN layer a graph convolution operation, as defined in equation 1.35. A complete overview of the GWN architecture is shown in figure 1.17. In the beginning the input graph signal $\mathbf{X}^{(in)} \in \mathbb{R}^{N \times T_p}$ is linearly transformed into a P-dimensional feature representation $\mathbf{X}^{(1)} \in \mathbb{R}^{N \times P \times T_p}$. This representation is passed through in total L layers, each containing a gated TCN followed by a graph convolution operation. To account for vanishing gradients, residual connections are also applied in each layer [64]. Due to the causal convolutions, each layer reduces the temporal dimension $T^{(in)}$ of its input state to $T^{(out)}$, and the number of layers can be chosen in a way to reduce the temporal dimension to $T^{(out)} = 1$ in the final layer. After each TCN operation, a skip connections is applied (as illustrated in figure 1.17) and the information from all L layers is finally aggregated by adding these skip connections up. This sum is passed through two fully connected layers with non-linear ReLU functions to generate the predicted graph signal $\mathbf{X}^{(out)} \in \mathbb{R}^{N \times T_f}$, thereby directly generating the prediction for all T_f future timepoints at once. Based on these above introduced machine learning techniques, the DCRNN and GWN model will provide a novel possibility to investigate the spatio-temporal dynamics in human brain networks. The neuroimaging techniques used for the acquisition of the data and the different concepts of brain connectivity will be discussed in the subsequent sections of this chapter.

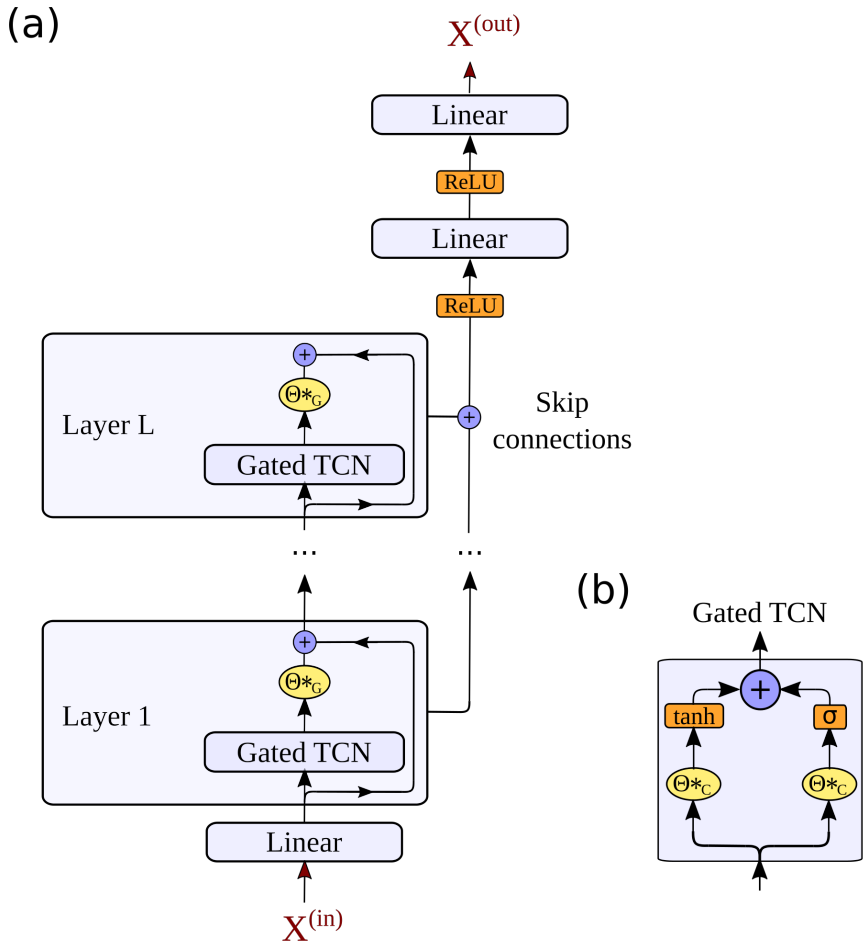

FIGURE 1.17: This figure shows an overview of the complete GWN architecture. The input graph signal $\mathbf{X}^{(in)}$ is at first linearly projected onto a P-dimensional feature representation (a). Then recursively gated TCNs and graph convolutions are applied in each layer $l = 1, \ldots, L$ of the GWN. The gating mechanism of the TCN is illustrated in more detail in (b). By using skip connections the information in each of the L layers is combined in a sum, and finally two nonlinear transformations are applied to generate the predictions of the temporal graph signal $\mathbf{X}^{(out)}$.

1.3 Magnetic Resonance Imaging

In this section a short introduction to the physical and physiological foundations of *magnetic resonance imaging* (MRI) will be provided. MRI is an imaging technique that allows us to collect high-resolution volumetric images of the human anatomy and different physiological processes in a non-invasive manner [71]. First in section 1.3.1 the physical foundations of MRI will be outlined. This imaging technique has been later extended to *functional magnetic resonance imaging* (fMRI), what made it possible to additionally study dynamic functions in the human brain in vivo [27]. The physiological origin of the signal obtained in fMRI will be then discussed in section 1.3.2. More recently, additional aspects of the brain structure became of interest, and techniques like *diffusion weighted imaging* (DWI) have been developed, which will be introduced in section 1.3.3. This imaging modality allows us to resolve bundles of white matter tracks, and to obtain an image of the structural connectivity within the human brain [61]. These different types of MRI modalities provide us different possibilities to define distinct concepts of brain connectivity, which will be later discussed in section 1.4. The following section on MRI relies mainly on the references [71, 27, 54, 61, 66, 53].

1.3.1 Magnetic Resonance Imaging Basics

MRI is based on a fundamental property of particles denoted as spin, which is characterized by a spin quantum number s, taking values of multiples of $\frac{1}{2}$. If they are unpaired, particles like protons, neutrons and electrons have a spin of $s = \frac{1}{2}$, and they thus carry a magnetic momentum $\boldsymbol{\mu}$. The signal observed in MRI is based on the interaction of a particle with non-zero net spin with a radio frequency pulse, and therefore nuclei with non-zero nuclear spin and high natural abundance are mainly of interest in MRI. One such element with high abundance in the human body is the isotope of the hydrogen nucleus ^1H [71], which will be referred to as proton for simplicity in the following. In presence of a magnetic field B_0 such a particle with non-zero net spin can interact with a photon. Modern MRI systems typically have field strengths between $B_0 = 1.5T$ and $7T$ for clinical routines and research studies. When a subject is placed in the MRI scanner, the static magnetic field $\mathbf{B}_0 = B_0\hat{\mathbf{e}}_z$ follows the direction of the body, pointing from feet to the head of the subject. The body axis will be defined as the z-axis in the following, while the y-axis is defined pointing from the back to the chest, and the x-axis from the left to the right hand of the subject. An illustration of this three axis in relation to the subject in the MRI scanner is provided in figure 1.18.

In the presence of the static magnetic field $\mathbf{B}_0$, the $\mathbf{z}$-component of the magnetic momentum $\boldsymbol{\mu}$ of the protons aligns either parallel or anti-parallel to the magnetic field. These two configurations are represented by two different energy states, whereby the number of particles in the lower energy state N^P

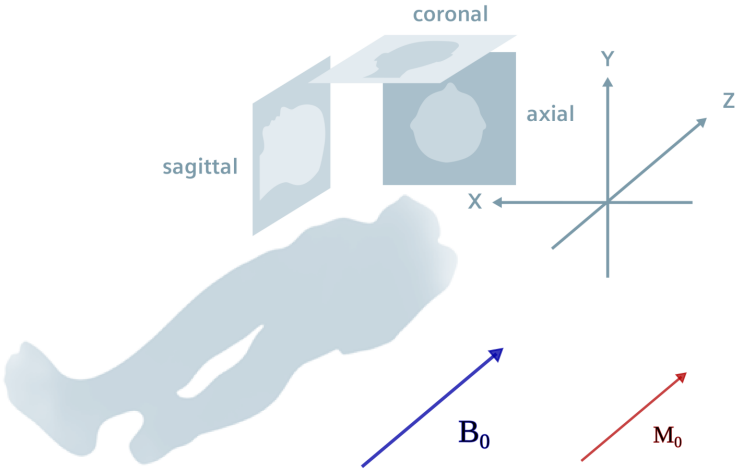

FIGURE 1.18: In this figure the three axis and the orientation of the static magnetic field $\mathbf{B}_0$ (depicted in blue) relative to the subject is illustrated. In addition the orientation of the three anatomical imaging planes are illustrated, referred to as the axial, sagittal and coronal plane. The equilibrium magnetization $\mathbf{M}_0$ (depicted in red) is oriented in parallel to the magnetic field along the z-axis. Adapted from [66].

(parallel alignment) slightly outnumbers the one in the higher state N^{AP} (anti-parallel alignment). In the equilibrium the ratio is described by the Boltzmann statistics [71]:

$$\frac{N^{AP}}{N^{P}} = e^{-\frac{E_\Delta}{k_B T}} \tag{1.65}$$

whereby E_Δ represents the energy difference between the two states, T the temperature of the system and $k_B = 1.3806 \times 10^{-23} \frac{J}{K}$ is the Boltzmann constant. To understand MRI it is helpful to first change to a more macroscopic view of the process. Multiple of such spins can be summarized in spin packets, and each spin packet can be represented by a magnetization vector $\mathbf{M}_0$. The equilibrium magnetization $M_0 \hat{\mathbf{e}}_z$ is parallel to the magnetic field, as depicted in figure 1.18, and its strength is determined by the number of spins aligned in parallel N^P or anti-parallel N^{AP} to the magnetic field. The equilibrium magnetization can be characterized by $M_0 \propto (N^P - N^{AP})$ and in the equilibrium state there exists no transverse magnetization $\mathbf{M}_x = \mathbf{M}_y = 0$. Now to generate a measurable signal, the magnetization $\mathbf{M}_0$ is tipped away from its equilibrium by applying a radio frequency pulse, whose energy matches the energy difference between the two states E_Δ. The magnetization then precesses around the z-axis with the so-called *Larmor frequency* [27]:

$$\omega_L = \gamma B_0 \tag{1.66}$$

where γ denotes the *gyromagnetic ratio*, which for Protons is $\gamma = 2.675 \times 10^8 \frac{rad}{s \cdot T}$. If the radio frequency pulse matches the frequency ω_L, the system can

be excited in resonance, and for a typical MRI scanner with a field strength of $3T$, the resonance frequency $f_L = \frac{\omega_L}{2\pi}$ is approximately $f_L = 128MHz$ [27]. This rotation of the magnetization generates an oscillating magnetic field, which can in turn be detected by a receiver coil by measuring the inductive voltage. After the excitation, the magnetization vector then returns back to its initial state again in a so-called relaxation process. This leads to a decay in the signal amplitude measured by the coil, which is denoted as *free induction decay* (FID). When the magnetization vector was tipped away from its equilibrium state along $\mathbf{B}_0$, the magnetization vector can be described by two components. The magnetization starts to relax back into its equilibrium state M_0, and the relaxation process of the longitudinal component of the the magnetization as a function of time t can be described by the following equation:

$$M_z = M_0 \left(1 - e^{-\frac{t}{T_1}}\right) \tag{1.67}$$

In this equation T_1 characterizes the time, after which the difference between the longitudinal magnetization M_z and its equilibrium state M_0 is reduced by a factor of e, which can be mainly related to spin-lattice interactions [71]. Directly after the excitation with the radio-frequency pulse, the spin packets are in phase and precess around the z-axis with a magnetization of M_{xy_0}. Mainly due to spin-spin interactions, these spin packets begin to dephase over time, leading to a decrease of the transverse magnetization M_{xy} over time t:

$$M_{xy} = M_{xy_0} e^{-\frac{t}{T_2}} \tag{1.68}$$

The constant T_2 determines thus the time after the transverse magnetization is decreased by a factor of e. In addition to the relaxation due to molecular interactions, captured in T_2, also inhomogeneities in the static magnetic field B_0 lead to a faster decrease in the transverse magnetization. The impact of the inhomogeneous magnetic field can be characterized by a second time constant $T_2^{(in)}$, which leads to an effective relaxation time $T_2^{(*)}$:

$$\frac{1}{T_2^{(*)}} = \frac{1}{T_2} + \frac{1}{T_2^{(in)}} \tag{1.69}$$

To obtain a signal that lasts long enough to allow for the reconstructing of a complete image, multiple fine-tuned radiofrequency pulses can be applied to the system. A radiofrequency pulse can flip the magnetization at a certain angle θ, depending on its duration t_P and its magnetic field component B_1:

$$\theta = 2\pi\gamma t_P B_1 \tag{1.70}$$

For most MRI sequences two flip angles are commonly employed. At first the 90° pulse rotates the equilibrium magnetization $\mathbf{M}_0$ vector by 90 degrees into the xy-plane, resulting in a precision of the magnetization vector around the z-axis. This precision can induce a current into a nearby coil, but due to the transversal relaxation, the signal strength would decay quite rapidly. To

recover some of the signal, a second type of pulse is applied to the system. The 180° pulse flips the net magnetization vector by 180 degrees. If the pulse is applied in parallel to the magnetization, it would orient the magnetization vector into its opposite direction. The 180° pulse thus also reverses the order of the spins, what allows a rephasing of the magnetization of spin packets. This trick can thereby be used to recover some of the original signal magnitude, as illustrated in figure 1.19. If the 180° pulse is applied after time τ the restored signal reaches its maximum at 2τ, what is referred to as the *echo time* (TE).

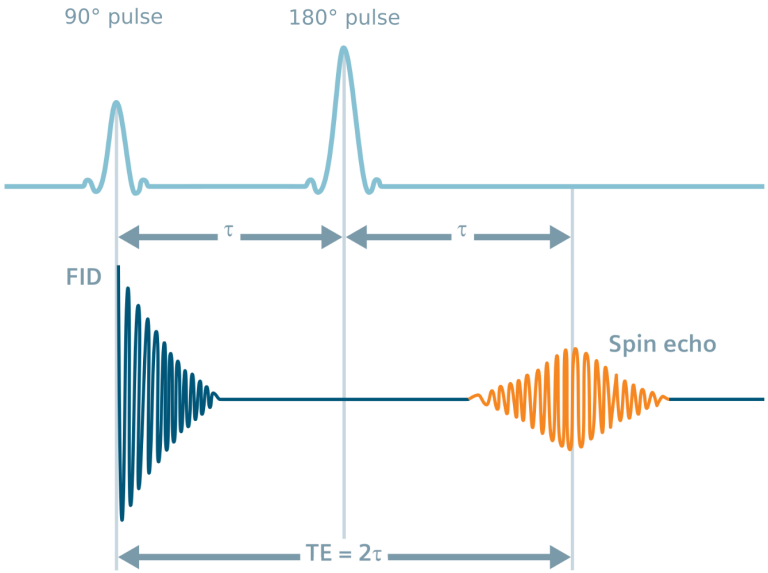

FIGURE 1.19: The figure depicts the process of the so-called spin echo. After applying the 90° pulse, the free induction decay (FID) decreases with time constant $T_2^{(*)}$. The original signal amplitude can be partially recovered by applying a 180° pulse at run time τ, what allows the spins to rephase again, until 2τ. The time between the 90° pulse and the maximum of the recovered signal as denoted as echo time (TE). Adapted from [66].

Such a sequence of pulses would only yield a signal generated by the average transverse magnetization of all spins in the system, but to obtain a spatially resolved signal, an additional sequence of gradients has to be introduced. The goal in MRI is to obtain high-resolution volumetric images of the body part, and in anatomical MRI it is possible to resolve a cubic unit (voxel) with a spatial resolution of around $1mm \times 1mm \times 1mm$, while in temporally resolved fMRI the typical resolution is in the range of $3mm \times 3mm \times 3mm$. The idea is therefore to exploit the linear dependence of the Larmor frequency ω_L on the magnetic field strength B, based on the relation described in equation 1.66. By turning on a linear gradient in z-direction, the Larmor frequency will therefore show a linear dependence on the location along the z-axis $\omega_L \propto z$. The gradient in z-direction is denoted as slice selection gradient, because

due to the induced spatial dependency of the Larmor frequency the radiofrequency pulse will excite only spins in a specific slice in the axial plane. The thickness of the slice is determined by the bandwith of the pulse, as well as the steepness of the gradient. The next step to resolve a volumetric image is to introduce a second gradient in the x-direction. After the slice selection gradient determined the axial examination area, the frequency encoding gradient along the x-axis is switched on during the readout of the signal. This leads to a linear dependence of the precession frequency ω_L along the x-axis. After the signal is acquired, which is composed of different frequency compartments emitted by spatially distinct voxels, this signal can be projected back into the spatial domain by applying a Fourier transformation. Finally for the third spatial dimension, a so-called phase-encoding gradient is employed. This gradient induces a phase shift in the rotation of the transverse magnetization along the y-axis, whereas the phase depends on the location along the y-direction. To recover the signal by using Fourier transformation, the phase-encoding gradient has to be switched on with different magnitude for every row of voxels along the y-axis [66]. An illustration of the frequency and phase encoding is provided in figure 1.20 (a), and the timing diagram of the gradient sequence is shown in figure 1.20 (b). The time interval between two such acquisitions is referred to as the *repetition time* (TR) of the MRI sequence. Based on these basic physical principles, MRI allows us to resolve volumetric images of the human brains with high spatial resolution in a non-invasive manner. Its different variants allow us to gain contrasts for different types of tissues and physiological processes observed in the human brain. In the following two for this thesis relevant imaging modalities, namely fMRI and DWI, will be discussed in more detail.

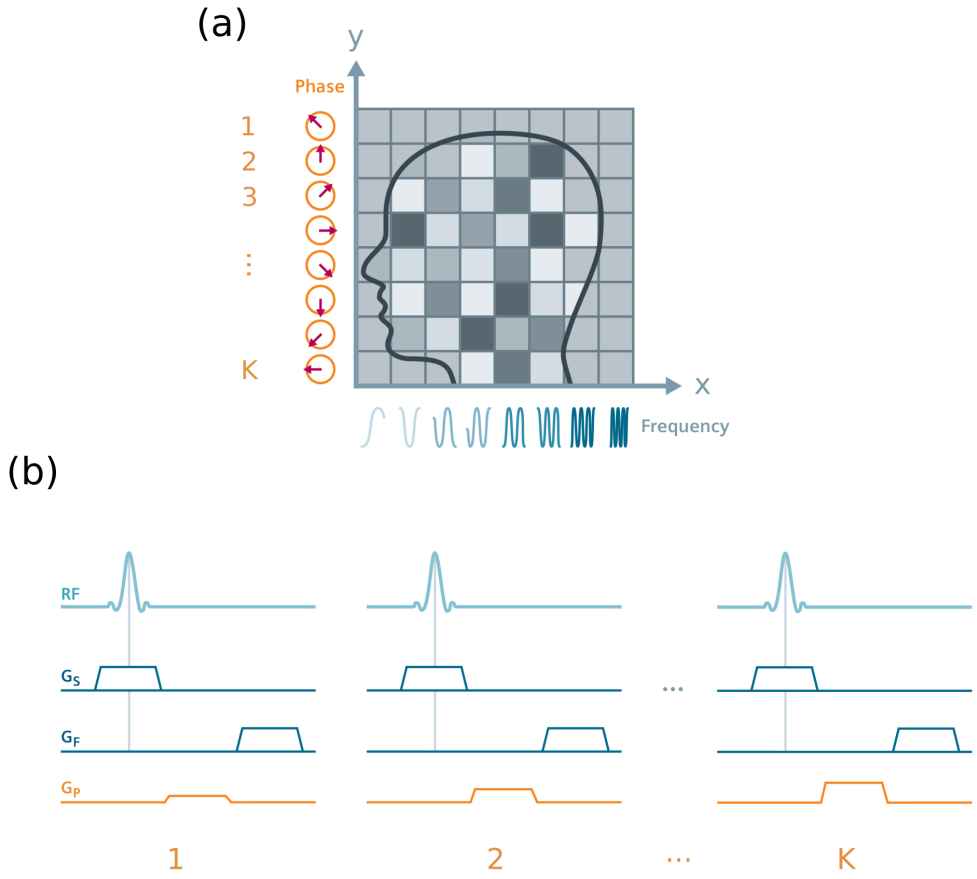

FIGURE 1.20: Figure (a) illustrates the frequency and phase encoding in one slice of voxels. By switching on the frequency encoding gradient, each voxel along the x-direction exhibits a specific Larmor frequency ω_L. Employing a phase encoding gradient along the y-axis, the phase of the precessing transversal magnetization becomes dependent on the y-direction. These local dependencies allow us then to spatially resolve the measured signal. Figure (b) depicts the sequence of the three gradients. At first, simultaneously with the radio frequency pulse (RF) the slice selection gradient is switched on (G_S), so only one axial slice gets excited by the RF pulse. In a next step the phase encoding gradient (G_P) is turned on, encoding the voxel intensities along the y-axis. In a third step the frequency encoding gradient (G_F) is employed to encode the third dimension along the x-axis. These three steps are repeated for each of the K voxels along the y-axis, each time using a phase encoding gradient with different magnitude. Adapted from [66].

1.3.2 Functional Magnetic Resonance Imaging

Functional MRI (fMRI) allows us to temporally resolve regional modulations of neural metabolism [53]. If a region in the brain is activated (due to visual stimulation for example) the region requires a larger amount of energy, leading to a higher *cerebral metabolic rate of oxygen* ($CMRO_2$) in that particular region [28]. To offset the rise of the $CMRO_2$, the bloodflow increases in that area to restore the local amount of O_2 [53]. First this process denoted as hemodynamic response leads to a decrease in oxygenated hemoglobin, and an increase in deoxygenated hemoglobin, while due to the vasodilatory response, afterwards the oxygenated hemoglobin level increases again and deoxygenated hemoglobin level decreases.

The contrast which is typically exploited in fMRI is based on magnetic properties of hemoglobin, and is denoted as the *blood oxygen level dependent* (BOLD) contrast. In its oxygenated state, hemoglobin is diamagnetic, and cannot be distinguished from the brain tissue, but in its deoxygenated state it becomes highly paramagnetic. This change in the magnetic susceptibility leads to magnetic field distortions and causes local gradients in the magnetic field, whereby the gradients magnitude depend on the amount of deoxygenated hemoglobin [53]. Nearby (water) protons are affected by these local distortions, which modifies their transversal relaxation times $T_2^{(in)}$ and $T_2^{(*)}$ [92]. As described in the previous section 1.3.1 the MRI signal strength depends on the strength of the transverse magnetization. Due to the decrease of deoxygenated hemoglobin this signal decay is slowed down, which results in a stronger signal in regions with higher neural activity [27].

Exploiting this kind of contrast, fMRI allows us to observe fluctuations in the BOLD signal in the brain with a temporal resolution of around 2 seconds and a spatial resolution of around $3mm \times 3mm \times 3mm$. After the fMRI data have been collected, commonly several steps for preprocessing and denoising are applied to the data [54]. Typical steps include slice timing correction, to account for the time delay between the acquired axial slices, and re-alignment of the collected volumetric images, to correct for movements of the head during the experiment. For further analysis the data of different subjects are usually transformed into a common space, using non-linear registration methods to transform the data in the volumetric MNI space or employing cortical surface constrained methods [50]. An example of pre-processed fMRI data is depicted in figure 1.21.

1.3.3 Diffusion Weighted Imaging

While fMRI allows us to temporally resolve neural activity distribution across the human brain, more recently it became of interest to additionally understand how these dynamic patterns are constrained by the structure of the brain. *Diffusion weighted imaging* (DWI) is a method that allows us to reconstruct bundles of white matter tracks, providing insights into the structural

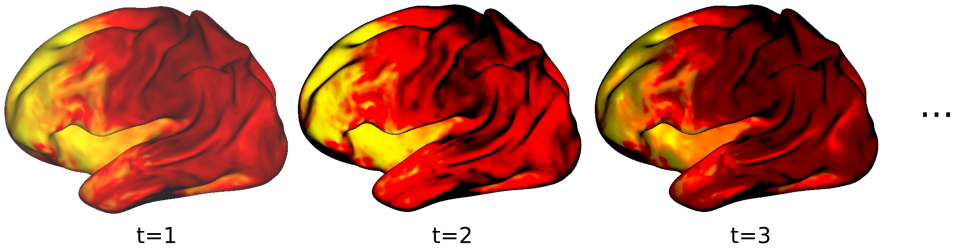

FIGURE 1.21: The figure shows a sequence of volumetric images as obtained from fMRI. With this functional imaging technique the BOLD signal can be obtained across the whole brain, sampled at different timepoints $t = 1, 2, 3, \ldots$ during an experiment. The signal values can be projected onto the surface of the brain and the intensity of neural activity values can be encoded using different colors, as illustrated in this figure. The resting-state fMRI data was obtained from a subject provided by the human connectome project database [122] and the figure was created employing the *Connectome Workbench* software (version 1.4.2).

connections between different brain areas. This imaging modality relies on the idea of measuring the diffusion of water in the human brain. In the absence of any boundaries, water molecules at room temperature would move randomly, without any preferred direction, which is denoted as isotropic diffusion [31]. But in neural tissue we can find aligned bundles of axons that constrain the movement perpendicular to their orientation [61]. This constraint results in an anisotropic diffusion of water predominantely in a direction parallel to the axon orientation. This diffusion of water protons can be detected in MRI by introducing two additional gradients. First the dephasing gradient induces a phase shift, depending on the position of the spin at $t = 0$. This dephasing gradient is applied before the 180° pulse in the MRI sequence, so the 180° pulse would reverse the phase shift caused by the first dephasing gradient [61]. A second gradient pulse applied after the 180° pulse would therefore rephase the spins again, but only if the spins at $t = \Delta$ are in the same position as before. On the other hand, spins that undergo some kind of diffusion would have changed their position during that time. Due to the spatial dependency of the gradients, spins that have changed their location would experience now a different gradient strength and the second gradient would not fully rephase them again. This would result in a signal loss in areas of high diffusion motion, which results in a contrast exploited in DWI. This process of spin dephasing is additionally illustrated in figure 1.22.

The difference in the signal intensity caused by the diffusion can be characterized by the so-called *Stejskal-Tanner* equation [31]:

$$S(b) = S_0 e^{bD} \tag{1.71}$$

Whereby $S(b)$ denotes the measured signal after applying the DWI gradients. The factor S_0 is the signal strength without applying the diffusion gradients,

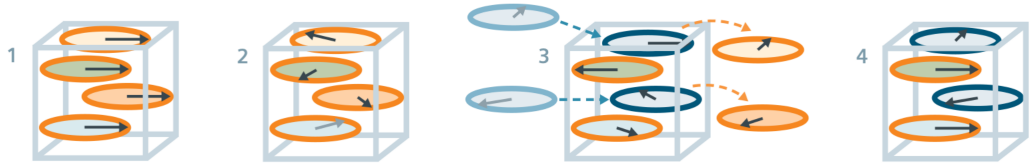

FIGURE 1.22: This figure illustrates the process of spin dephasing utilized in DWI. In this example of a single voxel, in the initial state (1) all spins are in phase. The first gradient pulse dephases the spins (2), and due to the diffusion, some of the spins move out of the voxel, while some spins remain in their original position (3). The second gradient pulse would completely rephase the stationary spins again, but due to the spatial dependency of the gradient, spins which diffused in or out of the voxel, will experience a different magnetic field strength and will not fully rephase again (4). This will result in a weaker signal in voxels with higher motion of spins, creating a diffusion weighted contrast. Adapted from [66].

and D denotes the diffusion coefficient, describing the magnitude of the water diffusion. The diffusion gradients are characterized by their so-called b-values, which depend on the following quantities:

$$b = \gamma^2 G^2 \delta^2 \left(\Delta - \frac{\delta}{3} \right) \tag{1.72}$$

Here γ denotes the gyromagnetic ratio of a hydrogen proton, G is the diffusion gradient magnitude, δ the duration of the applied gradient and Δ the time span between the dephasing and rephasing gradient. These quantities determine the amount of signal loss due to the diffusion of protons and by adjusting these parameters the contrast in an image can be influenced. An example of two diffusion weighted images acquired with $b = 0 \frac{s}{mm^2}$ and $b = 1000 \frac{s}{mm^2}$ is shown in figure 1.23 (a). By switching on a diffusion weighted gradient along a certain direction, the amount of molecular diffusion in this specific direction can be estimated. Modern DWI sequences sample multiple of such images with diffusion gradients along numerous directions and using different b-values. Based on these images the predominant diffusion directions can be estimated, and by randomly generating tracks following these directions, fiber-tracking algorithms are able to reconstruct paths of axon bundles [118, 14, 119], as shown in an example in figure 1.23 (b). In the context of fiber tracking, such a DWI sequence is also often referred to as *diffusion tensor imaging* (DTI) [11].

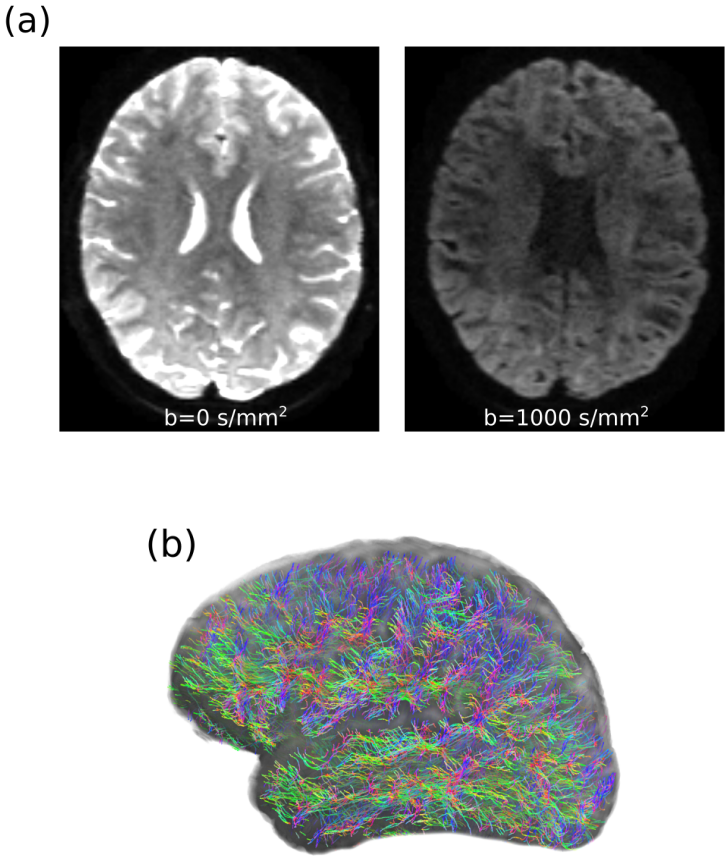

FIGURE 1.23: In (a) two axial image slices without diffusion weighting ($b = 0\frac{s}{mm^2}$) and with a diffusion gradient strength of $b = 1000\frac{s}{mm^2}$ are shown. Areas with stronger diffusion motion of protons exhibit a stronger signal loss and appear correspondingly darker in the image. The illustration in (b) shows a reconstruction of fiber tracks based on the acquired DWI data. In this example the spherical deconvolution model and porbabilistic fibertracking was employed to estimate the orientation of the depicted fiber bundles [118]. The different colors encode different orientations of white matter tracks. The DWI data were obtained from a subject provided by the Human Connectome Project database [122] and the figure was created employing the *MRtrix3* (version 3.0) software package [119].

1.4 Concepts of Brain Connectivity

A central question in neuroscience is how distinct neural populations in the brain exchange information when a participant is stimulated, performs a task, or simply is at rest. For studying such networks of communication in the human brain, research on the notion of *brain connectivity* has emerged, which currently gains increasingly more attention in the field of neuroscience [113, 78]. Different concepts of connectivity have emerged over the past years, ranging from anatomical links based on fiber tracks to statistical or potentially causal dependencies based functional neuroimaging data [113]. These types of connectivity can be roughly divided in two categories, denoted as *structural connectivity* (SC) and *functional connectivity* (FC). These concepts will be discussed in more detail in the subsequent sections, which are based on the literature in [78, 113] and our recently published survey on brain connectivity [127].

1.4.1 Structural Connectivity

As introduced in section 1.3.3, DWI (or DTI) can be used to reconstruct structural anatomical networks in the human brain. This network describes the spatial layout of white matter tracks, linking cortical and sub-cortical structures in the brain [127]. In most MRI studies the first step for constructing a connectome is to find a proper definition of the nodes in the brain network. Such pools of neurons can be aggregated in distinct areas defined by a brain atlas, as shown in figure 1.24 (a). But also data driven methods like independent component analysis (ICA) can be used to identify functionally independent areas in subject groups [30, 125]. Such areas of brain networks would thus represent the nodes in the graphical representations of brain networks. The edge strength in the graphical model can be defined by incorporating information on the brain structure derived from DTI. The most common quantification of SC strength is to simply count the number of fiber tracks, generated by probabilistic tracking methods, which connect two areas in the brain atlas, as illustrated in figure 1.24 [14, 119]. This brain network can now be formally defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A}_w)$, with $\mathcal{V}, |\mathcal{V}| = N$ denoting a set of vertices (or nodes), $\mathcal{E}$ representing the edges, and $\mathbf{A}_w \in \mathbb{R}^{N \times N}$ symbolizing the weighted adjacency matrix. In our context one entry $w_{nn'}$ of the adjacency matrix would characterize the anatomical connectivity between two nodes n and n'. This type of connectivity can be summarized and visualized in a SC matrix as shown in figure 1.24 (c).

The anatomy of the structural brain network is characterized by its substantial plasticity on longer time scales, usually due to its natural development, aging or disease [10, 62]. On shorter time scales, like the duration of a single fMRI experiment, it usually can be considered as static when comparing it to rapid fluctuations in functional brain activity [29]. Activity distribution, as observed in fMRI are mediated via propagating action potentials.

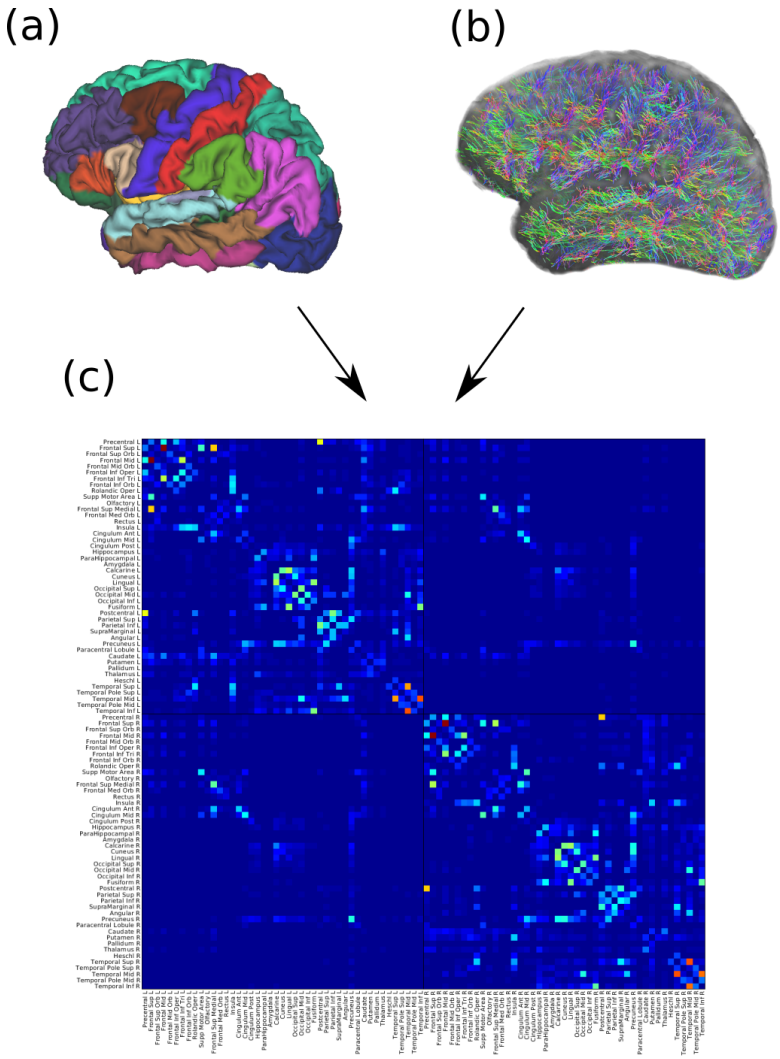

FIGURE 1.24: The figure illustrates how a structural connectivity matrix is reconstructed from DTI data. An atlas can be used to define segregated brain regions, which will represent the nodes in the structural brain network (a). The connectome of fiber tracks (b), reconstructed from DTI data, determines the edge strength between the nodes in the structural graph, so each pair of the N brain regions can be associated with a anatomical connection strength. These values can be collected in a $N \times N$ structural connectivity matrix (c).

Therefore the structural organization of neuron assemblies and their dendritic and axonal connections are considered to be the underlying physical substrate for information processing [127]. Following this idea, it became of interest to study the relationship to the functional organization of the brain, which can be described in the notion of functional connectivity.

1.4.2 Functional Connectivity

Functional connectivity (FC) characterizes the coherency of temporal activity fluctuations in two brain regions and is often considered as the dynamic

counterpart of SC [113]. As described in section 1.3.2, with fMRI we can temporally resolve the activity patterns in different spatial locations of the brain. In analogy to SC, we can also first use a brain atlas to define the nodes in our brain network, as illustrated in figure 1.25 (a). Then the activity of one region n at a certain timepoint t is usually computed as the average activity across all voxels within that region. If during one fMRI session T images are collected, we would thus obtain for each of the N regions an activity timecourse $x_n^{(t)}$ with $t = 1, \ldots, T$. Functional connectivity is expressed as a statistical dependence between such temporal activity patterns, and most commonly Pearson correlation is used to quantify the strength of FC between two regions n and n' [78]:

$$r_{x_n x_{n'}} = \frac{\sum_{t=1}^{T}(x_n^{(t)} - \bar{x}_n)(x_{n'}^{(t)} - \bar{x}_{n'})}{\sigma_{x_n}\sigma_{x_{n'}}} \tag{1.73}$$

whereby $\bar{x}_n$ describes the mean of activity values $x_n^{(t)}$ over the time t, and $\sigma_{x_n} = \sqrt{\sum_t (x_n^{(t)} - \bar{x}_n)^2}$ their variance over time. Besides Pearson correlation, also partial correlation or mutual information are common choices to derive statistical dependencies of temporal dynamics between different ROIs [113, 109]. If such a connectivity measure is computed for every pair of brain regions n and n', we can interpret those values as the edge weights in our graphical model of the brain, and visualize them by collecting them into a $N \times N$ matrix as shown in figure 1.25.

As in SC, with this coherency based definition of FC we obtain a symmetric measure of connectivity, so we cannot identify if a brain region A drives region B or vice versa. For such directed relations a third category of directed connectivity measures was introduced, studied in the notion of *directed functional connectivity* or *effective connectivity* [113]. This concept of directed connectivity analysis is in addition illustrated in figure 1.26. Even if the directed functional connectivity and effective connectivity have the same goal of inferring potentially causal dependencies in networks, they conceptually differ in their methodology [47].

Effective connectivity is typically derived from *dynamic causal modeling* (DCM), which is based on a mechanistic input-state-output model of neurons, with the goal to replicate the effective coupling strength between brain areas [46]. In DCM experimental conditions and stimuli can be encoded in predefined input functions, and the model output can then be related to hemodynamic responses as observed in fMRI. A Bayesian framework then estimates the effective couplings of neural populations, which can provide us with a neurophysiological perspective on potentially causal relationships between areas in brain networks. But due to the relatively high computational complexity of this type of modeling, the connectivity analysis with DCM is typically limited to a few pre-defined regions in the brain only. This limitation again might lead to the neglect of relevant areas for the brain connectivity

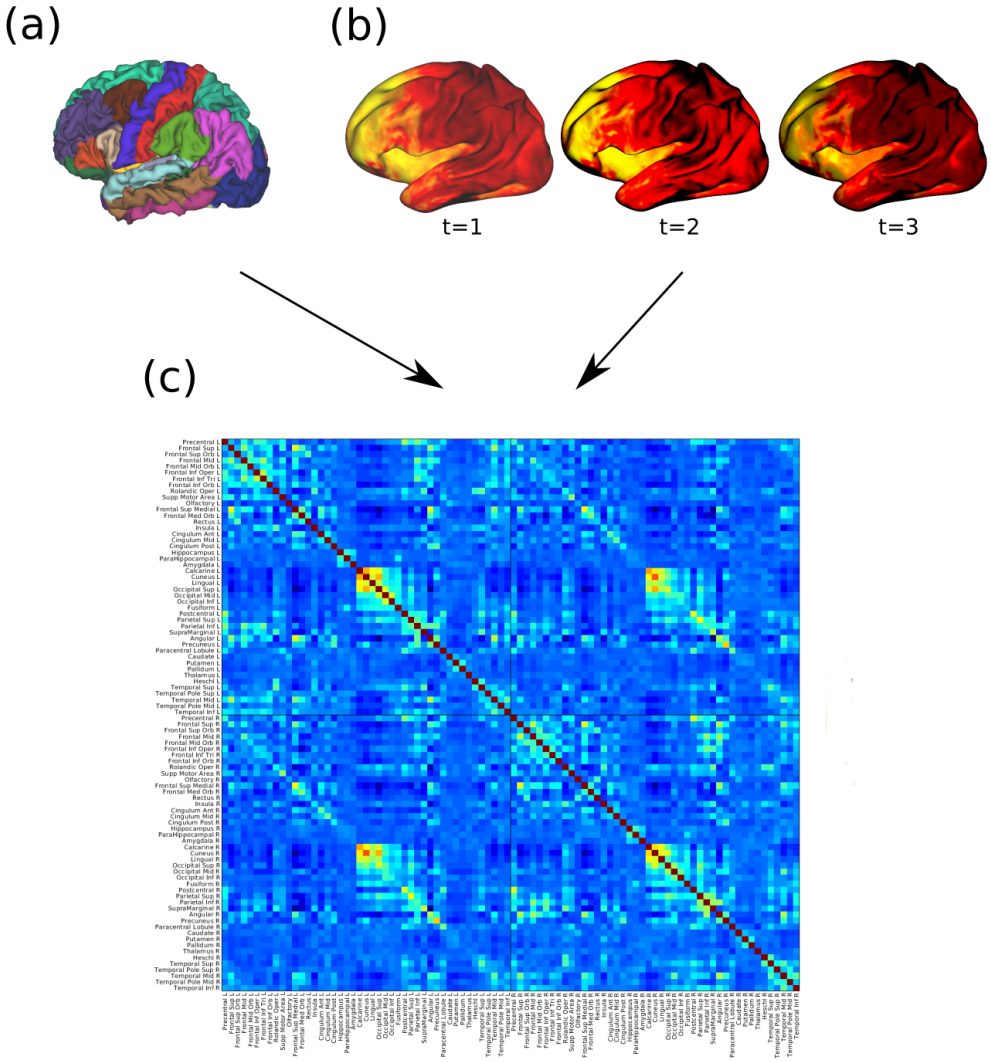

FIGURE 1.25: The figure illustrates how a functional connectivity matrix can be derived from fMRI data. A brain atlas can be used to identify brain regions, which define the nodes in the functional network (a). The temporally resolved activity patterns observed in fMRI (b) can be incorporated to characterize the activity within each brain region at different timesteps $t = 1, 2, \ldots, T$. Then statistical measures like Pearson correlation can be used to quantify the temporal coherency of neural activity in a pair of regions x_n and $x_{n'}$. The values of all $N \times N$ region pairs can be collected in a $N \times N$ functional connectivity matrix (c).

analysis [37], so it would be favorable to have an approach which also scales to large networks.

The second approach denoted as directed functional connectivity follows a simple idea proposed by the British econometrician Clive Granger [56]. If a certain event A causes another event B, then event A would precede event B, and information about the occurrence of A could contribute to the prediction of the occurrence of B. The temporal dependencies in fMRI data between the neural activity timecourses are usually described in the predictive framework

of a multivariate *vector auto regressive* (VAR) model [103]. With the objective to make accurate inferences about the temporal evaluation of neural signals, Granger causality tests if adding information about activity in one brain region B can help to improve the prediction of the activity in another region A (and vice versa). The VAR model is based on an linear autoregressive process, which assumes that a time series $x^{(t)}$ can be described as a superposition of the first T_p of its lagged values [85]:

$$x^{(t)} = \beta + \alpha_1 x^{(t-1)} + \alpha_2 x^{(t-2)} + \cdots + \alpha_p x^{(t-T_p)} + u^{(t)} \tag{1.74}$$

with coefficients $\alpha_1, \ldots, \alpha_p$, the intercept β and an error term $u^{(t)}$. The formulation of the auto regression model can be extended to a multivariate VAR model, with in total N time series $\mathbf{x}^{(t)} = [x_1^{(t)}, \ldots, x_N^{(t)}]^T$ as [85]:

$$\mathbf{x}^{(t)} = \mathbf{b} + \mathbf{A}_1 \mathbf{x}^{(t-1)} + \mathbf{A}_2 \mathbf{x}^{(t-2)} + \cdots + \mathbf{A}_p \mathbf{x}^{(t-T_p)} + \mathbf{u}^{(t)} \tag{1.75}$$

In this multivariate formulation the coefficients are stored in matrices $\mathbf{A} \in \mathbb{R}^{N \times N}$, and the intercepts and errors are represented by vectors $\mathbf{b} \in \mathbb{R}^N$ and $\mathbf{u}^{(t)} \in \mathbb{R}^N$. In our context of functional MRI, the goal of this model is to predict the BOLD signal $\mathbf{x}^{(t)}$ in all N brain regions from its past values $\mathbf{x}^{(t-1)}, \ldots, \mathbf{x}^{(T_p)}$. A measure for the directed connectivity strength in the notion of Granger causality can then be derived by comparing the prediction errors in ROI n with and without the information of activity in another ROI n'. This can confirm if n' contains some additional information on the activity in n and can provide a basis to identify a possible causal relation between n' and n [9].

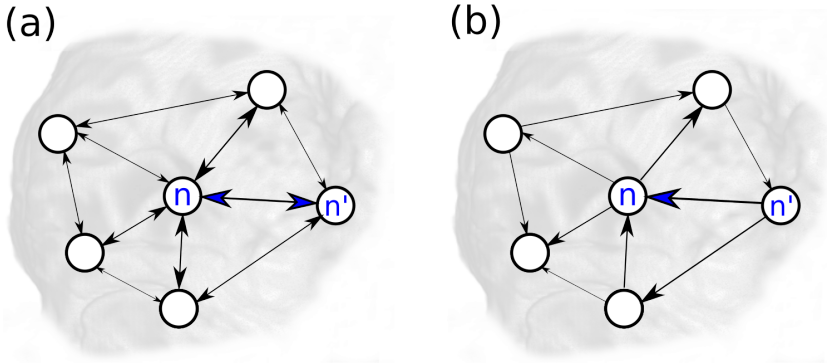

FIGURE 1.26: The figure illustrates the difference between undirected functional connectivity and directed functional/effective connectivity. Correlation based functional connectivity (a) provides us an undirected measure for connectivity strength, so we can not identify if a brain region n drives region n' or vice versa. Measures of directed functional connectivity or effective connectivity based on Granger causality or dynamic causal modeling try to reconstruct directed relations in brain networks (b). In this manner a change in the activity level in region n' can be identified as a potential cause for a change in the activity observed in another region n. Adapted from [127].

Chapter 2

Spatio-Temporal Graph Neural Networks for Brain Connectivity Analysis

2.1 Materials and Methods

Comprehending the interplay of spatial and temporal dynamics of neural activity is one key aspect for understanding how information is distributed and processed in the human brain. This chapter will discuss how spatio-temporal graph neural networks (STGNNs) can contribute to the study of such neural dynamics in complex brain networks. In general a brain network can be first defined by segregating the brain into distinct regions based on a brain atlas, thereby characterizing the nodes in our graphical model of the brain. In this graphical representation, the time-varying functional activity in the individual brain regions can be interpreted as a temporal signal in the nodes of our network. The edges in our network, shaping the interactions between the different regions, can be characterized by including spatial information on the brain anatomy. Based on this idea, such a neural activity distribution in a brain network can be interpreted as a graph-structured, time-varying signal. For this kind of geometric data structures the above discussed STGNN architectures provide novel possibilities to account for such spatial and temporal dependencies we observe in dynamic brain networks. The following section 2.1.1 will describe in which manner the STGNNs, introduced in section 1.2.6, can be applied to the analysis of spatio-temporal dynamics in brain networks. The subsequent section 2.1.2 provides a more detailed description of the acquisition protocols and preprocessing of the MRI datasets used in this thesis. Section 2.1.3 then discusses how the graph signals are generated from the functional and structural imaging data and finally section 2.1.4 outlines how the STGNN models are trained for the prediction of these neural signals. In the second part of this chapter, in section 2.2, the results of different applications for STGNN will be discussed. The following chapter is based on our publication [126] and the preprint of a manuscript [128].

2.1.1 Model Description

In the presented context of neuroimaging, the goal of the STGNN architectures will be to model the BOLD signal as accurately as possible, in order to precisely capture the underlying mechanisms of the spatio-temporal dynamics observed in brain networks. The objective of this learning task can be formalized by introducing a graph signal $\mathbf{x}^{(t)} \in \mathbb{R}^N$, representing the BOLD signal intensity in N different brain regions at a timestep t. Based on this idea, the STGNN will then try to learn to predict from an input sequence of T_p past neural activity states $t = 1, \ldots, T_p$ a sequences of T_f future activity states $t = T_p + 1, \ldots, T_p + T_f$. In addition to the temporal information in the BOLD signal $\mathbf{x}^{(t)}$, also prior knowledge on spatial dependencies can be included in these GNN architectures. The spatial layout which connects the N brain regions can be represented in the notion of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A}_w)$, composed of vertices (nodes) $\mathcal{V}$, with $|\mathcal{V}| = N$ and edges $\mathcal{E}$. The structure of the graph is defined by a weighted adjacency matrix $\mathbf{A}_w \in \mathbb{R}^{N \times N}$, whereby one entry $w_{nn'}$ of the matrix describes the spatial connection strength between brain region n and n'. An illustration of the graphical representation of a dynamic brain state is given in figure 2.1. Based on this definition of a brain state, the task of the STGNN models can be summarized in learning a function $f(\cdot)$ which predicts T_f future neural activity states from an input sequence of T_p past states:

$$[\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T_p)}; \mathcal{G}] \xrightarrow{f(\cdot)} [\mathbf{x}^{(T_p+1)}, \ldots, \mathbf{x}^{(T_p+T_f)}] \tag{2.1}$$

As the first predictive model, the diffusion convolution recurrent neural network (DCRNN) model will be studied in its ability to replicate empirically observed neural dynamics. This RNN based architecture provides one efficient way to detect patterns in such sequential data structures, like in our context the BOLD signal subsequently sampled at different timesteps t. In this sequence-to-sequence based architecture the encoder recursively processes an input sequence of T_p past neural activity states $\mathbf{x}^{(t)}$ and encodes the temporal information into a hidden state $\mathbf{H}^{(T_p)}$ [114]. The decoder part uses the information in $\mathbf{H}^{(T_p)}$ to generate a prediction for T_f future activity states. To account for vanishing gradients during training, the encoder and decoder of the DCRNN consist of gated recurrent unit (GRU) cells [34], which are modified to process graph-structured signals by invoking graph convolution operations. The detailed description of the DCRNN architecture DCRNN is provided in the first chapter in section 1.2.6.

As an alternative to this RNN based approach, patterns in sequential data structure can also be efficiently detected by convolutional neural network (CNN) models, as introduced in section 1.2.3. This principle is implemented in the graph WaveNet (GWN) architecture, which will be additionally included as a candidate to model spatio-temporal activity distributions in brain networks. By incorporating one-dimensional convolutions in the time domain, this model is able to capture dependencies in the temporal dynamics

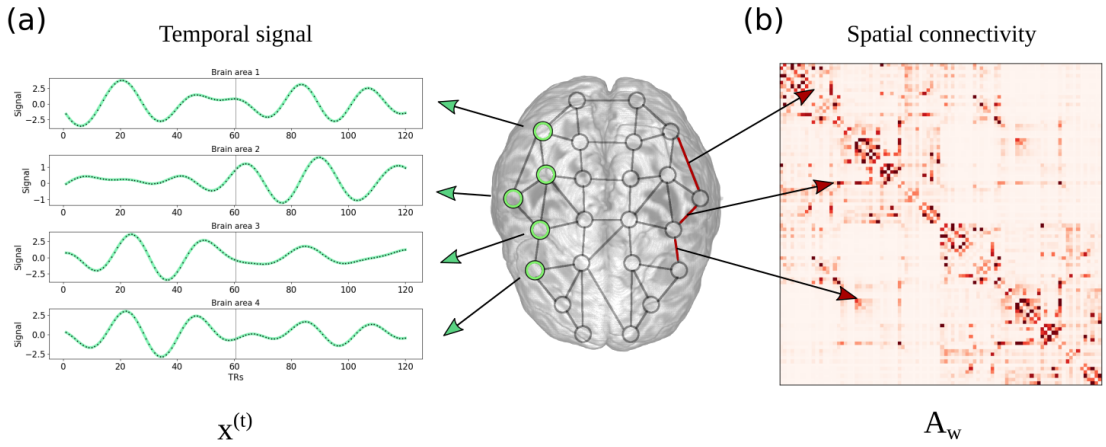

FIGURE 2.1: The figure shows the spatio-temporal representation of a graph-structured signal in a brain network. The temporal component of the signal is here marked in green (a). It is represented by the BOLD signal $\mathbf{x}^{(t)} \in \mathbb{R}^N$ in N brain regions, measured at different timesteps t. The edge connections of the graph signal, are highlighted in red (b). The connection strength between brain region n and n' is defined as an entry $w_{nn'}$ of the weighted adjacency matrix $\mathbf{A}_w \in \mathbb{R}^{N \times N}$, which characterizes the spatial relations between all N brain areas.

of neural signals. To account for long-term dependencies in temporal data, a WaveNet (WN) architecture serves as the foundation for this spatio-temporal model [121]. The WN exploits dilated causal convolution operations to generate a large receptive field with relatively few network layers, which alleviates the processing of long temporal input horizons. This principle of CNN based temporal modeling is then combined with graph convolution operations to account for the spatial dependencies in the graph-structured signal. The complete GWN architecture is described in detail in section 1.2.6.

In these two presented STGNN architectures, graph convolution operations model the propagation of information between adjacent nodes in graph-like signals $\mathbf{x}^{(t)}$ [40]. Here the neighborhood structure of the vertices or nodes $\mathcal{V}$ in the network are captured in an adjacency matrix $\mathbf{A}_w$. In the following further possibilities for defining this spatial layout for the information propagation between brain regions will be compared. As the first choice for an adjacency matrix the structural connectivity $\mathbf{A}_{SC}$ will be employed, which can be reconstructed from DTI data as described in section 1.3.3. This choice is motivated by the idea that white matter tracks obtained from DTI establish the anatomical layout for information exchange between brain areas. As an alternative approach, Rosenthal et al. [99] proposed to use the node2vec algorithm for projecting the nodes of the structural network $\mathbf{A}_{SC}$ into a continuous vector space. As discussed in section 1.2.4, such node representations can capture meaningful topological relations between different areas in a network. This so-called generated *connectome embeddings* (CEs) can capture long range and inter-hemispheric homotopic connections, which are usually only weakly expressed in DTI based anatomical connectivity [115]. In the context

of anatomical MRI, this technique is used to represent the edge weight $w_{nn'}$ in the adjacency matrix as the Pearson correlation-based similarity between the vector representations of two nodes n and n'. The pairwise similarities between all N brain areas can be collected in an adjacency matrix which will be denoted in the following as $\mathbf{A}_{CE}$. Based on this definition the information is propagated between brain regions which possess a high similarity regarding to their neighborhood role within the anatomical layout. Finally these techniques will be compared to the case when the model is given the freedom to learn the spatial dependencies between the N regions itself during training. In this case the adjacency relation is characterized by a self adaptive transition operator $\mathbf{A}_{Adap} \in \mathbb{R}^{N \times N}$, as described in section 1.2.4 (equation 1.38). In the following the effectiveness of these different spatial and temporal modeling approaches will be evaluated by comparing their predictive performance on empirical MRI data.

2.1.2 Datasets

In this study of STGNNs, two different MRI datasets will be incorporated for the evaluations [126, 128]. The first one was provided by the *Human Connectome Project* (HCP) data repository [68, 122]. The HCP S1200 release contains resting-state fMRI sessions, each with a duration of 14.4 min, whereby 1200 volumes were collected per session. Customized *Siemens Connectome Skyra* MRI scanners with a field strength of $B_0 = 3T$ and using multi-band (factor 8) acceleration were employed for the measurements [91, 44, 104, 135]. The data was collected using a gradient-echo echo-planar imaging (EPI) sequence with a repetition time of $TR = 720\ ms$ and an echo time of $TE = 31.1\ ms$. In total $N_s = 72$ slices with a field of view $FOV = 208\ mm \times 180\ mm$ and with a thickness of $d_s = 2\ mm$ were acquired, containing voxels with a resolution of $2\ mm \times 2\ mm \times 2\ mm$. The standard HCP preprocessing pipeline includes motion-correction, structural preprocessing and ICA-FIX denoising [50, 73, 72, 45, 110, 101, 58]. For the definition of the ROIs, the multi-modal parcellation scheme proposed by Glasser et al. [49] was used to divide each hemisphere into 180 segregated regions. Then the average of the BOLD signal in each brain region was taken to compute the overall temporal activity in each area. Next global signal regression was applied to the timeseries, firstly because it can effectively account for movement artifacts in HCP datasets [25]. Also in an application of directed connectivity analysis, the main objective is to extract the additional information, which certain regions contain about the activity in other regions, so that local interactions rather than global modulations in the signal are of interest for this analysis. The time courses were further bandpass filtered in the $0.04 - 0.07Hz$ frequency range. In a summary of several studies that account for different artifacts in the BOLD signal related to MRI scanner drift [106], respiratory and cardiac frequencies [20, 13, 18], and fluctuations in arterial carbon dioxide level [132], Glerean et al. [51]

and have found this $0.04 - 0.07Hz$ frequency band to be most reliable and relevant for gray matter activity in resting-state fMRI [2, 139, 24].

In the HCP, diffusion MRI data was acquired in 6 runs, whereby during each run approximately 90 diffusion directions were sampled, employing three shells with b-values of $b = 1000, 2000$, and $3000\ s/mm^2$, including 6 $b = 0$ images [111]. A spin-echo EPI sequence was incorporated for the image acquisition, with a repetition time of $TR = 5520\ ms$, a echo time of $TE = 89.5\ ms$, and using a multi band factor of 3. The volumetric images included in total $N_s = 111$ slices, with field of view size of $FOV = 210\ mm \times 180\ mm$ and a voxel resolution of $1.25\ mm \times 1.25\ mm \times 1.25\ mm$. The DTI preprocessing incorporates intensity normalization across runs, EPI distortion correction, removing motion artifacts, eddy-current corrections, and gradient non-linearity corrections [50, 112, 5, 7, 6]. The definition of regions in the structural brain network were in accordance to the functional preprocessing also based on the multi-modal cortical parcellation [49]. For the probabilistic reconstruction of fiber tracks, the *MRtrix3* software package was incorporated [119]. Multi-shell multi-tissue constrained spherical deconvolution [74] was used to compute the response functions for fiber orientation distribution estimation [118, 117]. Then anatomical constrained tractography was used to sample 10 million streamlines [107] and further spherical-deconvolution informed filtering was applied [108], reducing the number of streamlines to 1 million. The structural connectivity was defined as the number of streamlines connecting two brain regions, additionally normalized by the region volumes. The group structural connectivity matrix was computed as an average across the first 10 subjects, because the variance in the anatomical connection strength is typically very low across subjects [138], while probabilistic tractography methods are computationally very demanding. For the HCP dataset, including only young and healthy probands, the similarity in their structural connectivity profiles was relatively high. When computing the Pearson correlation between every possible pair of the 10 subjects, the correlation between the SC values was on average 0.91. But in other applications, when comparing considerably different subject groups to each other, for example in studies including healthy and patients with brain disorders, the SC matrices should be computed separately for each group.

The second MRI dataset was acquired at the Brain Imaging Center of the University of Regensburg (UR) at a *Siemens Magnetom Prisma* scanner with a field strength $B_0 = 3\ T$. Resting-state fMRI data of 10 different subjects was acquired with a scanning time of 7.3 min, thereby collecting 600 volumetric images per session. All subjects have provided written informed consent and the study was approved by the local ethics committee of the University of Regensburg. All methods were performed in accordance with the relevant guidelines and regulations. An EPI sequence was used with multi-band (factor 8) acceleration, employing a repetition time of $TR = 730\ ms$ and an echo time of $TE = 31\ ms$. In total $N_s = 72$ slices with thickness of $d_s = 2\ mm$

were collected, with a field of view of $FOV = 208\ mm \times 208\ mm$, and containing voxels with a resolution of $2\ mm \times 2\ mm \times 2\ mm$. For the preprocessing, the HCP pipeline (version 4.0.0) was incorporated, as proposed by Glasser et al. [50]. To obtain a good correspondence between the HCP and the UR dataset, the subsequent processing was implemented as described for the HCP data. The average of the BOLD signal was taken within each brain region of the multi-modal parcellation atlas [49], and global signal regression was applied to the timecourses. Again those time courses were then bandpass filtered within the range of $0.04 - 0.07\ Hz$.

The diffusion MRI data was collected in 4 runs by sampling approximately 90 diffusion directions, incorporating two shells with b-values of $b = 1500$ and 3000 s/mm^2, including 7 $b = 0$ images. For data acquisition a spin-echo EPI sequence was used with a repetition time of $TR = 3222\ ms$ with an echo time of $TE = 89.2\ ms$, additionally using a multi-band (factor 4) acceleration. In total $N_s = 92$ image slices were collected, including a field of view of $FOV = 210\ mm \times 210\ mm$, and containing voxels with a resolution of $1.5\ mm \times 1.5\ mm \times 1.5\ mm$. The further DTI preprocessing was based on the HCP pipelines [50], and for reconstructing the structural connectivity strength, again constrained spherical deconvolution was incorporated, as provided in the *MRtrix* software package [119]. Finally the group SC matrix was computed as the average over the 10 subjects.

2.1.3 Data Preparation

The following section will outline how the training, validation and test samples for the STGNNs are generated from the MRI datasets. To define the nodes in our brain network, each hemisphere was segregated into 180 regions based on the multi-modal parcellation proposed by Glasser et al. [49]. The signal in these network nodes are represented by the average BOLD activity within each region, so thereby in total $N = 360$ time courses were obtained (180 per hemisphere). During one resting-state fMRI session $T = 1200$ images were collected, and here the activity timecourses can be collected in a data matrix $\mathbf{X} \in \mathbb{R}^{N \times T}$.

After applying global temporal signal regression [25] and filtering the timecourses within the $0.04 - 0.07Hz$ range [51, 24, 21, 2], pairs of input and output samples were generated from the timeseries data in $\mathbf{X}$ [127]. The idea of the different models is to replicate the spatio-temporal dynamics in brain networks by learning to predict from a past sequence of brain states a sequence of future neural states. Therefore input and output pairs were generated from the data by selecting windows of length T_p to obtain input sequences of neural activity states $[\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T_p)}]$, and respective target sequences of length T_f denoted as $[\mathbf{x}^{(T_p+1)}, \ldots, \mathbf{x}^{(T_p+T_f)}]$. The time index t can be propagated through each session dataset, so in total $T - T_p - T_f + 1$ input-output samples were generated per fMRI session. The first 80% of those pairs were used for training the STGNN models, the subsequent 10% as a validation set, and the last

10% have been incorporated for testing. For the following evaluations of the models, the length of the input and output sequences were selected to be $T_p = T_f = 60$, which corresponds to a time span of roughly 43 s, based on a sampling interval of $TR = 0.72$ s [120]. The length of this time window has been shown to be long enough to be sufficiently challenging for the different models and to make clear the differences in their prediction accuracy. Likewise, the window of 60 timepoints is short enough for them to make reasonable, non-random forecasts of the signal.

In addition to the information on functional dynamics, which was derived from fMRI, knowledge on the spatial connectivity between regions can be obtained from DTI data. Here the DTI dataset provided by the HCP was processed using the multi-shell, multi-tissue constrained spherical deconvolution model [74], and the number of fiber tracks connecting two regions was used to determine the structural connectivity strength between the regions. The structural connectivity values between all N regions were then collected in a adjacency matrix $\mathbf{A}_{SC} \in \mathbb{R}^{N \times N}$.

As an alternative to this adjacency relation defined by the original structural connectivity ($\mathbf{A}_{SC}$), connectome embeddings allow us generate representations of nodes that capture higher order topological features of nodes in the structural network [99]. As introduced in section 1.2.4, the idea of a graph embedding is to represent each node in the graph by a Q-dimensional feature vector. In such an embedding subspace similar embeddings characterize the k-step ($k = 1, 2, \ldots, K$) relation between the vertices and their k-step neighbors [99, 59]. This technique was used to embed each brain region n of the SC network into a $Q = 64$-dimensional vector representation. Therefore the *gensim* python package [140] was incorporated, using the skip-gram model to learn the node representations [89]. In our application, the objective of the skip-gram model is to predict from a target node in the structural network its neighborhood context, whereby a sequence of neighboring nodes is sampled by performing a biased random walk on the graph, as described in section 1.2.4. For generating the node sequences, in total 100 random walks were performed for each node, with each walk consisting of 80 nodes. The return parameter of the random walk was set to $p = 2$ and in-out parameter to $q = 1$. To quantify the similarity of brain regions based on their role within the anatomical brain network, the Pearson correlation was used to compute a similarity score between all N brain regions. The correlation coefficients between all pairs of regions were then collected into an adjacency matrix $\mathbf{A}_{CE} \in \mathbb{R}^{N \times N}$.

2.1.4 Model Training

This section will describe how to the DCRNN and GWN are trained to make neural signal inference. For both models, the mean absolute error (MAE) was used as a cost function, quantifying the difference between the true BOLD signal intensity $\mathbf{x}^{(t)} \in \mathbb{R}^N$ and predicted signal $\hat{\mathbf{x}}^{(t)}$:

$$\mathrm{MAE}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{T_f} \sum_{t=1}^{T_f} |x_n^{(t)} - \hat{x}_n^{(t)}| \tag{2.2}$$

Here MAE is obtained as the average across all N brain regions and all T_f predicted signal values. The DCRNN model, which is based on a RNN architecture, was trained with the backpropagation through time (BPTT) algorithm [130], with the objective to maximize the likelihood of predicting true BOLD signal states. In addition, to account for a potential mismatch between training and testing distributions, a scheduled sampling technique was incorporated during the training of the DCRNN [82, 15]. The probability of using a true label during training as an input for the decoder decayed accordingly to:

$$\epsilon(i) = \frac{\tau}{\tau + \exp(i/\tau)} \in (0, 1) \tag{2.3}$$

whereby $\tau > 0$ denotes the decay parameter and $i \in \mathbb{N}$ counts the training iteration. In this learning task instances to be predicted are represented by the empirically observed BOLD signal. For the optimization of the DCRNN, the Adam algorithm [76] (as described in section 1.2.2) was used. The model was trained in total for 70 epochs on batches consisting of 16 samples. To improve convergence of the gradient-descent based optimization, an annealing learning rate was employed, initialized as $\eta = 0.1$, and decreased by a factor of 0.1 at epochs 20, 40 and 60, or if the validation MAE did not improve for more than 10 epochs. Every time before decreasing the learning rate in this manner, the weights with lowest validation error were restored, in order to escape local optima. The influence of the DCRNN model hyperparameters on the prediction accuracy are discussed in more detail in appendix B.1 (figure B.1). For the application of the DCRNN model, the hyperparameters were chosen to yield a reasonable trade-off between model accuracy and computational requirements. The encoder and decoder of the DCRNN consist to 2 diffusion convolution GRU layers each, and the hidden state size was set to 64. The experiments were performed using a *Nvidia RTX 2080 Ti* GPU, running on a desktop PC with an *Intel(R) Core(TM) i7-9800X* CPU under *Linux Ubuntu 20.04*. One epoch of the DCRNN on a dataset including 25 subjects and predicting the activity within one hemisphere (180 regions) took approximately 3.4 minutes with this setup.

Like the DCRNN, the GWN model was also trained using the Adam optimizer [76] to minimize the forecasting error of the BOLD signal defined in equation 2.2. It was found sufficient to train the GWN for 30 epochs with a

batch size of 8 samples, thereby initializing the learning rate as $\eta = 0.0001$ and decreasing it by a factor of 0.1 at epochs 10 and 20. The influence of the hyperparameters of the GWN is evaluated in appendix B.1 (figure B.2), and a good trade-off between model accuracy and computational complexity could be found using 32 feature maps in each CNN layer. In this architecture 2 causal convolution layers were used per block, with a total number of 12 blocks. With this setup one epoch on a dataset with 25 subjects including 180 ROIs took around 12.2 minutes.

2.2 Results

In the following different aspects of STGNNs in will be studied in their application for brain network analysis in MRI. At first in section 2.2.1, two different temporal modeling strategies will be compared to each other: The RNN based model, as implemented in the DCRNN, with the WN based architecture, as implemented in the GWN. In addition to these temporal models, different possibilities to account for the spatial information exchange between brain regions will be studied. Therefore the structural connectivity as a substrate for information propagation is compared to the structural connectome embedding similarity and to a self-adaptive adjacency relation. Based on these comparisons the most efficient STGNN architectures will be identified in order to model the spatial and temporal dynamics as observed in brain networks.

In a subsequent step in section 2.2.2, the STGNN based approaches are then contrasted to a currently popular data-driven approach for modeling directed relationships between brain areas. Granger causality is usually based on a vector auto regressive (VAR) model for multivariate timeseries inference, as introduced in section 1.4.2. In a brain network including N ROIs the parameters in a VAR model grow with N^2, so for large brain networks it can be challenging to accurately fit the model if only limited data are available in a study. Thus, for their application it would be practical to have a model for neural dynamics that is able to learn functional interactions between all areas of interest, and in addition it should scale to larger brain networks. The STGNN approaches will be therefore compared to the classical VAR model on a number of network and dataset sizes, to compare model predictions for different applications.

Moreover, spatial interactions between brain regions, which were learned by the STGNN models will be studied in more detail in section 2.2.3. It will be discussed how a perturbation based approach can be used to reconstruct directed relations between ROIs captured in these STGNN models. By integrating prior knowledge on the brain anatomy in form of structural connectivity or based on connectome embeddings, these models are able to provide us a multi-modal perspective on directed dependencies between brain areas.

Finally, the concept of transfer learning, as introduced in 1.2.2, will be presented for an application of connectivity analysis with STGNNs. It will be demonstrated in section 2.2.4 that by pretraining the DCRNN on 100 subjects of the HCP, the model accuracy on a smaller dataset from a different study, consisting of 10 subjects, can be improved. By transferring some of the learned characteristics of neural dynamics, this strategy can help us to a achieve a high model accuracy also on datasets from studies with limited sample sizes.

2.2.1 Spatial and Temporal Modeling in GNNs

Before studying the performance of the above-described STGNN architectures on a larger variety of MRI datasets from different experiments, we first focus on the effects of the temporal and spatial modeling in STGNNs. For this analysis, a dataset with a sample size of a medium sized fMRI study including data from 25 subjects was incorporated. From each resting-state fMRI session windowed input and output samples were created, as described in section 2.1.3, and the generated training, validation and test samples were then aggregated across all 25 fMRI sessions. The signals of regions within the right hemisphere were included in the following comparison, consisting of $N = 180$ ROIs based on the atlas proposed by Glasser et al. [50]. First the prediction accuracy of the different temporal modeling strategies will be evaluated, thereby comparing the recurrent neural network (RNN) based sequence-to-sequence learning with the convolutional neural network based WaveNet (WN) model. The STGNN hyperparameters, which are used for the following comparisons were discussed in the previous section 2.1.4 'Model training'. The BOLD signal data was scaled to zero mean and unit variance for the following evaluations, to obtain values of a magnitude that is easier to interpret. Figure 2.2 (a) shows the test mean absolute error (MAE) between the predicted and the true activity values. The error was averaged across all test samples, brain regions and the 60 predicted time points (corresponding to roughly 43s of activity). The comparison reveals that the RNN and WN model have very similar capabilities in predicting the BOLD signal. Despite their conceptual differences in their architecture, this points out that the RNN and WN based approach are able to both recover a comparable and consistent amount of temporal information from the fMRI data.

In a next step we will investigate the impact of adding information on spatial dependencies between the different regions in the brain network. This will be implemented by including graph convolution operations to the temporal prediction models. The definition of a adjacency matrix determines how information is propagated between the different nodes in the brain network, and in the following three conceptually different approaches will be compared to each other. First the structural connectivity as derived from DTI will be incorporated as the substrate for information exchange between ROIs. This SC based adjacency matrix $\mathbf{A}_{SC}$ is illustrated in figure 2.2 (b). The information can propagate along direct connections in the network $(K = 1)$, but also higher orders $(K = 2, 3, \ldots)$, describing the influence of indirect connections, can considerably contribute to interactions between ROIs [12, 83, 16]. A walk order of $K = 0$ represents the case when accounting for no spatial information exchange between network areas, exclusively integrating temporal information for the predictions. Figure 2.2 (c) shows the test MAE in dependence of the walk order K when using the SC derived from DTI as substrate for the information propagation. The RNN based model in combination with graph convolution operations was referred to as DCRNN [82] (section 1.2.6) and the

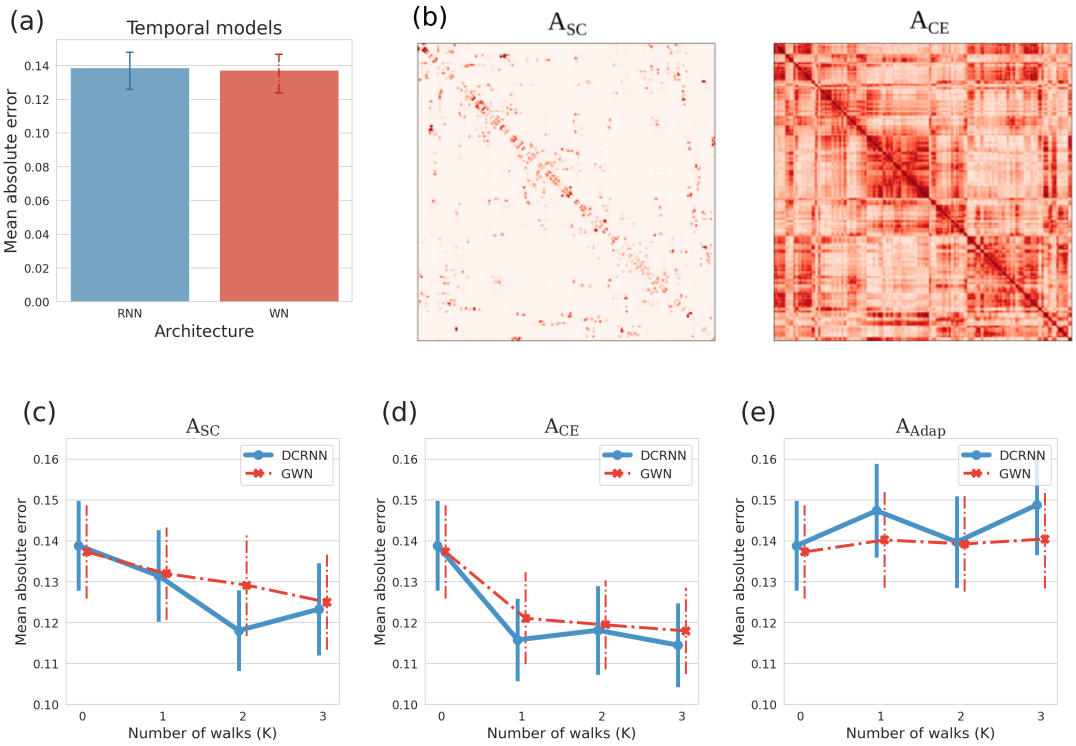

FIGURE 2.2: A comparison of the two different temporal modeling strategies for the BOLD signal is shown in (a), comparing the test MAE of the recurrent neural network (RNN) and the WaveNet (WN). The test error was obtained as an average across samples, brain regions and subject sessions. The error bars represent the standard deviations of the test MAEs across subjects. Spatial relations are added to the temporal models in form of graph convolution operations, and the spatio-temporal extensions of the RNN and WN models are respectively denoted as diffusion convolution recurrent neural network (DCRNN) and graph WaveNet (GWN) [82, 133]. Spatial transitions are captured in the weighted adjacency matrix, which is either based on structural connectivity ($\mathbf{A}_{SC}$), connectome embedding similiarity ($\mathbf{A}_{CE}$), or adapted during model training ($\mathbf{A}_{Adap}$). In (b) the adjacency matrix $\mathbf{A}_{SC}$ based on structural connectivity within the 180 regions of the right hemisphere is illustrated, together with the adjacency matrix $\mathbf{A}_{CE}$ based on structural connectome embedding similarities. The regions in this illustration are ordered according to the atlas proposed by Glasser et al. [49]. Figure (c), (d) and (e) show the forecasting errors of the DCRNN and GWN model in dependence on the walk order K. Note that $K = 0$ represents the case, when no spatial information exchange between regions is considered. In figure (c) the overall test MAE is shown when including the SC as an adjacency matrix $\mathbf{A}_{SC}$, figure (D) displays the test error when employing CEs in an adjacency matrix $\mathbf{A}_{CE}$ to define spatial relations, and (E) depicts the case when using a self-adaptive weight matrix $\mathbf{A}_{Adap}$.

MAE of its predictions, averaged across test samples, brain regions and predicted timepoints is shown here in blue. Figure 2.2 (c) points out that the DCRNN has the lowest test MAE when incorporating walks on the structural graph up to a order of $K = 2$. The WN incorporating graph convolution operations is denoted as GWN [134] (section 1.2.6) and its average test MAE is

depicted in red in figure 2.2 (c). The dependency of the walk order K on the GWN accuracy suggests that its performance can be successively improved by including first-order connections, followed by the second- and third-order connections. As an alternative to the original SC, the structural similarity between ROIs can be characterized based on their CE similiarity $\mathbf{A}_{CE}$, as also illustrated in figure 2.2 (b). A comparison between $\mathbf{A}_{CE}$ and the structural connectivity matrix $\mathbf{A}_{SC}$ reveals that in the adjacency relation based on structural CEs, long range connections between areas are considerably more pronounced. Figure 2.2 (d) shows the test MAE of the STGNNs when using $\mathbf{A}_{CE}$ in the graph convolution operations. In this case we can observe for both models a sharp drop in the error at walk order $K = 1$, what suggests that the similarity of node embeddings inherently account for higher order relations between nodes in the brain network. Finally, in figure 2.2 (e) the test MAE is shown when treating spatial connections between nodes as adaptive weights. In this case we do not observe an improvement in the error, which indicates that it is rather challenging to freely learn all N^2 connections between brain regions without prior knowledge. In general both STGNN architectures could profit the most when using CEs to characterize the spatial layout for functional interactions between brain areas. The DCRNN model had a test error of $MAE = 0.1388$ when including no information from other brain regions in the network, which could be then reduced to $MAE = 0.1158$ (for $K = 1$) when incorporating CEs to model the spatial information exchange within the brain network. To test the significance of this improvement on the accuracy in comparison to the case incorporating no structural modeling ($K = 0$), the overall test MAE for each subject was computed, and based on a paired t-test the impact of structural modeling has shown to be significant with $p \leq 0.0001$ for both STGNN models and both structural adjacency relations ($\mathbf{A}_{SC}$ and $\mathbf{A}_{CE}$). Although the performance differences between the GWN and DCRNN are quite small in general, the DCRNN slightly outperformed with a test error of $MAE = 0.1158$ the GWN with a test error of $MAE = 0.1211$ (also significant with $p \leq 0.0001$). This observation can show us that around 17% more information on functional dynamics can be directly retrieved from nodes with a similar higher-order context within the anatomical network. In contrast, using the SC to model transitions could only reduce the test error of the DCRNN by 5% at $K = 1$, which supports the idea that the structural node embeddings can directly strengthen the relationship between structural data derived from DTI with functional data observed in fMRI [99]. When applying a paired t-test, the improvement in the model performance when using the CE similarity in comparison to the original SC became for both, the DCRNN and GWN model, significant with $p \leq 0.0001$. By inherently capturing higher order transitions in $\mathbf{A}_{CE}$, only a low walk order K is only needed to account for information from structurally connected ROIs. In this way, this technique can contribute to efficiently reducing the number of parameters in STGNN models.

2.2.2 Model Accuracy and Network Scaling

In this section we have a closer look on the prediction accuracy of the above introduced STGNN based approaches. We evaluate their performances on different MRI study scenarios and compare their accuracy to the VAR, which is currently a popular method for directed functional connectivity analysis [47, 9, 17]. In real applications of such methods, the amount of available fMRI data may vary depending on the project size and on the type of subject cohort. Also the size of the brain network of interest can range from a few areas in a specific functional network to a large-scale whole brain analysis. For this purpose in the following different scenarios will be considered for analyzing the models' performance in dependence of the brain network size and the fMRI dataset size. For these evaluations one larger dataset including resting-state fMRI sessions from 50 subjects will be incorporated, one medium sized dataset of 25 subjects and one smaller dataset only consisting of data from 10 subjects. In addition, the size of the analyzed brain network will be varied. The first smaller network consists of 22 ROIs per hemisphere involved in visual processing as defined by the Glasser parcellation [49]. The complete list of selected ROIs is provided in the appendix B.2. The second, medium-sized network includes the regions within one hemisphere, and for this purpose the 180 ROIs within the right hemisphere based on the Glasser atlas were selected [49]. Finally the whole brain network consisting of in total 360 regions was used for the evaluation. As described in section 2.1.3 'Data preparation', windowed input and output sequence pairs were created from the timeseries data. The goal of the different forecasting models is accordingly to predict $T_f = 60$ timepoints of neural activity from the past $T_p = 60$ activity values. The hyperparmeters used for the STGNNs in this comparison are described in section 2.1.4 'Model training'. Further in this evaluation the CE similarity $\mathbf{A}_{CE}$ with transition order of $K = 1$ was used in the STGNN models, which has shown to improve the GNNs forecasting accuracy with low computational cost, as discussed in the previous section 2.2.1.

The VAR model was fitted to the BOLD signal timecourses using the ordinary least squares (OLS) method as implemented in the multivariate Granger causality (MVGC) toolbox [9], and for each dataset, the VAR model with order p that achieved the best MAE on the test set was selected. To check for stationarity of the BOLD signals, an augmented Dickey-Fuller test for unit roots was applied [63, 86], using a p-value of $p < 0.01$. For the 25 subjects dataset, roughly 10.0% of the BOLD timecourses did not fulfill the stationarity criteria of the augmented Dickey-Fuller test ($p > 0.01$) when using a high lag order of $T_p = 60$. But as the criterion for the following evaluations is the prediction accuracies of the models, the VAR model with the highest accuracy is chosen here for comparison with the STGNNs.

Figure 2.3 illustrates the test accuracy of the VAR, DCRNN and GWN model in dependence on the dataset size and brain network size. It is apparent in figure 2.3 (a) that if a large dataset of 50 subjects is available, all

three models are able to accurately predict the BOLD signal with a very low test MAE. A notable increase in the test error only becomes present for the VAR model, when it is fitted to the large whole brain network. Figure 2.3 (b) shows the test MAE when data from 25 subjects is incorporated. In that case the error of the VAR model starts to increase already noticeably when modeling activity distributions within a single hemisphere, and becomes quite large when including the whole brain network. In contrast to the VAR, the prediction accuracies of the DCRNN and GWN models remain stable in all cases. In the last case, when only 10 subject datasets are available, the test MAE of VAR model strongly depends on to the analyzed network size, as illustrated in figure 2.3 (c). On the contrary, the DCRNN and GWN model can still achieve a high accuracy also in this case when data are limited and the network size of interest is relatively large. Based on a paired t-test, the accuracy improvement of the DCRNN and GWN in comparison to the VAR were shown to be in all cases highly significant with $p \leq 0.0001$, except when the VAR is only fitted to the single visual network, where it still could make highly reliable forecasts.

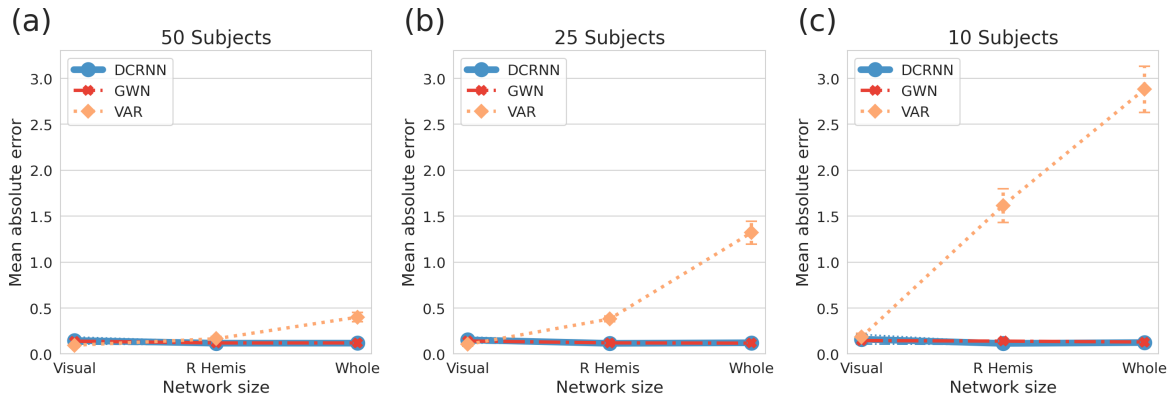

FIGURE 2.3: The figure shows a comparison of the prediction accuracies of the three models when varying the amount of data and the size of the network. The test MAE of the VAR is in this figure depicted in orange, the MAE of the DCRNN in blue and the MAE of the GWN in red. The MAE was computed as an average across brain regions, timesteps and test samples. In (a) the test MAE when employing a dataset including 50 subjects is shown for the visual network, the network within the right hemisphere and the whole brain network [49]. Figure (b) and (c) depict the test performances in dependence of the network size when using the 25 and 10 subject dataset, respectively. The error bars represent the standard deviations of the test MAEs across subjects, which are very small for the STGNN models, but clearly notable for the predictions of the VAR, when the datasize is limited.

To illustrate the prediction accuracies of the different forecasting models in more detail, an example of the predictions incorporating the dataset with 25 subjects, and modeling the activity within one hemisphere is shown in figure 2.4. Figure 2.4 (a) depicts the MAE of the models computed as an average across test samples and ROIs in dependence of the forecasting horizon. Within the first 15 timesteps all three models can generate very accurate predictions,

but after that period the test error of the VAR model starts to accumulate, while the GNN based approaches remain considerably more stable and precise. The predicted BOLD signals of the different models in a few representative samples are depicted in figures 2.4 (b), (c) and (d). To further validate the results, the analysis was replicated using a frequency filter for the BOLD signal in the $0.02 - 0.09Hz$ frequency range, and the respective results are presented in appendix B.3. In addition, this evaluation were replicated using a different dataset provided by the Brain Imaging Center of the University of Regensburg (UR), and the findings are discussed in appendix B.4.

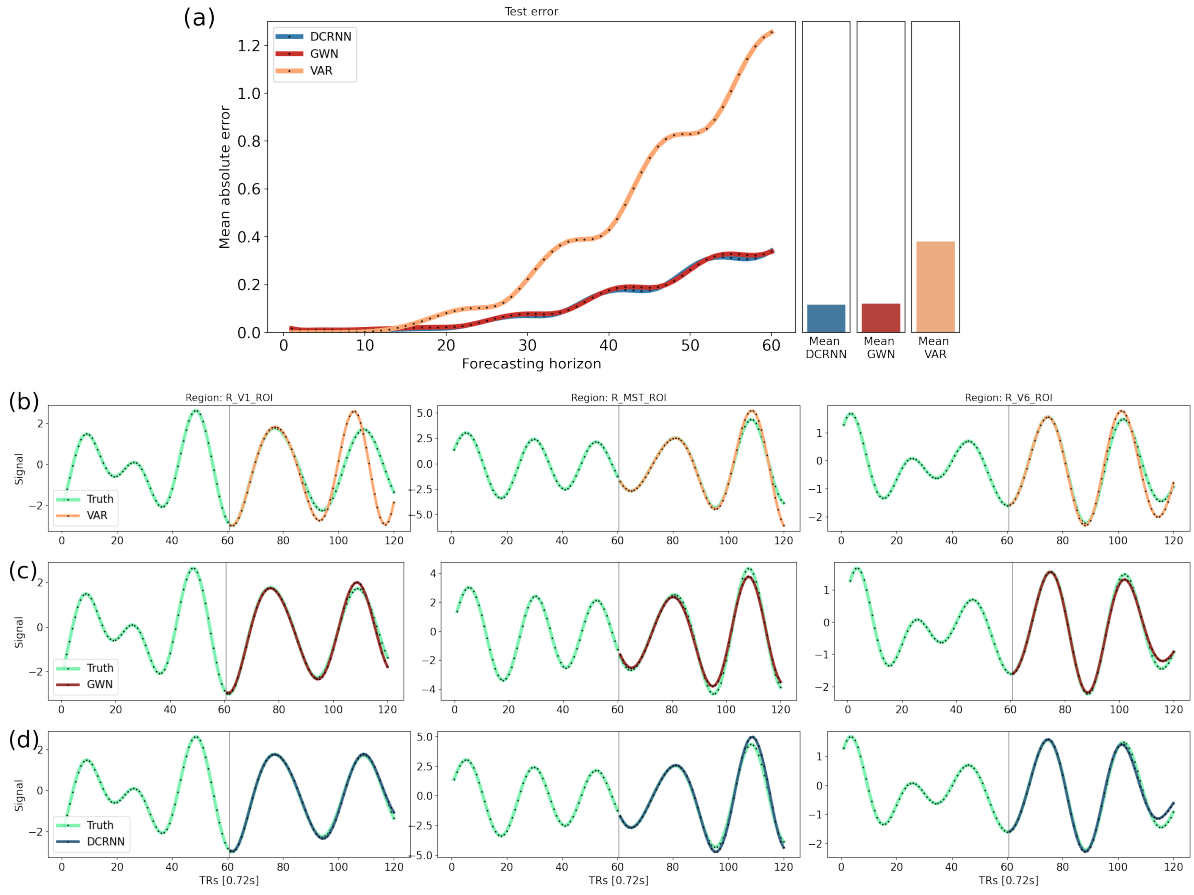

FIGURE 2.4: The prediction accuracy of the three models is presented in more detail for the 25 subject dataset and the brain network including the ROIs within the right hemisphere [49]. In (a) the test MAE in dependence of the forecasting horizon is shown, computed as an average across test samples and brain regions. Figure (b) depicts a representative example of predictions generated by the VAR model, and the error of the predictions in this example are with $MAE = 0.376$ slightly below its overall test MAE. Figure (c) illustrates an example of GWN predictions and the error in this example is with $MAE = 0.137$ slightly higher than its average MAE. Finally (d) shows the predictions of the DCRNN and the error is with $MAE = 0.120$ slightly higher than its average error.

Another interesting aspects of the predictive models is the dependence of the forecasting error on the respective analyzed brain region. Figure 2.5 illustrates the test MAE of the DCRNN, GWN and VAR model in dependence of the brain region within the right hemisphere. For all three models there appears a consistently greater prediction error in the posterior cingulate cortex and medial orbitofrontal cortex, which could possibly point towards a more complex temporal dynamic in those regions. Alternatively, the prediction accuracy might be affected by a lower signal-to-noise ratio observed in medial brain regions [93].

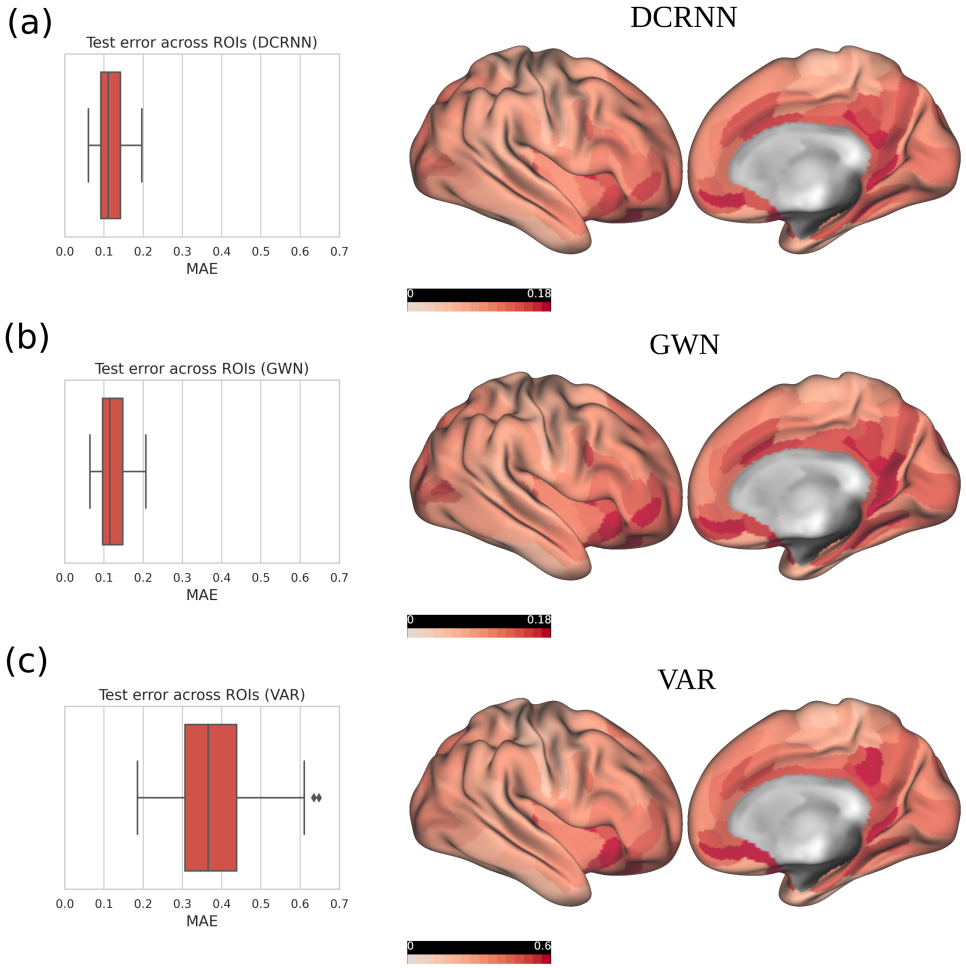

FIGURE 2.5: The distribution of the test error across the cortical surface is illustrated. In (a) the test MAE across brain regions of the DCRNN is first visualized in a boxplot, as shown on the left side. Accordingly on the right side of the figure, the MAE values projected onto the cortical surface are shown for the right hemisphere. The colormap was linearly scaled between 0 and 0.18. Respectively in (b) the MAE of the GWN is shown across regions and in (c) the MAE values of the VAR model. For the VAR model, the colormap was adjusted to account for larger error values by scaling it between 0 and 0.6.

2.2.3 Multi-Modal Directed Connectivity

Different approaches were compared in section 2.2.1 for the spatial modeling of dynamic interactions between regions in the brain network. The results have shown that adding information on the spatial relation between regions in the form of structural connectivity ($\mathbf{A}_{SC}$) or connectome embedding similarity ($\mathbf{A}_{CE}$) could considerably improve the prediction accuracy of the STGNN models. This points out that STGNNs are able learn relevant and functional informative transitions of neural activity based on the structural scaffold. By following the idea of Granger causality that the observation of one certain event A carries information about about the future occurrence of another event B, this could represent initial evidence for a potentially causal relation between A and B [56]. In this spirit, propagating the information between ROIs based on their SC or structural CE similarity would provide us a multi-modal perspective of such a directed and potentially causal relationship between brain areas. A perturbation based approach can be utilized to reconstruct the amount of information one ROI carries about other ROIs in the network [136, 126]. By learning a function $f(\cdot)$, the STGNN models try to infer from an input sequence of activity states $[\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T_p)}]$ a sequence of future states $[\hat{\mathbf{x}}^{(T_p+1)}, \ldots, \hat{\mathbf{x}}^{(T_p+T_f)}]$, whereby $\mathbf{x}^{(t)} \in \mathbb{R}^N$ denotes the neural activity at timestep t in all regions $n = 1, \ldots, N$. For inducing a perturbation into the model of neural dynamics, all information on activity in a specific ROI n' is removed by setting its activity values to the sample mean $x_{n'} = 0$. In a next step, by using the perturbed timeseries as an input for our trained model $f(\cdot)$, the model generates then a prediction $[\hat{\mathbf{x}}'^{(T_p+1)}, \ldots, \hat{\mathbf{x}}'^{(T_p+T_f)}]$. Finally, to reconstruct the directed influence of ROI n' on ROI n in our STGNN model, the overall difference between the original prediction and the prediction with perturbation in the input can be quantified as described in the following:

$$I_n(n') = \frac{1}{S} \sum_{s=1}^{S} \frac{1}{T_f} \sum_{t=1}^{T_f} |\hat{\mathbf{x}}_n^{(t)}(s) - \hat{\mathbf{x}}_n'^{(t)}(s)| \tag{2.4}$$

where $I_n(n')$ denotes the impact of ROI n' on n. Here $\hat{\mathbf{x}}_n^{(t)}(s)$ and $\hat{\mathbf{x}}_n'^{(t)}(s)$ denote the predictions in ROI n with and without the perturbation in n' of one test sample s at a time step t.

In the following this proposed measure of directed influence $\mathbf{I}(n')$ will be compared to classical undirected types of brain connectivity. First it will be contrasted to structural connectivity as derived from DTI, characterizing the number of fiber tracks connecting two brain regions (as described in section 1.4.1). Then the functional connectivity will be incorporated, defined as the Pearson correlation of BOLD signal timecourses between two brain areas (section 1.4.2). The above introduced GWN will be used in the following example to obtain a multi-modal measure of directed connectivity $\mathbf{I}(n')$, as defined in equation 2.4. First by employing the SC as substrate for information

propagation, captured in $\mathbf{A}_{SC}$, and then also using the similarity of CEs, represented by $\mathbf{A}_{CE}$. In the following example the connectivity of V1 within the right hemisphere will be studied incorporating the medium-sized 25 subjects dataset. For the comparison, all connectivity values are rescaled by normalizing them between 0 and 100. Then the connectivity values are visualized by projecting them onto the cortical surface as displayed in figure 2.6. In figure 2.6 (a) the structural connectivity is illustrated and the target region V1 is marked here in light blue. The strength of connectivity to all other regions is encoded in red color. Figure 2.6 (a) illustrates that we can mainly observe a pronounced structural connectivity between V1 and V2 and some structural connections leading to V3. Figure 2.6 (b) shows the undirected functional connectivity pattern in resting-state. In this variant of connectivity we can observe predominantly correlations to the functional activity in V2 and V3, but also a notable connectivity strength to V3, V4 and V6. In figure 2.6 (c) the directed connectivity strength $\mathbf{I}(n')$ is depicted, when using the SC as spatial backbone for the information exchange between brain regions in the STGNN model. In comparison to the plain SC, in addition to V2 a more pronounced relationship to areas V3 and V4 can be observed, and also to some anatomically more distant areas like V6 and the ventromedial visual area VMV1. This shows that this multi-modal type of brain connectivity additionally reflects the role of indirect structural connections by modeling higher order transitions on the structural scaffold, which are captured by the STGNN model. As an alternative to the SC, figure 2.6 (d) shows the directed connectivity patterns when using CE similarity as the spatial layout in the GWN. In this case we can recognize an even stronger integration of V1 within the visual network, which is in agreement with the observation that CEs can capture higher order topological relations in the anatomical connectivity [99]. Appendix B.6 illustrates additionally the spatial relations between brain areas captured by the DCRNN model. In this DCRNN based connectivity distribution a pronounced similarity to the directed connectivity pattern learned by the GWN architecture can be noticed, revealing more pronounced connectivity to areas like V3 and V4. Based on this observation, such a STGNN based connectivity approach can serve as a link between structural and functional connectivity and as such it can provide us a multi-modal perspective on directed dependencies between individual areas in brain networks.

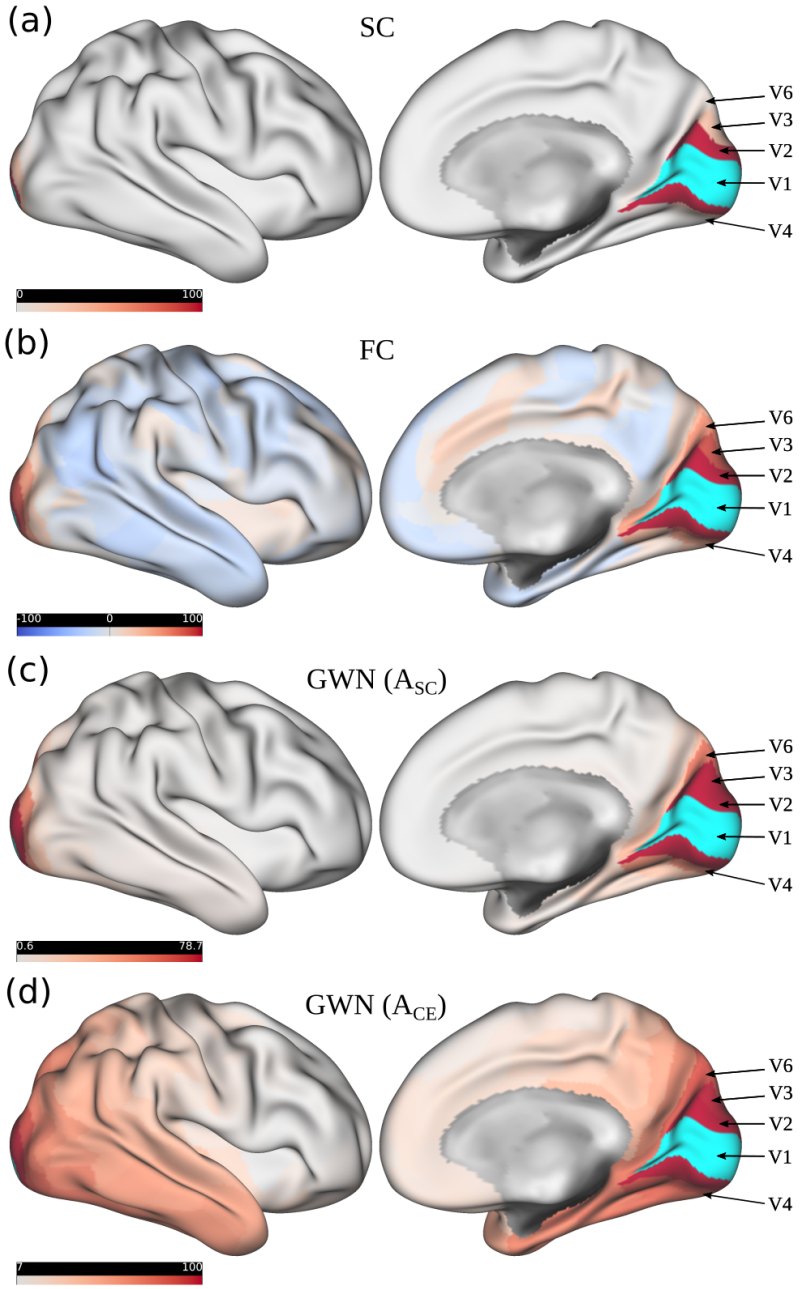

FIGURE 2.6: Different types of connectivity between V1 and all other regions are illustrated within the right brain hemisphere. In (a) the structural connectivity is depicted, whereby the target region V1 is marked here in light blue and the connectivity strength is encoded in red. In (b) the correlation-based functional connectivity is shown, which was computed as an average across the 25 subjects. Further (c) illustrates the measures of directed influence $\mathbf{I}(n')$, derived from the GWN model using the SC for information propagation. Finally figure (d) depicts the influence when incorporating CEs for the information exchange between ROIs. The values of the connectivity measure were linearly mapped between 0 and 100 (and between -100 and 100 for FC). The default scaling of the color values provided by the *connectome workbench* (version 1.4.2) was used, adjusting the colormap between the $2th$ and $98th$ percentile of the values respectively.

2.2.4 Model Generalization

Often a limitation in applications of more complex machine learning models is the amount of data available to properly train them. Especially in MRI studies it can be time-consuming and costly to acquire such large datasets. To address such issues, the concept of transfer learning was proposed in machine learning [94]. As introduced in section 1.2.2, the basic idea behind transfer learning is that if there are only limited amounts of data available for model training, one can pretrain the model on a large-scale dataset of a similar task. In a subsequent step, the feature representations learned on the large database can be used as an initialization for learning the desired target task. If the feature representation of the source domain is diverse enough, model performance can be improved in comparison to starting the training without any prior knowledge, e.g. relying on a random initialization of the model weights [94].

To investigate if transfer learning might also be suitable for spatio-temporal modeling in MRI, the capabilities of the DCRNN to generalize across different datasets will be studied in the following. Therefore the DCRNN was pretrained using the large-scale dataset provided by the HCP [122], as described in the section 2.1.2 'Datasets'. The input and forecasting horizons of the BOLD signals have been here selected to be of length $T_p = T_f = 30$ for the following analysis [126]. The DCRNN model was pretrained for in total 70 epochs on 100 resting-state fMRI sessions (4 sessions from 25 subjects), in addition using their structural connectivity as reconstructed from DTI. Then the dataset acquired at the University of Regensburg (UR) was used, as described in section 2.1.2. Here 10 different subjects participated in a resting-state fMRI sessions, including a DTI session. Each resting-state session of the UR dataset lasted 7.3 min, whereby 600 fMRI images were collected during each session. In correspondence to the larger HCP dataset, the UR data were further processed by windowing the average BOLD signals in the ROIs defined by Glasser et al. [49], thereby obtaining windows with an input and output length of $T_p = T_f = 30$ timepoints. The first 80% of these input-output samples were used for training, the subsequent 10% for validation and the final 10% for testing. Then the DCRNN, pretrained on the HCP data, was fine-tuned in a next step by training it for 70 more epochs on the UR dataset. The second training was initialized with a smaller learning rate of 0.001. This pretrained model was compared to the model only trained on the UR dataset, and with weight parameters initialized randomly with Xavier/Glorot initialization (as introduced in section 2.1.4).

A comparison between relying on standard training with a random initialization, and utilizing transfer learning is illustrated in figure 2.7. Figure 2.7 (a) depicts the training and validation error during learning when starting with a random initialization of the weights in red. This model was trained in total for 140 epochs on the UR dataset only. In blue the training and validation error is depicted of the model, initially pretrained on the larger HCP dataset for 70 epochs, and then fine tuned on the UR dataset for the subsequent 70 epochs.

Figure 2.7 (a) shows that at onset, the training error on the UR data is relatively high, but as the pretrained model adapts to the new dataset, the MAE becomes considerably smaller than without the pretraining. In figure 2.7 (b) the test MAE in dependence of the prediction horizon is illustrated. In total 540 test samples from 10 different subjects were used for the model testing on the UR dataset. The overall test error could be reduced by 27% from 0.0388 to 0.0284 by encompassing transfer learning. In this way, the model accuracy on the small UR dataset, containing 10 sessions a 7.3 min, becomes comparable to the accuracy on the large HCP dataset with 100 sessions a 14.4 min with a $MAE = 0.0279$. In addition, to evaluate the significance of this improvement across subjects, the test MAE with and without pre-training the model was computed for each of the 10 subjects. Then a paired t-test was incorporated and the difference was significant with $p \leq 0.0001$.

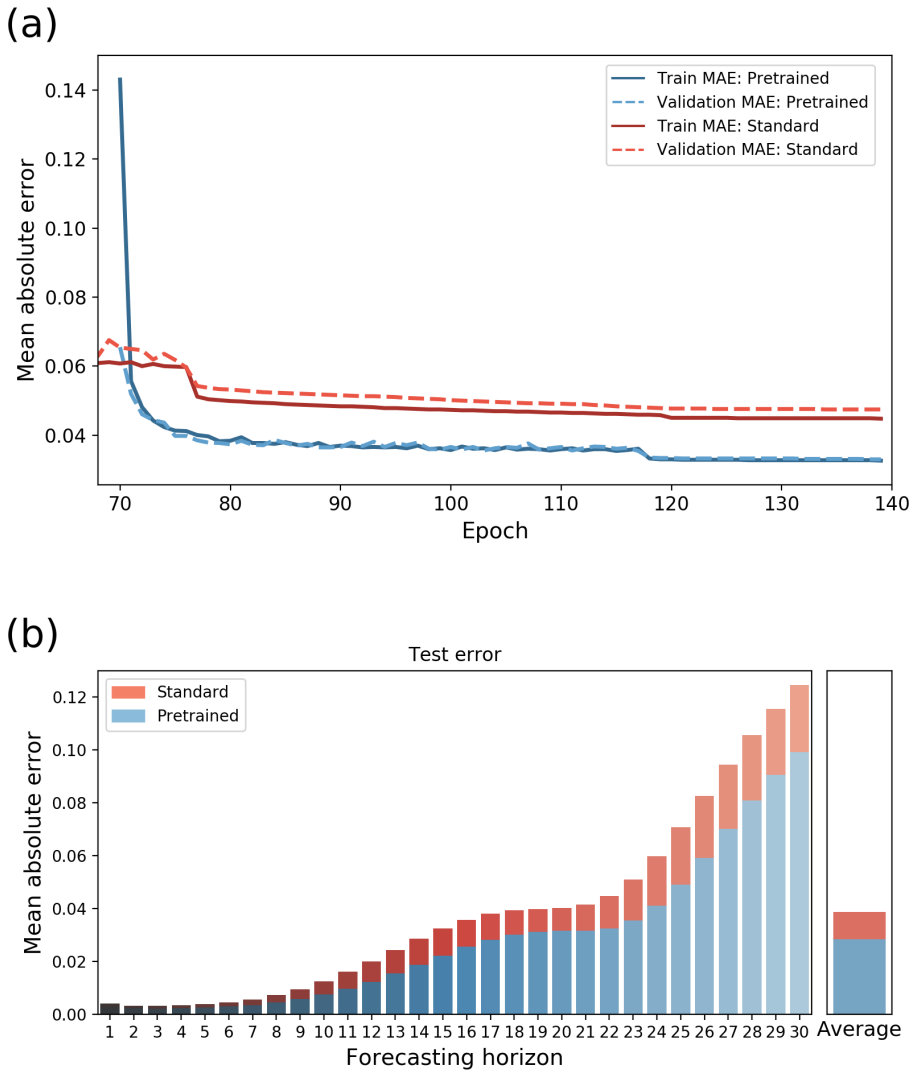

FIGURE 2.7: The performance difference between standard training and encompassing transfer learning is shown. Figure (a) illustrates the validation and training MAE during model training from epoch 70 onwards. The errors with and without pretraining are depicted in blue and red respectively. The error values were computed as the average over all subjects, sessions, brain regions and test samples. At the very beginning of fine tuning, the error of the pretrained model is relatively high, but decreases after the model adapts to the new UR dataset. In figure (b) the final test MAE of both models is shown in dependence on the forecasting horizon. Adapted from our publication [126].

Chapter 3

Conclusion

3.1 Discussion

In this thesis a novel technique based on spatio-temporal graph neural networks (STGNNs) was presented for studying the characteristics of spatial and temporal dynamics in complex brain networks. After deriving the theory of STGNNs, the basics of MRI and current concepts of brain connectivity in chapter 1, several aspects and applications of STGNN for brain connectivity analysis were introduced in chapter 2. At first, different artificial neural network architectures for replicating the temporal dynamics in the BOLD signal have been studied in section 2.2.1. The comparison could reveal that a RNN based model and a WN based model have very similar capabilities in detecting temporal patterns in the neural activity dynamics. Despite their conceptual differences in their architectures, they demonstrated almost the exact same prediction accuracy, which indicates that they are both very consistent in capturing the temporal information in functional imaging data. In a subsequent step the impact of adding spatial dependencies was examined, which was realized by adding graph convolution operations to these temporal models. Different spatial layouts have been compared to account for the information propagation between brain regions, either based on the structural connectivity ($\mathbf{A}_{SC}$), the CE similarity ($\mathbf{A}_{CE}$), or a self-adaptive adjacency matrix ($\mathbf{A}_{Adap}$). While the model performance of the GWN and DCRNN steadily improved by including higher walk orders K on the anatomical substrate, a more pronounced improvement was already observed when incorporating structural CEs with a walk order of only $K = 1$. This embedding strategy turns out to be therewith also interesting in applications of STGNNs, because it has the potential to effectively incorporate indirect structural connections with a considerably lower learning complexity, which depends linearly on the walk order K [40]. These observed characteristics of CEs in this proposed application support the ideas of Rosenthal et al. [99], which have demonstrated in their study that node embeddings of the structural network can naturally capture higher order topological relations between ROIs. Also in the presented context of modeling spatio-temporal dynamics this method has shown to strengthen the relationship between brain structure and functional dynamics.

The STGNN models were then compared to the current popular approach for directed brain connectivity analysis. In Granger causality analysis the VAR model is still predominantly used for the inference of directed relationships between brain regions [47, 9, 17]. For this comparison in section 2.2.2, the accuracy of the different approaches was evaluated on a variety of brain network sizes and dataset sizes to account for different possible scenarios in their applications in MRI studies. The results could demonstrate that if a sufficiently large cohort of 50 subjects is available in a study, a VAR model is able to make very reliable long-term predictions, and only for a large network consisting of $N = 360$ regions there is a notable increase in the prediction error. But the dependency of the accuracy on the size N of the brain network becomes more apparent when data from only 25 subjects are used to fit the VAR model. Finally in a case where only 10 subjects are available for the analysis, the error of the VAR grows strongly with N. This demonstrates that a VAR is a fast and reliable model for fMRI studies with a sufficiently large test subject size and for connectivity studies including a limited amount of pre-defined regions. But in certain cases it might be desirable to include a larger amount of brain areas into the brain connectivity analysis, in order to avoid omitting relevant areas in the studied network of interest. Also in some MRI studies it can become very costly and time-consuming to collect a large amount of data, which is, for example especially challenging in studies on rare neurological disorders. While the number of parameters in a VAR based approach grow with an order N^2, the spatial modeling in STGNNs, based on localized graph convolutions, is independent of the number of ROIs N. This property enables the presented STGNN based approaches to make very robust inferences also on large networks and when only limited data are available, thereby providing a considerably more flexible method for different network analysis scenarios.

In a subsequent step, spatial interactions between regions learned by the STGNN models have been studied in more detail in section 2.2.3. By integrating information on the anatomical substrate into the STGNNs, a multi-modal measure of connectivity strength could be derived from these models for identifying directed and potentially causal relationships between brain regions. When comparing this measure of directed influence to classical structural connectivity, some transitions along higher order structural connections in the brain network could be additionally observed in this multi-modal measure. The STGNN models could detect links between $V1$ and $V2$, but also additional prominent connections to $V3$ and $V4$. The influence of such higher-order connections became even more apparent, when a CE based similarity $\mathbf{A}_{CE}$ was incorporated to define spatial node relations. In that case a very high integration of $V1$ within the visual system could be observed in the connectivity profiles. In this manner, this spatio-temporal analysis approach based on STGNNs can serve as a link between features observed in structural and functional imaging data for studying brain connectivity from a multi-modal

perspective. Due to the relatively low temporal resolution in fMRI [47], and the indirect measurement of the underlying neural signals based on their hemodynamic response [124], one should also be aware of these limitations in the inference of directed and potentially causal connections in fMRI studies [109]. A lag-based predictive approach based on STGNN models might therefore also be affected by the same limitations as classical Granger causality in fMRI. On the other hand, a combined fMRI-MEG study by Mill et al. [90] and different computational simulations of fMRI data [123, 103, 129, 42] could meanwhile establish evidence that Granger causality is still able to identify meaningful directed relationships between brain regions in fMRI, despite the indirect inference based on the hemodynamic response. As an alternative, deconvolution based approaches can have the potential to reconstruct from the measured BOLD signals the underlying neural timeseries [26, 90] for assessing *effective* brain connectivity, rather than only estimating *directed* functional connectivity from the original BOLD signals [17]. But the estimation of the underlying hemodynamic response from the data might come with the cost of introducing additional assumptions and uncertainties into the analysis [96, 17]. Despite these current potential limitations in fMRI, a multi-modal GNN based approach allows us to join structural and functional imaging data in a new manner, and reveals thereby its potential for supplementing current analysis methods in brain connectivity research [95].

Finally, in section 2.2.4 an approach was presented, which can improve the model performance on smaller MRI datasets. It was demonstrated that the concept of transfer learning [94] finds also an application in our context of spatio-temporal modeling in MRI. Features learned from the large-scale data of the HCP repository [122] could be well transferred to a smaller dataset, acquired with a *Siemens Magnetom Prisma 3T* at the UR. This strategy made it possible to achieve almost the same accuracy on the smaller UR dataset including 10 fMRI sessions (each 7.3 minutes in duration) as with a large dataset including 100 sessions (each 14.4 minutes in duration). The acquisition and preprocessing protocols of the two datasets were relatively comparable in this example, so in other cases with larger differences in the temporal resolutions of the fMRI data, downsampling one dataset might be necessary in order to achieve a higher similarity between them and to obtain comparable feature representations. In this manner, in studies with a limited amount of data available, this pre-training strategy has the potential to improve the accuracy of STGNN models in such challenging cases.

3.2 Outlook

For applications and investigations based on STGNNs, several conceptual and methodological aspects might be of interest in neuroimaging research in the future. In the presented analysis in section 2.2.3, a perturbation based approach was used to reconstruct the spatial dependencies between ROIs which were learned by the STGNN models. Alternative ways to detect such dependencies among the models input variables could be provided by recent approaches proposed in the notion of *explainable artificial intelligence* (XAI) [116]. Techniques developed for artificial neural networks like sensitivity analysis [105] or layer-wise relevance propagation [79] might be interesting alternatives to explain the relations between brain regions learned by STGNN models.

Further these whole-brain models might be of interest for clinical research questions. These spatio-temporal models could provide a possibility for studying neural dynamics in the diseased brain and could be utilized to investigate how functional interactions between different areas might be affected by pathological brain states. Similarly, these multi-modal STGNN models could also be applied to simulate the impact of a structural lesion to investigate the effects of such lesions on the brain functions [3].

Besides applications of STGNN in fMRI studies, alternative functional neurophysiological techniques like electroencephalography (EEG) or magnetoencephalography (MEG) might be interesting for analyzing temporal dynamics in the high frequency range. This could allow us to study dynamic functional interactions with a considerably higher temporal resolution, and could provide us a more detailed perspective on directed and causal dependencies in brain networks. Further, structural imaging techniques like neurite orientation dispersion and density imaging (NODDI) [137] might capture additional interesting aspects the brain structure, which could be included as structural spatial information in these STGNN based models. In general GNNs still comprise a relatively new field in machine learning research and recent developments in this area have the potential to likely make further interesting contributions to our understanding of information processing in brain networks [60].

3.3 Epilogue

In this thesis a new approach based on STGNNs was developed for modeling spatial and temporal dynamics observed in complex brain networks. One of the main advantages of this method is its effective scaling to large brain networks. This property of STGNNs allows us to study large-scale neural dynamics, also in cases where the amount of MRI data is very limited. The second main contribution of STGNNs to brain connectivity research is that they provide us with a new possibility to link structural and functional neuroimaging data. Based on dynamic functional interactions, constrained by the structural backbone, directed relations captured in STGNN models allow us to study the structure-function coupling in brain networks from a new viewpoint. The codes and a demo version for the DCRNN and GWN model, modified for the analysis of brain connectivity, are publicly available under:

https://github.com/simonvino/DCRNN_brain_connectivity
https://github.com/simonvino/GraphWaveNet_brain_connectivity

This STGNN based method for brain connectivity analysis was first published in:

> Wein, S., Malloni, W., Tomé, A.M., Frank, S., Henze, G-I., Wüst, S., Greenlee, M., Lang, E.. A graph neural network framework for causal inference in brain networks. *Scientific Reports*. 11. https://doi.org/10.1038/s41598-021-87411-8 (2021).

In a follow up study, different spatial and temporal GNN architectures were compared in their application of brain connectivity analysis. A preprint of this study is available under:

> Wein, S., A. Schüller, W., Malloni, Tomé, A.M., Greenlee, M., Lang, E.. Modeling Spatio-Temporal Dynamics in Brain Networks: A Comparison of Graph Neural Network Architectures. *Preprint at arXiv::2112.04266*. https://arxiv.org/abs/2112.04266 (2021).

A literature review on current methods in brain connectivity analysis with a focus on machine learning techniques was published in:

> Wein, S., Deco, G., Tomé, A.M., Goldhacker, M., Malloni, W., Greenlee, M., Lang, E.. Brain Connectivity Studies on Structure-Function Relationships: A Short Survey with an Emphasis on Machine Learning. *Computational Intelligence and Neuroscience*. 2021. 1-31. https://doi.org/10.1155/2021/5573740 (2021).

Based on my master thesis, a data-driven approach for identifying functional independent regions in brain networks was published in:

> Wein, S., Tomé, A.M., Goldhacker, M., Greenlee, M., Lang, E.. A Constrained ICA-EMD Model for Group Level fMRI Analysis. *Frontiers in Neuroscience*. 14. https://doi.org/10.3389/fnins.2020.00221 (2020).

Appendix A

Neural Networks

A.1 Backpropagation Algorithm

Based on the description in the book of Bishop [19] the backpropagation algorithm for a feedforward neural network is here outlined in more detail. As defined in equation 1.1 in a feedforward neural network architecture each unit computes a weighted sum of its inputs:

$$\nu_q = \sum_{p=1}^{P} w_{qp} z_p \tag{A.1}$$

where z_p denotes the activation or input for an arbitrary layer, which is connected to a unit q via a weight w_{qp}. In order to not have to deal with the biases b_q explicitly, they can be represented by an additional input unit with constant activation at $+1$ [19]. Then the activation of unit q can be computed by transforming the sum in A.1 with a nonlinear function $\Phi(\cdot)$:

$$y_q = \Phi(\nu_q). \tag{A.2}$$

To evaluate the derivative of our cost function J with respect to a weight w_{qp}, the chain rule for partial derivatives can be applied:

$$\frac{\partial J}{\partial w_{qp}} = \frac{\partial J}{\partial \nu_q} \frac{\partial \nu_q}{\partial w_{qp}} \tag{A.3}$$

Further we can then introduce the definition:

$$\delta_q \equiv \frac{\partial J}{\partial \nu_q} \tag{A.4}$$

In this context the δ's are often referred to as *errors*. By evaluating A.1 we then obtain:

$$\frac{\partial \nu_q}{\partial w_{qp}} = z_p \tag{A.5}$$

And by substituting A.4 and A.5 into A.3 we get the expression:

$$\frac{\partial J}{\partial w_{qp}} = \delta_q z_p \tag{A.6}$$

Equation A.6 illustrates that the derivative can be computed by multiplying δ_q at the output end of the weight with the value at the input z_p. So the derivatives can be obtained by evaluating δ_q for all hidden and output units in the network and then applying equation A.6. If we want to obtain the value of δ_q of a hidden unit q which is connected to the networks output units r in a subsequent layer, we can use the chain rule for partial derivatives:

$$\delta_q \equiv \sum_{r=1}^{R} \frac{\partial J}{\partial \nu_r} \frac{\partial \nu_r}{\partial \nu_q} \tag{A.7}$$

which yields a sum over all units r which are connected to unit q. Finally we can substitute the definition of δ in A.4 into A.7, and use A.1 and A.2 to derive the following expression for the *backpropagation* of the output error:

$$\delta_q = \Phi'(\nu_q) \sum_{r=1}^{R} w_{rq} \delta_r \tag{A.8}$$

which shows that the value of δ can be computed by propagating the values of δ from higher units backwards through the network. So by recursively applying the rule in A.8, every value for δ for all hidden units in a feedforward neural network can be evaluated.

A.2 Backpropagation Through Time Algorithm

To outline the principle of the backpropagation through time (BPTT) algorithm we can consider a basic one-dimensional RNN with some parameters $\boldsymbol{\theta}$, which has the goal to learn a mapping from an input sequence $x^{(1)}, x^{(2)}, \ldots, x^{(T)}$ to a sequence of target values $\hat{y}^{(1)}, \hat{y}^{(2)}, \ldots, \hat{y}^{(T)}$. Then, in order to describe the difference between the outputs of our model $y^{(t)}$ and corresponding targets $\hat{y}^{(t)}$, we can define a cost function for all T target values as:

$$J(x^{(1)}, \ldots, x^{(T)}, \hat{y}^{(1)}, \ldots, \hat{y}^{(T)}, \boldsymbol{\theta}) = \frac{1}{T} \sum_{t=1}^{T} J(x^{(t)}, \hat{y}^{(t)}, \boldsymbol{\theta}) \tag{A.9}$$

Now if we want to compute the derivative of J with respect to for example a certain weight w, we obtain the expression:

$$\frac{\partial J}{\partial w} = \frac{1}{T} \sum_{t=1}^{T} \frac{\partial J}{\partial y^{(t)}} \frac{\partial y^{(t)}}{\partial h^{(t)}} \frac{\partial h^{(t)}}{\partial w} \tag{A.10}$$

In section 1.2.5 we defined the state of a RNN $h^{(t)}$ depending on its previous state $h^{(t)} = \Phi_h(wh^{(t-1)} + ux^{(t)} + b)$. Because the RNN shares its parameters across the whole sequence, $h^{(t-1)}$ also depends on the parameter w again, and we have to backpropagate the derivative with respect to w in time by computing $\frac{\partial h^{(t-1)}}{\partial w}, \frac{\partial h^{(t-2)}}{\partial w}, \ldots$ recursively. This recursive computation of the gradient across the T (time) steps is denoted as backpropagation through time.

Appendix B

Appendix: Spatio-Temporal Graph Neural Networks for Brain Connectivity Analysis

B.1 Influence of Hyperparmeters

In this appendix the influence of the STGNN hyperparameters on their performance is discussed. The hyperparameters of the DCRNN and GWN are chosen as outlined in section 2.1.4 and are held constant, while only the hyperparameter of interest is varied in the following evaluation. Figure B.1 and B.2 show that the performance of he DCRNN and GWN can in general slightly be improved when using a larger number of model parameters. However as the computation time and memory requirements linearly grow with the number of parameters, the STGNN hyperparameters are chosen as described in 2.1.4 to yield a reasonable trade-off between model performance and computational requirements.

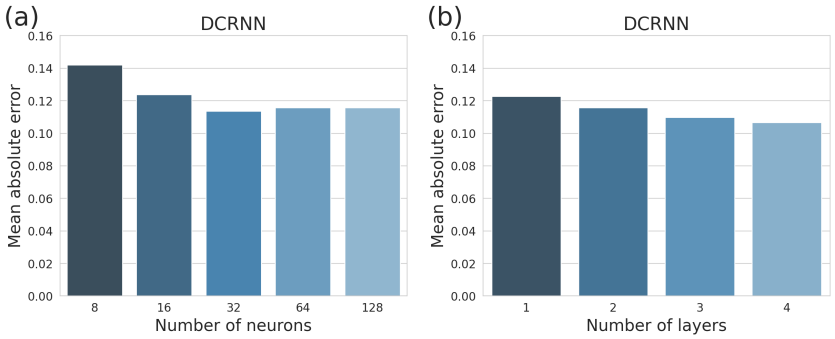

FIGURE B.1: Influence of hyperparameters on the prediction accuracy of the DCRNN model. In (a) the test MAE is shown in dependence of the number of neurons in each layer, and in (b) the error in dependence of the number of DCGRU layers used in the encoder and decoder of the DCRNN.

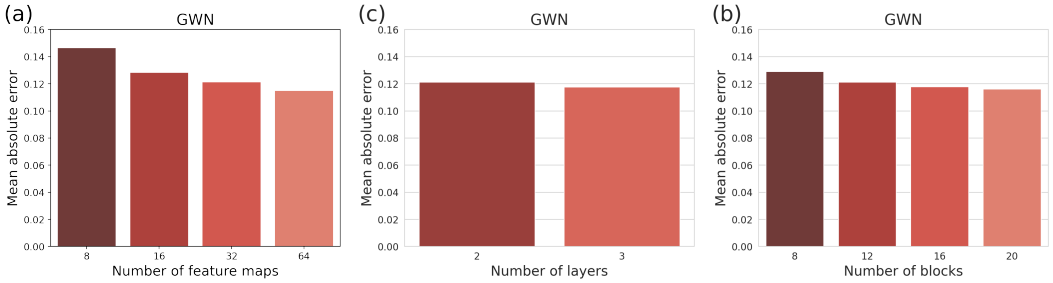

FIGURE B.2: Influence of the GWN hyperparameters on the prediction accuracy. In (a) the test MAE in dependence of the number of feature maps is illustrated. The GWN architecture is organized in blocks of dilated causal convolution (DCC) layers, whereby the dilation factor is doubled in every subsequent layer as $d = 1, 2, 4, \ldots$, and reset to $d = 1$ in the first layer of each block (section 1.2.6). Figure (b) shows the influence of the number of DCC layers used in each block, and (c) shows the impact of the number blocks on the GWN performance.

B.2 List of ROIs in Visual Network

TABLE B.1: In this table a list of ROIs involved in visual processing according to the multi-modal parcellation proposed by Glasser et al. [49]. The table lists the index of the region in the atlas for the right/left hemisphere including the name of the region.

Index	Name
1/181	V1
2/182	MST
3/183	V6
4/184	V2
5/185	V3
6/186	V4
7/187	V8
13/193	V3A
16/196	V7
19/199	V3B
20/200	LO1
21/201	LO2
22/202	PIT
23/203	MT
152/332	V6A
153/333	VMV1
154/334	VMV3
156/336	V4t
158/338	V3CD
159/339	LO3
160/340	VMV2
163/343	VVC

B.3 Accuracy in the 0.02 - 0.09 Hz Frequency Range

In figure B.3 the prediction accuracies of the DCRNN, GWN and VAR model are illustrated for the BOLD signal filtered in the $0.02 - 0.09$ Hz frequency range. By including more frequency compartments of the BOLD signal, the temporal dynamic of the signal is more complex and becomes correspondingly harder to predict for the models. All three models are able to generate accurate predictions for the first 15 TRs (approximately 11s) but for longer forecasing horizons, the DCRNN and GWN remain considerably more accurate in comparison to the VAR.

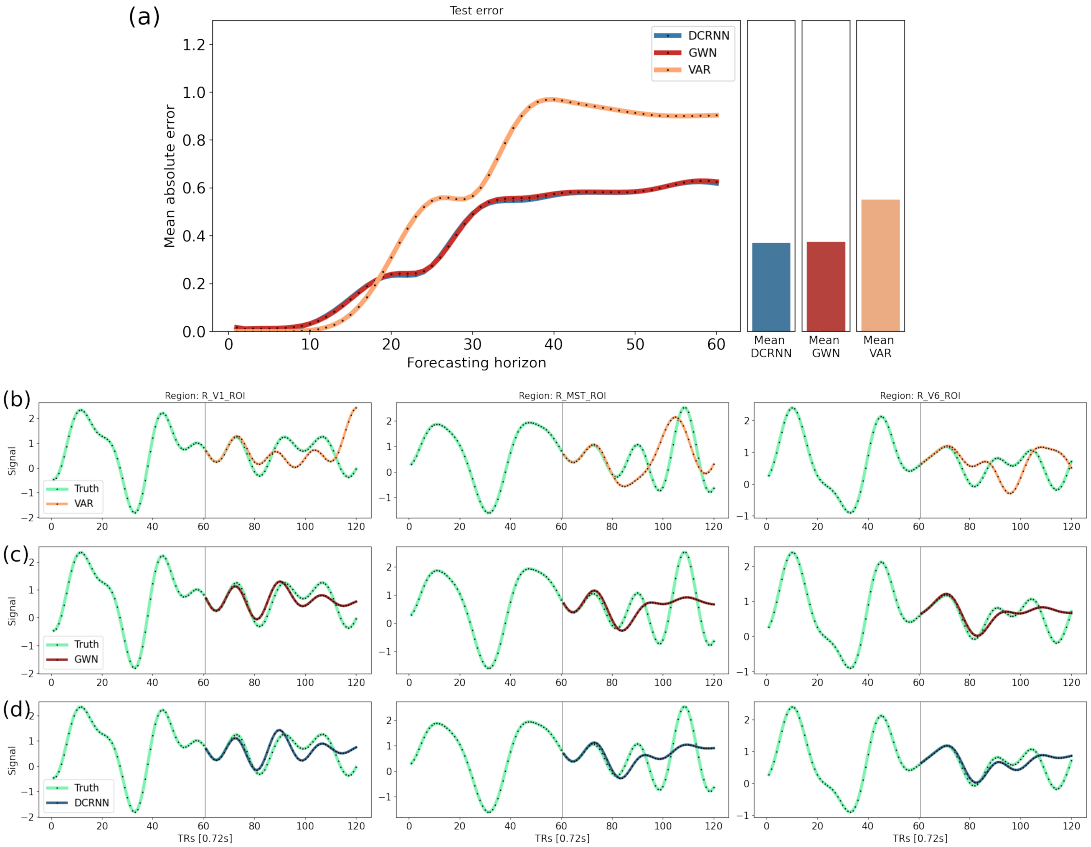

FIGURE B.3: The prediction accuracy of the three models is shown here for BOLD signal filtered in the $0.02 - 0.09$ Hz frequency range. The 25 subject dataset and the brain network including the 180 ROIs within the right hemisphere was incorporated for the comparison [49]. In (a) the test MAE in dependence of the forecasting horizon is shown, computed as an average across test samples and brain regions. Figure (b) depicts a representative example of predictions generated by the VAR model, figure (c) illustrates an example of GWN predictions and figure (d) shows the predictions of the DCRNN. The examples were chosen to be representative for the average prediction accuracy of the models, by selecting a test samples on which the MAE of the three models maximally deviates by ± 0.03 from their average MAE.

B.4 Accuracy UR Dataset

In figure B.4 the prediction accuracies of the DCRNN, GWN and VAR model are illustrated for the UR dataset. The acquisition parameters and data preprocessing of the UR dataset is described in detail in section 2.1.2. Due to the smaller number of datasets, including data from only 10 subjects, and due to the shorter duration of the resting-state fMRI sessions (600 images per session) in comparison to the HCP data (1200 images per session), the number of training samples used to fit the different models is relatively small (3850 samples in total). With this smaller amount of training data, it is very challenging to fit the VAR model to the BOLD signal of all $N = 180$ within one hemisphere [49], and the test MAE is accordingly relatively large, as shown in figure B.4. In contrast, the DCRNN and GWN generate very accurate predictions also on this smaller MRI dataset.

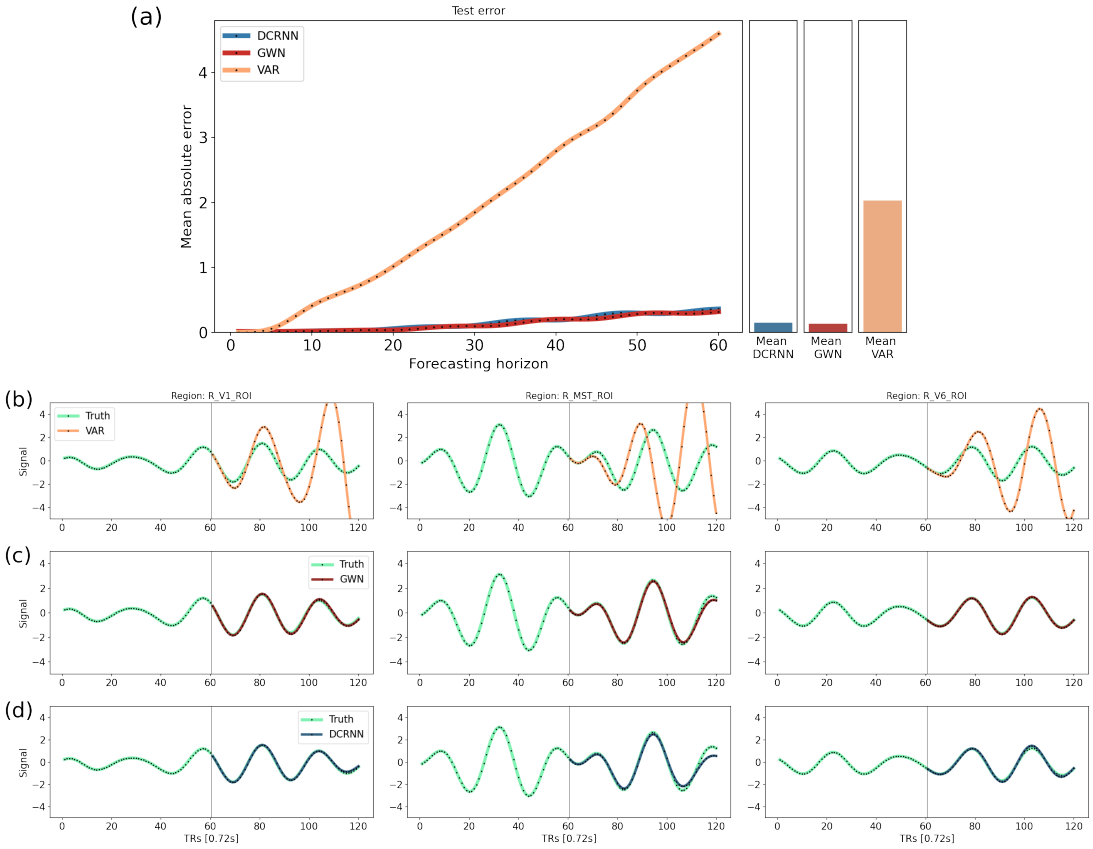

FIGURE B.4: The prediction accuracy of the three models is shown for the UR dataset. The brain network within the right hemisphere including the 180 ROIs was incorporated for this comparison [49]. In (a) the test MAE in dependence of the forecasting horizon is shown, computed as an average across test samples and brain regions. Figure (b) depicts a representative example of predictions generated by the VAR model, figure (c) illustrates an example of GWN predictions and figure (d) shows the predictions of the DCRNN. The examples were chosen to be representative for the average prediction accuracy of the models, by selecting test samples for which the MAE of the three models maximally deviates by ± 0.1 from their average MAE.

B.5 Comparison with DCGRU and DCLSTM

To further emphasize the efficiency of the DCRNN and GWN architecture, these two models will be compared with simpler graph convolution based machine learning models. For this purpose the DCRNN and GWN will be first compared to a simple DCGRU model. The DCGRU is based on a GRU [32], modified by including diffusion convolution operations, as described in section 1.2.4. In this simple DCGRU architecture, no sequence-to-sequence learning is employed as in the DCRNN model. As another baseline, a LSTM model was implemented, as described in section 1.2.5, also including diffusion convolution operations. Accordingly, the gating mechanisms in a LSTM cell are obtained as:

$$\mathbf{f}^{(t)} = \sigma\left(\boldsymbol{\Theta}_f *_G \left[\mathbf{x}^{(t)}, \mathbf{H}^{(t-1)}\right] + \mathbf{b}_f\right) \tag{B.1}$$

$$\mathbf{g}^{(t)} = \sigma\left(\boldsymbol{\Theta}_g *_G \left[\mathbf{x}^{(t)}, \mathbf{H}^{(t-1)}\right] + \mathbf{b}_g\right) \tag{B.2}$$

$$\mathbf{o}^{(t)} = \sigma\left(\boldsymbol{\Theta}_o *_G \left[\mathbf{x}^{(t)}, \mathbf{H}^{(t-1)}\right] + \mathbf{b}_o\right) \tag{B.3}$$

$$\tilde{\mathbf{s}}^{(t)} = \tanh\left(\boldsymbol{\Theta}_{\tilde{s}} *_G \left[\mathbf{x}^{(t)}, \mathbf{H}^{(t-1)}\right] + \mathbf{b}_{\tilde{s}}\right) \tag{B.4}$$

$$\mathbf{s}^{(t)} = \mathbf{f}^{(t)} \odot \mathbf{s}^{(t-1)} + \mathbf{g}^{(t)} \odot \tilde{\mathbf{s}}^{(t)} \tag{B.5}$$

$$\mathbf{H}^{(t)} = \mathbf{o}^{(t)} \odot \tanh(\mathbf{s}^{(t)}) \tag{B.6}$$

This modified LSTM variant will be denoted as *diffusion convolution LSTM* (DCLSTM) in the following. In correspondence to the DCRNN, the DCLSTM and DCGRU were trained using the Adam optimizer [76], and the architectures were composed of 2 layers with a hidden state size of 64. Figure B.5 shows a comparison of the forecasting error of the different models on the 25 subjects dataset including the brain regions within the right hemisphere, as defined by Glasser et al. [49]. The comparison reveals that within the first 5 predicted timesteps, the DCLSTM and DCGRU are also able to make very accurate predictions of the BOLD signal, but for larger forecasting horizons the errors of the DCRNN and GWN remain considerably lower. This highlights the effectiveness of the sequence-to-sequence learning as used in the DCRNN architecture, and the multi-timestep inference implemented in the GWN architecture for making reliable long-term predictions of the BOLD signal.

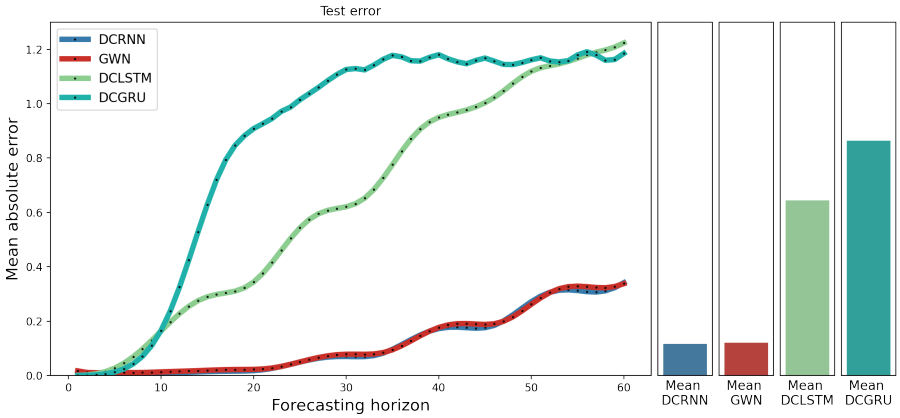

FIGURE B.5: Comparison of the prediction accuracy of the DCRNN and GWN with a DCLSTM and DCGRU architecture. The brain network within the right hemisphere including the 180 ROIs was incorporated for this comparison [49]. The test MAE is shown in dependence on the forecasting horizon, computed as an average across test samples and brain regions.

B.6 Multi-Modal Directed Connectivity DCRNN

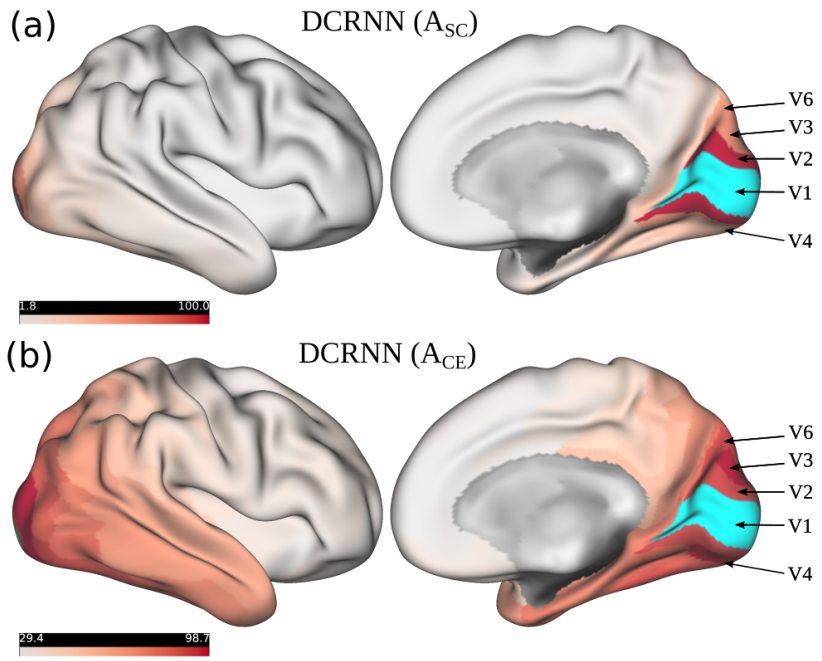

FIGURE B.6: Directed spatial relations learned by the DCRNN model are displayed. Figure (a) shows the measures of directed influence $\mathbf{I}(n')$, derived from the DCRNN model when using the SC for information propagation and figure (b) depicts the influence when incorporating CEs for the information exchange. The values of the connectivity measures were linearly mapped between 0 and 100 and the default scaling of the color values provided by the *connectome workbench* (version 1.4.2) was used, adjusting the colormap between the $2th$ and $98th$ percentile of the values respectively.

Bibliography

[1] Farras Abdelnour et al. "Functional brain connectivity is predictable from anatomic network's Laplacian eigen-structure". In: *NeuroImage* 172 (2018), pp. 728–739. DOI: 10.1016/j.neuroimage.2018.02.016.

[2] Sophie Achard et al. "A Resilient, Low-Frequency, Small-World Human Brain Functional Network with Highly Connected Association Cortical Hubs". In: *The Journal of Neuroscience* 26 (2006), pp. 63 –72. DOI: 10.1523/JNEUROSCI.3874-05.2006.

[3] Jeffrey Alstott et al. "Modeling the Impact of Lesions in the Human Brain". In: *PLOS Computational Biology* 5.6 (June 2009), pp. 1–12. DOI: 10.1371/journal.pcbi.1000408.

[4] Enrico Amico and Joaquín Goñi. "Mapping hybrid functional-structural connectivity traits in the human connectome". In: *Network Neuroscience* 2.3 (Sept. 2018), pp. 306–322. ISSN: 2472-1751. DOI: 10.1162/netn_a_00049.

[5] Jesper Andersson, Stefan Skare, and John Ashburner. "How to correct susceptibility distortions in spin-echo echo-planar images: Application to diffusion tensor imaging". In: *NeuroImage* 20 (2003), pp. 870–88. DOI: 10.1016/S1053-8119(03)00336-7.

[6] Jesper Andersson and Stamatios Sotiropoulos. "An integrated approach to correction for off-resonance effects and subject movement in diffusion MR imaging". In: *NeuroImage* 125 (2015), pp. 1063–1078. DOI: 10.1016/j.neuroimage.2015.10.019.

[7] Jesper Andersson and Stamatios Sotiropoulos. "Non-parametric representation and prediction of single- and multi-shell diffusion-weighted MRI data using Gaussian processes". In: *NeuroImage* 122 (2015), pp. 166–76. DOI: 10.1016/j.neuroimage.2015.07.067.

[8] Salim Arslan et al. "Graph Saliency Maps through Spectral Convolutional Networks: Application to Sex Classification with Brain Connectivity". In: *GRAIL/Beyond-MIC@MICCAI*. 2018.

[9] Lionel Barnett and Anil Seth. "The MVGC Multivariate Granger Causality Toolbox: A New Approach to Granger-causal Inference." In: *Journal of neuroscience methods* 223 (2013), pp. 50–68. DOI: 10.1016/j.jneumeth.2013.10.018.

[10] George Bartzokis et al. “White matter structural integrity in healthy aging adults and patients with Alzheimer disease: A magnetic resonance imaging study”. In: *Archives of Neurology* 60 (2003), pp. 393–8. DOI: 10.1001/archneur.60.3.393.

[11] Peter J. Basser, James Mattiello, and D. LeBihan. “MR diffusion tensor spectroscopy and imaging.” In: *Biophysical journal* 66 1 (1994), pp. 259–67.

[12] Cassiano Becker et al. “Spectral mapping of brain functional connectivity from diffusion imaging”. In: *Scientific Reports* 8 (Dec. 2018). DOI: 10.1038/s41598-017-18769-x.

[13] Christian Beckmann et al. “Investigations into resting-state connectivity using Independent Component Analysis”. In: *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 360 (June 2005), pp. 1001–13. DOI: 10.1098/rstb.2005.1634.

[14] Timothy Edward John Behrens et al. “Probabilistic diffusion tractography with multiple fibre orientations: What can we gain?” In: *NeuroImage* 34 (2007), pp. 144–55. DOI: 10.1016/j.neuroimage.2006.09.018.

[15] Samy Bengio et al. “Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks”. In: NIPS’15. Montreal, Canada: MIT Press, 2015, 1171–1179.

[16] Ruggero G. Bettinardi et al. “How structure sculpts function: Unveiling the contribution of anatomical connectivity to the brain’s spontaneous correlation structure”. In: *Chaos* 27 (Apr. 2017), p. 047409. DOI: 10.17863/CAM.11160.

[17] Natalia Bielczyk et al. “Disentangling causal webs in the brain using functional Magnetic Resonance Imaging: A review of current approaches”. In: *Network Neuroscience* 3 (Feb. 2019). DOI: 10.1162/netn_a_00062.

[18] Rasmus Birn et al. “Separating respiration-variation-related fluctuations from neural-activity-related fluctuations in fMRI”. In: *NeuroImage* 31 (July 2006), pp. 1536–48. DOI: 10.1016/j.neuroimage.2006.02.048.

[19] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738.

[20] Bharat Biswal, Edgar A. Deyoe, and James S. Hyde. “Reduction of physiological fluctuations in fMRI using digital filters”. In: *Magnetic Resonance in Medicine* 35.1 (1996), pp. 107–113. DOI: https://doi.org/10.1002/mrm.1910350114.

[21] Bharat B. Biswal et al. “Functional connectivity in the motor cortex of resting human brain using echo-planar MRI.” In: *Magnetic resonance in medicine* 34 4 (1995), pp. 537–41.

[22] Michael M. Bronstein et al. "Geometric Deep Learning: Going beyond Euclidean data". In: *IEEE Signal Processing Magazine* 34 (2017), pp. 18–42.

[23] Joan Bruna et al. "Spectral networks and locally connected networks on graphs". In: *International Conference on Learning Representations (ICLR2014)*. CBLS, 2014.

[24] Randy Buckner et al. "Cortical Hubs Revealed by Intrinsic Functional Connectivity: Mapping, Assessment of Stability, and Relation to Alzheimer's Disease". In: *The Journal of neuroscience: the official journal of the Society for Neuroscience* 29 (2009), pp. 1860–73.

[25] Gregory Burgess et al. "Evaluation of Denoising Strategies To Address Motion-Correlated Artifact in Resting State fMRI Data from the Human Connectome Project". In: *Brain Connectivity* 6 (2016).

[26] Keith Bush et al. "Improving the Precision of fMRI BOLD Signal Deconvolution with Implications for Connectivity Analysis". In: *Magnetic resonance imaging* 33 (July 2015). DOI: 10.1016/j.mri.2015.07.007.

[27] Richard Buxton. "The physics of functional magnetic resonance imaging (fMRI)". In: *Reports on progress in physics. Physical Society (Great Britain)* 76 (Sept. 2013), p. 096601. DOI: 10.1088/0034-4885/76/9/096601.

[28] Richard B. Buxton and Lawrence R. Frank. "A Model for the Coupling between Cerebral Blood Flow and Oxygen Metabolism during Neural Stimulation". In: *Journal of Cerebral Blood Flow & Metabolism* 17.1 (1997), pp. 64–72. DOI: 10.1097/00004647-199701000-00009.

[29] Joana Cabral, Morten L Kringelbach, and Gustavo Deco. "Functional Connectivity dynamically evolves on multiple time-scales over a static Structural Connectome: Models and Mechanisms". In: *NeuroImage* 160 (Mar. 2017), p. 10. DOI: https://doi.org/10.1016/j.neuroimage.2017.03.045.

[30] Vince Calhoun et al. "A Method for Making Group Inferences Using Independent Component Analysis of Functional MRI Data: Exploring the Visual System". In: *Neuroimage* 13 (June 2001), pp. 88–88. DOI: 10.1016/S1053-8119(01)91431-4.

[31] Geetha Soujanya Chilla et al. "Diffusion weighted magnetic resonance imaging and its recent trend-a survey". In: *Quantitative imaging in medicine and surgery* 5 (June 2015), pp. 407–22. DOI: 10.3978/j.issn.2223-4292.2015.03.01.

[32] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: (June 2014). DOI: 10.3115/v1/D14-1179.

[33] Fan R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[34] Junyoung Chung et al. *Empirical evaluation of gated recurrent neural networks on sequence modeling*. English (US). 2014.

[35] Radoslaw M. Cichy and Daniel Kaiser. "Deep Neural Networks as Scientific Models". In: *Trends in Cognitive Sciences* 23.4 (2019), pp. 305–317. ISSN: 1364-6613. DOI: https://doi.org/10.1016/j.tics.2019.01.009.

[36] Jessica S. Damoiseaux et al. "Chapter 21 - Applications of MRI connectomics". In: *Advanced Neuro MR Techniques and Applications*. Ed. by In-Young Choi and Peter Jezzard. Vol. 4. Advances in Magnetic Resonance Technology and Applications. Academic Press, 2021, pp. 323–338. DOI: https://doi.org/10.1016/B978-0-12-822479-3.00034-8.

[37] Jean Daunizeau, Olivier David, and K.E. Stephan. "Dynamic causal modelling: A critical review of the biophysical and statistical foundations". In: *NeuroImage* 58 (2009), pp. 312–22. DOI: 10.1016/j.neuroimage.2009.11.062.

[38] Yann N Dauphin et al. "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014.

[39] Gustavo Deco, Mario Senden, and Viktor Jirsa. "How anatomy shapes dynamics: a semi-analytical study of the brain at rest by a simple spin model". In: *Frontiers in computational neuroscience* 6 (2012), p. 68. DOI: 10.3389/fncom.2012.00068.

[40] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering". In: *NIPS*. 2016, pp. 3837–3845.

[41] Fani Deligianni et al. "NODDI and Tensor-Based Microstructural Indices as Predictors of Functional Connectivity". In: *PloS one* 11 (Apr. 2016), e0153404.

[42] Andrea Duggento et al. "Multivariate Granger causality unveils directed parietal to prefrontal cortex connectivity during task-free MRI". In: *Scientific Reports* 8 (Apr. 2018).

[43] Anthony Faivre et al. "Depletion of brain functional connectivity enhancement leads to disability progression in multiple sclerosis: A longitudinal resting-state fMRI study". In: *Multiple Sclerosis Journal* 22 (Feb. 2016). DOI: 10.1177/1352458516628657.

[44] David Feinberg et al. "Multiplexed Echo Planar Imaging for Sub-Second Whole Brain FMRI and Fast Diffusion Imaging". In: *PloS one* 5 (2010), e15710.

[45] Bruce Fischl. "FreeSurfer". In: *NeuroImage* 62.2 (2012), pp. 774 –781. DOI: 10.1016/j.neuroimage.2012.01.021.

[46] Karl Friston, L. Harrison, and W. Penny. "Dynamic causal modelling". In: *NeuroImage* 19.4 (2003), pp. 1273–1302. ISSN: 1053-8119. DOI: https://doi.org/10.1016/S1053-8119(03)00202-7.

[47] Karl Friston, Rosalyn Moran, and Anil Seth. "Analysing connectivity with Granger causality and dynamic causal modelling." In: *Current opinion in neurobiology* 23 (2013), pp. 172–178.

[48] Kunihiko Fukushima. "A neural network model for the mechanism of selective attention in visual pattern recognition". In: *Systems and Computers in Japan* 18 (1987), pp. 102 –113. DOI: 10.1002/scj.4690180110.

[49] Matthew Glasser et al. "A multi-modal parcellation of human cerebral cortex". In: *Nature* 536 (2016).

[50] Matthew Glasser et al. "The Minimal Preprocessing Pipelines for the Human Connectome Project". In: *NeuroImage* 80 (2013).

[51] Enrico Glerean et al. "Functional Magnetic Resonance Imaging Phase Synchronization as a Measure of Dynamic Functional Connectivity". In: *Brain connectivity* 2 (2012), pp. 91–101.

[52] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Journal of Machine Learning Research - Proceedings Track* 9 (2010), pp. 249–256.

[53] Gary Glover. "Overview of Functional Magnetic Resonance Imaging". In: *Neurosurgery clinics of North America* 22 (Apr. 2011), pp. 133–9. DOI: 10.1016/j.nec.2010.11.001.

[54] Markus Goldhacker. "Frequency-resolved dynamic functional connectivity and scale stability of connectivity-states". PhD thesis. University of Regensburg, 2017.

[55] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[56] Clive William John Granger. "Investigating causal relations by econometric models and cross-spectral methods". In: *Econometrica* 37 (1969), pp. 424–438.

[57] Klaus Greff et al. "LSTM: A Search Space Odyssey". In: *IEEE Transactions on Neural Networks and Learning Systems* 28.10 (2017), pp. 2222–2232. DOI: 10.1109/TNNLS.2016.2582924.

[58] Ludovica Griffanti et al. "ICA-based artefact removal and accelerated fMRI acquisition for improved resting state network imaging". In: *NeuroImage* 95 (2014), pp. 232 –247. DOI: 10.1016/j.neuroimage.2014.03.034.

[59] Aditya Grover and Jure Leskovec. "node2vec: Scalable Feature Learning for Networks". In: vol. 2016. July 2016, pp. 855–864. DOI: 10.1145/2939672.2939754.

[60] Pim de Haan, Taco S Cohen, and Max Welling. "Natural Graph Networks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 3636–3646.

[61] Patric Hagmann et al. "Understanding diffusion MR imaging techniques: from scalar diffusion-weighted imaging to diffusion tensor imaging and beyond." In: *Radiographics: a review publication of the Radiological Society of North America, Inc* 26 (2006), pp. 205–23.

[62] Patric Hagmann et al. "White matter maturation reshapes structural connectivity in the late developing human brain". In: *Proceedings of the National Academy of Sciences of the United States of America* 107 (Oct. 2010), pp. 19067–72. DOI: 10.1073/pnas.1009073107.

[63] James D. Hamilton. *Time Series Analysis*. Princeton, NJ.: Princeton University Press, 1994.

[64] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.

[65] Kaiming He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 1026–1034.

[66] Alexander Hendrix. *Magnets, Spins, and Resonances : An Introduction to the Basics of Magnetic Resonance*. Siemens Medical Solutions, 2003.

[67] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[68] Michael Hodge et al. "ConnectomeDB – Sharing Human Brain Connectivity Data". In: *NeuroImage* 124 (2015). DOI: 10.1016/j.neuroimage.2015.04.046.

[69] Christopher J. Honey et al. "Predicting human resting-state functional connectivity from structural connectivity". In: *Proceedings of the National Academy of Sciences of the United States of America* 106 6 (2009), pp. 2035–40.

[70] Andreas Horn et al. "Connectivity Predicts deep brain stimulation outcome in Parkinson disease". In: *Annals of Neurology* 82 (June 2017). DOI: 10.1002/ana.24974.

[71] Joseph P. Hornak. *The Basics of MRI*. URL: https://www.cis.rit.edu/htbooks/mri/index.html (visited on 08/02/2021).

[72] Mark Jenkinson et al. "FSL". In: *NeuroImage* 62.2 (2012), pp. 782 –790. DOI: 10.1016/j.neuroimage.2011.09.015.

[73] Mark Jenkinson et al. "Improved Optimization for the Robust and Accurate Linear Registration and Motion Correction of Brain Images". In: *NeuroImage* 17 (2002), pp. 825 –841. DOI: 10.1006/nimg.2002.1132.

[74] Ben Jeurissen et al. "Multi-tissue constrained spherical deconvolution for improved analysis of multi-shell diffusion MRI data". In: *NeuroImage* 103 (2014), pp. 411–426.

[75] Byung-Hoon Kim and Jong Chul Ye. "Understanding Graph Isomorphism Network for rs-fMRI Functional Connectivity Analysis". In: *Frontiers in Neuroscience* 14 (2020), p. 630. DOI: 10.3389/fnins.2020.00630.

[76] Diederik Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: 2014.

[77] Sofia Ira Ktena et al. "Metric learning with spectral graph convolutions on brain connectivity networks". In: *NeuroImage* 169 (2018), pp. 431–442.

[78] Elmar Lang et al. "Brain Connectivity Analysis: A Short Survey". In: *Computational intelligence and neuroscience* (2012). DOI: 10.1155/2012/412512.

[79] Sebastian Lapuschkin et al. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". In: *PLoS ONE* 10 (July 2015), e0130140. DOI: 10.1371/journal.pone.0130140.

[80] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521 (May 2015), pp. 436–44. DOI: 10.1038/nature14539.

[81] Xiaoxiao Li et al. "Graph Neural Network for Interpreting Task-fMRI Biomarkers". In: *Medical Image Computing and Computer Assisted Intervention - MICCAI 2019*. 2019, pp. 485–493. DOI: 10.1007/978-3-030-32254-0_54.

[82] Yaguang Li et al. "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting". In: *International Conference on Learning Representations*. 2018.

[83] Hualou Liang and Hongbin Wang. "Structure-Function Network Mapping and Its Assessment via Persistent Homology". In: *PLOS Computational Biology* 13 (Jan. 2017). DOI: 10.1371/journal.pcbi.1005325.

[84] Yaou Liu et al. "Functional Brain Network Alterations in Clinically Isolated Syndrome and Multiple Sclerosis: A Graph-based Connectome Study". In: *Radiology* 282 (Aug. 2016), p. 152843. DOI: 10.1148/radiol.2016152843.

[85] Helmut Luetkepohl. *The New Introduction to Multiple Time Series Analysis*. Springer, 2005. DOI: 10.1007/978-3-540-27752-1.

[86] James Mackinnon. "Approximate Asymptotic Distribution Functions for Unit-Root and Cointegration Tests". In: *Journal of Business and Economic Statistics* 12 (1994), pp. 167–76. DOI: 10.1080/07350015.1994.10510005.

[87] Warren S. McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *Bulletin of Mathematical Biophysics* 5 (1943), 115–133. DOI: https://doi.org/10.1007/BF02478259.

[88] A. Messé et al. "A closer look at the apparent correlation of structural and functional connectivity in excitable neural networks". In: *Scientific Reports* 5 (2015), p. 7870.

[89] Tomas Mikolov et al. "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems* 26 (Oct. 2013).

[90] Ravi D. Mill et al. "Empirical validation of directed functional connectivity". In: *NeuroImage* 146 (2017), pp. 275–287. DOI: 10.1016/j.neuroimage.2016.11.037.

[91] Steen Moeller et al. "Multiband multislice GE-EPI at 7 tesla, with 16-fold acceleration using partial parallel imaging with application to high spatial and temporal whole-brain fMRI." In: *Magnetic resonance in medicine* 63 5 (2010), pp. 1144–53.

[92] Seiji Ogawa and Yul-Wan Sung. "Functional magnetic resonance imaging". In: *Scholarpedia* 2.10 (2007). revision #151126, p. 3105. DOI: 10.4249/scholarpedia.3105.

[93] Cheryl A. Olman, Lila Davachi, and Souheil J. Inati. "Distortion and Signal Loss in Medial Temporal Lobe". In: *PLoS ONE* 4 (2009).

[94] Sinno Pan and Qiang Yang. "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22 (2010), pp. 1345 – 1359. DOI: 10.1109/TKDE.2009.191.

[95] Andrew Reid et al. "Advancing functional connectivity research from association to causation". In: *Nature Neuroscience* 22 (Oct. 2019), pp. 1–10. DOI: 10.1038/s41593-019-0510-4.

[96] Alard Roebroeck, Elia Formisano, and Rainer Goebel. "The identification of interacting networks in the brain using fMRI: Model selection, causality and deconvolution". In: *NeuroImage* 58 (Sept. 2009), pp. 296–302. DOI: 10.1016/j.neuroimage.2009.09.036.

[97] Xin Rong. *word2vec Parameter Learning Explained*. 2016. arXiv: 1411.2738 [cs.CL].

[98] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65 6 (1958), pp. 386–408.

[99] Gideon Rosenthal et al. "Mapping higher-order relations between brain structure and function with embedded vector representations of connectomes". In: *Nature Communications* 9 (Dec. 2018).

[100] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323 (1986), pp. 533–536.

[101] Gholamreza Salimi-Khorshidi et al. "Automatic denoising of functional MRI data: Combining independent component analysis and hierarchical fusion of classifiers". In: *NeuroImage* 90 (2014), pp. 449 –468. DOI: 10.1016/j.neuroimage.2013.11.046.

[102] Tom Schaul, Ioannis Antonoglou, and David Silver. "Unit Tests for Stochastic Optimization". In: *CoRR* abs/1312.6055 (2014).

[103] Anil Seth, Paul Chorley, and Lionel Barnett. "Granger causality analysis of fMRI BOLD signals is invariant to hemodynamic convolution but not downsampling". In: *NeuroImage* 65 (2012), pp. 540–55. DOI: 10.1016/j.neuroimage.2012.09.049.

[104] Kawin Setsompop et al. "Blipped-controlled aliasing in parallel imaging for simultaneous multislice echo planar imaging with reduced g-factor penalty." In: *Magnetic resonance in medicine* 67 5 (2012), pp. 1210–24.

[105] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*. 2014.

[106] Anne M. Smith et al. "Investigation of Low Frequency Drift in fMRI Signal". In: *NeuroImage* 9 (1999), pp. 526–533.

[107] Robert Smith et al. "Anatomically-constrained tractography: Improved diffusion MRI streamlines tractography through effective use of anatomical information". In: *NeuroImage* 62 (2012), pp. 1924–38. DOI: 10.1016/j.neuroimage.2012.06.005.

[108] Robert Smith et al. "SIFT: Spherical-deconvolution Informed Filtering of Tractograms." In: *NeuroImage* 67 (2013), pp. 298–312. DOI: 10.1016/j.neuroimage.2012.11.049.

[109] Stephen M. Smith et al. "Network modelling methods for FMRI". In: *NeuroImage* 54 (2011), pp. 875–891. DOI: 10.1016/j.neuroimage.2010.08.063.

[110] Stephen M. Smith et al. "Resting-state fMRI in the Human Connectome Project". In: *NeuroImage* 80 (2013), pp. 144 –168. DOI: 10.1016/j.neuroimage.2013.05.039.

[111] Stamatios Sotiropoulos et al. "Advances in diffusion MRI acquisition and processing in the Human Connectome Project". In: *NeuroImage* 80 (2013), p. 125. DOI: 10.1016/j.neuroimage.2013.05.057.

[112] Stamatios Sotiropoulos et al. "Effects of Image Reconstruction on Fibre Orientation Mapping from Multi-channel Diffusion MRI: Reducing the Noise Floor Using SENSE". In: *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine* 70 (2013). DOI: 10.1002/mrm.24623.

[113] Olaf Sporns. "Brain connectivity". In: *Scholarpedia* 2.10 (2007). revision #91084, p. 4695. DOI: 10.4249/scholarpedia.4695.

[114] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. "Sequence to Sequence Learning with Neural Networks". In: *NIPS*. 2014.

[115] Cibu Thomas et al. "Anatomical accuracy of brain connections derived from diffusion MRI tractography is inherently limited." In: *Proceedings of the National Academy of Sciences of the United States of America* 111 46 (2014), pp. 16574–9.

[116] Erico Tjoa and Cuntai Guan. "A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI". In: *IEEE Transactions on Neural Networks and Learning Systems* 32 (2021), pp. 4793–4813.

[117] Jacques-Donald Tournier, Fernando Calamante, and Alan Connelly. "Robust determination of the fibre orientation distribution in diffusion MRI: Non-negativity constrained super-resolved spherical deconvolution". In: *NeuroImage* 35 (2007), pp. 1459–72. DOI: 10.1016/j.neuroimage.2007.02.016.

[118] Jacques-Donald Tournier et al. "Direct estimation of the fiber orientation density function from diffusion-weighted MRI data using spherical deconvolution". In: *NeuroImage* 23 (2004), pp. 1176–85. DOI: 10.1016/j.neuroimage.2004.07.037.

[119] Jacques-Donald Tournier et al. "MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation". In: *NeuroImage* 202 (2019). DOI: 10.1101/551739.

[120] Kamil Uğurbil et al. "Pushing spatial and temporal resolution for functional and diffusion MRI in the Human Connectome Project". In: *NeuroImage* 80 (2013). DOI: https://doi.org/10.1016/j.neuroimage.2013.05.012.

[121] Aäron van den Oord et al. "WaveNet: A Generative Model for Raw Audio". In: *Proc. 9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*. 2016, p. 125.

[122] David Van Essen et al. "The WU-Minn Human Connectome Project: an overview". In: *NeuroImage* 80 (2013). DOI: https://doi.org/10.1016/j.neuroimage.2013.05.041.

[123] Huifang E. Wang et al. "A systematic framework for functional connectivity measures". In: *Frontiers in Neuroscience* 8 (2014). DOI: 10.3389/fnins.2014.00405.

[124] J. Taylor Webb et al. "BOLD Granger Causality Reflects Vascular Anatomy". In: *PLOS ONE* 8.12 (Dec. 2013). DOI: 10.1371/journal.pone.0084279.

[125] Simon Wein et al. "A Constrained ICA-EMD Model for Group Level fMRI Analysis". In: *Frontiers in Neuroscience* 14 (2020). DOI: 10.3389/fnins.2020.00221.

[126] Simon Wein et al. "A Graph Neural Network Framework for Causal Inference in Brain Networks". In: *Scientific Reports* 11 (Apr. 2021). DOI: 10.1038/s41598-021-87411-8.

[127] Simon Wein et al. "Brain Connectivity Studies on Structure-Function Relationships: A Short Survey with an Emphasis on Machine Learning". In: *Computational Intelligence and Neuroscience* 2021 (May 2021), pp. 1–31. DOI: 10.1155/2021/5573740.

[128] Simon Wein et al. *Modeling Spatio-Temporal Dynamics in Brain Networks: A Comparison of Graph Neural Network Architectures*. 2021. arXiv: 2112.04266 [q-bio.NC].

[129] Xiaotong Wen, Govindan Rangarajan, and Mingzhou Ding. "Is Granger Causality a Viable Technique for Analyzing fMRI Data?" In: *PloS one* 8 (July 2013), e67428. DOI: 10.1371/journal.pone.0067428.

[130] Paul Werbos. "Backpropagation through time: what it does and how to do it". In: *Proceedings of the IEEE* 78 (Nov. 1990), pp. 1550 –1560. DOI: 10.1109/5.58337.

[131] D. Wilson and Tony Martinez. "The general inefficiency of batch training for gradient descent learning". In: *Neural networks : the official journal of the International Neural Network Society* 16 (Jan. 2004), pp. 1429–51. DOI: 10.1016/S0893-6080(03)00138-2.

[132] Richard G. Wise et al. "Resting fluctuations in arterial carbon dioxide induce significant low frequency variations in BOLD signal". In: *NeuroImage* 21 (2004), pp. 1652–1664.

[133] Zonghan Wu et al. "A Comprehensive Survey on Graph Neural Networks". In: *IEEE Transactions on Neural Networks and Learning Systems* (2020), pp. 1–21.

[134] Zonghan Wu et al. "Graph Wavenet for Deep Spatial-Temporal Graph Modeling". In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. IJCAI'19. AAAI Press, 2019, pp. 1907–1913.

[135] Junqian Xu et al. "Highly Accelerated Whole Brain Imaging Using Aligned-Blipped-Controlled-Aliasing Multiband EPI". In: *Proceedings of the 20th Annual Meeting of ISMRM* (2012), p. 2036.

[136] Matthew Zeiler and Rob Fergus. "Visualizing and Understanding Convolutional Neural Networks". In: *ECCV 2014, Part I, LNCS 8689* 8689 (2013). DOI: 10.1007/978-3-319-10590-1_53.

[137] Hui Zhang et al. "NODDI: Practical in vivo neurite orientation dispersion and density imaging of the human brain". In: *NeuroImage* 61 (2012), pp. 1000–16. DOI: https://doi.org/10.1016/j.neuroimage.2012.03.072.

[138] Joelle Zimmermann et al. "Subject-Specificity of the Correlation Between Large-Scale Structural and Functional Connectivity". In: *Network Neuroscience* (2019), pp. 1–35. DOI: 10.1162/NETN_a_00055.

[139] Qihong Zou et al. "An improved approach to detection of amplitude of low-frequency fluctuation (ALFF) for resting-state fMRI: Fractional ALFF". In: *Journal of neuroscience methods* 172 (Aug. 2008), pp. 137–41. DOI: 10.1016/j.jneumeth.2008.04.012.

[140] Radim Řehůřek and Petr Sojka. "Software Framework for Topic Modelling with Large Corpora". In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. May 2010, pp. 45–50. DOI: 10.13140/2.1.2393.1847.

Acknowledgements

I would like to sincerely thank the following people, which provided their support during the period of my thesis:

Prof. Dr. Elmar W. Lang for his dedicated supervision, constant helpfulness and long-term guidance during the last few years. I am very grateful I had the possibility to pursue my thesis with him on this topic and I enjoyed the many hours we spend together discussing ideas, carrying out research and writing papers together. Also I appreciated very much his invitations to stay and work with him at the University of Aveiro, which was a valuable support for our projects and in addition a personally nice experience to see Portugal.

Prof. Dr. Mark W. Greenlee for making this interdisciplinary cooperation possible and providing me the exciting opportunity to join his research group in experimental neuroscience. I would like to thank him for proposing the topic of my thesis on multi-modal brain connectivity, and for constantly guiding me towards the current interesting topics in neuroscience research. The warm welcome in his group and his caring and patient supervision have contributed to a very pleasant and productive working atmosphere during the period of my thesis.

Prof. Dr. Ana Maria Tomé for supporting my work with her expertise on signal processing and with her helpful feedbacks on our research projects. My stays at the University of Aveiro have been an enrichment for our projects and her great hospitality contributed to a memorable time during my stays in Portugal.

Dr. Wilhelm M. Malloni for the always friendly and helpful support on many technical questions on MRI data processing. Our numerous discussions and his feedback were a valuable help for our common projects.

As well I would like to thank my former masterstudent Alina Schüller for her reliable and helpful work on the GWN model optimization, which was valuable contribution to our research project. I would like to thank Christian, Florian and Marinus for the numerous entertaining coffee breaks. Finally I want to thank all other people in the group of Prof. Greenlee and Prof. Lang, which have contributed to a stimulating and inspiring research time, and also my friends and family for their support during the last years.

Declaration

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe des Literaturzitats gekennzeichnet. Weitere Personen waren an der inhaltlich-materiellen Herstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe eines Promotionsberaters oder anderer Personen in Anspruch genommen. Niemand hat von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Die Arbeit wurde bisher weder im Innoch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Regensburg, 14.03.2022

Simon Wein