UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

EEG to fMRI Synthesis

David António Cóias Calhas

Supervisor: Doctor Rui Miguel Carrasqueiro Henriques

Thesis approved in public session to obtain the PhD degree in

Information Systems and Computer Engineering

Jury final classification: **Pass with Distinction**

2023

TÉCNICO LISBOA

UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

EEG to fMRI Synthesis

David António Cóias Calhas

Supervisor: Doctor Rui Miguel Carrasqueiro Henriques

Thesis approved in public session to obtain the PhD degree in

Information Systems and Computer Engineering

Jury final classification: **Pass with Distinction**

Jury

Chairperson: Doctor Ana Teresa Correia de Freitas, Instituto Superior Técnico, Universidade de Lisboa;
Members Committee:
Doctor César Alexandre Domingues Teixeira, Faculdade de Ciências e Tecnologia, Universidade de Coimbra;
Doctor Sérgio Rafael Mano Pereira, Oncology Group, Lunit, South Korea;
Doctor Helena Isabel Aidos Lopes Tomás, Faculdade de Ciências, Universidade de Lisboa;
Doctor Rui Miguel Carrasqueiro Henriques, Instituto Superior Técnico, Universidade de Lisboa.

Funding Institution
FCT - Fundação para a Ciência e Tecnologia

2023

To my mom

Acknowledgments

I want to thank Fundação para a Ciência e a Tecnologia for providing funding of my PhD, through the scholarship with reference SFRH/BD/05762/2020.

I want to mention my parents. My Father for the great education and values that I still hold dear and will for the future. My Mother for being my best friend and the biggest pillar of my life. I am lucky for being their son. I also want to thank my sister, Sara, and Guido for not only being always there, but also for the support given when I was going through tough times.

My supervisor Rui Henriques who was crucial during my Master degree to stay and follow this path. I also want to thank him not only for the educational help but also life in general.

Pedro Orvalho for overall always being there. I can always count on him, we helped each other and it is amazing that we continued through the same path.

João Silveira the same as Pedro. We did not follow the same path, but in addition to our connection, it is amazing that life had my mother and your father as friends and 40 years later we meet each other.

Laura Palma, you are a big influence on how I approach things in life and most importantly to follow what I want to do.

Francisca Ribeiro, you also had a big influence on me. Thank you for introducing Avatar to me.

Daniel Gonçalves for being an amazing person and always being there when shit happens.

My friends who are always there for any of us for the good and the bad.

The people that were at INESC-ID: Pedro Orvalho, Rodrigo Graça, Margarida Ferreira, João Cortes, Martim Zannati, João Rico, Daniel Gonçalves and Leonardo Alexandre.

I would like to thank Manuela Sado and Vanda Fidalgo who were a huge help throughout my stay at INESC-ID.

Abstract

Electrophysiological activity is related to haemodynamical activity, resting on the theory that the blood flows to where it is needed. As neurons fire, producing action potentials, blood irrigation is required to provide necessary molecules and remove the released toxins. Recent computational advances withheld information at high capacities of computational processing and storage, allowing a comprehensive analysis of neuroimaging records. Complementarily to Electroencephalography (EEG), the emergence of functional Magnetic Resonance Imaging (fMRI) has paved neuroscience discoveries related with haemodynamical brain activity. These modalities provide different stances on brain functions and show structurally different spatiotemporal resolutions.

This work maps EEG to fMRI using neural processing principles. With such a mapping, different applications can be explored: assist ambulatory diagnostics of brain diseases; support longitudinal studies via proxy of the EEG; and increase the discriminative and interpretative power of decision support systems. In this work, we propose a automatic generation of neural architectures to give new insights to the structure of existing models and removing the human bias from the problem. To this end, simultaneous EEG and fMRI records at resting and task-based states were processed. We then converge on a neural architecture that is able to map EEG to fMRI, surpassing the state-of-the-art in this task by 13% in RMSE. Our results suggest that EEG electrode relationships (such as between Oz and PO9) are pivotal to retain information necessary for haemodynamical activity retrieval. Following, we addressed the need for uncertainty quantification in our synthesis task. The proposed solution consisted on a spectral manipulation at the output of the network, which lowers the spatial resolution of the synthesis and benefits the original versions ($+0.21$pp in RMSE).In the end, we show that our proposed neural network is innate at extrapolating to a classification setting, due to its shift invariant properties and spectral inspired mechanics. Using schizophrenia as a proof of concept case, we were able to prove that our synthesized view is discriminative (0.77 area under the receiver operating characteristic curve) and has a good interpretability power over this neuropsychiatric disorder.

Keywords: neuroimaging, machine learning, electroencephalography, functional magnetic resonance imaging, neural architectures

Resumo

A atividade eletrofisiológica está intricadamente relacionada com a atividade hemodinâmica, sendo a irrigação sanguínea essencial para promover a homeostasia celular onde necessário. Avanços na computação permitem uma elevada capacidade de processamento, despoletando uma análise profunda de neuroimagem médica. Complementarmente à electroencefalografia (EEG), a emergência da resonância magnética funcional (fMRI) permitiu avanços na compreensão da hemodinâmica cerebral. Estas modalidades captam funções cerebrais diferentes e têm resoluções espaço-temporais diferentes.

Este projeto mapeia EEG em fMRI através de princípios de processamento neuronal. Através deste mapeamento, diferentes aplicações podem ser exploradas: suporte de diagnósticos ambulatórios de doenças cerebrais; estudos longitudinais recorrendo ao EEG como proxy; e aumento da discriminabilidade e interpretação de sistemas de apoio à decisão médica. Neste trabalho, propomos a geração automática de arquitecturas neuronais para dar novas perspectivas para a estrutura de modelos existentes e remover o viés humano do problema. Dada uma arquitectura neuronal óptima que é capaz de mapear EEG para fMRI, melhorando o estado da arte na tarefa em 13% RMSE. Os nossos resultados sugerem que os eletrodos de EEG contêm relações (como entre Oz e PO9) essenciais para reter informação necessária para recuperação de atividade hemodinâmica. Adicionalmente, olhamos para a necessidade de quantificar a incerteza na tarefa de síntese. A solução proposta consiste em manipular o espaço espectral na saída da rede, o que reduz a resolução espacial da síntese e melhora as versões originais. No final, mostramos que a nossa rede está apta a extrapolar a problemas de classificação, através da sua invariância a desvios e mecânicas espectrais. Usando esquizofrenia como prova de conceito, observamos um notável poder discriminativo da vista sintetizada (0.77 na área sobre a curva descritora) e interpretabilidade acrescida.

Palavras Chave: Neuroimagem, Aprendizagem Automatica, Electroencefalogradia, Imagem por ressônancia magnética funcional, Arquitecturas neuronais

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The human brain is a complex system whose structural and functional characteristics are still largely unknown. The study of the brain is mediated by techniques that are able to record part of its characteristics, some capture structural properties (diffusion tensor imaging, structural T1 weighted magnetic resonance imaging), others capture functional information through haemodynamics (**functional magnetic resonance imaging** (fMRI), positron emission tomography) or neuronal activity (magnetoencephalography, **electroencephralography** (EEG)). These neuroimaging techniques all have in common its source of information (the brain), raising the question of whether it is possible to predict the information of a neuroimaging modality (e.g., fMRI) with the information of a more accessible or less expensive (e.g., EEG). The content retrieval between modalities that capture information from a common source has been done before (Zanardi et al. [1]), showing quality benefits. With this motivation, this study focuses on corresponding EEG to fMRI. By comparing these two neuroimaging modalities, one can enumerate differences in terms of spatial and temporal resolution, recorded functionality, cost, exposure time (short versus long duration), availability in ambulatory settings, among others. A special emphasis is given to the cost reduction and hypothesized additional information benefits, that a correspondence between EEG and fMRI, would provide to users.

1.1 Broader impact of corresponding EEG to fMRI

In retrospective, neuroscience research has been impacted by the evolution of techniques that capture heterogeneous dynamics at different timescales, which is the case of EEG and fMRI. EEG, first reported by Richard Caton in 1875, appeared as an useful tool that is nowadays known to contain relevant information. For instance, consider advances on psychiatric or neurodegenerative disorders [2–4]. On the other side of the spectrum, fMRI, a successor of the MRI technique, was discovered by Seiji Ogawa, in 1990, a notable mark in neuroscience. and is widely acknowledged by the community. EEG and fMRI capture characteristics related to the functionality of the brain, therefore being known as functional techniques, and stand as top modalities used in impactful studies. Take, for instance, these works of Taghia et al. [5], Pisauro et al. [6], Mohr et al. [7], Daly et al. [8], published in Nature. EEG can be recorded in ambulatory settings, under long-term exposure protocols (e.g., daily activity monitoring), and requires significantly less preparation than an fMRI recording, with the latter reportedly costing up to hundreds of dollars [9], excluding the cost of the laboratory equipment (an MRI machine can cost from 1 to 3 millions of dollars).

Notwithstanding these advantages, the cost reduction of using EEG to retrieve the corresponding fMRI is seen as an additional driver for this work. This difference is key for the less privileged communities that could be positively impacted by a quality correspondence of EEG to fMRI. Ogbole et al. [10] surveyed the availability of MRI machines in West Africa. A worlwide estimate of 35,000 is reported, accounting for approximately ≈ 4.67 MRI units per million people, not taking into account the limitations of mobility and politics that unable the reality of this number. Taking these restrictions in consideration, people in West Africa have an offer of ≈ 0.22 MRI units per million people. It is worth noting that MRI machines are a key factor for the diagnosis of a number of pathologies, consequently extending life expectation and quality of life if used accordingly [11]. In addition to the social impact, successful EEG to fMRI mappings can yield a notable role in: health care by supporting the ambulatory diagnostic of diseases and enabling longitudinal studies; EEG research by increasing both performance and interpretability of decision support systems that use EEG; and in neuroscience by allowing a better identification of the state of the brain when combining both modalities.

1.2 Why machine learning?

Simultaneous EEG and fMRI recordings were first present in a published study in 1999 by Bonmassar et al. [12] in the NeuroReport journal, with the first ever simultaneous recording taking place in the 90s decade. Thanks to the rise of this neuroimaging fusion technique, not only key discoveries were made in the neuroscience field, but also this thesis is feasible with methods that require such a setup. Complementarily, machine learning has seen a spurt in the last decades [13], with applications in classification, regression, dimensionality reduction, among others. These advances have been recently propelled by the use of the backpropagation algorithm of machine learning, which is the backpropagation algorithm that enabled the convergence of parametrized models using gradient descent optimization. Respectively, in 1986, Rumelhart et al. [14], and latter in 1989, LeCun et al. [15], validated backpropagation to be applied on neural networks. The natural fit of machine learning algorithms in a regression task allows its use in the task of corresponding the input feature information from EEG into the corresponding fMRI. Simultaneous EEG and fMRI enables this task as the input and output are clearly aligned in the recording.

We hypothesize that dissimilarity in structure and representation, between the EEG and fMRI, can be bridged through the use of techniques descibred in this thesis, whose roots are present in recent artificial intelligence advances (Chen et al. [16], Chakraborty et al. [17], Liu et al. [18], Kendall and Gal [19], Tancik et al. [20], LeCun et al. [15]).

1.3 Hypotheses and contributions

The main research question being answered in this work, and consequently the driver hypothesis, is: To what extent can a mapping function successfully transform EEG to fMRI? To answer this question, different ways of addressing such task will be taken into account. Nonetheless, subsequent hypotheses, about such mapping, are drawn:

- What is the discriminative power of a synthesized fMRI representation to support clinical decisions? And how does it compare to other EEG perspectives? See Chapter 9.

- Why are automated machine learning methods promising candidates to bridge the gap between these modalities? See Chapter 6.
- Given a synthesized fMRI, how is the model formulating its decision based on an EEG signal? Considering the functional nature of these modalities, what functional connectivity properties are relevant for the model? See Chapter 7.
- What is the ability, of developed mapping functions, to quantify uncertainty for the predictions? What alternatives make this risk assessment feasible? Which regions of the fMRI are more prone to fail for prediction? See Chapter 8.

These questions will be addressed using machine learning techniques. Fundamentally, the proposed methodology aims at:

- modelling a mathematical function that captures the style features of the fMRI to produce representations that contain haemodynamical characteristics;
- using well known neural architectures to perform the encoding mapping function between EEG and fMRI;
- developing automated machine learning techniques to avoid domain-driven biases, so that the mapping between EEG and fMRI purely relies on the computational and data-centric perspective;
- novel spectral based uncertainty quantification for EEG to fMRI synthesis, allowing an easy plug and play to alternative neural processing models;
- extrapolation of the predictive approach from a synthesis regime to a classification task, maintaining the learned fMRI style and incorporating new discriminative properties from different data sources.

1.4 Thesis outline

This document is divided in three parts: *The Foundations* (Part I), *The Problem* (Part II) and *The Proposed Solution* (Part III). The first part provides the reader with the state-of-the-art on EEG, fMRI and simultaneous EEG and fMRI studies in each of their applications, such as motor imagery, disease diagnosis and connectivity dynamics (Chapter 2), along with the necessary machine learning background, where operations such as convolutions, attention mechanism, and Fourier features are described (Chapter 3) and needed to understand the following content. The second part, describes the problem by clearly setting the variables, target and functions (Chapter 4), followed by a description of the experimental setup, specifically the datasets to be considered in this study, the quantitative evaluation metrics, the qualitative evaluation that is set with explainability and uncertainty quantities, and the decision support setting to evaluate the effectiveness of the target synthesis task (Chapter 5). The third part describes the computational solutions, experimental results, and subsequent discussion. A neural architecture generation framework is proposed (Chapter 6). First, the problem formulation is described. Here, the reader will find the encoding formula that respects the arithmetic of convolutions for a specified input and output space, following the experimental setup description. Resnet [21] is used to test the framework. Finally, the results are presented and the uniformity trait of the generated neural architecture space is ultimately validated. Then, the neural network

that maps EEG to fMRI is introduced (Chapter 7). The machine learning foundations are discussed on its fit for the mapping function between the two brain signals. We assess how the proposed model compares with the state-of-the-art. Further, we analyze thoroughly the explanations, given by algorithms, that justify the neural network's predictions. Uncovering EEG relationships that go in accordance with related work on simultaneous EEG and fMRI data. Next, we dive into the uncertainty quantification in the proposed neural network (Chapter 8). Inspired on its internal mechanisms, we compare its function to a transform, with the latent representations operating in the neural network's own spectral domain. This analogy allows us to propose a novel method to quantify uncertainty, that is easily able to generalize to different models. We also found that the proposed method relaxes the spatial resolution of the problem, which inherently comes in agreement with the spatial resolution of the EEG signal. Given the model that can synthesize fMRI from an EEG, we hop from a regression to a classification task (Chapter 9). Here, we test the ability of the neural network to extrapolate the learned fMRI style to a diagnostic setting, with schizophrenic individuals and healthy controls. Therefore, we set an experimental setting that allows us to assess the discriminative and interpretative power of the synthesized fMRI. Finally, the concluding remarks and future directions are drawn (Chapter 10).

Note that, as of now, the work of this thesis produced publications such as Calhas et al. [22], Calhas and Henriques [23], Calhas and Henriques [24], Calhas and Henriques [25], and Calhas and Henriques [26], with the first being the most relevant for the solution proposed. The following contributions are currently under review: Chapter 7, condensed in a preprint that can be found in the Calhas and Henriques [24] study, proposes a model for EEG to fMRI Synthesis and explores the explanations that made the prediction, ultimately analyzing the connection between these two modalities; Chapter 8 culminates in a contribution that explores the degree of uncertainty of the mapping function, of the network proposed in Chapter 7, and proposes a spatial relaxation of the problem that goes in accordance with the spatial resolution of the EEG; and Chapter 9 produced a study that analyzes the discriminative and interpretative power of the synthesized fMRI modality for schizophrenia.

Part I

The Foundations

Chapter 2

The State of Neuroimaging Research

Here we introduce the reader to the state of neuroimaging research by discussing the recent studies that use electroencephalography (EEG), functional magnetic resonance imaging (fMRI), and simultaneous EEG-fMRI. The outline is as follows: Section 2.1 offers an overview of EEG research advances; Section 2.2 covers fMRI research advances; Section 2.3 discusses the emergence of studies that combine both modalities to either improve success in certain domains or even aid the understanding of the human brain; Section 2.4 contains a quantitative comparison of the studies presented in this chapter that either use EEG or fMRI in disease diagnostic settings; finally, Section 2.5 provides a brief summary of the theoretical concepts and essential claims that the reader should remind for a better understanding of the contributions of our work.

2.1 Electroencephalography

EEG is an attractive neuro sensory technique due to its simple setup when compared to other neuroimaging laboratory setups. It became popular with its wide use and application on identifying epileptic seizures [27]. Acharya et al. [28] was one of the pioneer studies in the application of deep learning in neuroimaging data settings, specifically in automated detection of epilepsy. The methodology proposed, achieved an accuracy of 88.7%. However, as it happens with a large amount of studies that use deep learning and other machine learning techniques with limited neuroimaging data observations, the study appears to have overfitted the dataset. This is resembled with the description of the dataset used: 5 individuals with a total of 100 EEG signals recorded of 23 seconds each. Since then, researchers have found more critical applications where EEG is used to support decisions. Oh et al. [29] applied a similar neural network architecture to Acharya et al. [28], which in the task of identifying individuals with Parkinson's disease achieved an accuracy of 88.25%. They show that the EEG signature of Parkinson's is not able to be identified at the naked eye, as opposed to an epileptic seizure. As such, the non linearity and convolutional nature of operations applied by the neural network were able to compute discriminative features from the preprocessed EEG signal. Although the achieved accuracy was high, it is worth noting that the dataset had a limited number of observations, with only 20 individuals and 5 minute EEG recordings. In addition to the hypothesis that once again the method overfitted the input data, no additional results were shown in other datasets, and the method was unable to reach state-of-the-art predictive performance for the task, which according to the authors is attained by support vector machines. Ieracitano et al. [30] performed a multinomial classification task of Alzheimer

using EEG recording, differentiating between Alzheimer's disease, mild cognitive impairment and healthy control. They claim that their method is multimodal in respect to the variable types and not driven by the heterogeneity of data sources. This study was validated on a population of 189 individuals equally distributed among the defined classes, and EEGs were recorded in a resting state with closed eyes setting (4 minute recordings with 19 channels). The two types of features extracted consisted of continuous wavelet transform and bispectral features. These were concatenated and given as input to a classifier. The classifier that performed best was a fully connected neural network, with 80% accuracy considering all classes and reaching 90% in 2 classes settings (healthy controls vs cognitive impairment).

When learning from EEG data, typically the norm is to extract frequency features, such as frequency band intensities. However, the feature engineering ability of neural networks has shown that raw data can be fed as input and still competitive results are achieved. The work of Gemein et al. [31] is a pragmatic case, where the use of convolutional operations on the raw EEG data enable the extraction of relevant features which are then commonly fed to a fully connected hidden layer (assuming a classification task). The study differentiated between healthy controls and individuals, claiming an accuracy of 86.16%.

In non health care related settings, EEG is also shown to be an important neuroimaging modality. Song et al. [32] used EEG to retrieve emotion characteristics from individuals in a task specific setting. A graph neural network, with each EEG channel set as a node of the graph, dynamically learned relationships between the defined edges, which with the addition of graph convolutions was able to compute rich compound features from the properties of the EEG frequency bands. Results shown in the study support the claim that a graph neural network is better suited to aggregate EEG data due to its spatial discontinuity. At this stage it is worth noting that other machine learning branches outside of deep learning offer state-of-the-art results in EEG-driven tasks. Such is the case with flexible analytic wavelet transform proposed by Gupta et al. [33], a decomposition algorithm, combined with a random forest or a support vector machine, that shows pivotal results for emotion assessment from EEG data. On a distance based perspective, Wang et al. [34] developed a framework that is able to extract distance based rich features that, when fed to an SVM, are able to correctly specify an emotional state. These states were modelled using a hidden Markov model capturing the underlying brain-emotion dynamics from labelled data.

Motor imagery has drawn a lot of attention, especially with the surge of attendable EEG devices. EEG is a natural fit to the motor imagery task due to its high temporal resolution that is necessary for a fast response. Neuper et al. [35] is one of the former studies achieving significant results in this task. With the increasing relevance of neural processing methods, more studies were published, showing the capacity of achieving/surpassing the state-of-the-art. Müller et al. [36] was one of the pioneer studies that applied machine learning methods on neuroimaging data, specifically EEG, to perform a motor imagery task. An et al. [37] showed that a different branch of deep learning, restricted Boltzmann machines, are also able to accurately classify motor imagery classes. Nakagome et al. [38] is another relevant study that made a comprehensive comparison on a variety of algorithms in a motor imagery task with EEG data. Among the algorithms, there was a special focus on deep learning with convolutional neural networks (specifically Temporal Convolutional Networks), as well as recurrent (LSTM and GRU). The authors found that the state-of-the-art alternatives proposed by Unscented Kalman filter [39], still outperformed the deep learning techniques used as baselines. The study conducted by Nakagome et al. [38] does not necessarily take away importance from the increasing usage of neural networks, but it rather claims the need to not exclusively

look at deep learning. Amin et al. [40] is another study that shows the ability of convolutional neural networks to perform feature extraction from the raw EEG signal, achieving a competitive performance with state-of-the-art in the EEG decoding task. The method proposed is based on a multi layer convolutional neural network, whose hidden activations are fused using a fully connected layer that outputs the logits. With such a method, they claim each convolutional layer learns a specific representation that, when fused with all layers, forms a very discriminative motor imagery feature set. The study was validated in two datasets: BCI Competition IV 2a and High Gamma datasets with 75.7% and 95.4% accuracy, respectively. There is a wide variety of methods applied to EEG data. Recently, there has been a push to apply Bayesian learning methods that are able to quantify uncertainty, which is key to making real life decisions. Wu et al. [41] provides an excellent overview of such methods in an EEG setting. Dai et al. [42] also employed a deep learning method on the BCI Competition IV 2a dataset. They used a convolutional neural network that was followed by a variational encoder. The variational encoder is described by a multi layer network, whose weights are sampled from a Gaussian distributions. The parameters of the distributions are trained using the computed gradients. This type of approach is seen preferable due to the increasing necessary care of computing the uncertainty associated with the predictions. The input of the model was the STFT representation concatenated accross the channels corresponding to the motor cortex region. The results are not easily comparable as the work is not focused on predictive accuracy.

2.2 Functional Magnetic Resonance Imaging

The previous section described a wide variety of EEG applications. Disease diagnosis, emotion retrieval and motor imagery tasks were covered. However, to the best of our knowledge there is only one common fMRI application, disease diagnosis. In this context, a comparison between the two modalities is limited. This may be due to lack of portability of an fMRI recording equipment. As such, this section focused on assessing the role of fMRI for disease diagnostic studies, studies focused on uncovering brain dynamics and the use of fMRI in regression settings.

In contrast to EEG, fMRI has not been traditionally considered to address epilepsy. Still, up to the early 2000's, some studies focused on the role of fMRI to study epilepsy [43]. More recently, in the context of neurodegenerative disorders, Li et al. [44] developed a neural network architecture composed of convolutional and recurrent operations that were able to accurately distinguish between the three standard groups in an Alzheimer setting. The method is composed of a convolutional neural network that extracts features along a time series of volumes (independently from each volume). The extracted features are fed as a sequence to a long short term memory layer, followed by an affine transformation to the logits used for classification. The input data had a total of 116 individuals with Alzheimer disease, 99 in a mild cognitive impairment state, and 174 healthy individuals. A cross validation was run to discover the hyperparameters of the model. The study claimed a 89.47% accuracy when accounting for the three groups. Hojjati et al. [45], in resemblance with the previous work, performs classification of Alzheimer's disease. For this, the recordings of 177 individuals were used, where 34 had Alzheimer, 94 had mild cognitive impairment, and 49 were healthy controls. With the resting state fMRI and structural fMRI, local and global properties of the built graphs were extracted and filtered (see study for further details [45]), then used as features for classification with a support vector machine. The filtering of the features was done using a sequential

feature selection, which according to the authors is able to select features yielding maximal statistical dependency based on mutual information. They claim an accuracy of 56% considering all classes. Kazemi and Houghten [46] employed a well known architecture, AlexNet [47], that achieved a remarkable accuracy of 97.63%. The dataset used was a subset of the Alzheimer's Disease Neuroimaging Initiative, that contains resting state fMRI recordings. A total of 197 individuals were considered and each volume was sliced, producing 64×64 images. These images were fed as input to the architecture. This study is paradigmatic example on how deep learning methods can surpass the traditional algorithms for the classification of images.

fMRI has been widely used to understand the human brain, uncovering important findings. In contrast with the previous studies, we will now cover notable descriptive studies that were found relevant to this topic. EEG and fMRI measure different brain processes that evolve at different time scales. One of the disadvantages of fMRI against EEG, is that it can not capture brain processes evolving at time scales of 10ms, which EEG is able to do. On the other hand, it is still difficult to uncover brain processes evolving in some sub-cortical regions (although not impossible according to Daly et al. [8]) due to its poor spatial resolution. Recently, advances were made in neuroimaging techniques that are able to have a spatial resolution as good as fMRI, with a much better sampling rate [48]. Despite the advances in this field, these new techniques continue to be expensive and only available to a small portion of the human population. Miller et al. [49] provide an extensful discussion on fMRI spatial, temporal, functional and connectivity dynamics. They hypothesize that we should not concentrate on whether fMRI dynamics exist, but rather on how they are represented, e.g., by means of differentiation over temporal, spatial and functional scales. The presence and susceptibility to artifacts in fMRI is further discussed as a human brain that is in a state of wakefulness is constantly receiving functional stimuli. There is still a limited understanding on which signal variations have influence on a shift on functions of the brain. Huang et al. [50] studies the transitions between the default mode network and the dorsal attention network, claiming that consciousness is present at a macro scale (fMRI temporal resolution scale). To this end, a Markov process is modelled with coactivation patterns defined as states (please refer to the original work by Huang et al. [50] for more details). After the learning session, where the transition probabilities are estimated, thorough conclusions were taken. Chang and Glover [51] studied the brain dynamic changes in a resting state setting (recordings of 12 to 15 minutes long with eyes closed) setting on fMRI default mode networks, using of a wavelet transform. The study was conducted on a dataset of 12 individuals. The dynamic behaviour of specific brain regions of interest were examined in order to check which were more strongly and consistently negatively correlated with the default mode network [51]. The Wavelet Transform Coherence algorithm was implemented to analyze the coherence and lag between two time series, so as to build a network through correlation metrics. Statistically significant results were found when comparing resting state networks. Taghia et al. [5] analyzed brain dynamics using the hidden states of a hidden Markov model (HMM). Specifically, Bayesian switching dynamical systems is suggested to optimize the number of states to model the dynamics. The transitions between these states are easily interpretable. Taghia et al. [5] study is highly essential in the context of this manuscript as it uses machine learning methods to model dynamics. Although it does not implicitly model a function describing the behaviour of brain dynamics, it studies the transitions between defined states (through the use of HMMs). An important breakthrough on the modelling of fMRI dynamics was attained by Tagliazucchi et al. [52] being the first to model large scale brain network dynamics. Several conclusions

were taken from this study [52]: 1) blood oxygen level dependent fluctuations contain rich information about brain dynamics; and 2) brain dynamics can be taken from data subjected to a high degree of reduction (lossy compression). In particaular, [52] a reduction of $> 94\%$ was performed without significant impact.

Casey [53] performed a regression of the activity of voxels/multiple voxels from music stimuli. The study gathered 20 subjects that listened to 25 music clips (from 5 genres) each. The hypothesis was that distinct musical attributes could be encoded in different voxel spaces. Although the method applied does not perform a regression of the activity of the whole brain, it does perform regression of low level dynamics, which in this case is the activity of a subset of multiple voxels. Nunez-Elizalde et al. [54] perform a regression task of brain image (fMRI) and words. The study recorded fMRI of individuals during a naturally spoken narrative of 2 hours. Using this data, word2vec embeddings were used to perform Ridge and Tinokov regression of encoded fMRI images. This is a type of study that falls into the low level dynamics hierarchy. Although the approach is simple, it needs a second modality to be able to regress fMRI data. Radhakrishnan et al. [55] used stochastic differential equations to understand the cerebellar neuro vascular coupling. The neuro vascular coupling is related to changes in neural activity and blood flow in the brain. The work falls into the category of low level dynamics, however it is not able to be compared with ours, as the changes being analyzed are related to a substance (Nitric Oxide), that is not observable in fMRI data. Jain and Huth [56] explored the ability of latent recurrent representations to perform regression to haemodynamic activity. To this end, a recurrent neural architecture was developed with a standard number of layers and each latent activation was used independently to perform the regression. It was found that the activations of top level layers are the best layers for this task. Adding to it, the proposed recurrent activations outperformed natural language processing baselines in this task. Jain and Huth [56] is an impactful work that bridged the gap between stimuli (natural language) and haemodynamic activity. The work was extended by Jain et al. [57], which explored the significance of different timescales (in natural language timescales can be defined at the word level, sentence level, etc) across the regions of the brain. The exploration of the timescales was done with a recurrent neural network, parametrized with different values for the forget gate of an LSTM. The latter is able to analyze different timescales. Vaidya et al. [58] analyzed how speech associated with cortical activity. The experimental setting involved: fMRI data; various machine learning models for speech feature selection; and the correspondings word embeddings annotated for the audio. Their findings suggest that there are associations with audio speech and cerebral activity, but only limited to the temporal cortex. Further, cortical activity associated with semantic understanding of the brain was more correlated with word related features, showing that there are various hierarchies that participate in the process that the human cortex does for speech.

2.3 Simultaneous EEG and fMRI

The firing of neurons spends molecules that are necessary for these cells to work. These molecules are given to the neurons from the intracerebral arterioles and capillaries that transport oxygen and glucose. While neuronal activity is captured by the EEG, fMRI captures this blood flow. One is related to the other in the sense that the blood flows to where it is needed [59]. Note however, that some neurons present inhibitory activity, that is activity that surpresses action potentials and that do not reflect in the electrical field, and they also spend energy. The blood flow is able to resemble this type of activity [60]. EEG is able to capture

rapid temporal processes, while fMRI captures fine spatial processes. As these two modes capture different characteristics of brain activity, in the past decade there has been a surge of studies that unprecedently combined both. For instance, this joint force has been studied to retrieve cognitive functions, Cichy and Oliva [61] claim that the identification of neural activity in space and time enables characterization of cognition. To support this observation, they linked fMRI with EEG/MEG using representational similarity matrices by computing the similarities of voxel activations (fMRI) and sensor activation patterns (EEG). The study was carried with fMRI and EEG/MEG data recorded in an experiment that carried a visual object processing. Further, the hypothesis that EEG and fMRI capture common processes was tested with success at specific brain locations and time intervals/periods. One of the main barriers of EEG is its poor spatial resolution, which makes retrieving data from sub cortical regions difficult. Daly et al. [8] is the first study to show that the activity in sub-cortical regions can be retrieved directly from EEG dynamics. This, until then, was considered to not be possible using EEG, as there is noise inherent due to intermediate regions being captured at the scalp. Specifically, results in [8] showed that prefrontal EEG asymmetry changes reflect activity in sub-cortical brain regions. Mann-Krzisnik and Mitsis [62] studied how EEG and BOLD simultaneously change. For this, they apply a Matrix Factorization (MF) method, decoupling both EEG and BOLD signals at the same time. They claim that the applied MF method outputs features containing temporal, spectral and spatial factors. Although the EEG and BOLD data contain those type of features (spatial, temporal and spectral), their preservation in MF is not guaranteed. Liu et al. [63] perform regression mappings between EEG and fMRI on both directions. They applid a cycle generative adversarial network without the use of generated instances, and were able to perform regression of fMRI from EEG and vice versa. The neural network is composed of convolutional layers (for encoding) and transposed convolutional layers (for decoding). This is another low level dynamic method, that also uses a second modality to be able to better describe fMRI dynamics. Cury et al. [64] performed a regression task between neurofeedback scores of EEG and fMRI, with the goal of enriching EEG neurofeedback sessions with the information of fMRI. The study gathered a total of 17 individuals and recorded simultaneously EEG and fMRI. EEG data was gathered from 64 channels sampled at 5kHz and fMRI was recorded in a 3 Tesla scanner. They found an improvement by assessing the Pearson correlation between the EEG baseline and the model trained with simultaneous EEG and fMRI. A relevant finding of this study is that the model was trained using two modalities, EEG and fMRI, and afterwards it is capable of retrieving rich information only from EEG. Philiastides et al. [65] provide a good discussion on the capabilities and challenges faced by fusion methods of EEG and fMRI. One of the mentioned challenges is the clear difference in the structures of each modality, which even after the feature extraction process are still highly dissimilar. A turnaround for this problem is to perform an affine transformation to a shared space, where both modalities can be related and participate in simple mathematical operations. Pisauro et al. [6] took advantage of simultaneous EEG and fMRI to accurately identify the source locale (fMRI region) of an observed accumulation of activity in the EEG electrodes. This shows that one is able to predict the spatial coordinates of observed neuronal activity. One of the difficulties faced with fMRI data is the variability of functional measurements that ends up producing ill results, due to noise artifacts, such as small movements and blood flow. Wirsich et al. [66] assess the impact of these effects, combining EEG and fMRI to accurately fit connectivity measures of function and structure. The study recorded simultaneous EEG and fMRI, as well as structural MRI. They hypothesized that functional connectivity should resemble structural connectivity if one averages

the function. However, this is not observed using fMRI only, but with the addition of neuronal activity, specifically EEG spectral features, a better fit of connectivity was achieved. This study showed that neuronal activity contains rich information about the source. Portnova et al. [67] provides an extensive analysis of how different features of the EEG signal correlate with BOLD fluctuations. The method used for correlation was a non parametric *t-Student* test, between covariates of EEG features and BOLD fluctuation (a widely used technique in neuroimaging research). Spectral analysis was considering for extracting EEG features, which strengthens the motivation towards the use of spectral features when relating to BOLD fluctuations.

2.4 Comparison between individual EEG and fMRI studies

This section undertakes a comparison between the disease-related studies covered in Sections 2.1 and 2.2. In total, there are five studies covering Alzheimer's (AD) with EEG. These studies show an mean accuracy 0.770 ± 0.200. The comparison made with the studies on the same setting, but undertaken using the fMRI modality, tell us that on average studies that use EEG to classify AD have a slightly worse performance (-0.039 in accuracy). Similarly, studies that use fMRI to classify AD (a total of three), have a mean accuracy of 0.810 ± 0.000. Comparing to EEG, fMRI similarly shows accuracy gains of $+0.040$, meaning these may be preferred in the fMRI perspective to classify AD. Moving to the studies on Parkinson's (PD), there is a total of one study that uses EEG to diagnose PD, showing a 0.882 accuracy. In comparison to fMRI, EEG yields a difference on accuracy of -0.013 when compared to fMRI. From the fMRI perspective, there is a total of two studies with mean 0.895 and deviation $+0.0013$. When comparing them to studies that use EEG to perform PD classification, there is a difference $+0.013$, meaning once again that fMRI is preferable to EEG in the fMRI perspective.

Since the comparison made in the previous paragraph does not take into account statistical significance, one cannot claim that fMRI is better than EEG to perform diagnosis in a clinical setting. Nonetheless, the purpose of this section is to give the reader an overview of the state-of-the-art studies in neuroimaging.

(Setting) Study	EEG	fMRI	Accuracy	Diff Intra	Diff Inter
(EP) Acharya et al. [28]	✓		0.887	*NA*	*NA*
(AD) Amin et al. [40] I	✓		0.757	−0.013	−0.053
(AD) Dai et al. [42] I	✓		0.564	−0.206	−0.246
(AD) Tabar and Halici [68] I	✓		0.776	0.006	−0.034
(AD) Amin et al. [40] II	✓		0.954	0.184	0.144
(AD) Ieracitano et al. [30]	✓		≈ 0.800	0.030	−0.010
(PD) Oh et al. [29]	✓		0.882	0.000	−0.013
(ALL) Gemein et al. [31]	✓		0.861	*NA*	*NA*
(AD) Hojjati et al. [45]		✓	0.560	−0.250	−0.210
(AD) Kazemi and Houghten [46]		✓	0.976	0.166	0.206
(AD) Li et al. [44]		✓	0.894	0.084	0.124
(PD) Long et al. [69]		✓	0.869	−0.026	−0.013
(PD) Dehsarvi and Smith [70]		✓	0.921	0.026	0.039

Table 2.1: Claimed results of EEG and fMRI on disease diagnostics.

2.5 Summary

- EEG data analysis has solid applications in disease diagnosis, motor imagery and emotion recognition. Many of this applications rest on the trait of a good temporal resolution;
- fMRI data analysis has notable applications in uncovering brain dynamics and to allow a better understanding of the human brain;
- The neural correlates between electrophysiology and haemodynamic happen because blood supply oxygenates the cells, to make up for molecule consumption of the neurons when they fire. However blood supply is also related with inhibitory signals, which are not present in the brain electrophysioligy recorded by the EEG;
- It is known that EEG is able to accurately retrieve the source of neuronal activity. This claim along with other claims of neural correlates with haemodynamic response motivate a mapping between EEG and fMRI.
- Simultaneous EEG and fMRI is a recording technique that joins these two modalities. EEG-informed fMRI is an example that preserves the good traits of each, while removing some of the drawbacks;
- Finally, provisory comparisons show that fMRI might be preferred to EEG for some computer-aided medical decisions. None-theless, it is an expensive neuroimaging technique that may not be easily available and further inappropriate for long-term monitoring protocols.

Chapter 3

The Machine Learning Foundations

Let us start by focusing on neural processing techniques, such as fully/densely connected, local and non-local transformations. Consider the following setup: given an instance $\vec{\mathbf{x}} \in \mathbb{R}^n$, with $n > 1$, and a linear classifier $l_c : f(W_c^\top \cdot \vec{\mathbf{x}} + b_c)$, with $f(z) = 1$, if $z \geq 0$, and $f(z) = -1$, if $z < 0$. We can use a gradient descent method to optimize the parameters of the linear classifier $\{W_c, b_c\}$. The feature engineering ability of this type of classifier is limited. One way to improve its performance is to simply represent $\vec{\mathbf{x}}$ as pairs of its own features, such that $\vec{\mathbf{x}}_r = \{\forall i, j \in \{1, \ldots, n\} : \vec{\mathbf{x}}_i \vec{\mathbf{x}}_j\} \in \mathbb{R}^{n \times n}$. The latter procedure relates to why neural networks work well.[1] Now, consider a neural network with 2 layers, $L = \{L_1, L_2\}$, where the input, $\vec{\mathbf{x}}$, is first processed by $L1 : L1(\vec{\mathbf{x}}) = W_{L_1}^\top \cdot \vec{\mathbf{x}} + b_{L_1} \in \mathbb{R}^h$, with $W_{L_1} \in \mathbb{R}^{n \times h}, b_{L_1} \in \mathbb{R}^h$, and then processed by $L2 : L2(L1(\vec{\mathbf{x}})) = W_{L_2}^\top \cdot L1(\vec{\mathbf{x}}) + b_{L_2} \in \mathbb{R}^c$, with $W_{L_2} \in \mathbb{R}^{h \times c}, b_{L_2} \in \mathbb{R}^c$. The hidden representation, $L_1(\vec{\mathbf{x}})$, called hidden due to being positioned between the input and output representations, is seen as a feature engineering technique, optimized through the parameters W_{L_1}, b_{L_1} by targeting a classification task. Each hidden feature, $\forall i \in \{1, \ldots, h\} : L_1(\vec{\mathbf{x}})_i$, is formulated as a weighted sum of all input features, $\forall j \in \{1, \ldots, n\} : \vec{\mathbf{x}}_j$, as $L_1(\vec{\mathbf{x}})_i = \sum_{j=1}^{n} W_{L_1 ji} \times \vec{\mathbf{x}}_j + b_{L_1 i}$. This is the feature representation formulation of neural networks, using densely connected transformations, but other types of transformations may be used. Recalling the first example, this relates to the pairwise multiplication example as it represents the original features in a space that contain more information about the input, using the parameters.

In this chapter, we provide an overview on the machine learning (ML) techniques used along this thesis. First, the convolution arithmetic is introduced, along with local and pooling operations, in Section 3.1. In Section 3.2, an introduction to a simple attention mechanism is given. Finally, in Section 3.3 we introduce Fourier features. This chapter is closed with a brief summary in Section 3.4.

3.1 Convolution

Convolutional layers had their major breakthrough in LeCun et al. [15], using backpropagation along with convolution arithmetic operations. This operation is one of the most used techniques in computer vision still to this day. To understand the inner workings, of a convolutional layer in machine learning, consider a 1-dimensional instance, $\vec{\mathbf{x}} \in \mathbb{R}^{w \times c}$, with w being the length of the signal. The constant c is the number of

[1] Example taken from the DSML fall 2020 course from DEEC

channels, e.g. it can be thought of as the representation of a color in RGB, where one needs 3 channels. A convolution is represented by:

- k_1, the kernel size;
- s_1, the stride size;
- f, the number of filters.

Some machine learning software libraries include a parameter p_1, which represents the padding size, i.e. the number of zeros (in the case of zero padding) added to the boundaries of $\vec{\mathbf{x}}$. For the sake of simplicity, we only consider two types of padding techniques are considered: *valid* and *same*. *Valid* padding corresponds to having $p_i = 0$, whereas *same* corresponds to having p_i, such that the signal after being processed by the convolution still has the same dimensions as

$$C_l(\vec{\mathbf{x}}) \in \mathbb{R}^{w \times f} : f = c, \tag{3.1}$$

where C_l is the convolution operation. To extend this operation to more dimensions, let a convolutional layer be formulated as $C_l(\vec{\mathbf{x}}; f, k, s, p)$, with the special case of a *valid* padding as $C_l(\vec{\mathbf{x}}; f, k, s, \vec{\mathbf{0}}) = C_l(\vec{\mathbf{x}}; f, k, s)$. The variables k, s and p are a composition of all the values for each dimension of $\vec{\mathbf{x}}$. Equations 3.2 and 3.3 represent the *valid* and (some) type padding, respectively.

$$O_i = \frac{I_i - k_i}{s_i} + 1 \tag{3.2}$$

$$O_i = \frac{I_i + 2p_i - k_i}{s_i} + 1 \tag{3.3}$$

Notice that the second is just an extension of the first equation, being the first the particular case when $p_i = 0$. In sum, a convolution is a shifting window (kernel), that jumps with a step size (stride) along a signal (for the 1-dimensional case), that may or may not be padded (padding). The same reasoning extends to the multidimensional case, consider the following formalization: the input shape $I = I^{(1)} \times \cdots \times I^{(K)}$, kernel size $k = k^{(1)} \times \cdots \times k^{(K)}$ and stride size $s = s^{(1)} \times \cdots \times s^{(K)}$. For each dimension $d \in \{1, \ldots, K\}$, the same reasoning applies. Figure 3.1 illustrates how a convolution works for a specific case. In the next

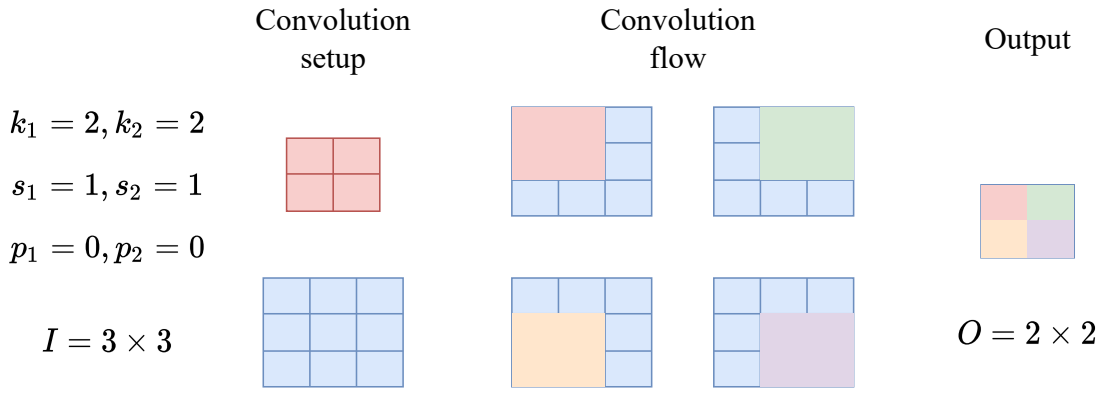


Figure 3.1: Example of a 2-dimensional convolution with parameters $I = 3 \times 3$, $k_1 = 2, k_2 = 2, s_1 = 1, s_2 = 1, p_1 = 0, p_2 = 0$, that when plugged in Equation 3.3 produces $O_1 = \frac{3+2\times0-2}{1} + 1 = \frac{1}{1} + 1 = 2$, the same goes for O_2.

section, we describe two *local* types of operations that respect the convolution arithmetic.

3.1.1 Local operations

Local convolutional operations apply the same operational kernel to all regions of the input it hovers through (in Figure 3.1, the input regions analyzed are the colored ones). The specific local operations considered are: **convolution** and **pooling**. These operations only differ in the type of kernel used.

A convolution applies a kernel, with weights $W \in \mathbb{R}^{k_1 \times \cdots \times k_K}$, to each region

$$I[i_1 s_1 : k_1 + i_1 s_1, \ldots, i_K s_K : k_K + i_K s_K] \in \mathbb{R}^{k_1 \times \cdots \times k_K}, \tag{3.4}$$

producing an output

$$o_i = \phi(\sum_j W_j I[i_1 s_1 : k_1 + i_1 s_1, \ldots, i_K s_K : k_K + i_K s_K]_j) \in O. \tag{3.5}$$

It is local, because the same set of weights, W, applies to all regions of the input. Making a convolution one of the operations with fewer number of parameters (when compared with a densely connected layer). Similarly, a pooling operation implements the convolution arithmetic as well. However, since it does not apply weights, it is a non parametric operation. Examples of pooling operations are: average pooling and max pooling [71]. Average pooling takes the average of the values in $I[i_1 s_1 : k_1 + i_1 s_1, \ldots, i_K s_K : k_K + i_K s_K]$, whereas max pooling computes the maximum.[2] The concept of locality comes from the application of the same operation to different regions of a signal. Figure 3.2 illustrates the operations introduced in this section.

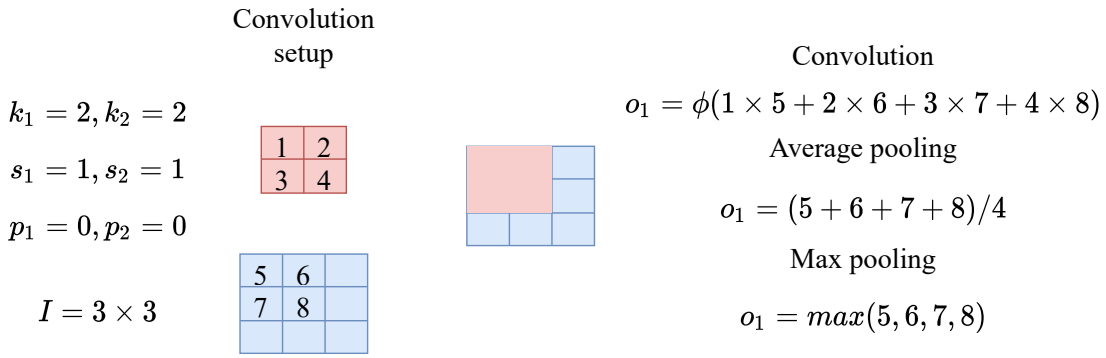


Figure 3.2: The kernel, in red, with $w = [1, 2, 3, 4]$, in a convolution setup multiplies the weights with all values of a region $I[i_1 s_1 : k_1 + i_1 s_1, \ldots, i_K s_K : k_K + i_K s_K] = I[0 \times 1 : 2 + 0 \times 1, 0 \times 1 : 2 + 0 \times 1] = [5, 6, 7, 8]$ and sums them, following it applies a function ϕ. An average pooling operation takes the value of the regions and computes the average value. A max pooling operation takes the values of the region and computes the maximum value.

3.2 Self attention

Let $\vec{\mathbf{x}} \in \mathbb{R}^{l \times c}$ be a 1-dimensional signal, such that c is the channel dimension and $\forall i \in \{1, \ldots, n\} : \vec{\mathbf{x}}_i \in \mathbb{R}^c$. Recall, from the beginning of the chapter, a densely connected layer, now formulated as

$$\phi(A^\top \vec{\mathbf{x}}), \tag{3.6}$$

[2]Note that max pooling acts as average pooling operation in order to compute gradients since it needs to be differentiable to propagate gradients [13].

where $A \in \mathbb{R}^{c \times f}$, f is the hidden size and ϕ is the activation function. In contrast to a densely connected layer, the concept of attention in machine learning consists on $A \in \mathbb{R}^{n \times n}$, now the attention weight matrix, defining attention scores $\alpha_i : i \in \{1, \ldots, l\}$ to different locations $j \in \{1, \ldots, l\}$ of the input signal. Let us define

$$\vec{\mathbf{x}} = \begin{bmatrix} x_1 \\ \vdots \\ x_l \end{bmatrix} \in \mathbb{R}^{l \times c}, A = \begin{bmatrix} a_1 \\ \vdots \\ a_l \end{bmatrix} \in \mathbb{R}^{l \times c}.$$

An attention mechanism starts by computing context vectors, $c_i \in \mathbb{R}^l$, such that[3]

$$\forall i \in \{1, \ldots, l\} : c_i = a_i \cdot \vec{\mathbf{x}}^\top, \tag{3.7}$$

Each context vector is then normalized to produce a weight vector $\alpha_i \in \Delta^l : \sum_k \alpha_k = 1$ as

$$\alpha_i = \frac{e^{c_i}}{\sum_j e^{c_j}}. \tag{3.8}$$

By doing this process $\forall i \in \{1, \ldots, l\}$, one ends up with an attention processed weight matrix, $W \in \mathbb{R}^{l \times l}$,

$$W = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_l \end{bmatrix}. \tag{3.9}$$

With this weight matrix, the final step consists on giving context to each $i \in \{1, \ldots, l\}$ as

$$\sum_j W_{ij} \odot \vec{\mathbf{x}}_j. \tag{3.10}$$

Note that there are various types and ways of performing attention. Attention was first introduced to the machine learning community by Bahdanau et al. [72], using it in a neural machine translation task to provide context in a vanilla recurrent neural network with an encoder-decoder architecture. Nowadays, alternative complex attention mechanisms have been proposed. The mechanism described in this section is simple and suffices for the reader to understand the next chapters. More complex mechanisms, such as multi head attention proposed by Vaswani et al. [73] and Devlin et al. [74], are not considered, since they demand significant computational power, requiring expensive computational resources to run such a model. Recall from Chapter 1 that the main goal is to provide rich information without increasing costs.

3.3 Fourier features

In this section, we describe a projection, that is used in state-of-the-art computer vision studies [20, 75], to synthesize signals. It is called Fourier features and owes its name to the analogy it has with the sinusoid basis functions of Fourier transform. Consider a set of random Fourier features [75] as

$$\cos(\omega \cdot \vec{\mathbf{x}} + b), \tag{3.11}$$

[3]Note that $x_i \odot A_i$ is the element wise multiplication and not the dot product.

where ω is a random variable of size d, b is an uniform random variable and $\vec{\mathbf{x}}$ is the set of original features. Rahimi et al. [75] claim that a kernel, e.g., Gaussian kernel $k(\vec{\mathbf{x}}, \vec{\mathbf{y}}) = \exp(-\gamma||\vec{\mathbf{x}} - \vec{\mathbf{y}}||^2)$, can be approximated with $\boldsymbol{z}(\vec{\mathbf{x}})\boldsymbol{z}(\vec{\mathbf{y}})$, with

$$\boldsymbol{z} = \sqrt{\frac{2}{D}} \left[\cos(\omega_1 \cdot \vec{\mathbf{x}} + b_1) \quad \dots \quad \cos(\omega_D \cdot \vec{\mathbf{x}} + b_D) \right], \tag{3.12}$$

given that $\omega \sim \mathcal{N}(0, 1)$, which is the Fourier transform of the kernel, k, and $b \sim \mathcal{U}(0, 2\pi)$. Intuitively, this consists on D random projections of $\vec{\mathbf{x}} : \omega \cdot \vec{\mathbf{x}} + b$ to the unit circle given by the sinusoidal transformation (cosine). This type of features are able to approximate a kernel, consequently applying a non linearity training only a scale parameter of each projection ω. Tancik et al. [20] showed that fast convergence and high resolution are enabled in settings using this technique. The latter is due to it learning to produce sinusoidal basis function that are then multiplied by an affine projection to produce an image. The affine projection acts as the spectral coefficients of the image produced.

3.4 Summary

- Neural networks are innate to perform feature engineering, being seen as a way of processing features;
- Convolutional neural networks are still a state-of-the-art operation in computer vision. With the rise of more complicated tasks, non local operations were hypothesized to outperform local ones, however convolutions still outperform these operations in various tasks;
- A simple attention mechanism has been introduced. This operation is widely used in natural language processing task [72, 74];
- Fourier features emulate the sinusoids basis functions of a transform. When followed by an affine projection, which acts as the spectral coefficients, images/kernels of high resolutions are produced.

Part II

The Problem

Chapter 4

Problem Definition

The problem of mapping EEG to fMRI is defined as a regression task between a set of features, $\vec{\mathbf{x}} \in \mathbb{R}^E$, extracted from the EEG signal and an fMRI volume, $\hat{\vec{\mathbf{y}}} \in \mathbb{R}^M$. The main focus of this thesis is in developing a function, F, that performs the mapping from EEG to fMRI, i.e. $F : \mathbb{R}^E \rightarrow \mathbb{R}^M$ yielding $\vec{\mathbf{y}} = F(\vec{\mathbf{x}})$, as illustrated in Figure 4.1. The spatial structures of $\vec{\mathbf{x}}$ and $\vec{\mathbf{y}}$ are described by E and M, respectively, which are of the form $\forall n \in \mathbb{N} : K = K_1 \times \cdots \times K_n$, where n is the number of dimensions and $K_j \in \mathbb{N}$ is the size of the jth dimension. Let n_E and n_M be the number of dimensions of $\vec{\mathbf{x}}$ and $\vec{\mathbf{y}}$, respectively. As we will see in Section 4.1, each dimension is not similar across modalities, i.e. $\forall i \in \{1, \ldots, n_E\}, j \in \{1, \ldots, n_M\} : \sum \mathbb{1}_{E_i = M_j} \ll n_E \wedge \sum \mathbb{1}_{E_i = M_j} \ll n_M$. The latter expresses the structural dissimilarity of the space representations of EEG and fMRI, i.e. for all combinations of dimensions, E_i, M_j, the total number of equal structure, $\sum \mathbb{1}_{E_i = M_j}$, is much less than the total number of dimensions, n_E, n_M, of E and M.

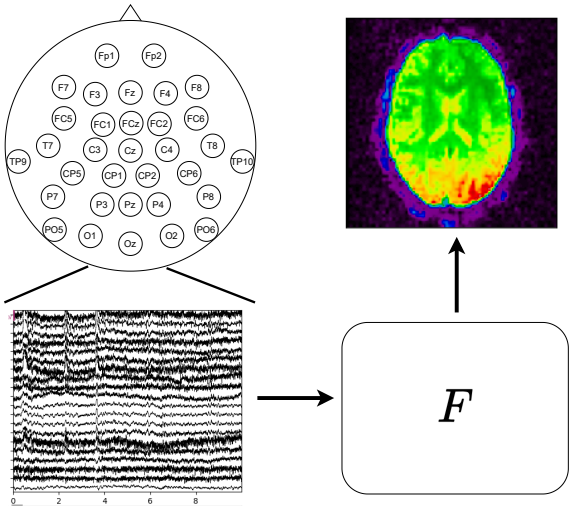


Figure 4.1: A simplistic illustration of the problem. A set of features is extracted from the EEG signal, $\vec{\mathbf{x}}$, and are processed by a function, F, that outputs the corresponding fMRI volume, $\vec{\mathbf{y}}$.

4.1 EEG-fMRI Structural Dissimilarity

When building this function, F, one faces several problems. First, EEG and fMRI evaluate very different brain processes at different time scales [9]. EEG evaluates electrophysiological processes that change at a

rapid time scale. In addition, being a mix of the activity electric field, produced by billions of neurons firing, one does not get the specific activity of one neuron or a small group of neurons. One other fact is that high frequencies are correlated with activity with sources nearby the electrodes, which does not happen with low frequencies that are capable of travelling longer distances and are not as affected as high ones in the presence of obstacles, such as the skull. This mesh, of different frequencies and sources, introduces complexity upon analyzing the source of neuronal activity [9]. To give the reader more context, EEG frequencies typically range from 0.5 Hertz (Hz), depending on the window size used to extract frequency features, to the maximum value the device takes from sampling frequency (modern devices have a sampling rate $\approx$ 1000 Hz). Usually, researchers filter frequencies higher than 60-100Hz and frequencies below 0.1Hz, as it is sufficient to retrieve the information needed for the task at hand and the electrophysiological activity is in the range of 0.1 to 50 Hz [9]. Filtering frequencies is called pass filtering. A high band pass filter removes high frequencies and a low bandpass filter removes low frequencies. These methods are used to remove the mean of the signal and long term drifts in the raw EEG signal. Regarding the fMRI modality, it is generally sampled approximately every 2 seconds, known as the Time of Response, which allows spectral activity to be detected with a sampling rate as high as 0.5Hz. This frequency value is further limited by the physics behind the acquisition of the signal. To record an fMRI, one needs a nuclear magnetic resonance machine. The individual is fit into it and an head coil is placed on his/her head. Two magnetic fields are applied during the recording, one that oscillates and one that is static. These magnetic fields interact with hydrogen mainly and the imaging signal retrieved is taken from the energy that is emitted from these hydrogen protons, that interact with the magnetic fields applied. This emission of energy is correlated with the spin direction of the proton and the direction of the magnetic field. When both are aligned, energy is emitted. Because this (computationally/hardware exhaustive) process has to repeat multiple times, the acquired signal has a low temporal resolution in time. The aforementioned differences, on the temporal resolution of both signals, difficult the mapping between electrophysiological and haemodynamic processes/dynamics. EEG is able to capture processes that vary at a rapid timescale, whereas the frequency sampling of fMRI does not allow the recording of these processes. Similarly, fMRI captures slow varying blood oxygenation (that can equate to slow varying brain processes, in the range of 0.03 to 0.1Hz), something EEG is not able to measure[1] due to those frequencies ($<$1Hz) being in a band that is filtered to remove artifacts. As a result, we note that E and M also differ in the temporal dimension.

EEG spatial resolution is not implicitly defined in its structural representation, E. The spatial information can be retrieved if one knows the exact coordinates of each electrode given a referential set of axes. Distances between channel coordinates cannot be adequately encoded using an Euclidean representation, such as a matrix or a tensor. Instead one needs a geometrical representation, e.g. a graph or even a higher level topographical structure [76]. This kind of representation goes beyond the scope of this work. Nonetheless, E has a dimension that encodes the spatial information and researchers refer to it as the electrodes/channels dimension. It is worth noting, that the principle of locality mentioned in Section 3.1 does not hold for every point of the mentioned dimension of E. Reshifting the attention to fMRI, consider the dimensional space defined in Figure 4.2. The spatial resolution can be naturally encoded in an Euclidean space representation, with a total of 3 dimensions. This is because, it is already present in the nature of

[1] Slow activity can in fact be retrieved in EEG setups with an amplifier called Direct Amplifier, but typically this range of frequencies is mixed with noise and is not of interest to retrieve electrophysiological activity [9]

the fMRI recording this Euclidean property, since an fMRI volume is a set of adjacent 2-dimensional slices taken across the vertical axis, Z. In this representation, the principle of locality holds for every point across the different spatial dimensions.

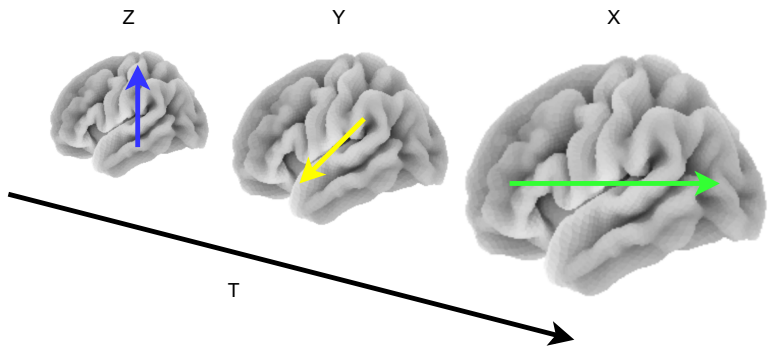


Figure 4.2: An fMRI recording can be represented in a 4-dimensional space, where T refers to the temporal dimension, Z to the vertical spatial dimension, Y to the depth wise spatial dimension, and X to the side wise spatial dimension.

Both modalities, $\vec{\mathbf{x}}$ and $\vec{\mathbf{y}}$, are georeferenced multivariate time series. However, the structural dissimilarity of the EEG and fMRI signals poses as challenge to perform the mapping between the two. In addition to the latter, as previously mentioned, the two modalities measure different processes, being another obstacle in the problem.

4.2 Defining the Mapping Function

In the previous section, the differences between the input, $\vec{\mathbf{x}}$, and the output, $\vec{\mathbf{y}}$, were described. In a regression task, where $F : \mathbb{R}^E \rightarrow \mathbb{R}^M$ is posed as the function that performs the mapping, these differences need to be tackled. In this section, we enumerate the properties F must satisfy to potentially aid the targeted regression task:

- it is present in literature of EEG and fMRI studies [9] that haemodynamic activity is correlated with neural activity. However, not only is the correlation not linear, but also the measured neural activity and blood supply have a drift in time;

- specifically, regarding the spatial representation encoded in E, one needs to introduce an operation that is able to approximate the relation of the different points (that have a topographical relation) in an Euclidean space;

- recall from chapter 3 that densely connected layers are capable of manipulating the size of the output space. Similarly, in section 3.1, we learned that through the arithmetic of convolutions one is also able to map an instance to a desired output space. Because these two operations mutate the structure, they are seen as the main candidates to be used in F, such that E is transformed accordingly, to bridge the structural dissimilarity gap.

4.3 Classification Setting

Given an EEG signal, $\vec{\mathbf{x}}$, we augment it by predicting its corresponding fMRI signal, $\hat{\vec{\mathbf{y}}}$, without a ground truth fMRI pair, $\vec{\mathbf{y}}$. The quality of the synthesis can be addressed with a distance based metric, between the synthesized and ground truth fMRI. On the other hand, withtout the true fMRI, we are unable to objectively assess the quality, only the plausibility of the overall synthesized fMRI state. Nonetheless, consider the available EEG instances to be paired with a one hot encoding of classes $y_c \in \{0,1\}^C : \sum_i y_{c,i} = 1$, which specify a characteristic of the individual (e.g., individual with a pathology or healthy), we may evaluate the quality of the predicted fMRI signal in a decision making setting. The latter can be mathematically defined as the negative log likelihood, $\mathcal{L}_C = -y_c \times log(f_C(F(\vec{\mathbf{x}}))) = -y_c \times log(f_C(\hat{\vec{\mathbf{y}}}))$, where $f_C : \mathbb{R}^M \rightarrow \Delta^C$ is a classifier and Δ^C the simplex of size C meaning $\sum_i p_i = 1, \forall p \in \Delta^C$.

4.4 Summary

- The problem is formulated as a regression task, that maps $\vec{\mathbf{x}} \in \mathbb{R}^E$ to an estimate of $\vec{\mathbf{y}} \in \mathbb{R}^M$, according to a function $F : \mathbb{R}^E \rightarrow \mathbb{R}^M$;
- In addition to differences in structure, the content captured by EEG and fMRI also differs, i.e. they encode different brain dynamics. More specifically, fMRI is not able to measure the processes captured in an EEG recording, and the contrary is also true as low frequencies are filtered from EEG for artifact removal;
- The structural dissimilarity of the encoded space representations, E and M, poses a challenge to the formulation and learning of F;
- The mapping function, F, requires structure manipulation for capturing topographical relations in the Euclidean space, and extrapolation of non-linear fMRI dynamics.

Chapter 5

Experimental Setting

This chapter describes the experimental setting done for the validation of the computational contributions proposed along the research project. First, the datasets are described in Section 5.1. Following, evaluation metrics are introduced in Section 5.2.

5.1 Datasets

A total of five datasets are considered. These datasets will be used in the experiments of this thesis. Four datasets have simultaneous EEG and fMRI recordings and one dataset has EEG recordings of healthy controls and pathology individuals. The simultaneous EEG and fMRI datasets are:

- NODDI dataset described in section 5.1.1;
- Oddball dataset described in section 5.1.2;
- CN-EPFL dataset described in section 5.1.3;
- and CHUR-Xp2 dataset described in section 5.1.4.

The first dataset considers a resting state monitoring protocol, while the last three datasets were recorded in a task based experiment. For all the datasets, from above, we considered a 80/20 partition for training and test data. Regarding the EEG-only data for decision making purposes:

- Fribourg dataset described in section 5.1.5.

All the datasets considered are publicly available and incorporate experiments of published studies.

5.1.1 NODDI

The NODDI dataset [77, 78] contains 10 individuals, with an average age of 32.84 ± 8.13 years. Simultaneous EEG and fMRI recordings were acquired considering a resting state with eyes open (fixating a point). The EEG was recorded at 1000 Hz with a total of 64 channels, arranged in line with the modified combinatorial nomenclature [79]. The fMRI acquisition, using a T2-weighted gradient-echo Echo-planar imaging (EPI) sequence, was performed with: 300 volumes, TR of 2160 ms, TE of 30 ms, 30 slices with 3.0 millimeters (mm), voxel size of $3.3 \times 3.3 \times 4.0$ mm and a field of view of $210 \times 210 \times 120$ mm. For a

more detailed description we refer the reader to Deligianni et al. [77] and Deligianni et al. [78]. The dataset is publicly available to download[1]. Each individual's recording is divided into 24 equally sized time series of fMRI volumes. Each time series is 28 seconds long and resampled to a 2 second period. The *training set* is composed of 8 individuals and the *test set* of 2 individuals. Additionally, we chose this dataset for a demonstration of our work and made a compact version of it that is available for download[2].

5.1.2 Oddball

The Oddball dataset [80–82] contains 10 individuals. Simultaneous EEG-fMRI recordings were performed while the subjects laid down. Stimuli of auditory and visual nature were given to the subjects, which makes this a stimuli based dataset. The EEG was recorded at 1000 Hz with a total of 49 channels. The fMRI acquisition was made with a 3T Philips Achieva MR Scanner with: single channel send and receive head coil, EPI sequence, 170 TRs per run with a TR of 2000 ms, a TE of 25 ms, a voxel size of $3 \times 3 \times 4$ mm and 32 slices with no slice gap. For a more detailed description of the dataset please refer to Walz et al. [80]. The dataset is publicly available to download.[3] Each individual recording is divided into 12 equally sized time series of fMRI volumes, each time series is 28 seconds long, sampled at 2 seconds period. The *training* set is composed of 8 individuals and the *test set* 2 individuals.

5.1.3 CN-EPFL

The CN-EPFL dataset [83] was recorded by the Center for Neuroprosthetics of the École Polytechnique Fédérale de Lausanne (CN-EPFL). A total of 25 individuals (12 females and 13 males, with mean age of $\approx$24 years old) took part in the experiment: the setting consisted on task based recording session. The individuals registered their confidence of decisions: made by them; or decisions they observed. The participants did not have any diagnosed neuronal disorder at the time. EEG recordings were done at 5000 Hz using a 63 channel setup. fMRI was recorded at 3T using a Prisma Siemens scanner with a 32-channel head coil. Data was retrieved using an echo-planar imaging sequence with TR of 1280 milliseconds (ms), TE of 31 ms, and a flip angle of 64 degrees. A total of 64 slices were acquired with $2 \times 2 \times 2$mm voxel size and a field of view of 215 mm. Structural data was also acquired with TR of 2300 ms, TE of 2.32 ms and a field of view of 8 degrees. From the 25 individuals only 20 were considered due to artifacts present in fMRI and/or EEG recordings. Of these 20, the *training set* is composed of 16 and the *test set* 4 individuals. The dataset is publicly available to download[4].

5.1.4 CHUR-Xp2

The CHUR-Xp2 [84] is a simultaneous EEG and fMRI dataset that contains recordings for a total of 17 individuals. The recording sessions took place at the Centre Hospitalier Universitaire de Rennes. Individuals were subject to a motor imagery task, where they first imagined a movement and then followed three repetitions with neurofeedback support for the imagined movement. EEG was set with 64 electrodes, distributed in accordance with the 10-10 system, and sampled at 5000 Hz. A low pass filter of 200Hz was applied at

[1] `https://osf.io/94c5t/`
[2] `https://web.ist.utl.pt/ist180980/eeg_to_fmri/datasets/01.zip`
[3] `https://legacy.openfmri.org/dataset/ds000116/`
[4] `https://openneuro.org/datasets/ds002158`

preprocessing time. The fMRI acquisition was made with 1 second TR, 23 millisecond TE, voxels of size $2 \times 2 \times 4$ millimeters, resulting in a $108 \times 108 \times 16$ volume size. The *training set* consists of 13 individuals and the *test set* 4 individuals. The dataset is publicly available to download[5].

5.1.5 Fribourg

The Fribourg dataset contains EEG recordings of 43 individuals. Of these 24 were healthy controls and 19 were diagnosed with schizophrenia. The recordings were setup with a task, where each individual played a game. The EEG was set with 128 electrodes, distributed according to the 10-10 system. The sampling rate was 2048 Hz. The *training* and *test sets* were setup according to a leave-one-individual-out schema (see sections 5.2.3 and 9.3.1). The dataset is publicly available to download[6]. Similar to the NODDI dataset, we also chose this dataset for a demonstration and a compact version is available for download[7].

5.2 Evaluation

In this section, quantitative and qualitative metrics to evaluate the efficacy of synthesized (predicted) fMRI signals are introduced. Three main types of evaluation are described: synthesis accuracy (Section 5.2.1), qualitative metrics (Section 5.2.2) and decision support (Section 5.2.3).

5.2.1 Synthesis Accuracy

Quantitative metrics analyze synthesized signals of the form $\hat{\vec{\mathbf{y}}} = F(\vec{\mathbf{x}})$. Consider N and $|M|$ as the number of instances and features (of $\vec{\mathbf{y}}$), respectively. Let $\vec{\mathbf{y}}$ be the ground truth (of fMRI signals), then we formulate the errors, e, such that $e_{\text{metric}}(\vec{\mathbf{y}}, \hat{\vec{\mathbf{y}}}) = \frac{1}{N}\sum_i^N e_{\text{metric}}(\vec{\mathbf{y}}_i, \hat{\vec{\mathbf{y}}}_i)$, i.e. the mean of all errors. The rest of this section is dedicated to the instantiation of selected metrics, e_{metric}.

Root Mean Squared Error (RMSE)

The RMSE provides the root of the mean of the squared residuals,

$$e_{\text{RMSE}}(\vec{\mathbf{y}}_i, \hat{\vec{\mathbf{y}}}_i) = \sqrt{\frac{1}{|M|}\sum_j^{|M|}\left(\vec{\mathbf{y}}_{ij} - \hat{\vec{\mathbf{y}}}_{ij}\right)^2}. \tag{5.1}$$

Structural Similarity Index Measure (SSIM)

The SSIM [85] measures the quality of a signal against its noise free version. In our setting, the noise free version is the ground truth, $\vec{\mathbf{y}}$, and the signal subject to comparison is our prediction, $\hat{\vec{\mathbf{y}}}$,

$$e_{\text{SSIM}}(\vec{\mathbf{y}}_i, \hat{\vec{\mathbf{y}}}_i) = \frac{(2\mu_{\vec{\mathbf{y}}_i}\mu_{\hat{\vec{\mathbf{y}}}_i} + c_1)(2\sigma_{\vec{\mathbf{y}}_i\hat{\vec{\mathbf{y}}}_i} + c_2)}{(\mu^2_{\vec{\mathbf{y}}_i} + \mu^2_{\hat{\vec{\mathbf{y}}}_i} + c_1)(\sigma^2_{\vec{\mathbf{y}}_i} + \sigma^2_{\hat{\vec{\mathbf{y}}}_i} + c_2)}, \tag{5.2}$$

with $\mu_{\vec{\mathbf{y}}_i}$ being the average value of $\vec{\mathbf{y}}_i$, $\mu_{\hat{\vec{\mathbf{y}}}_i}$ the average value of $\hat{\vec{\mathbf{y}}}_i$, $\sigma_{\vec{\mathbf{y}}_i}$ the standard deviation of $\vec{\mathbf{y}}_i$, $\sigma_{\hat{\vec{\mathbf{y}}}_i}$ the standard deviation of $\hat{\vec{\mathbf{y}}}_i$ and $\sigma_{\vec{\mathbf{y}}_i\hat{\vec{\mathbf{y}}}_i}$ the covariance between $\vec{\mathbf{y}}_i$ and $\hat{\vec{\mathbf{y}}}_i$. The constants c_1 and c_2 avoid high values for this metric when the denominator is much lower than the numerator. These constants are

[5] `https://openneuro.org/datasets/ds002338`
[6] `https://openneuro.org/datasets/ds004000/`
[7] `web.ist.utl.pt/ist180980/eeg_to_fmri/datasets/ds004000.zip`

defined by parameters, k_1, k_2 and L, usually set to $k_1 = 0.01$, $k_2 = 0.03$ and $L = 2^{p-1}$, with p being the number of bits per pixel, for $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$ [86].

5.2.2 Qualitative

In the previous Section, quantitative metrics were introduced to evaluate the synthesized signal. Although the advantage of these metrics is that they give a quantized value of how close is the prediction against the ground truth, these quantized values do not give a qualitive understanding of what is being synthesized. To tackle this observation, this section covers two main ways of evaluating the quality of a prediction. First, we hypothesize that matching the relevance, of predicted features with the relevance reported in previous studies, may serve as a way of measuring the quality of a signal. And second, we introduce uncertainty as a quality measure. These are used in addition to the manual visual quality inspection. The concept of quality comes from the fact that both approaches are able to be plotted and manually interpreted.

5.2.2.1 Layer-wise Relevance Propagation

Explainability in artificial intelligence has been a growing topic in the last decade with the need of explaining to a user why a complex model, such as a neural network, made a certain decision. For instance, a setting given as an example can be: a doctor providing the rationale behind a specific diagnosis grounded on fMRI patterning, given an fMRI recording. In this section, we describe the Layer-wise Relevance Propagation (LRP) algorithm, first proposed in [87]. For a more detailed description, Montavon et al. [88] give a comprehensive explanation of this method.

Using a set of rules, LRP is able to propagate the output logits of a neural network, layer by layer. These rules use the activations of the neurons of each layer to propagate the *relevance*, to each neuron of the previous layer, through the activation of a neuron in the next layer. Consider a chained structured neural network, L, that has l layers, represented by $L = \{L_1, \ldots, L_l\}$, being L_1 the input layer and L_l the output layer. Each layer has an arbitrary number of neurons. Let each layer be densely connected, for the sake of simplicity. A neuron, j, has a relevance, R_j, associated to it. The relevance of all the neurons of the output layer is by default the output logits. The relevance of all layers, $L_i : 1 \leq i < l \wedge i \in \mathbb{N}\backslash 0$, is computed by the backpropagation of relevances using the rule stated in Equation 5.3. Let j be a hidden neuron of layer L_i, and k a neuron of layer L_{i+1}, then the relevance of j, R_j, can be computed as

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_j a_j w_{jk}} R_k. \tag{5.3}$$

This process can be quite time consuming for wide and/or deep networks. Fortunately, Montavon et al. [88] provide the necessary procedures to compute this propagation with an auto differentiation software, available through Tensorflow [89][8].

As mentioned before, explainability methods are becoming popular with the need of knowing the "why a complex model is making a certain decision". Böhle et al. [90] is an interesting follow up work to [87], that applied this method to extract the regions with high relevance in a classification task of Alzheimer's disease. This method is used, in the scope of our work, to check the relevant features of the EEG, that contributed to

[8] Implementation of LRP in tensorflow available at `https://stackoverflow.com/questions/62555103/layer-wise-propagationlrp-in-keras-neural-network/69836885`.

a certain fMRI volume prediction. We hypothesize that if the reported input relevance goes in accordance with the EEG correlates of fMRI from related work (e.g., Rojas et al. [91]), then the synthesized signal is retrieving fMRI functions. This is done using the metrics introduced in Section 5.2.1 and the relevances of the input for the synthesized signal.

5.2.2.2 Aleatoric and Epistemic Uncertainty

In this section, the notion of quantifying uncertainty is introduced. Probabilistic models are preferred over deterministic ones [19], due to their better modulation of the stochasticity present in the real world. For motivation, consider the example of a classification task with a total of c classes. If the model is given an input that does not properly fit into the classes the model has been trained on, it may misclassify it. However, probabilistic models enable the user to quantify uncertainty measures that tells us how sure the model is of a certain prediction. In a perfect setting, the probabilistic model gives a high uncertainty quantity to the instance that does not belong to any of the classes present during the training session. With such a setting, one may introduce the notion of having an extra class that corresponds to the *reject* label, where the model is able to not label an instance if the level of uncertainty is high. In many settings, the *reject* class is useful. An illustrative case is a medical setting where the set of available treatments for prescription might be considerably dangerous for a patient.

Two types of uncertainty are introduced: *epistemic* and *aleatoric* uncertainty. Epistemic uncertainty refers to the uncertainty of a model to make a certain prediction, specifically what is the probability of the model parametrized by θ making *that* prediction for *that* instance. Aleatoric uncertainty captures the noise of the observations, i.e. given two instances of the same class, it measures whether the model returns the same prediction. Aleatoric uncertainty captures data noise and epistemic captures the uncertainty of the model. To measure aleatoric uncertainty, one places a prior on the outputs of the model. To measure epistemic uncertainty one places a prior over the parameters, θ, of the model. These two measures are proposed to serve as quality metrics of the synthesized signal. When evaluating a prediction, the following questions are asked:

- What is the level of *epistemic* uncertainty for the original signal and the synthesized one?
- Does the proposed process enable the synthesized signal to have a lower level of uncertainty?

5.2.3 Decision Support

Ultimately, the objective of this thesis is to provide informative diagnostics at lower cost and higher accessibility than diagnostics with MRI requirements. To that end, we work with EEG recordings and map each recording to their corresponding fMRI. The hypothesis is that learning at the fMRI data space provides an alternative structure and additional information relevant for interpretation and decision making. Therefore, the final product is an fMRI informed EEG modality. An fMRI informed EEG can be a simple fMRI volume prediction or an augmented view of the EEG (e.g., concatenation of spectral EEG features with fMRI predicted features), as long its source is only an EEG instance. The fMRI prediction needs, however, two learning phases: 1) learn the fMRI style, from corresponding EEG and fMRI pairs; and 2) adapt the learned fMRI style to EEG only data from a set of annotated EEG instances. While (1) is fulfilled with simultaneous

EEG and fMRI datasets, (2) needs EEG recordings paired with labels. Datasets containing EEG recordings and labels resemble a real life application of this methodology, since they do not have fMRI pairs.

We define the setting as EEG data pairs in a set of individuals $S = \{s_1, \ldots, s_{|S|}\}$. Each individual, $s_i = (\vec{\mathbf{x}}_i, y_i), \forall i \in \{1, \ldots, |S|\}$, has an EEG instance $\vec{\mathbf{x}}_i$ associated with a label $y_i \in \{0, 1\}^C$, where C denotes the number of groups in S. Let F be trained on simultaneous EEG and fMRI data, where θ denotes the parameters of F. After being trained to approximate an fMRI volume, such that $F(\vec{\mathbf{x}}; \theta) \approx \vec{\mathbf{y}}$, the goal is to apply $F(\vec{\mathbf{x}}; \theta)$ in a classification setting without $\vec{\mathbf{y}}$. This is done by first processing the predicted fMRI with a classifier, $f_C : \mathbb{R}^M \rightarrow \Delta^C$, parameterized by θ_C and then minimizing the objective

$$\mathcal{L}_C(s) = \mathcal{L}_C(\vec{\mathbf{x}}, y) = -y \times log\left(f_C(F(\vec{\mathbf{x}}; \theta_C); \theta)\right), \tag{5.4}$$

w.r.t. θ_C. The dataset considered for the experiments in the decision support setting is the Fribourg dataset (see section 5.1.5). In this setting, we consider a leave-one-individual-out cross validation schema, where each individual is left out as a fold for testing. This can be represented in terms of accuracy as

$$\frac{1}{|S|} \sum_{i}^{|S|} 1_{y_i == f_C\left(\vec{\mathbf{x}}_i; argmin_{\theta_C}(\mathcal{L}_C(S \setminus s_i))\right)}. \tag{5.5}$$

This setting allows us to assess if the synthesized fMRI is able to correctly classify groups of individuals, showing its applicability in health care settings. Not only should the fMRI prediction be explained (see section 5.2.2.1), as well as provide a quantification of its prediction risk (see section 5.2.2.2), but also should its decision support applicability be assessed. All of these qualitative evaluations give a thorough report on if such a setting/methodology could one day be applied in a real life setting.

5.3 Summary

- A total of four simultaneous EEG-fMRI datasets are considered to validate our work. All of them are made publicly available and take part in the experiments of related studies. The NODDI dataset was recorded in a resting state setting, whereas the Oddball, CN-EPFL and CHUR-Xp2 datasets are task based ones;
- Regarding evaluation metrics, traditional synthesis accuracy measures used in regression tasks are covered;
- In addition to manual human model inspection, quantification of uncertainty and explainability methods can also evaluate quality;
- A decision support pipeline assesses the role of the synthesized signal in a clinical decision context using a classification task, by hypothesizing that $\hat{\vec{\mathbf{y}}} = F(\vec{\mathbf{x}})$ contains information to correctly recognize groups of individuals. This validation is done with the Fribourg dataset.

Part III

The Proposed Solution

Chapter 6

Generation of Neural Architectures

In machine learning, hyperparameter search is a common and pivotal step that allows the optimization of the parameters of a method [92]. In addition, when working with neural networks, there must be specific network architectures yielding properties of interest (number of hidden units, kernel size, etc) for a certain task (classification, regression, etc). In this chapter, a neural architecture (NA) generation algorithm is proposed. The method is developed for a specific use case: a regression task from an input to an output space; this method is hypothesized to generate neural architectures that automatically bridge the structural dissimilarity gap between two modalities (EEG and fMRI in the case of this project).

Often properties that define the structure of a network, such as the number of layers, kernel and stride sizes, are included in tuning algorithms (grid search [92], random search [93], Bayesian optimization [94]). However, these algorithms either explore invalid permutations of parameters or do not explore the full range of the search space. Take convolutional layers [95] for instance. In their simplified definition, they are defined by a kernel size and a stride size. The output of a convolutional layer varies with different values for the kernel and stride, which are hyperparameters that highly impact the performance (either in accuracy or resource usage) of a neural network [96]. The same goes for max-pooling [97], average-pooling [97], locally connected [98], transposed convolution [95] and others that use kernel and stride like parameters. To tackle this problem, the hyperparameter search can be divided in two steps: 1) perform a neural architecture search (NAS), 2) run a search algorithm for the hyperparameters. Recently, there have been advances in NAS [99–102], where first neural network architectures are collected, either scrapped from previous works [103] or manually built by the user [104], and then subjected to a validation process to find the best architecture. All these methods have a common liability: the defined samples, assembling a NAS space, of architectures are limited and biased.

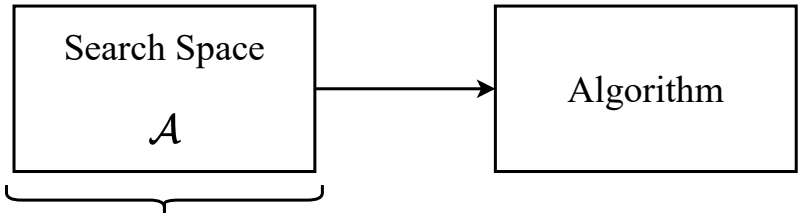


Figure 6.1: A NAS algorithm typically consists on exploring a defined space of neural architectures using a search algorithm. The approach, introduced in this manuscript, defines the search space, $\mathcal{A}$, for a NAS algorithm. This NAS algorithm flow is based on the one introduced in [104].

We propose that instead NAs[1] are generated using an algorithm capable of producing a non-biased and uniform architecture space. *Non-biased*, as the search space is built automatically by an algorithm as opposed to a human, and *uniform* because of the characteristics of the approach proposed (more details in Section 6.4). This qualities are achieved using constraint programming, more specifically formulating NA generation as a Boolean Satisfiability (SAT) problem. Given an input and output space, a chained structured neural architecture can be defined by a formula, representing the kernel and stride of each layer as variables. A satisfiable assignment to these variables defines an NA.

We find that the number of NAs that satisfy the encoded formula increases exponentially with the number of layers and dimensions (SAT formulas have an exponential solution space growth as the number of variables increase [105]). As such, logical XOR constraints, theoretically proven to restrict the solution space of a SAT formula and allow for a near-uniform sampling [106], are used.

The contributions of this work are:

- To the best of our knowledge, the proposed approach is the first to automatically generate NAs without an induced human bias, introducing a new view and possibility to discover different networks by formulating an NA search space;
- Use of constraint programming to formulate a neural architecture encoding.

This chapter is organized as follows: Section 6.1 relates the work to other studies. Section 6.2 provides a description of the problem at hand. Section 6.3 defines the encoding of the formula that describes a chained neural network composed of operations that respect the convolution airthmetic. In Section 6.4, we provide a description of the method used to allow for a near uniform sampling of the encoded formula. Section 6.5 describes how the evaluation of the generated NAs was done. In Section 6.6, the results are shown. In Section 6.7, concluding remarks of the work are provided. In Section 6.8 summarizes of the contents introduced in this chapter.

6.1 Related Work

Studies that consider NAS for hyperparameterization are often seen. In this section, a description of some of the most recent works on NAS is given. Sukthanker et al. [107] define a continuous space of semi positive definite matrices, where a neural network layer/operation is represented as a point. The defined space respects defined rules in order to associate a layer to it. The chosen operation is retrieved by discretization (taking the argmin of all operations) to the closest (defined distance) defined operation in that space. All candidate architectures are optimized using the relaxation of the *softmax* [18]. Zhao et al. [108] explore the use of supernets in NAS, which consists on using a neural architecture that encompasses a representation of all networks in a search space. By picking multiple supernets from a search space, one can apply a search algorithm on this subset of architectures, which automatically takes less time than exploring the entire space. Although the methodology reduces by many orders the search, there is still a bottleneck to where/how are the architectures from/generated[2]. Nayman et al. [109] perform search in a space configured with different convolutions, i.e., different kernels (strides are fixed). However, they do not explore all the

[1] Please note that NAs stands for Neural Architectures, whereas NAS stands for Neural Architecture Search
[2] The NAS-Bench-101 was used as the neural architecture search space.

combinations, which in consequence do not account with different dimension exploration (always apply a square shaped convolution). In terms of search algorithm, they propose a space of one hot variables, that once optimized, the optimal architecture is drawn by taking the argmax. The optimization is done using gradient descent methods on the softmax relaxed operations, that in consequence act as probability distributions of operations. At each step of the optimization procedure, an architecture is sampled from the distributions, by gathering an operation from each vector. Kandasamy et al. [103] approach the problem of NAS using Bayesian optimization [94], which is feasible through the definition of an acquisition function. This function chooses the network that is most similar to the best network acquired, up to certain timestep of the optimization process. Grathwohl et al. [110] much like Liu et al. [18], explore the concept of NAS with gradient based optimization. Differing from these contributions, our work futher incorporates different convolutional structures (kernel and stride size).

6.2 Problem Description

Consider an input space, $\mathbb{R}^I$, and output space, $\mathbb{R}^O$. I and O are K-dimensional spaces, that is $I = I^{(1)} \times \cdots \times I^{(K)}$, similarly $O = O^{(1)} \times \cdots \times O^{(K)}$, with $K \in \mathbb{N}\backslash\{0\}$. The k-th dimension of I is referred to as $I^{(k)}$. Let a convolutional layer, C_l, be a mapping function, f_{C_l}, between two spaces. A convolutional neural network, C, is set of L layers, $C = \{C_1, \ldots, C_L\}$, which is also seen as a mapping function, f_C. The latter is a chain of mapping functions, $f_C : \mathbb{R}^I \to \mathbb{R}^O$, $f_C = f_{C_L}(\ldots f_{C_1}(I)) \ldots) = O$.

6.3 Problem Formalization

Convolutional operations can be applied in 1-dimensional (e.g. signal), 2-dimensional (e.g. images), 3-dimensional (e.g. videos) spaces and so on. For the sake of simplicity, consider two 1-dimensional spaces, $\mathbb{R}^I$ and $\mathbb{R}^O$, which are referred to as *input space* and *output space*, respectively.

The input and output of a convolutional layer[3] can be related with the kernel size, k, and stride, s, as

$$O = \frac{I - k}{s} + 1 \Leftrightarrow I = (O - 1) \times s + k, \tag{6.1}$$

such that $\forall I, O \in \mathbb{N} : I \geq O$ iff $k > 0 \wedge s > 0$. A convolutional neural network (CNN) in its simplest form[4] is a set of $N \in \mathbb{N}$ convolutional layers, $C = \{C_{L_1}, \ldots, C_{L_N}\}$. Every layer is characterised by its kernel size and stride,

$$\forall n \in \{1, \ldots, N\} : C_{L_n} = (k_n, s_n). \tag{6.2}$$

A CNN transformation from an input space, I, to an output space, O, is represented as: $f_C : \mathbb{R}^I \to \mathbb{R}^O$. Similarly, a convolutional layer, $C_{L_n} \in C$, transformation is defined as $f_{C_{L_n}} : \mathbb{R}^{I_i} \to \mathbb{R}^{O_o}$, with $I \preceq I_i \preceq O_o \preceq O$. The network function f_C in its decomposed form respects $f_C(I) = f_{C_{L_n}}(f_{C_{L_{n-1}}}(\ldots(f_{C_{L_1}}(I))) \in \mathbb{R}^O$. If $f_{C_{L_0}} = I \wedge \cdots \wedge f_{C_{L_N}} = O$, this can be represented as a Satisfiability Modulo Theory (SMT) for-

[3] Convolutions have more parameters that are not described in this document, such as number of channels and padding. These parameters are not considered in this description, because they are not being used in the context of my work.

[4] In the machine learning community CNN studies also call networks with more types of layers, in addition to convolutions, CNNs (e.g. networks with convolutional layers and fully connected layers).

mula with

$$\left(f_{C_{L_N}} = O\right) \wedge \bigwedge_{l=1}^{N} \left(f_{C_{L_l}} \leq f_{C_{L_{l-1}}} \wedge k_l > 0 \wedge s_l > 0\right). \tag{6.3}$$

If Equation 6.4 is satisfiable, an SMT solver gives us a set of N tuples of (k, s) characterizing convolutional layers from

$$\left(\frac{f_{C_{L_{N-1}}} - k_N}{s_N} + 1 = O\right) \wedge \bigwedge_{l=1}^{N} \left(\frac{f_{C_{L_{l-1}}} - k_l}{s_l} + 1 \leq f_{C_{L_{l-1}}} \wedge k_l > 0 \wedge s_l > 0\right). \tag{6.4}$$

K-dimensional Setting - to extend the problem to K dimensions one just needs to apply Equation 6.4 to all K dimensions. With $x^{(k)}$ corresponding to the k-th dimension of an instance x in a K-dimensional space, such a setting is represented as

$$H_o = \left(\frac{f^{(k)}_{C_{L_{N-1}}} - k^{(k)}_N}{s^{(k)}_N} + 1 = O^{(k)}\right),$$

$$H_h = \left(\frac{f^{(k)}_{C_{L_{l-1}}} - k^{(k)}_l}{s^{(k)}_l} + 1 \leq f^{(k)}_{C_{L_{l-1}}} \wedge k^{(k)}_l > 0 \wedge s^{(k)}_l > 0\right),$$

$$\bigwedge_{k=1}^{K} H_o \wedge \bigwedge_{l=1}^{N} H_h, \tag{6.5}$$

where H_o refers to an output layer and H_h refers to a hidden layer. An SMT solver would explore a space of infinite satisfiable solutions for Equation 6.5 since $(k, s) = (1, 1)$ characterises a convolutional layer that does not mutate the dimensions from the input to the output. To avoid this, the number of layers to be accounted for is lower, n, and upper, N, bounded.

Pseudo boolean optimization - to enumerate solutions that satisfy an SMT formula with lower and upper bounds for the number of layers, we need to have auxiliary variables that define which constraints are participating, i.e. which lth kernel and stride are eligible for the solution. For that pseudo-Boolean [111] variables, X, are introduced, where $X = \{x_1, \ldots, x_N, x_{N+1}\}$. All $x \in X$ should obey $\forall 2 \leq i \leq N : x_i = 1 \Rightarrow x_{i-1} = 1$ and $\forall 1 \leq i \leq N - 1 : x_i = 0 \Rightarrow x_{i+1} = 0$. The first means if $x_i = 1$ then all the previous layers are activated, therefore it implies $x_{i-1} = 1$ and the same goes for x_{i-1} until x_1. For the second statement, if $x_i = 0$, the ith layer is not activated and does not participate in the solution, then the same goes for x_{i+1} which is 0, until x_N. In addition, $x_{N+1} = 0$ is always true. Every time one wants a solution of l layers, a constraint, c, should be true, where $c = 1 \left[\sum_{i=1}^{N} x_i = l\right]$ is equivalent to $(\forall 1 \leq i \leq l : x_i = 1) \wedge (\forall l < i \leq N : x_i = 0)$ or simply $x_l = 1 \wedge x_{l+1} = 0$. With this we define F as

$$\forall n = 1, \ldots, N : \bigwedge_{k=1}^{K} \left((\neg x_n \vee x_{n+1} \vee H_o) \wedge (\neg x_n \vee \neg x_{n+1} \vee H_h)\right). \tag{6.6}$$

Solution enumeration - to enumerate all solutions that satisfy the SMT formula, one needs to specify the SMT solver to not give us an already given solution again. This is done by adding the solution to the SMT formula in the form of negation. If, for a given x_i, the formula is no longer satisfiable, then $x_{i+1} = 1$, while $i < N$. All the solutions are enumerated if $i = N$ and the formula is no longer satisfiable.

6.4 The Uniform Enumeration Problem

The goal is to generate a limited number of reasonably different architectures that qualitatively represent the space of all possible networks in terms of performance. We hypothesize that similar solutions (in terms of structure) are similar in performance. Therefore, to maximize exploration we want solutions that are maximally heterogeneous. However, F (described in Equation 6.6), when solved by a well known Solver (Z3-Python [112]), follows a bias that makes consecutively enumerated solutions (CES) very similar. For instance, if $n = 2$ and $N = 5$, the enumerated solutions will have the following order: first all solutions of 2 layers will be enumerated, then all solutions of 3 layers will be enumerated, so on and so forth. In addition, F has a exponential growing number of solutions (see Figure 6.5) making $\mathcal{A}$ (recall $\mathcal{A}$ as the set of NAs explored in a NAS algorithm, see Figure 6.1) untractable.

A workaround is to limit the solutions sampled using the natural enumeration bias of a solver. This would cause $\mathcal{A}$ to lack uniformity, which in consequence makes the algorithm that explores $\mathcal{A}$ to have a small exploration trait in the solution space, $\mathcal{S}$, of F. Please note that $\mathcal{S}$ is the set of all the solutions of F, whereas, $\mathcal{A}$ is a subset of $\mathcal{S}$, where the NAS algorithm operates. For instance, consider a solution space $S = \bigcup_{i=1}^{9} s_i$, where s_i, s_{i+1} are highly similar, whereas s_i, s_j, with $j \gg i$, having a higher degree of difference. To enumerate 3 solutions from $\mathcal{S}$, we define two approaches: a *biased* approach, that once a solver finds the first solution, it performs atomic changes to a minimum set of variables to give a different solution, this is the behaviour described in Figure 6.2(a); and a *near-uniform* approach, that uses techniques to uniformly sample solutions from a solution space, as described in Figure 6.2(b). In the rest of this section, we describe the technique that is able to do the latter.

(a) A distribution of 3 solutions among a solution space, following the *biased* enumeration mechanism of a SAT solver.

(b) A *near-uniform* distribution of 3 solutions among a solution space.

Figure 6.2: Comparison between sampling from a solution space according to a biased versus uniformly sampling.

Chakraborty et al. [17] proposed the UniWit algorithm that is able to enumerate uniformly distant solutions. This algorithm uses XOR constraints to restrict the solution space, $\mathcal{S}$, of a formula, F, and randomly chooses a solution on the restricted solution space. The XOR constraints enable the sampling of solutions with a near uniform distribution among $\mathcal{S}$ [106]. UniWit starts by defining a pivot variable, $p = 2|V|^{\frac{1}{k}}$, with V being the set of variables in ϕ and k a constant, recommended by the original work to be set to $k = 2$. It then iteratively involves more variables in XOR constraints that all together make an hash function,

$$h = \bigwedge_{j=1}^{i-l} \left(\left(\bigoplus_{h=1}^{n} (V[j] \wedge a[j+h-1]) \oplus b[j] \right) \Leftrightarrow \alpha[j] \right). \tag{6.7}$$

The variables involved in this equation, $\alpha, a, b \in \{0, 1\}$, are uniformly generated with length $i - l$, $|V| + i - l - 1$, $i - l$, respectively. In addition, auxiliary variables, l, i, are initialized with $\frac{1}{k} log_2 |V|$ and $l - 1$, respectively. UniWit iteratively increments i and generates an hash function, h, until the formula $F \wedge h$ has a solution space, $\mathcal{S}$, with a total number of solutions lesser than p, $|\mathcal{S}| < p$. When $|\mathcal{S}| < p \wedge |S| > 0$ the algorithm stops and returns a random choice from $\mathcal{S}$. If $|\mathcal{S}| < 1$ the algorithm returns a null solution. This

corresponds to sampling one solution from F, therefore to sample M *near-uniform* solutions from F this process is repeated at most M times (UniWit may fail and return a null solution). For more details please refer to the original work [17].

6.5 Evaluation

We experimentally assess three methods:

- *All*: Solutions are enumerated until F is *unsat*, using the internal enumeration bias of Z3-Solver.
- *Limited-S*: Solutions enumerated using the internal enumeration bias of Z3-Solver. Solutions are enumerated until either the S solutions are enumerated or the formula, F, becomes *unsat*.
- *Limited_Uniform-S*: Solutions are enumerated until either the S solutions are enumerated or $F \wedge h$ becomes *unsat* (recall h the hash function defined in Equation 6.7). The order in which solutions are enumerated, ends up following a uniform solution picking among the solution space of F as described in Section 6.4.

In this section, two evaluation methodologies are introduced. One assesses the dissimilarity between CES. The other evaluates the generated search space using a state-of-the-art NAS algorithm [18].

6.5.1 Uniformity evaluation metric

In order to evaluate how similar are CESs, a bit-wise logical xor is used,

$$\text{adj\_}T = \frac{1}{M-1} \sum_{m=0}^{M-1} \left[\sum y_m \oplus y_{m+1} \right]. \tag{6.8}$$

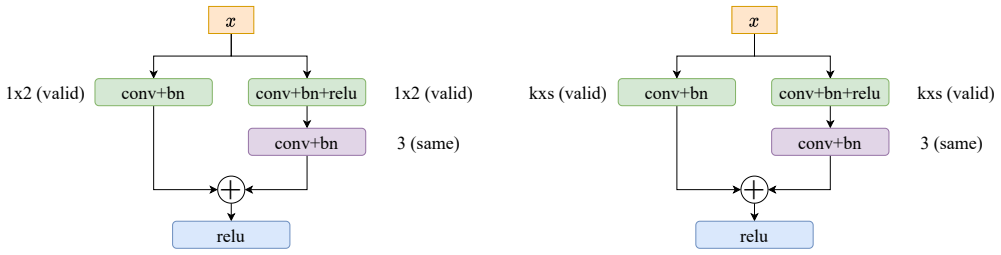
The term $y_m \oplus y_{m+1}$ refers to a bit-wise logical xor. The sum of all positions of the bit-array, $\sum y_m \oplus y_{m+1}$, is the manhattan distance between two vectors, $y_m, y_{m+1} \in \{0, 1\}$. To assess the similarity of CES, the mean distance of all CES (y_m, y_{m+1}) is taken.

6.5.2 Automatic generation based on Resnet-18

Resnet-18 [21] is used to address the quality of the generated architectures from both limited approaches. It is an attractive and well known architecture, and it does not consume as much computational power as its bigger versions. The way this architecture was used for automatic generation is the following: by analyzing the mutation of shapes from layer to layer, one can observe that it goes from shape 8×8 to the final shape 1×1 (specific for input with shape 28×28), being mutated by a total of 3 times corresponding to $8 \times 8 \rightarrow 4 \times 4 \rightarrow 2 \times 2 \rightarrow 1 \times 1$. Resnet-18 has a total of 18 layers, however the shape is only mutated in 3 of them. As such, we specify, in the defined encoding of Equation 6.6, that $n = 3 \wedge N = 3 \wedge I = 8 \times 8 \wedge O = 1 \times 1$. Given a limited number of solutions, the kernel and stride values $\forall l \in [0, N] : k_l, s_l$ are used to obtain a modification of the Resnet-18.

The changes to the default Resnet block are shown in Figure 6.3. The input, $\vec{x}$, is split in two flows, with one being processed by a convolution with $k = 1 \wedge s = 2$ followed by a convolution with $k = 3 \wedge s = 1$

that keeps the dimensions of the input (*same* padding scheme [95]) and the other only being processed by a convolution with $k = 1 \wedge s = 2$. Each convolution is followed by either a *bn* layer (stands for Batch Normalization [113]) or *relu* activation (Rectified Linear Unit [114]) or both, with *relu* always following *bn* and the latter the convolution operation. In the end, both flows are joined in an addition operation followed by a *relu* activation.



(a) Original Resnet block.

(b) Redefined block to fit our framework.

Figure 6.3: Comparison between the original Resnet block and the redefined block. All of the architectures submitted to evaluation have the redefined block of Figure 6.3(b). It is considered as the original Resnet, a neural network that integrates the redefined block with $k = 1 \wedge s = 2$ in the downsampling blocks.

6.5.3 Assessing the quality of the generated NA space

In addition to comparing the final test accuracy of each generated network, and the original Resnet, it is pertinent to see how they evolve along the training session. As such, we use the method proposed in [18], which consists on treating all of the NAs as a single model that outputs a prediction. Let:

- a_0 be the original Resnet-18;
- $\forall i \in \{1, \ldots, M\} : a_i$ be the set of M generated NAs by one of the limited approaches (*Limited-*M or *Limited_Uniform-*M);
- $\alpha \in \mathbb{R}^{M+1}$ the weights attributed to the NAs predictions.

Then, as defined in [18]

$$\hat{y} = \sum_{i=0}^{S} \frac{e^{\alpha_i}}{\sum_{j=0}^{S} e^{\alpha_j}} a_i(\vec{\mathbf{x}}), \tag{6.9}$$

where $\vec{\mathbf{x}} \in \mathbb{R}^{L \times W}$ is an image with label $y \in \{0, 1\}^C$, L the height of the image, W the width and C the number of classes of the classification problem. Each NA performs a mapping $a_i : \mathbb{R}^{L \times W} \rightarrow \mathbb{R}^C$, whose output is $a_i(\vec{\mathbf{x}})$. With such a setting, Equation 6.9 can be interpreted as the sum of the *softmax* normalized α_i multiplied with $a_i(\vec{\mathbf{x}})$. The *softmax* activation provides a probability value $\frac{e^{\alpha_i}}{\sum_{j=0}^{S} e^{\alpha_j}} \in [0, 1]$ that can be interpreted by how much importance the $a_i(\vec{\mathbf{x}})$ prediction has for the final $\hat{y}$. By optimizing α with a gradient descent method: NAs, with poor predictions, are assigned probability $\frac{e^{\alpha_i}}{\sum_{j=0}^{S} e^{\alpha_j}} \rightarrow 0$, whereas the best NAs have probability $\frac{e^{\alpha_i}}{\sum_{j=0}^{S} e^{\alpha_j}} \rightarrow 1$. The best NA is derived from α, by taking the $argmax(\alpha)$ at the end of the training session.

Due to limited GPU availability, we only take zero order gradients, i.e. given a loss function, $\mathcal{L} : [\mathbb{R}^C, \mathbb{R}^C] \rightarrow \mathbb{R}$, α is optimized with respect to $a_i(\vec{\mathbf{x}})$ by taking gradients $\nabla_\alpha \mathcal{L}(y, \hat{y})$ and the weights, w_i, of each NA, a_i, are optimized with respect to the input, $\vec{\mathbf{x}}$, with gradients $\nabla_{w_i} \mathcal{L}(y, a_i(\vec{\mathbf{x}}))$. The chosen loss function, $\mathcal{L}$, is the negative log likelihood. Figure 6.4 provides an illustration of the flow (forward pass

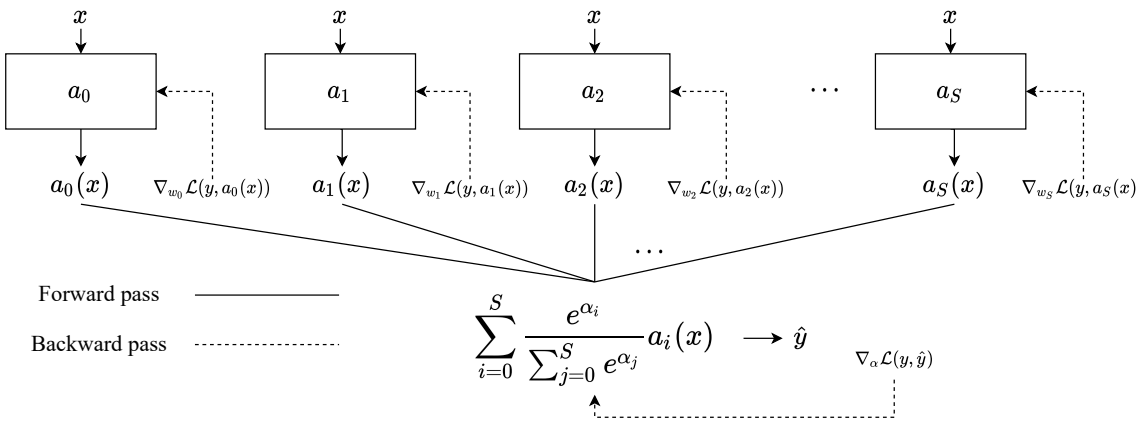


Figure 6.4: All of the networks, $\forall i \in \{0, 1, \ldots, S\} : a_i$, the input, x, is processed and the gradients of each network, a_i, are taken independently with respect to its weights, w_i, such that $\nabla_{w_i}\mathcal{L}(y, a_i(\vec{\mathbf{x}}))$. Following, all of the networks prediction are joined using the softmax activation on α, producing the final prediction, $\hat{y}$. In the final backward pass, using $\hat{y}$ and the same loss, $\mathcal{L}$, the gradients are taken with respect to α, giving $\nabla_{\alpha}\mathcal{L}(y, \hat{y})$

and backward pass) introduced in this Section. Please refer to Appendix B for a description of the code implementation.

It is worth noting that, in contrast with [18], we are performing NAS in a space that has different geometric architectures, instead of different local operations (e.g. check which operation between max pooling and average pooling is best). Please refer to the original work for more details [18].

6.6 Results

In this section, we provide the results from experimentally assessing: in Section 6.6.1 the number of solutions of the formula defined in Section 6.3; in Section 6.6.2 the quality of the CES of *Limited-S* and *Limited_Uniform-S* according to the metric described in Equation 6.5.1; in Section 6.6.3 a performance comparison between *Limited-S* and *Limited_Uniform-S* in a classification task setting; and in Section 6.6.4 an analysis of the time performance related to the approaches defined in the beginning of Section 6.5.

6.6.1 Number solutions

In Figure 6.5, we see how the number of solutions increases with the difference between the input and output, $I - O$. The generated NAs setup a search space for a NAS algorithm, and as with any algorithm, the bigger the search space the harder it is to converge to the global optima, especially when evaluating a state takes a long time, which is the case in NAS. As a consequence, the full solution space, $\mathcal{S}$, can not be fully explored, but a subset, $\mathcal{A}$, with M solutions can.

6.6.2 Quality

Adding to the need of selecting a limited number of solutions from F, the chosen subset of solutions, $\mathcal{A}$, should be representative of $\mathcal{S}$. In this Section, we show the results on two approaches that select a limited number of solutions: *Limited-*M and *Limited_Uniform-*M. The first follows the standard bias of the Z3-Solver and the second uses XOR constraints to promote a uniform sampling of solutions. We compare the

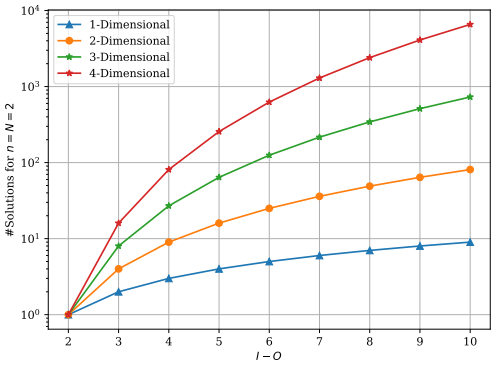


Figure 6.5: Number Solutions *All*. $I - O$ refers to the difference between the input and output. All input, I, dimensions had value of 30, however the exact value of I does not impact the number of solutions, but the $I - O$ does.

two with the evaluation metric described in Equation 6.5.1, that is capable of evaluating how distant are CES. Results are reported in Table 6.1

Table 6.1: Adj_T metric for both limited approaches. Entries are formatted as *Limited-*M*/Limited_Uniform-*M for a better comparison. For all settings, *Limited_Uniform-*M had the bigger Adj_T, producing a uniform like CES.

Number Layers	1D	2D	3D
2	3.375/**4.125**	3.895/**7.632**	5.684/**10.789**
3	3.579/**5.000**	3.632/**10.053**	3.211/**13.842**

Results were gathered for 1, 2 and 3 dimensional settings with $I = 30$ and $O = 20$. The number of layers considered was $N = \{2, 3\}$. More settings were not considered as the average sampling time increased, for a higher number of layers and dimensions, becoming unfeasible to run in real time, as shown in Table 6.5. Table 6.2 shows the same comparison of Table 6.1, but with the addition of pooling layers after each convolution, i.e. after each layer, a layer with $k = 2$ and $s = 1$ is added, emulating a pooling layer. This setting is widely used in computer vision. Inserting already setup layers in the middle of the convolutional layers decreased significantly the solving time. This made feasible in real time the generation of neural networks with as much as 5 convolutional layers, each one of them followed by a pooling layer (giving a total of 10 layers).

Table 6.2: Adj_T metric for both limited approaches, with pooling layers following each convolutional layer. Entries are formatted as *Limited-*M*/Limited_Uniform-*M for a better comparison.

Number Layers	1D	2D	3D
2	**4.286**/3.286	3.750/**7.150**	4.600/**10.350**
3	3.933/**4.267**	3.500/**7.750**	4.450/**11.400**
4	**3.600**/3.200	3.800/**6.000**	5.400/**9.300**
5	0.000/0.000	0.000/0.000	0.000/0.000

We observe that *Limited-*M had a slighty bigger distance of CES than *Limited_Uniform-*M in the 1D case for 2 and 3 layers. In the other cases, 2D and 3D for 2, 3 and 4 layers the *Limited_Uniform-*M showed that it promotes CES with a higher degree of dissimilarity than *Limited-*M. The 0.000 in the 5 layers case are explained by the formula only having one solution in $\mathcal{S}$.

6.6.3 Classification using Resnet as a generation baseline

To address the quality of the generated architectures, the setup, introduced in Section 6.5.2, is used to evaluate the algorithm with the MNIST [115] dataset. The experiments were ran with $M = 20$. Figures A.1 and A.2 provide an illustration of the gathered results. All networks, including the weights, α, were trained with a learning rate of 0.0001, batch size of 512, 0.0001 weight decay and trained for 10 epochs. Recall that the zero order gradient propagation, defined by Liu et al. [18], was used for optimization. Table 6.3 shows the results gathered from the experiments.

Table 6.3: Comparison of the results of Resnet-18, *Limited_Uniform-*M and *Limited-*M.

Number Layers	Dataset	Accuracy	Best	Worst	α deviation
Resnet	MNIST	0.9843	NA	NA	NA
Limited_Uniform-M	MNIST	0.9856	0.9903	**0.9801**	**0.0028**
Limited-M	MNIST	**0.9865**	**0.9909**	0.9835	0.0019

The original Resnet version achieved an accuracy of 0.9843. In comparison, the generated architectures, using the *Limited_Uniform-*M approach, achieved an average accuracy of 0.9856 with 0.0028 standard deviation. The best and worst architecture had accuracies 0.9903 and 0.9801, respectively, meaning a $+0.0060$ and -0.0042 difference against the original Resnet. As for the *Limited-*M approach, an average accuracy of 0.9865 with 0.0019 standard deviation. The best and worst architecture achieved 0.9909 and 0.9835, respectively, with a $+0.0086$ and $+0.0012$ difference against the original Resnet. *Limited_Uniform-*M had higher accuracy standard deviation than *Limited-*M, with 0.0028 against 0.0019. The latter indicates uniformity in performance, however the differences between *Limited_Uniform-*M and *Limited-*M, at least in accuracy, are not statistically significant (using a *t*-test between the two sets of predictions).

Figure 6.6 shows the weight deviation along the epochs, during the training session. The weight deviation against the Resnet is defined as $\sqrt{\frac{\sum_{i=1}^{S}(\alpha_i - \alpha_0)^2}{S-1}}$, α_0 is the weight related to the original Resnet-18 and $\forall i \in \{1, \ldots, S\} : \alpha_i$ the weights associated to the generated architectures. The *Limited_Uniform-*M deviation strictly increased along the epochs, whereas *Limited-*M stayed constant.

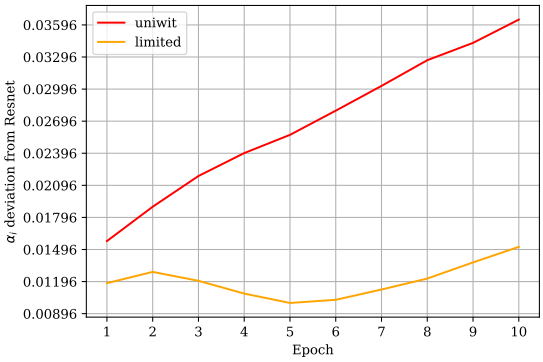


Figure 6.6: α_i deviation from the original Resnet-18. The plot shows that the *Limited_Uniform-*M generated architectures increase deviation with the epochs, which is not seen with the *Limited-*M architectures that stay with the same deviation against the Resnet-18.

6.6.4 Time

Table 6.4 reports the total time it took to gather $M = 20$ solutions, with $I = 30$, $O = 20$ and pooling layers (similar to Table 6.2) for the *Limited_Uniform-M* amd *Limited-M* approaches.

Table 6.4: Total time (seconds) to gather $M = 20$ solutions for both limited approaches, with pooling layers following each convolutional layer. Entries are formatted as *Limited-M*/*Limited_Uniform-M* for a better comparison.

Number Layers	1D	2D	3D
2	**0.262**/0.789	**0.676**/6.293	**0.983**/13.676
3	**0.611**/5.418	**1.821**/99.775	**1.645**/1204.123
4	**1.731**/6.888	**17.338**/508.337	**394.587**/11509.547
5	123.670/**35.412**	497.815/**14.912**	1885.051/**66.540**

Limited-M is much faster than *Limited_Uniform-M*, which was expected due to the solving time bottleneck for each solution sampled, whereas *Limited-M* performs atomic changes at the bottom leafs of the search tree to provide the next solution.

Solving time refers to the time spent solving a formula, ϕ, i.e. the time the solver takes until returning the first solution that satisfies ϕ. As seen in Figure 6.7, the solving time increases exponentially with the increase of dimensions.

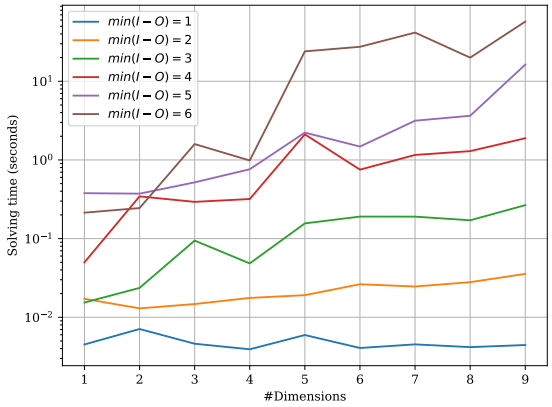


Figure 6.7: Solving time for *All* setting with varying dimensionality.

The higher the difference between the input and output space, the more complex the problem gets and therefore the solving time increases as well (see Figure 6.7). However since all approaches solve the same formula, the solving time is the same for *Limited-M* and a sub-estimate for *Limited_Uniform-M* (the hash function introduced increases the solving time).

Finally, we analyze the *average sampling time* of the *Limited_Uniform-M* approach, which is shown in Table 6.5. The setting is the same that was used for Table 6.2.

Table 6.5: Average sampling time (seconds) of the *Limited_Uniform-M* approach.

Number Layers	1D	2D	3D
2	0.0493	0.255	0.619
3	0.288	4.9219	60.128
4	0.606	25.344	575.392
5	34.252	5.526	50.457

The results show that *Limited_Uniform-M* is not able to generate deep neural networks (networks with

a high number of layers) that have a high gap between the input and output space, $I - O$. However, a neural network does not have to be deep to have a good performance. In fact, Guo et al. [116] show that compressing neural networks not only increases the computation time, but in some cases promotes better efficacy.

6.7 Conclusion

In this chapter, a framework capable of generating NAs automatically, specifically chained neural architectures, is introduced. This framework uses SAT and SMT techniques for this task. Further, due to the dimensionality of the solution space and the enumeration bias present in the Z3-Solver, we used XOR constraints that are known to restrict a solution space and perform a near uniform solution sampling. The results show that the framework is capable of generating NAs that are relatively distant from each other, in other words a stratified and near uniform set of NAs. In terms of quality, the *Limited_Uniform-*M space of architectures was evaluated in the MNIST dataset and it showed uniformity in terms of accuracy when compared with the *Limited-*M approach. This validated the hypothesis that uniform sampling of architectures from the solution space, produces networks that are uniform in terms of performance.

6.8 Summary

- Respecting the arithmetic of convolutions and given an input space, I, and an output space, O, solutions for the kernel, k, and stride, s, are given by an SMT solver. By extending the problem, one can retrieve solutions of multiple layers from a defined encoding, F, culminating in automatic generation of neural architectures;
- The encoding has a number of solutions that increases exponentially with the number of layers and $I - O$ distance. This produces a search space that is not feasible for NAS;
- Through the use of XOR constraints, the solution space of F can be limited and uniform;
- The solution space produced using the XOR constraints, proved to have not only more uniform solutions in terms of neural architecture properties and uniformity in terms of performance, but also increasing weight deviation of the DARTS procedure along the epochs of the training session.

Chapter 7

EEG to fMRI

In the previous chapters we set the foundations that contribute to the targeted end of this thesis: EEG to fMRI synthesis. Automated machine learning (Chapter 6), neural processing and attention techniques (Chapter 3) make the ingredients for such a model. This thesis has so far focused on how the structure dissimilarity can be tackled and chapter 6 presented a method that not only automatically generates chained structured neural architectures, but also optimizes non chained structured like architectures. Consequently, it allows the generated architectures to perform better (overall) than the manually tweaked ones. All this poses the proposed methodology as a natural candidate to tackle the structure dissimilarity between EEG and fMRI. Additionally, a mapping from EEG to fMRI requires filtering information present in the neuronal activity signal (EEG), which is necessary under the noise and spatial organization of this modality. We hypothesize that an attention mechanism enables the *reorganization* of the electrode dimension and filters which electrodes better predict haemodynamics.

First, the type of features extracted from each modality, EEG and fMRI, are described in Section 7.1. In Section 7.2, the structural dissimilarity problem is tackled with the use of the framework introduced in Chapter 6. Section 7.6 provides a compact summary of the contents presented in this chapter.

7.1 What feature setup do we need?

An electroencephalogram (EEG), in its raw form, consists in a mutivariate time series of electrophysiological activity recorded at different scalp positions (electrodes). It is hypothesized that the extraction of frequency domain features, such as taking the short time Fourier transform (STFT) with a defined window of t_{STFT} seconds, can bridge neuronal activity with haemodynamical response. fMRI, highly dimensional yet organized in a *spatially interpretable structure*, is the target modallity of this work. The set of volumes that compose an fMRI recording are not processed, i.e. no further feature extraction is applied. As mentioned in Chapter 4, the structure of the EEG signal is characterized by E. Every dataset described in Section 5.1 provides simultaneous EEG-fMRI recordings, such that $E = E_1 \times E_2$, where E_1 represents the number of channels and E_2 the duration of the recording, a proxy for the number of samples to be used in the learning process. This structure, E, is inevitably mutated when applying the STFT to extract frequency features, this is illustrated in Figure 7.1.

The new structure of the set of extracted features is denoted as $E' = E'_1 \times E'_2 \times E'_3$, with E'_1 denoting

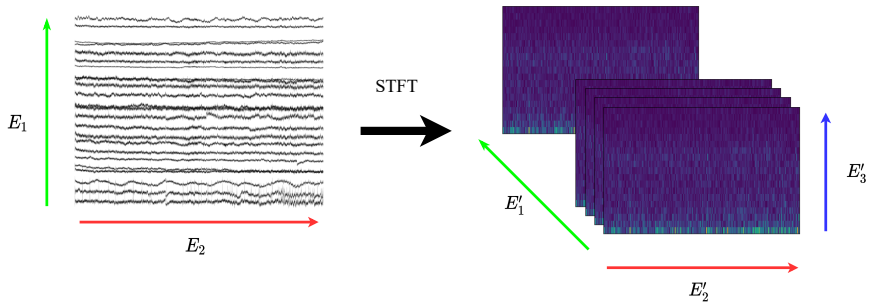


Figure 7.1: EEG frequency feature extraction illustration using the STFT, which mutates the original structure of the feature space.

the number of channels, E'_2 the temporal dimension (number of sliding windows for the STFT), and E'_3 the additional dimension, which is the frequency dimension (spectrum bands). An instance $\vec{x} \in \mathbb{R}^{E'}$ is interpreted as a collection of a multivariate time series. fMRI, on the other end, is structurally characterized

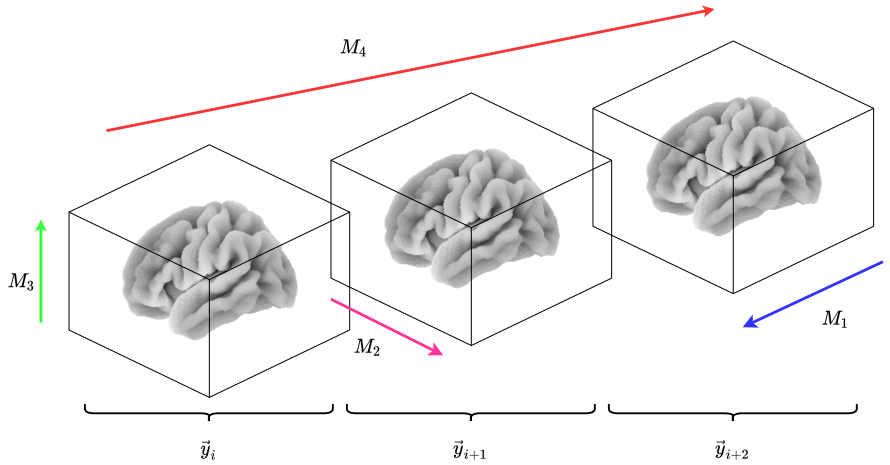


Figure 7.2: Structural nature of an fMRI recording and its dimensions.

by $M = M_1 \times M_2 \times M_3 \times M_4$. With M_1, M_2 and M_3 being the x-axis, y-axis and z-axis directions, respectively, specifying the spatial resolution and therefore $\prod_{i=1}^{3} M_i$ the number of voxels in an fMRI volume. M_4 denotes the temporal dimension, specifying the number of volumes recorded. The set of features gathered, used to represent an fMRI instance, are all of the voxels of **a single fMRI volume**. Please note, that an instance is only an fMRI volume, characterized by $M' = M_1 \times M_2 \times M_3$, and not a set of volumes, whereas in the case of EEG, the temporal dimension is considered for an instance. Figure 7.2 illustrates what is an fMRI instance.

In sum, the EEG space, $\mathbb{R}^{E'}$, and the fMRI space, $\mathbb{R}^{M'}$, form the input and output spaces for the targeted regression task. This is treated as a supervised task where each point observed in the EEG space, $\vec{\mathbf{x}} \in \mathbb{R}^{E'}$, has a pair in the target space, $\vec{\mathbf{y}} \in \mathbb{R}^{M'}$. The pairing between $\vec{\mathbf{x}}$ and $\vec{\mathbf{y}}$ is setup according to the estimation made by Liao et al. [117], claiming that the neuronal activity recorded is reflected in the haemodynamical signal with a delay of $t_{b_s} \approx [5.4, 6]$ seconds. The $\vec{\mathbf{x}}$ and $\vec{\mathbf{y}}$ pairs should therefore have a shift of t_{b_s} seconds, such that $\vec{\mathbf{x}}$ sampled at time t_e corresponds to $\vec{\mathbf{y}}$ sampled at time $t_b = t_e + t_{b_s}$. This is done by manipulating the time dimension of both E' and M, taking into account the size of the time window of the STFT, t_{STFT}, and the time response (TR) of the fMRI recording session. In addition, there is the need to specify a t_{STFT} large enough to adequately decode the lower frequency bands from the EEG signal, without breaking the

natural temporal resolution of the EEG when compared to the fMRI.

7.2 Methodology

In the previous section, the feature space of each modality was introduced and one of the main takeaways is that the feature spaces show structural and representational dissimilarity. In accordance, the methodology described in Chapter 6 is now put in perspective to guide the solution this problem. First, a latent space to represent an fMRI instance is discussed. Following, we represent automatic generation of neural architectures, to bridge the structural gap between EEG and fMRI. This gives us the setting to introduce the operations that together serve as the transfer function from EEG to fMRI, namely attention (see section 7.2.3), Fourier features (see section 7.2.4) and latent style induction (see section 7.2.6). Additionally, we provide a full description of the neural flow, coming from the EEG and fMRI inputs, to the synthesized fMRI ouput (see section 7.2.5).

7.2.1 Discovering the latent space

Both an EEG and fMRI instance have a feature space represented by **three dimensions**. This allows one to maintain the three dimensional trait, and use local and non local operations that follow the convolution arithmetic principles to map both modalities to a similar space. We start by proposing the optimization of latent three dimensional space, to encode an fMRI instance. As such, an encoder decoder architecture fits the purpose, with the encoder composed of convolutional operations along with a decoder. See Figure 7.3 for an illustration of the encoder mechanism. The decoder will not integrate local operations, since non local operations have overtaken local ones in transcription tasks [72].

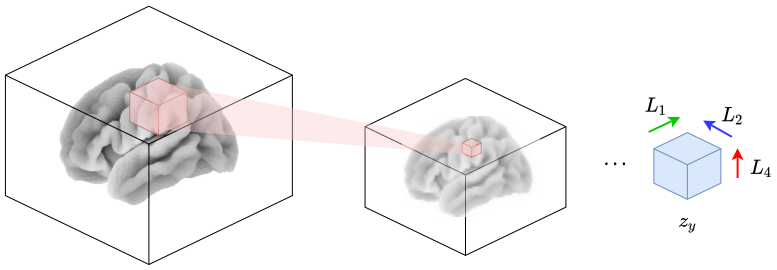


Figure 7.3: The encoder maps an fMRI instance, $\vec{\mathbf{y}}$, to the latent space, $\mathbb{R}^L$.

The encoder projects to a space, $\mathbb{R}^{M'} \rightarrow \mathbb{R}^L$, defined by $L = L_1 \times L_2 \times L_3$. In order to preserve structure, the following constraints are applied:

- $\forall i \in \{1, 2, 3\} : L_i > 1$;
- $\forall i, j \in \{1, 2, 3\} : L_i = L_j \wedge L_i < E'_i \wedge L_i < M_i$;

The first constraint refers to specifying at least 2 axes per dimension, in other words it maintains the multidimensionality trait. The second point, specifies that not only the latent dimension, L, takes the shape of a square, but also that all of its dimensions must be lower than the EEG, E', and fMRI, M', feature space dimensions (dimensionality reduction). A Bayesian optimization (BO) [94] hyperparameter search is hypothesized to discover, among other hyperparameters, the optimal latent dimension size L_i^*. Once the latent dimension is specified, the methodology of Chapter 6 can be applied.

The ability of resnet blocks to downsample the dimension of a representation (defined by the number of layers/blocks, as well as its kernel and stride sizes) enables its fit to the automatic neural architecture search framework, introduced in the previous chapter. A prerequired step, before generating the kernel sizes, strides sizes and number of layers, is discovering the latent space structure which these set of layers map the EEG and fMRI representations to. To this end, we ran a Bayesian optimization hyperparameter search, for the variable $L_i \in \{4, 6, 7, 8, 15, 20\}$. The optimal value is $L_i^* = 7$. Meaning that we can now generate neural architectures with $I = E' \vee M'$ and $O = L^*$.

7.2.2 Searching for an effective neural architecture

The automatic generation of neural architectures (NAs) method is applied to discover an NA capable of performing the mapping between the two spaces defined by E and L. Recall that, in addition to the structural dissimilarity of EEG and fMRI, one also is faced with the representational dissimilarity associated with the different functional processes recorded by the two modalities. Putting all together, the fMRI encoder,

$$E_y : \mathbb{R}^{M'} \rightarrow \mathbb{R}^L, \tag{7.1}$$

is optimized in a BO hyperparameter search and trained along with a decoder,

$$D_y : \mathbb{R}^L \rightarrow \mathbb{R}^{M'}, \tag{7.2}$$

that maps the latent instance, z_y, back to the approximated original space, $\hat{\mathbf{y}}$. Following, once the L_i is obtained, one runs the NAS approach introduced in the previous chapter and discovers the best NA (EEG encoder) from the generated NAs. The resulting architecture defines the EEG encoder,

$$E_x : \mathbb{R}^{E'} \rightarrow \mathbb{R}^L, \tag{7.3}$$

that maps $\vec{\mathbf{x}}$ to $\vec{\mathbf{z}}_x$, the latent representation.

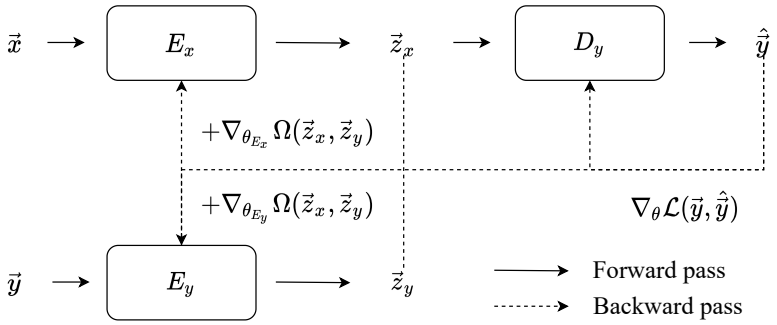


Figure 7.4: Computational flow across the proposed pipeline. The encoder components are trained with a regression loss, $\mathcal{L}$, adding a latent regularization term, Ω, that serves to approximate the latent representation of EEG and fMRI.

Finally, with all the components defined, a joint learning methodology is applied, where the encoders approximate a shared space, $L|\vec{\mathbf{z}}_x = \vec{\mathbf{z}}_y$, and the decoder maps $\vec{z}_x$ to the fMRI space, $\mathbb{R}^{M'}$. The functions E_x, E_y and D_y are parametrized by θ_{E_x}, θ_{E_y} and θ_{D_y}, respectively. For the sake of simplicity, please consider, $\theta = \theta_{E_x} \cup \theta_{E_y} \cup \theta_{D_y}$. Figure 7.4 illustrates the pipeline described. The mapping is defined and

at such a stage one has $\hat{\mathbf{y}}$, with a degree of success measured with the metrics described in 5.2.1.

EEG candidates. Starting with the EEG encoder, with input shape $I = 64 \times 134 \times 10$ and learnt latent space with $O = 7 \times 7 \times 7$, the properties, such as the kernel, stride and number of layers, of the architecture are presented in Table 7.1.

Candidate	Kernel $\times$ Stride ($\bigwedge_1^N k^{(1)}, k^{(2)}, k^{(3)} \times s^{(1)}, s^{(2)}, s^{(3)}$)	N
1	$11, 86, 2 \times 1, 1, 1 \wedge 17, 20, 2 \times 4, 2, 1 \wedge 2, 7, 2 \times 1, 1, 1$	3
2	$7, 37, 2 \times 3, 5, 1 \wedge 7, 7, 2 \times 2, 2, 1$	2
3	$9, 43, 2 \times 1, 2, 1 \wedge 11, 11, 2 \times 1, 2, 1 \wedge 9, 3, 2 \times 5, 2, 1$	3
4	$28, 15, 2 \times 1, 1, 1 \wedge 30, 77, 2 \times 1, 7, 1$	2
5	$7, 19, 2 \times 1, 1, 1 \wedge 20, 23, 2 \times 1, 4, 1 \wedge 23, 16, 2 \times 2, 1, 1$	3
6	$6, 29, 2 \times 1, 1, 1 \wedge 21, 33, 2 \times 1, 4, 1 \wedge 16, 11, 2 \times 3, 1, 1$	3
7	$32, 47, 2 \times 2, 4, 1 \wedge 4, 15, 2 \times 2, 1, 1$	2
8	$9, 16, 2 \times 3, 1, 1 \wedge 5, 2, 2 \times 1, 1, 1 \wedge 6, 81, 2 \times 1, 5, 1$	3
9	$23, 32, 2 \times 1, 1, 1 \wedge 11, 96, 2 \times 5, 1, 1$	2
10	$16, 31, 2 \times 1, 8, 1 \wedge 24, 6, 2 \times 4, 1, 1$	2

Table 7.1: From EEG input shape $64 \times 134 \times 10$ to output shape $K \times K \times K$ with $K = 7$. Each layer is followed by a max-pool operation with $2, 2, 1 \times 1, 1, 1$.

fMRI candidates. Following with the fMRI encoder, with input shape $I = 64 \times 64 \times 30$ and learnt latent space with $O = 7 \times 7 \times 7$, the properties of the architecture are presented in Table 7.2.

Candidate	Kernel $\times$ Stride ($\bigwedge_1^N k^{(1)}, k^{(2)}, k^{(3)} \times s^{(1)}, s^{(2)}, s^{(3)}$)	N
1	$16, 8, 8 \times 4, 2, 1 \wedge 2, 16, 9 \times 1, 1, 1 \wedge 3, 5, 6 \times 1, 1, 1$	3
2	$16, 6, 12 \times 2, 1, 1 \wedge 6, 4, 6 \times 1, 1, 1 \wedge 12, 47, 5 \times 1, 1, 1$	3
3	$8, 15, 3 \times 1, 4, 1 \wedge 38, 6, 21 \times 3, 1, 1$	2
4	$8, 7, 15 \times 1, 1, 1 \wedge 20, 5, 2 \times 1, 1, 1 \wedge 15, 10, 6 \times 3, 6, 1$	3
5	$6, 20, 2 \times 5, 1, 1 \wedge 5, 8, 16 \times 1, 6, 2$	2
6	$6, 44, 15 \times 1, 1, 1 \wedge 28, 7, 5 \times 1, 1, 1 \wedge 16, 6, 3 \times 2, 1, 1$	3
7	$14, 13, 5 \times 1, 1, 2 \wedge 18, 16, 2 \times 1, 1, 1 \wedge 11, 21, 3 \times 3, 2, 1$	3
8	$8, 11, 14 \times 1, 1, 1 \wedge 29, 19, 6 \times 1, 1, 1 \wedge 20, 27, 3 \times 1, 1, 1$	3
9	$7, 2, 7 \times 1, 1, 1 \wedge 29, 25, 9 \times 1, 1, 1 \wedge 21, 23, 7 \times 1, 2, 1$	3
10	$17, 28, 5 \times 1, 1, 1 \wedge 19, 16, 7 \times 1, 1, 1 \wedge 7, 6, 4 \times 3, 2, 2$	3

Table 7.2: From fMRI input shape $64 \times 64 \times 30$ to output shape $K \times K \times K$ with $K = 7$. Each layer is followed by a max-pool operation with $2, 2, 2 \times 1, 1, 1$.

7.2.3 EEG electrode selection

One good example that illustrates the EEG electrode dimension is a natural language context awareness. For instance, consider the sentence shown in Figure 7.5. The relationship of each word with its preceding and succeeding ones is not representable in an Euclidean space $\mathbb{R}^S$, with S being the length of the sentence. Since the proposed architecture has convolutional layers, local similarity/local properties need to be present in the electrode dimension.

If an attention mechanism processes that sentence, the result might be an embedding of "John playing Mark bed". This is the type of processing proposed to tackle the Euclidean representation of an EEG for the electrode dimension. Let $\vec{\mathbf{x}} \in \mathbb{R}^{C \times F \times T}$ be the EEG representation, where C refers to the electrode dimension, F to the frequency dimension and T to the temporal dimension, and $A \in \mathbb{R}^{C \times F \times T}$ is the attention weight matrix. The latter encodes the context of each electrode for a representation of an EEG

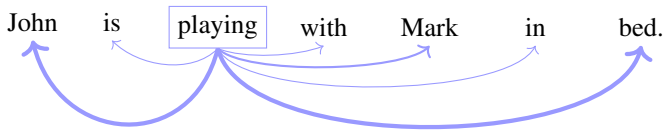
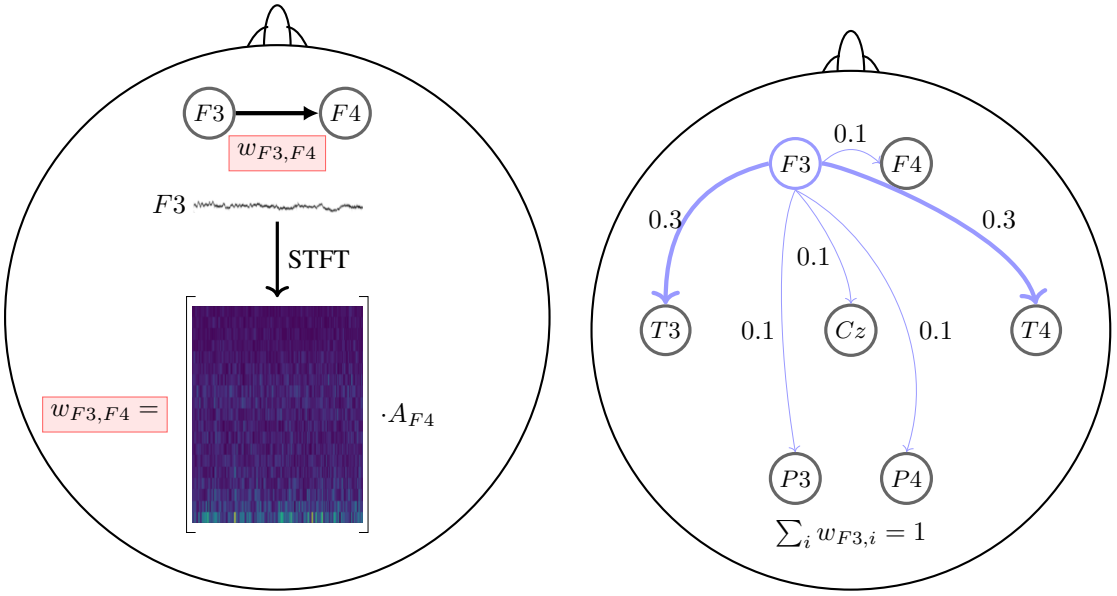


Figure 7.5: An example of attention in the context of natural language processing. In this example the word *playing* may have a different meaning if the sentence was instead: "*John is playing with Mark in the park.*". Playing in *bed* may not encompass the same actions as playing in the *park*.

signal. The dot product, with the $F \times T$ dimension flattened, gives $W \in \mathbb{R}^{C \times C}$, where

$$\forall i \in \{1, \ldots, C\} : W_i = \left[\vec{\mathbf{x}}_i^\top \cdot A_1, \ldots, \vec{\mathbf{x}}_i^\top \cdot A_C\right].$$



(a) Topographical attention.

(b) Attention mechanism of how an electrode is a linear combination of the other electrodes.

Figure 7.6: Attention by dot product for the reorganization of EEG channels.

Consider the diagram in Figure 7.6(a), the context/attention score given to the electrode F_3 relation with F_4 is given by the dot product $\vec{\mathbf{x}}_{F_3} \cdot A_{F_4}$. This type of processing enriches contextually/locally (as the representation is processed by convolutional layers) by normalizing the attention scores,

$$\forall i \in \{1, \ldots, C\} : B_i^\top = \left[b_{i1}, \ldots, b_{iC}\right] = \left[\frac{\exp(w_1)}{\sum_j \exp(w_j)}, \ldots, \frac{\exp(w_C)}{\sum_j \exp(w_j)}\right],$$

so that each electrode will consist of a linear combination of all electrodes, as

$$\vec{T}_{\vec{\mathbf{x}}_i} = \sum_j^C \vec{\mathbf{x}}_i \odot b_{ij}, \forall i \in \{1, \ldots, C\},$$

where $\vec{T}_{\vec{\mathbf{x}}}$ is the electrode rearranged attention representation of $\vec{\mathbf{x}}$. Grounded on empirical evidence, we hypothesize that such a representation is better able to be processed by a set of convolutional layers.

This processing pertains as a feature selection step and a reorganization of electrodes. Feature selection is inherent in the attention mechanism, because electrodes can be suppressed in B. For instance, if

$\exists i \in \{1,\ldots,C\} \wedge \forall j \in \{1,\ldots,C\} : b_{ji} \to 0$, it implies that $\vec{\mathbf{x}}_i$ will not have an influence in $\vec{\mathbf{T}}_{\vec{\mathbf{x}}}$ and therefore is discarded for the synthesis task. Learning from high dimensional data is challenging and the attention's feature selection property, although being memory expensive, can help restrict the learning space. On the reorganization of channels side, we pose it as being an induced property given the convolutional layer succeeding the reorganization process of the attention mechanism. The gradients, upon reaching the attention layer, will have influence from the convolutional layers. These layers have a unique trait of extractring local properties from a set of features (e.g. image). In our case, since one is processing an EEG signal representation, we are inducing the attention mechanism to organize channels in such a way, that for each window and step size of the convolution, there will be local properties in the $\vec{T}_{\vec{\mathbf{x}}}$ representation. All in all, there is an possibility that the patterns in $\vec{T}_{\vec{\mathbf{x}}}$ may be repeated, however pattern heterogeneity is not necessary for the task. Nonetheless, this is always conditioned on the resulting dimension of $\vec{T}_{\vec{\mathbf{x}}}$. For non repetitive patterns, one might need to perform dimensionality reduction (e.g. eigendecomposition).

Figure 7.6(b) illustrates, in similar fashion to the natural language example shown in the beginning of this section, the assignment of *softmax* normalized scores of an electrode to all other electrodes. These scores are interpreted as spatial relationships that are context enriched. Based on the example of Figure 7.6(b), if F_3 has the highest score for T_3 and T_4 then we say these two together $T_3 - T_4$ enrich the representation $\vec{T}_{\vec{\mathbf{x}}}$ to be processed by attentional layers. By inferring important connections, i.e. those with an attention score above a given threshold, we are extracting a relational electrode graph from EEG that is directly related to the ability to predict haemodynamics. Such relationships are important to the understanding of which activity in the EEG signal relates to the fMRI.

7.2.4 Fourier features

Fourier features [20] are a natural candidate for image synthesis tasks. This method has played a major role in computer vision tasks [20, 75, 118], because of its ability to capture functions/properties with high resolutions. Suffice to say that is the case for the application of this thesis. Nowadays, there are simultaneous EEG and fMRI datasets being recorded at 7 Tesla [119], which produce voxels of size $1 \times 1 \times 1$mm. Therefore high resolution learning capabilities are a need for the synthesis model being developed. Recall from chapter 3, given a vectorial representation $\vec{\mathbf{z}} \in \mathbb{R}^L$, then the Fourier projection takes the form

$$\cos(\omega_i \cdot \vec{\mathbf{z}} + b_i), \tag{7.4}$$

where $\omega_i \sim \mathcal{N}(0,1)^L$ and $b_i \sim \mathcal{U}(0, 2\pi)$. In practice, multiple sinusoids are projected, with $\omega \sim \mathcal{N}(0,1)^{L\times L}$ and $b \sim \mathcal{U}(0,2\pi)^L$, such that

$$\vec{\mathbf{z}}^* = \sqrt{\frac{2}{L}} \left[\cos(\omega_1 \cdot \vec{\mathbf{z}} + b_1) \quad \ldots \quad \cos(\omega_L \cdot \vec{\mathbf{z}} + b_L)\right]. \tag{7.5}$$

Adding to the ability of this sinusoid projection enabling the capture of high resolution functions, it also enables the ability to process samples with distribution shifts. Besides the presence of outliers in data, there is also the problem of processing data that was captured using different experimental settings. Neuroimaging data is particularly sensitive to these types of changes, even with the same sampling rate, the same electrode distribution system, etc; the distribution of the recorded data recorded will likely be different. This poses a

challenge for the application of this methodology in a real life setting for EEG data. Fortunately, because the *cosine* is periodic $\in [-1, 1]$, along its domain $\mathbb{R}$, it enables a model to be trained on $X_1 \sim \mathcal{X}_1$, EEG data, and $Y_1 \sim \mathcal{Y}_1$, and still be able to produce fMRI volumes similar to $\mathcal{Y}_1$ given $X_2 \sim \mathcal{X}_2 \wedge \mathcal{X}_1 \neq \mathcal{X}_2$, yet producing an fMRI identically distributed to $\mathcal{Y}_1$. This type of function is so called a shift invariant function and this solution is explored in chapter 9.

7.2.5 Neural flow

The resnet-18 block, already introduced in the previous chapter, is a big corner stone for the synthesis model. Its feature extraction ability and lower likelihood to produce vanished gradients, put this technique as one of the most used in computer vision tasks. Figure 7.7 illustrates the application to process both EEG and fMRI representations. The input of this layer is split into two flows. One that applies one convolution with $k \times s$ (these values are specified for the architecture in the next sections) with *valid* padding. And another that applies a 3×1 convolution with *same* padding, before applying a $k \times s$ with *valid* padding convolution. Both these flows produce representations with similar structure, allowing them to be joined by addition. The last step of the resnet block is a ReLU activation.

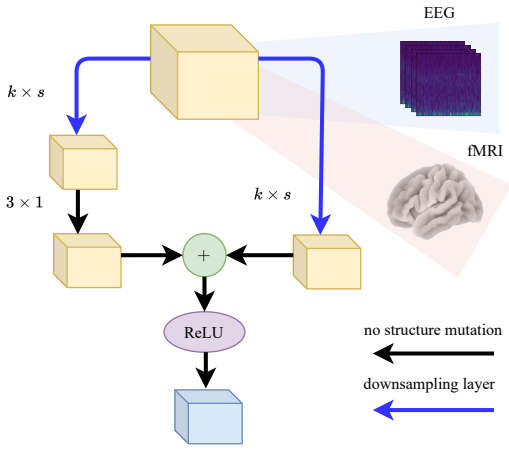


Figure 7.7: The inspired Resnet-18 block forks the input in two computational flows: (1) the first, represented in the left part of the figure, is processed by a convolutional layer with $k \times s$ as the kernel and stride sizes operate with *valid* padding, following the output goes through a convolutional layer with 3×1 with *same* padding; (2) the second flow, corresponds to the right arrow of the fork, processes the input with a convolutional layer with $k \times s$ with a *valid* padding. The representations of the fork are joined by the *addition* operation, which is followed by a ReLU activation [114]. Please note that max pooling [97] and batch normalization [113] layers are optional to follow each downsampling layer. EEG and fMRI feature representations are included in the figure for the reader to understand that this block structure is used to process EEG and fMRI, though differing in the values of $k \times s$ in each network.

Following the processing by the resnet blocks, the representation of EEG and fMRI go through an affine transformation. The results are the latent representations $\vec{\mathbf{z}}_x$ and $\vec{\mathbf{z}}_y$, referent to EEG and fMRI, respectively. Which are then used for a regularization term,

$$\Omega(\vec{\mathbf{z}}_x, \vec{\mathbf{z}}_y) = 1 - \frac{\vec{\mathbf{z}}_x \cdot \vec{\mathbf{z}}_y}{||\vec{\mathbf{z}}_x||_2^2 \cdot ||\vec{\mathbf{z}}_y||_2^2} \tag{7.6}$$

Fourier features, given the latent EEG representation $\vec{\mathbf{z}}_x$, are projected according to $cos(\omega \cdot \vec{\mathbf{z}}_x + \beta)$. These are then mapped to the fMRI output space using an affine projection. All these operations, together and according to the flow described in this section, account for the predicted fMRI, $\hat{\vec{\mathbf{y}}} \in \mathbb{R}^{M'}$.

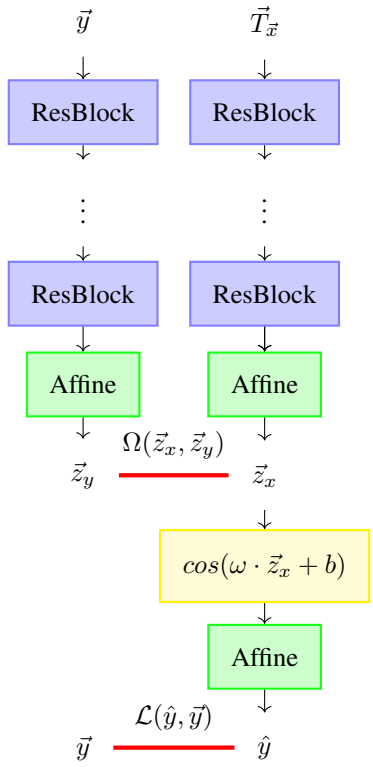


Figure 7.8: This architecture maps both EEG and fMRI representations to a space with the same structure, using a series of Resnet [21] blocks. In the latent space a regularization term is introduced, corresponding to $cos(\vec{z}_x, \vec{z}_y)$. The latent EEG representation is then projected to Fourier features, which are then processed by an affine layer that maps it to the predicted fMRI.

The problem of predicting an fMRI volume is solved with the final loss function, introduced at the output of the neural network,

$$\mathcal{L}(\vec{\mathbf{y}}, \hat{\vec{\mathbf{y}}}) = ||\vec{\mathbf{y}} - \hat{\vec{\mathbf{y}}}||_1^1, \tag{7.7}$$

which makes the gradients of the neural network being computed as described in Figure 7.4.

7.2.6 Style prior and posterior

Gu et al. [118] is one of the many studies that leverage the latent space to introduce *style* of an image/representation produced at the output of the model. Based on this principle, we propose two ways of adding style to the Fourier projection of EEG records, $\vec{\mathbf{z}}_x^* = cos(\omega \cdot \vec{\mathbf{z}}_x + \beta)$: they are the *style prior* and *style posterior*. The naming of the two alternatives is inspired by the Bayesian perspective. The style prior is defined as a vector, $\vec{\mathbf{z}}_w \in \mathbb{R}^L$, which is independent from the input/data (acts as a prior), $\vec{\mathbf{x}}$, such that $\vec{\mathbf{z}}_w \perp \vec{\mathbf{x}}$. This type of style has the advantage of maintaining the invariant shift property given by the cosine function. In other words, given a new dataset with a different EEG distribution, the model is able to produce an identically distributed fMRI volume, to the ones learned. On the other hand, the style posterior is dependent on the input, more specifically it is defined as

$$\vec{\mathbf{z}}_w | W, \vec{\mathbf{x}} : \vec{\mathbf{z}}_w = Q \cdot W,$$

where W is the attention weights computed at the beginning of the neural network, and $Q \in \mathbb{R}^{L \times C \times C}$ is a

trainable weight matrix that transforms W to the style posterior vector, $\vec{\mathbf{z}}_w$. This type of style,

$$\vec{\mathbf{z}}_x^* = cos(\omega \cdot \vec{\mathbf{z}}_x + \beta) \odot \vec{\mathbf{z}}_w, \tag{7.8}$$

is also inspired on the electrode connectivity claims by Rojas et al. [91] and other simulatneous EEG and fMRI studies that use graphs to relate both modalities [120, 121]. Indeed, the attention weights are a weighted graph representation, with EEG electrodes as its nodes. Note that given the prior and posterior definition, we are relating this variables as $\mathbf{P}(\vec{\mathbf{Z}}_w|W) \propto \mathbf{P}(W|\vec{\mathbf{Z}}_w) \times \mathbf{P}(\vec{\mathbf{Z}}_w)$.

7.3 Experimental setting

The experiments were ran for 3 distinct datasets, described in chapter 5, and a Bayesian optimization hyperparameter search was performed with the following hyperparameter ranges:

- learning rate $\in [1\text{e}-10, 1\text{e}-2]$;
- weight decay $\in [1\text{e}-10, 1\text{e}-1]$;
- filter size $\in \{2, 4\}$;
- max pooling layers (after each convolutional layer) $\in \{0, 1\}$;
- batch normalization layers (after each max pooling/convoutional layer) $\in \{0, 1\}$;
- skip layers (in Resnet block) $\in \{0, 1\}$;
- dropout of convolutional weights $\in \{0, 1\}$.

There were two phases of optimization: 1) only the fMRI encoder and decoder participated, forming an fMRI autoencoder; 2) the whole model participated and all the parameters were part of the search space except for L, which was already set in phase (1). The first phase discovered the latent dimension by mapping the EEG and fMRI representations, and the second phase discovered all the hyperparameters of the neural network. In the first phase, the batch size was set to 64 to decrease run time. The same was done for discovering the neural network setup, whose search space was generated by the framework described in the previous chapter. All these searches were performed on one dataset only, the **NODDI** dataset. The hyperparameters discovered were used for the experiments ran on the **Oddball** and **CN-EPFL** datasets. The latter is due to the searches being exhaustive and taking more than 2 months for the **NODDI**, since the other datasets contain more individuals and more features they would have a bigger search time.

The baselines subject to comparison with the state-of-the-art are:

- (i) Linear projection on the latent space representation, $\vec{\mathbf{z}}_x$;
- (ii) [with *style* posterior] Topographical attention on the EEG electrode dimension;
- (iii) Random Fourier feature [20] projection on the latent space representation, $\vec{\mathbf{z}}_x^*$;
- (iv) [with *style* posterior] Combination of (ii) and (iii), as topographical attention is applied in the EEG electrode dimension, as well as the random Fourier feature [20] projection on the latent space representation, $\vec{\mathbf{z}}_x^*$.

Additionally, experiments of (ii) and (iv), with no style and with a *style* prior learnable vector, are reported in section 7.4.

7.3.1 Data preprocessing

All the datasets considered in the experiments had preprocessed versions made publicly available. Although, these preprocessings were done with each study's goal in mind, we found beneficial to use them. Regarding the EEG, the STFT was applied with a rectangular window of size equal to the *time response* of the fMRI recording of each dataset. This was done, so one would have a direct temporal synchronization between the EEG and fMRI. This means frequencies were evaluated as low as 0.5Hz for the NODDI and Oddball, and 0.78Hz for the CN-EPFL. Despite the lower frequencies not being the most relevant correlations with haemodynamics [122], we found them required as input for the model.

EEG, $\vec{\mathbf{x}}$, and fMRI, $\vec{\mathbf{y}}$, representations were paired in segments of 20 seconds of EEG for one fMRI volume. In terms of start and end EEG time sets, only signal information prior to 6 seconds before the fMRI volume was considered. Liao et al. [117] claims neuronal activity is only reflected in haemodynamics 5.4 to 6 seconds after. In this context, $\vec{\mathbf{x}}$ is taken from $[t-26, t-6]$, being t the referenced time of the paired fMRI volume. Consequently, we formalize the EEG representation as $\vec{\mathbf{x}} \in \mathbb{R}^{C\times F\times 20}$.

7.3.2 Layer-wise relevance propagation

Bach et al. [87] proposed a method to propagate relevances from the output of a neural network to the input features. This provides relevance features, that have an informative explainability nature, assessing which ones were more relevant (either negatively or positively). Let j be a hidden neuron, following the proposed propagation rule, then its relevance is computed as

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_j a_j w_{jk}} R_k, \tag{7.9}$$

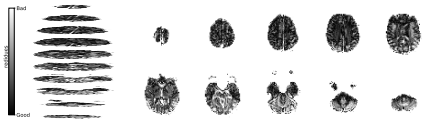
where a neuron, k, has a relevance, R_k, associated to it. The relevance of all the neurons of the output layer are by default the output logits and the relevance of all layers are computed by backpropagation of relevances using the rule stated in Equation 7.9. Note that, this rule does not apply to propagate through sinusoidal activations, which are used in this work (see Section 7.2.4). For EEG features, the relevances are propagated through the proposed *style* posterior (see Section 7.2.5), where standard layers, that enable the use of this rule, are used.

7.4 fMRI synthesis

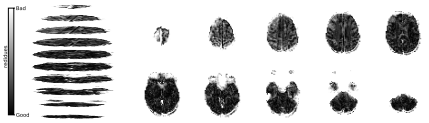
Figure 7.9 illustrates the distribution of residues (observed vs. estimated differences) on the fMRI volumes for the NODDI dataset. Clearly, by visual inspection, (iv) model has the darker and biggest area of shaded regions, which implies a better coverage across the brain regions and better synthesis quality. Models with topographical attention, (ii) and (iv), corresponding to Figures 7.9(b) and 7.9(d), respectively, significantly improve the synthesis, as shown by the darker and bigger areas against (i) and (iii) depicted in Figures 7.9(a) and 7.9(c), respectively. Particularly, we notice that models (i) and (iii) report difficulty in the retrieval of haemodynamical activity located in occipital and parietal lobes.

Model	RMSE			SSIM		
	NODDI	Oddball	CN-EPFL	NODDI	Oddball	CN-EPFL
(i)	0.5124 ± 0.0498	0.7419 ± 0.0290	0.5860 ± 0.0865	0.4329 ± 0.0054	0.1829 ± 0.0332	0.5037 ± 0.0734
(ii) (with *style* posterior)	0.4121 ± 0.0390	0.7728 ± 0.1184	0.5288 ± 0.0355	0.4724 ± 0.0096	0.1580 ± 0.0405	0.5221 ± 0.0707
(iii)	0.4333 ± 0.0448	0.7326 ± 0.0463	0.5282 ± 0.0614	0.4618 ± 0.0028	0.1963 ± 0.0388	0.5074 ± 0.0833
(iv) (with *style* posterior)	0.3972 ± 0.0186	0.7014 ± 0.0855	0.5166 ± 0.0560	0.4613 ± 0.0198	0.2004 ± 0.0172	0.5222 ± 0.0877
Liu and Sajda [123]	0.4549 ± 0.0806	0.8591 ± 0.0342	0.5915 ± 0.1083	0.4488 ± 0.0601	0.1885 ± 0.0380	0.5190 ± 0.1062

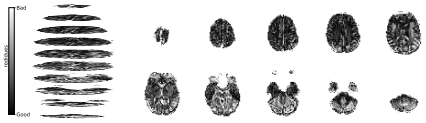
Table 7.3: Root mean squared error (RMSE) and structural similarity index measure (SSIM) of the target synthesis task for the proposed and state-of-the-art models across all datasets. (i) refers to the linear projection in the latent space, (ii) refers to topographical attention on the EEG channels dimensions with a linear projection in the latent space, (iii) implements a random Fourier feature projection in the latent space, and (iv) performs topographical attention on the EEG channels dimension with a random Fourier features projection in the latent space.



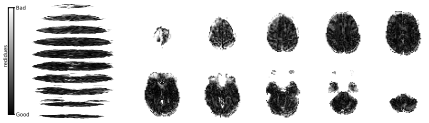
(a) (i) - Linear latent projection. RMSE of 0.5124 ± 0.0498 and SSIM of 0.4329 ± 0.0054.



(b) (ii) - Topographical attention on the EEG channels dimension, with linear latent projection (i). Attention scores are placed as a *style* posterior on the latent representation. RMSE of 0.4121 ± 0.0390 and SSIM of 0.4724 ± 0.0096.



(c) (iii) - Random Fourier feature latent projection. RMSE of 0.4333 ± 0.0448 and SSIM of 0.4618 ± 0.0028.

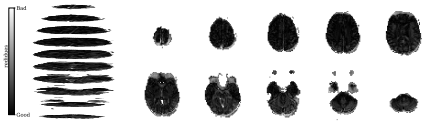


(d) (iii) - Topographical attention on the EEG channels dimension, with random Fourier latent projection (i). Attention scores are placed as a *style* posterior on the latent representation, as described in Equation 7.8. RMSE of 0.3972 ± 0.0186 and SSIM of 0.4613 ± 0.0198.

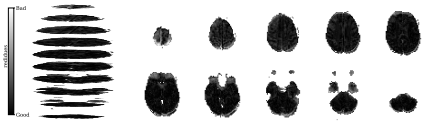
Figure 7.9: Mean absolute residues for each implemented models. Model (ii), implementing topographical attention with a *style* posterior, and model (iv), additionally transforming the latent features using the random Fourier feature projection (described in section 7.2.4), achieve the best performance relative to RMSE and SSIM metrics.

Table 7.3 contains the results obtained from running the target approaches ((i), (ii), (iii) and (iv)) and the state-of-the-art [123]. For all datasets considered in the experiments, model (iv) obtained the best RMSE values. Further, our baselines consistently outperform the state-of-the-art, according to the RMSE metric. From analyzing our baselines, we conclude that random Fourier features, described in Section 7.2.4, benefit models (i) and (ii) and the introduction of topographical attention also benefits both models (i) and (iii). The latter, shows the adaptability and robustness of introducing topographical relationships to the synthesis of fMRI. By assessing the experiments from the perspective of the SSIM metric, there is not a concordant superiority across all datasets, as observed with the RMSE. Nonetheless, the state-of-the-art is outperformed by at least one of our baselines on all datasets. Specifically, on the NODDI dataset (resting state), we observe that incorporation of topographical attention in model (ii), under a *style* posterior, achieves the best SSIM value.

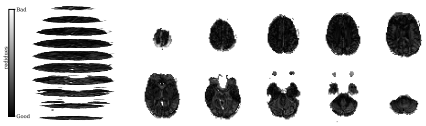
To better address which regions our baselines had more difficulty retrieving, the absolute residues were computed and are illustrated in Figure 7.10. Baselines – corresponding to models (i) and (ii), shown in Figures 7.10(a) and 7.10(b) respectively, which correspondingly implement a linear projection in the latent space and topographical attention –, have difficulty retrieving the prefrontal, occipital and parietal lobes, as the shade tends to a lighter grey in that region. Model (iv), shown in Figure 7.10(d), does not show a notice-



(a) (i) - Linear latent projection.



(b) (ii) - Topographical attention on the EEG channels dimension, with linear latent projection (i).



(c) (iii) - Random Fourier feature latent projection.

(d) (iii) - Topographical attention on the EEG channels dimension, with random Fourier latent projection (i).

Figure 7.10: Normalized mean absolute residues for the proposed models.

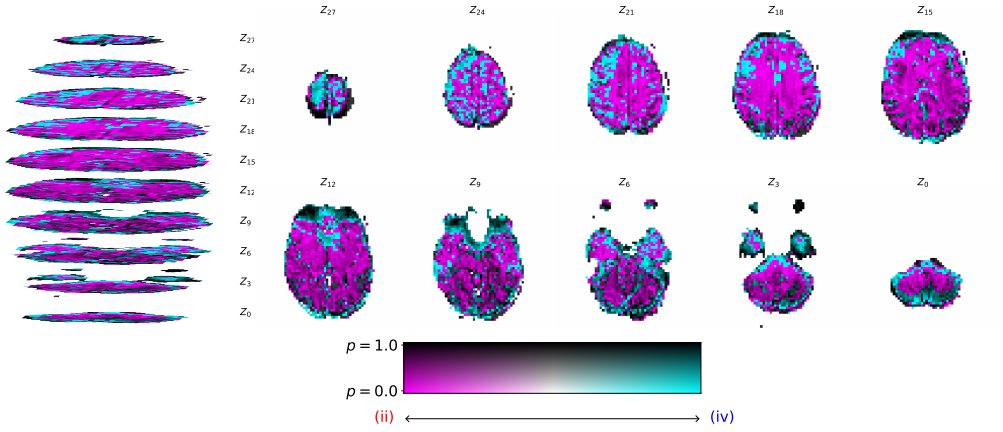


Figure 7.11: Region-sensitive comparison of models (ii) and (iv), both using *style* posterior, reporting the best model in each voxel according to predictive power (statistical significance under t-test). Although Table 7.3 shows that model (iv) outperforms (ii) regarding RMSE, this analysis shows that model (ii) achieves a significantly better synthesis capacity on the majority of the voxels.

able region with a lighter tone of grey, which implies no evident difficulty in retrieving haemodynamical activity across the different brain regions.

Figure 7.11 illustrates the voxel wise comparance, with statistical significance assessments, between (ii) and (iv). For the Oddball dataset, the RMSE and SSIM metrics report a worse synthesis ability for all methodologies compared to the other datasets. Our baselines outperform the state-of-the-art, and model (iv) with a *style* posterior is significantly superior to all baselines. Random Fourier projections, (iii), appear to better address the synthesis task than topographical attention alone, (ii). The SSIM is rather poor, with values generally below 0.2000 being the mean and only model (iv) surpassing this threshold with 0.2004 SSIM.

Models (ii) and (iv), considering topographical attention with a *style* posterior, show the best performance in terms of SSIM metric in the CN-EPFL dataset. In spite of the RMSE and SSIM not being in total accordance, the topographical attention superiority is consistent for the metrics considered. This supports our hypothesis that the use of topographical structures plays an important role when studying these two modalities and is hence preferable.

Role of topographical attention: in Table 7.3, reports the results of models (ii) and (iv), both im-

	RMSE			SSIM		
	NODDI	Oddball	CN-EPFL	NODDI	Oddball	CN-EPFL
(ii) w/o *style*	0.5119 ± 0.0494	0.9812 ± 0.0847	0.5458 ± 0.0596	0.4322 ± 0.0054	0.1930 ± 0.0543	0.5027 ± 0.0748
(iv) w/o *style*	0.4321 ± 0.0418	0.7221 ± 0.0411	0.5298 ± 0.0636	0.4621 ± 0.0027	0.1991 ± 0.0382	0.5063 ± 0.0830
(ii) (with *style* prior)	0.5159 ± 0.0477	0.9920 ± 0.8901	0.9920 ± 0.8901	0.4300 ± 0.0043	0.1760 ± 0.0402	0.4974 ± 0.1353
(iv) (with *style* prior)	0.4833 ± 0.0483	0.7394 ± 0.0377	0.5568 ± 0.0737	0.4388 ± 0.0069	0.1873 ± 0.0347	0.4960 ± 0.1084

Table 7.4: RMSE and SSIM scores in the absence and presence of prior styling, all considering the presence of a posterior style vector conditioned on the attention scores. The upper half of this table shows the results of implementing topographical attention, but without using the attention scores to add style to the latent space representation (w/o style). The bottom half, shows the use of a style prior vector, $\in \mathbb{R}^L$, that is not conditioned on any features, and serves to add learnable style features to the latent representation. The latter is widely used in computer vision research, with a recent study applying it to generate images [118].

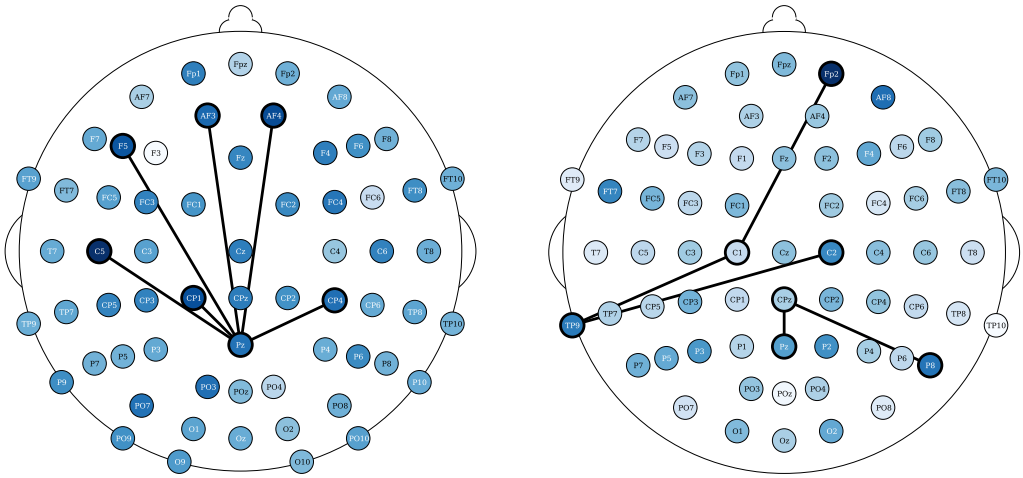
plementing a *style* posterior vector that is conditioned on the learned attention graph. This graph is a representation of the relationships between the EEG electrodes, learned during the optimization process, that inherently help the retrieval of haemodynamical activity. To validate this hypothesis, Table 7.4 shows the RMSE and SSIM metrics obtained from experiments ran on the following models:

- (ii) and (iv) with no *style* induction, but still performing attention in the EEG electrode dimension;
- (ii) and (iv) with *style* prior, reported on the bottom half.

From the previous section, we know that the topographical attention, inducing a *style* posterior on the latent representation (see Section 7.2.5), consistently benefits the regression task across all the datasets considered in our experiments. This holds for resting state (NODDI) and task-based (Oddball and CN-EPFL) settings. By comparing the results of models (ii) and (iv) reported in Table 7.3 with the ones presented in Table 7.4, the impact of conditioning the *style* posterior vector on the attention scores is quite noticeable. And it goes beyond the simple induction of *style* in the latent space, as Table 7.4 shows that placing a *style* prior can cause overfitting in some settings.

7.5 Discussion

EEG electrode attentional based relations dependency. The ran experiments with different types of *style*, $\vec{\mathbf{z}}_w$, in the latent representation (see Equation 7.8), tell us that conditioning the styling on the attention scores, an EEG electrode topographical representation, is beneficial for the fMRI synthesis task. Further, the fact that, in addition to not conditioning *style*, learning a *style* prior vector is not as informative (no dependency on $\vec{\mathbf{x}}$) for the neural network to better optimize the learning objective. This leads us to believe that a learnable unconditioned *style* acting as a prior, is prone to overfitting the training data, since it is not conditioned on $\vec{\mathbf{x}}$. Our experiments show that the projected random Fourier features (prior), $\vec{\mathbf{z}}_x \rightarrow \vec{\mathbf{z}}_x^*$, if multiplied (conditioned) by data dependent (EEG attention graph scores), Equation 7.8, not only reduces the empirical risk, but is also preferable to both multiplication of an unconditioned learnable *style* prior and no multiplication at all. Therefore, the placement of a *style* posterior, conditioned on EEG attention scores guides the random Fourier features and removes the inherent assumptions of a prior [124]. Adding to it, the topographical information retrieved from the attention scores contains information that is highly related to haemodynamical activity, this is in accordance with several neuroscience studies that use topographical structures, such as graphs, to relate EEG and fMRI, used in simultaneous EEG and fMRI studies [91, 120, 121].



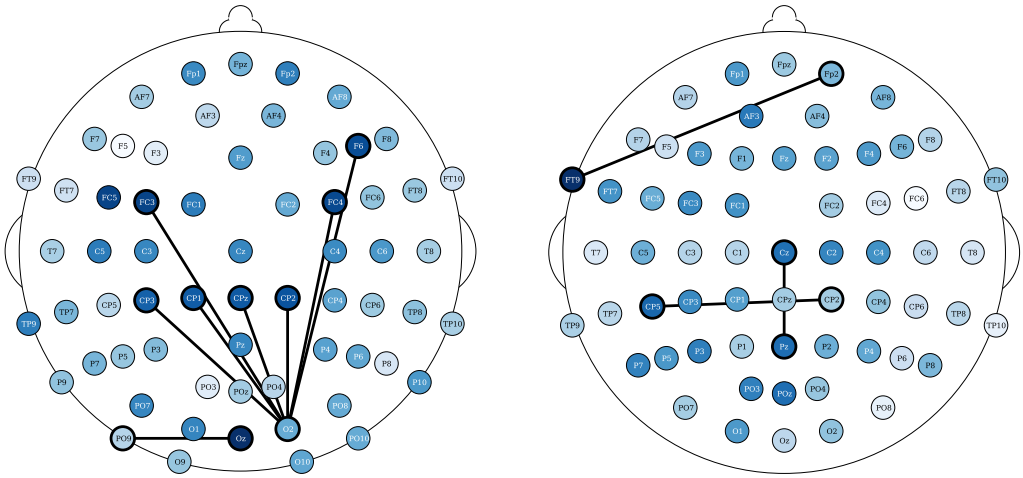
(a) (iv) - Topographical attention on the EEG electrodes dimension, with random Fourier feature projections in the latent space, in NODDI dataset.

(b) (iv) - Topographical attention on the EEG electrodes dimension, with random Fourier feature projections in the latent space, in CN-EPFL dataset.

Figure 7.12: EEG electrode attention score relevances for resting state NODDI and task based CN-EPFL datasets.

Most relevant electrode relations. Consider the relevance of the attention scores, computed from models (ii) and (iv), both having topographical attention at the EEG channel dimension, and model (iv) with projected random Fourier features in the latent space. These relevances were propagated, using the LRP algorithm [87] described in Section 7.3.2, through the attention style based posterior. Figure 7.12 shows the relevances plotted in a white to blue scale, from less relevant to most relevant, respectively. The latter only shows the edges that are above the 99.7 percentile. The presence of an edge between electrodes suggests that either this connection yields a Markovian property for the EEG instance or, otherwise, it is relevant to add fMRI style conditioned on these connections (recall from Section 7.2.3 that posterior $\vec{\mathbf{z}}_w$ conditions the latent EEG representation $\vec{\mathbf{z}}_x^*$ such that $\vec{\mathbf{z}}_x \odot \vec{\mathbf{z}}_w$). For resting state fMRI, Figure 7.12(a) show connections between visual cortex channels (O2 electrode in Figure 7.13(a) and Pz electrode in Figure 7.12(a)) with frontal and central channels to be the most relevant (above the 99.7 percentile of relevance). Figure 7.13(a) reports an additional connection between the Oz and PO9 electrodes, a correspondence between an occipital and a parietal-occipital electrode, which is in accordance with connectivity observations reported by Rojas et al. [91]. There were no reported relevances for the electrodes (T) placed in the temporal regions for resting state settings. In contrast, in task-based fMRI synthesis, relevant relationships between temporal (FT9 and TP9) and frontal/central (Fp2 and C1/C2, respectively) electrodes were reported, Figure 7.12(b). In both of these figures, connections between central and parietal electrodes were observed. Particularly, there were reported connections between Cz with Pz and CP5 and CP2 electrodes in Figure 7.13(b). And reported connections between Pz and P8 with CPz electrodes in Figure 7.12(b).

Converging to retrieve near scalp haemodynamical activity. One interesting phenomena that was observed by propagating relevances from the latent representations of the fMRI instance, $\vec{\mathbf{z}}_y$, to the input, $\vec{\mathbf{y}}$, was that the relevances in sub-cortical areas were neither positive nor negative, yielding residual relevance, as seen in Figure 7.14. This later observation suggests that haemodynamical activity from these areas does not significantly aid the targeted synthesis. Recall that the regularization term, $\Omega(\vec{\mathbf{z_x}}, \vec{\mathbf{z_y}}) = 1 - cos(\vec{\mathbf{z_x}}, \vec{\mathbf{z_y}})$, is used with the latent EEG and fMRI representations. This is in accordance with the fact that the retrievable



(a) (ii) - Topographical attention on the EEG electrodes dimension in NODDI dataset.

(b) (ii) - Topographical attention on the EEG electrodes dimension in CN-EPFL dataset.

Figure 7.13: EEG electrode attention score relevances for resting state NODDI and task based CN-EPFL datasets. Figures 7.13(a) and 7.13(b) report the attention relevances for the NODDI resting state dataset and the CN-EPFL dataset, respectively.

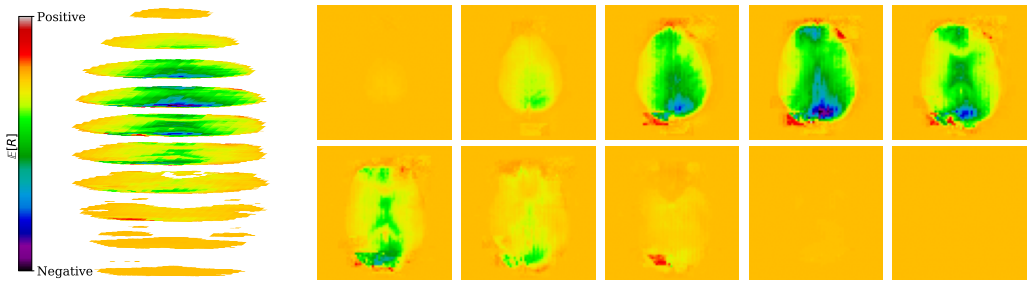


Figure 7.14: fMRI computed relevances for the NODDI dataset, starting from the latent fMRI representation, $\vec{z}_y$.

information is in its majority next to the scalp, where the electrodes are placed, and indeed de Beeck and Nakatani [9] discuss how high frequencies are not able to travel significant distances with obstacles, such as white matter and the scalp, in between. We also report negative relevances on the visual cortex and positive relevances on the occipital and prefrontal lobes. Please note that negative and positive relevances represent relevant features, whereas when one has zero relevance, it means a feature was not relevant for the task. Daly et al. [8] found that neuronal activity retrieved from EEG can reflect the haemodynamical changes in subcortical areas. Here we claim that haemodynamical activity information in areas next to the scalp are relevant to learn the shared latent space.

Laboratory setup impacts EEG to fMRI synthesis. The results show that it is more difficult, according to the RMSE metric, to synthesize task-based fMRI than resting state. This observation is in contrast with studies that report that resting state fMRI is inherently more complex than task based fMRI [125]. The SSIM metric, in contrast to the RMSE, shows less significant differences for the Oddball recordings in favor of fMRI synthesis in the resting state. However, the CN-EPFL dataset is not in accordance with the latter. This performance heterogeneity across the datasets may not only rise from the characteristics of the recording sessions, including the different nature of the performed task, yet may be also propelled by the different preprocessing techniques employed. Each dataset is publicly available and is supported with published studies, having unique equipment, experimental protocols, and algorithms. CN-EPFL dataset is

the most complete one, with a total of 20 individuals and with a resolution of $2 \times 2 \times 2$mm, which makes a total of $108 \times 108 \times 64$ voxels. These differences, caused by working with 3 Tesla (CN-EPFL dataset) versus 1.5 Tesla (NODDI and Oddball datasets) scanners, significantly impact the spatial resolution, which for the datasets NODDI and Oddball produce $64 \times 64 \times 30$ and $64 \times 64 \times 32$ voxels, respectively, with around $3 \times 3 \times 3$mm voxel size. One has to further account for the original recording artifacts and disruptions caused by the applied preprocessing techniques. For instance, Oddball dataset contains intra and inter individual wise misalignments across fMRI volumes. This may be the cause of poorer performance of all methods when compared to the other datasets. In addition, Oddball relies on a different EEG electrode positioning system, having a total of 43 electodes that were not placed in accordance with the 10-10 system [79]. Although NODDI and CN-EPFL recordings are in accordance with this system, each study selected unique electrode locations (see the different electrode placements between Figures 7.12(a) and 7.12(b)). Finally, the different EEG sampling frequencies, with 250Hz, 1000Hz and 5000Hz considered in NODDI, Oddball and CN-EPFL recordings, respectively, further affect architectural operations and subsequently impact the learning.

7.6 Summary

- EEG and fMRI instances are defined in 3 dimensional feature spaces with dissimilarity both in structure and representation. This dissimilarity is tackled by searching an optimal latent space, subject to restrictions. Given the optimal latent space, the proposed NAS framework is used to generate and search neural architectures. In resemblance with the methodology of Chapter 6, a neural architecture is obtained, completing the synthesis function, F, that maps EEG to fMRI;
- The function that maps EEG to fMRI needs a set of operations fit for the task: (topographical) attention, Fourier features, and latent style induction; these operations can be independently combined and yield a major role that goes from feature selection, reorganization of channels, shift invariance, to overfitting robustness traits;
- Our experiments conclude that attention-based scores, trained to give Markovian properties to the EEG representation and simultaneously add style features by usage of a posterior, significantly aid the learning task;
- We noticed that haemodynamical information in areas next to the scalp is predominantly considered to learn the shared latent space during the training, aiding fMRI synthesis.

Chapter 8

Quantifying uncertainty in synthesized fMRI

In previous chapters we prepared the reader for a clear understanding of the methodology involved in the electroencephalography (EEG) to functional magnetic resonance imaging (fMRI) synthesis developed and introduced in this thesis. Taking advantage of spectral features, neural processing techniques, automated machine learning frameworks and pairwise learning, we have been able to showcase a trustful synthesis with respect to quantitative metrics (root mean squared error and structure similarity index measure). All in all the synthesized fMRI modality requires exhaustive evaluation for its application in an health care setting. This type of evaluation has been partially done, by addressing the *why* is a certain hemodynamical activity (fMRI) being predicted from the neuronal activity recorded (EEG) through explainability methods (layer wise relevance propagation). This chapter extends this provisory evaluation of the synthesized fMRI by quantifying the uncertainty (risk) associated with a prediction made by an *EEG to fMRI synthesis* model.

We start by motivating the need for uncertainty quantification in the synthesis task (see section 8.1). Then we move on to the description of the discrete cosine transform (DCT) (see section 8.2), which plays a major role in the methodology proposed. Followed, we describe how coefficients are introduced in the DCT spectral domain (see section 8.3). We report on experiments ran with the methodology proposed and analyse the impact of different parameters that come along with it (see section 8.4). Following, we provide a thorough discussion of the results reported and relate them with previous studies (see section 8.5). Finally, we end this chapter with the main takeaways (see section 8.6).

8.1 Why do we need uncertainty in EEG to fMRI synthesis?

The answer to this question is simple: upon a decision making setting, the model needs to provide a measure of uncertainty associated with its decision. This type of information (uncertainty) enables a person (e.g., *doctor*) to reject the information made by the model if uncertainty is high. Consider two following stances. *Explainability* methods answer the question: "*Why do you say this?*". *Uncertainty* quantification methods answer the question: "*How sure are you of this?*". Let uncertainty be quantified by $r \in \mathbb{R}_{\geq 0}$, where $r = 0$ means the model is completely certain that its decision is the correct one, while for greater values of r certainty decreases.

For this synthesis task, we are asking the model $\forall i, j, k : i \in \{1, \ldots, M_1\} \land j \in \{1, \ldots, M_2\} \land k \in \{1, \ldots, M_3\}$: how sure are you (model) that $\hat{\vec{\mathbf{y}}}_{i,j,k} = \vec{\mathbf{y}}_{i,j,k}$? In other words, if the predicted value associated with this voxel is equal to the ground truth one. To answer this question, we need to define the types of uncertainty that we want to measure. In computer vision, the great work done by Kendall and Gal [19] defines two types of uncertainties:

- **Epistemic uncertainty**: this is the uncertainty associated with the model, whose parameters are inherently uncertain for a specific prediction;
- **Aleatoric uncertainty**: this is the uncertainty inherent to the data (e.g., monitoring protocol, instrumental, individual variability). If the data does not provide sufficient information for a certain decision, then this type of uncertainty is significantly high.

8.1.1 Epistemic and aleatoric uncertainty

Let F be the neural network that performs EEG to fMRI synthesis and θ_F its parameters. The set of observations, $\mathcal{D}$, can be described by the pairs $\forall n \in \{1, \ldots, N\} : \mathcal{D}_n = (\vec{\mathbf{x}}_n, \vec{\mathbf{y}}_n)$. Then we can define the goal of the synthesis from a Bayesian perspective [126] as

$$P(\theta_F|\mathcal{D}) = \frac{P(\mathcal{D}|\theta_F) \times P(\theta_F)}{P(\mathcal{D})}, \tag{8.1}$$

where $P(\theta_F|\mathcal{D})$ encodes the probability that θ_F is able to perform the regression/synthesis for the set of instances $\mathcal{D}$ (we want to discover θ_F that maximizes $P(\theta_F|\mathcal{D})$). The likelihood $P(\mathcal{D}|\theta_F)$ encodes the probability for all the observations $\mathcal{D}$, knowing the parameters θ_F. Suffice to say that the latter is computationally unfeasible given all of the infinite possibilities of θ_F, making $P(\mathcal{D})$ intractable. The prior $P(\theta_F)$ describes the probability of parameters θ_F happening. The latter is typically computed using a distribution, which we assume the parameters to follow. For now consider $\theta_F \sim \mathcal{N}(\mu_{\theta_F}, \sigma_{\theta_F})$ as proposed by Kendall and Gal [19]. Ultimately, we approximate the true posterior $p = P(\theta_F|\mathcal{D})$ using $\vec{\mathbf{y}}|\theta_F, \vec{\mathbf{x}} \sim Lap(F^{\theta_F}(x), b)$, assuming the a Laplacian distribution [126], which we can use by applying the logarithm and minimizing its negative as a loss, such that

$$\mathcal{L} = -log(f_{\vec{\mathbf{y}}|\theta_F,\vec{\mathbf{x}}}) = -log\left(\frac{1}{2b} \times \exp\left(-\frac{||\vec{\mathbf{y}} - F^{\theta_F}(\vec{\mathbf{x}})||_1^1}{b}\right)\right) = \frac{||\vec{\mathbf{y}} - F^{\theta_F}(\vec{\mathbf{x}})||_1^1}{b} + log(2b). \tag{8.2}$$

Throughout the rest of this chapter we will simplify what encompasses θ_F, as it is also valid to only put one layer's parameters as variational. For instance Gal and Ghahramani [127] show how a neural networks, that implement dropout [128], are able to quantify uncertainty by leaving the dropout computation on at test time. In a sense for the formulation presented, only the random variables that take part in the neural network flow formulate θ_F (**notation clash**: in future chapters θ_F refers to all the parameters of the neural network that performs synthesis).

Now, we can formulate how to quantify the uncertainties defined. For epistemic uncertainty, the uncertainty inherent in the model, it is computed as the variance of the residues, that is

$$\text{Var}[\vec{\mathbf{y}}_i] \approx \frac{1}{T}\sum^{T} F^{\theta_F}(\vec{\mathbf{x}}_i)^\top \cdot F^{\theta_F}(\vec{\mathbf{x}}_i) - \mathbb{E}(\vec{\mathbf{y}}_i)^\top \cdot \mathbb{E}(\vec{\mathbf{y}}_i). \tag{8.3}$$

Note that Monte Carlo simulation [129] is used to compute this type of uncertainty since random variable sampling is involved in its computation. The same happens for the computation of aleatoric uncertainty as it is computed as an additional output of a neural network, as $\hat{b}(\vec{\mathbf{x}}_i) = \exp(A \cdot F^{\theta_F}(\vec{\mathbf{x}}_i))$, where $A \in \mathbb{R}^{M' \times M'}$, and therefore it approximates the true noise in the data, by plugging it in the loss of equation 8.2 as

$$\mathcal{L} = \frac{||\vec{\mathbf{y}} - F^{\theta_F}(\vec{\mathbf{x}})||_1^1}{\hat{b}(\vec{\mathbf{x}}_i)} + log(2\hat{b}(\vec{\mathbf{x}}_i)), \tag{8.4}$$

when errors, $||\vec{\mathbf{y}} - F^{\theta_F}(\vec{\mathbf{x}})||_1^1$, are too high the aleatoric uncertainty, $\hat{b}(\vec{\mathbf{x}}_i)$, increases, since it is the denominator. At the same time the $log(2\hat{b}(\vec{\mathbf{x}}_i))$ term allows its minimization, so it does not continue its increase for any errors.

8.2 Variational decoder for uncertainty quantification

Consider the neural flow presented in the previous chapter. Let $A_M \in \mathbb{R}^{L^* \times M'}$ be a trainable matrix that performs the affine transformation from the latent space $\mathbb{R}^{L^*}$ to the synthesized fMRI space $\mathbb{R}^{M'}$, i.e. $\hat{\vec{\mathbf{y}}} = \vec{\mathbf{z}}_x^* \cdot A_M$. We can look at the matrix A_M as

$$A_M = \begin{bmatrix} a_{11} & \dots & a_{1M'} \\ \vdots & \ddots & \vdots \\ a_{L^*1} & \dots & a_{L^*M'} \end{bmatrix},$$

with each column $\forall i \in \{1, \dots, M'\} : [A_M]_i = \begin{bmatrix} a_{1i} & \dots & a_{L^*i} \end{bmatrix}^\top$ being a decision boundary for the voxel $\hat{\vec{\mathbf{y}}}_i$. With this we can write the equation for each voxel as

$$\hat{\vec{\mathbf{y}}}_i = \vec{\mathbf{z}}_x^* \cdot [A_M]_l = \sum_l^{L^*} a_{li} \times [\vec{\mathbf{z}}_x^*]_l = \sum_l^{L^*} (a_{li} \times [\vec{\mathbf{z}}_w]_l) \times cos(\omega_l \cdot \vec{\mathbf{z}}_x + \beta_l), \tag{8.5}$$

where $[\vec{\mathbf{z}}_x^*]_l$ and $[\vec{\mathbf{z}}_w]_l$ represent the lth entry of the vector $\in \mathbb{R}^{L^*}$. Recall from equation 7.8, that the style posterior and prior represent a latent vector that encodes *style*. If we unroll the vector mathematical notation to a sum, we get an expression just like in equation 8.5. By explaining this equation in words, we say that each voxel is a sum of *cosines* (representing different shifts and frequencies) each multiplied by a *coefficient*. These resembles a transform, as we will see in the next section.

As mentioned in the previous section, it is usually the case to only set a subset of the parameters of a neural network as random variables. We hypothesize that $A_M \sim \mathcal{N}(A_M^\mu, A_M^\sigma)$, enables an easy computation of the uncertainties introduced in the previous section. We make use of the reparametrization trick [130] to represent A_M as a stochastic variable. The trick consists of splitting the representation of A_M into two matrices $A_M^\mu \in \mathbb{R}^{L^* \times M'}$ and $A_M^\sigma \in \mathbb{R}^{L^* \times M'}$, such that the final matrix is defined as

$$A_M = A_M^\mu + A_M^\sigma \odot \epsilon, \tag{8.6}$$

where $\epsilon \sim \mathcal{N}(0,1)^{L^* \times M'}$ is a random variable that enables gradient backpropagation through this node of computation. This trick is done for all of the random variables represented in the neural network flow that

will be described in the next sections.

8.3 Variational spectral coefficients

The analogy made in the previous section with a transform motivated us to hypothesize the use of a well known transform used in image compression research, the discrete cosine transform (DCT) [131]. It is known for its application in the JPEG algorithm [132], an algorithm that is regarded as a state-of-the-art in image compression used across multiple platforms. The DCT is a linear function, described by $\mathcal{F} : \mathbb{R}^S \rightarrow \mathbb{R}^S$, where spectral coefficients, $X \in \mathbb{R}^S$, of a 1-dimensional signal $x \in \mathbb{R}^S$ (for simplicity's sake), are computed as

$$X_k = \mathcal{F}(x)_k = \sum_{s=0}^{S-1} x_s cos\left(\frac{\pi(2s+1)k}{2S}\right), \forall k \in \{0, \dots, S-1\}. \tag{8.7}$$

The frequency space of the DCT is used in the JPEG algorithm, by multiplying high resolution coefficients of 8×8 batches of the original image (signal) by a small factor and low resolutions with high factors. The latter is known as the process of quantization in JPEG [132] and it allows one to store less information without considerable corruption of the image in the perspective of the human eye. This transform also has its inverse, $\mathcal{F}^{-1} : \mathbb{R}^S \rightarrow \mathbb{R}^S$, that is able to retrieve the original space representation of the signal, x, given its spectral coefficients, X, as

$$x_k = \mathcal{F}^{-1}(X)_k = X_0 + 2\sum_{s=0}^{S-1} X_s cos\left(\frac{\pi s(2k+1)}{2S}\right), \forall k \in \{0, \dots, S-1\}. \tag{8.8}$$

Note that the resemblance of the two equations 8.5 and 8.8 arises when we isolate the components for the inverse DCT as

$$x_k = \sum_{n=0}^{S-1} \left(\frac{X_0}{Z_S} + 2 \times X_s\right) \times cos\left(\frac{\pi s(2k+1)}{2S}\right) \tag{8.9}$$

and associate the direct relation of the two equations between $\left(\frac{X_0}{Z_S} + 2 \times X_s\right)$[1] and $(a_{li} \times [\vec{\mathbf{z}}_w]_l)$ that represent the coefficients of the DCT and the latent coefficients of the neural network that performs *EEG to fMRI synthesis*, and the DCT sinusoids $cos\left(\frac{\pi s(2k+1)}{2S}\right)$ and the Fourier features $cos(\omega_l \cdot \vec{\mathbf{z}}_x + \beta_l)$. Motivated by this analogy and the fact that sinusoids do not impact the vanishing gradient phenomena, since $\frac{\partial}{\partial x} cos(x) = -sin(x) \in [-1, 1]$, we propose the introduction of random variables in the DCT spectral domain, X.

8.3.1 Introducing random variables in the DCT spectral domain

The neural flow is maintained, as in the whole process of the latent Fourier features followed by an affine decoder persists. Additionally, the last two layers of the neural network consist of the DCT and its inverse. The whole premise of this methodology is to allow the neural network to predict less output features, that is instead of $\hat{\vec{\mathbf{y}}} = \vec{\mathbf{z}}_x^* \cdot A_M$ with $A_M \in \mathbb{R}^{L^* \times M'}$, we now map it to a lower dimensional output feature space with $A_m \in \mathbb{R}^{L^* \times m} : m = m_1 \times m_2 \times m_3 \wedge m_i < M'_i, \forall i \in \{1, 2, 3\}$. This inherently impacts the neural

[1] Z_S corresponds to a normalization factor required to include X_0 in the sum and multiply it with the cosine.

network as it has to predict less features and relaxes the synthesis task. We hypothesize two main benefits from this approach, namely: increases generalization, since the number of parameters decreases; and EEG is characterized by its low spatial resolution and thus the prediction of an high resolution spatial signal such as the fMRI from an EEG, forces the synthesis model to predict non-informed features/functions, that are not present in the EEG signal.

Let $Z \in \mathbb{R}^m : Z = \vec{\mathbf{z}}_x^* \cdot A_m$ be the output of the network, illustrated in figure 7.8. This representation is then transformed by the DCT to its spectral domain[2] as

$$\overline{Z} = \mathcal{F}(Z), \tag{8.10}$$

where $\overline{Z} \in \mathbb{R}^m$. A straightforward perturbation that is introduced to $\overline{Z}$ is by multiplying it with random variables[3] as

$$\overline{Z} \odot c. \tag{8.11}$$

The latter is similar to what Khan et al. [133] did in their spectral dropout proposal, where the Bernoulli distributed mask, was applied in the spectral domain, according to equation 8.11.

Note that, the optimization problem is still optimized with the full resolution of the original fMRI volume, $\vec{\mathbf{y}} \in \mathbb{R}^{M'}$, according to equation 8.2. To match the dimension, upsampling is made in the spectral domain, $\overline{Z}$, by padding zeros for the higher frequencies as

$$\hat{\vec{\mathbf{y}}} = \text{concat}(\overline{Z} \odot c, \mathbf{0}^{M'-m}) : \mathbf{0}^{M'-m} = \begin{bmatrix} 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{M'-m}. \tag{8.12}$$

For $c_i = 1, \forall i \in \{1, \dots, m\}$, equation 8.12 is equivalent to upsampling without any perturbation and only filling zeros for the high resolutions. The latter is referred in the next section as *zero-padding*.

8.3.2 Von Mises distributed high coefficients

In this section, we go further and introduce a new sub dimension, where high frequency resolutions described by random variables $c^* \sim vM(\mu^*, \tau^*)^R$ are padded to the spectral domain representation, $\overline{Z}$, with $R \in [0, M' - m] \in \mathbb{N}$ being the number of high resolutions imputed into the signal. The von Mises (vM)

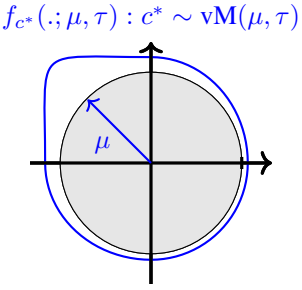


Figure 8.1: von Mises Distribution is normal distribution on a sphere. In practice, we propose an hypersphere of size R.

distribution, whose probability density function is described in figure 8.1, is known for its application in

[2]The DCT of a 3-dimensional signal consists on the concatenation of three sums, each over its respective dimension $m_i, \forall i \in \{1, 2, 3\}$

[3]Please consider from hereon until the end of this chapter that any random variable c is sampled according to the reparametrization trick described for equation 8.6 $c \sim \mathcal{N}(\mu, \sigma)^m$. In this case $c = \mu + \sigma \times \epsilon_c : \epsilon_c \sim \mathcal{N}(0, 1)^m$.

audio separation [134] and resolution enhancement [135] tasks, showing to be innate when applied in the spectral domain.

Sinusoid attention mechanism. The application of the random variables in the spectral domain is done using a simple attention mechanism as a decision dimension described by H, for the vM distributed coefficients. Let $W \in \mathbb{R}^{m \times H}$ be a weight matrix, used to compute the attention scores, $a \in \mathbb{R}^{H}$:

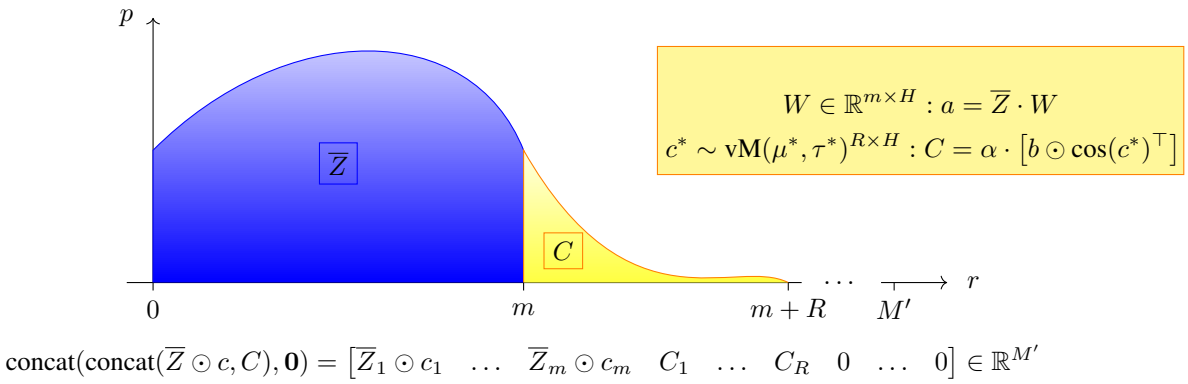


Figure 8.2: Coefficients begin imputed from a single dimension perspective.

$a = \overline{Z} \cdot W$, and consider the high frequency resolutions, redefined as $c^* \sim \text{vM}(\mu^*, \tau^*)^{R \times H}$. Figure 8.2 illustrates how the coefficients are integrated in the spectral domain. We introduce a *decision dimension*, H, where a is normalized to decide which distribution head/sinusoid it chooses according to the lower resolutions, that is $C = \alpha \cdot \left[\nu \odot \cos(c^*)^\top\right]$, where $\alpha = \frac{\exp(a)}{\sum_j \exp(a_j)}$, and $\nu \in \mathbb{R}^H$ is trainable and resembles voxel values in H. This allows the gradients w.r.t. $\mu^*, \tau^* \in \mathbb{R}^{R \times H}$ to be influenced by $\overline{Z}$. We can now define another version of computing uncertainty, integrated in the neural network of *EEG to fMRI synthesis*, as

$$\hat{\tilde{\mathbf{y}}} = \mathcal{F}^{-1}\left(\text{concat}(\text{concat}(\overline{Z} \odot c, C), \mathbf{0})\right), \forall m_i : i \in \{1, 2, 3\}. \tag{8.13}$$

8.3.3 Neural network agnostic uncertainty quantification

Here we show how the procedure presented in this section is easy to *plug and play* and is neural network agnostic. The idea is that any neural network that performs *EEG to fMRI synthesis*, referred to as F, is flexible to receive heterogeneous EEG (different number of channels, frequency sampling, recording duration) and fMRI setups (number of voxels, voxels size, time reponse). By plugging the DCT spectral coefficients, we are specifying a neural network to predict a lower resolution space (prediction of less voxels/features, consequently relaxing the problem). The work done by the DCT layers is to perturb the DCT spectral domain with coefficients treated as random variables.

Figure 8.3 shows that one just has to append the DCT layers to the neural network. With an automatic differentiation package, such as tensorflow [89], the gradients are easily propagated to θ_F. All this while learning the spectral domain coefficients with $\mu, \sigma, \mu^*, \sigma^*$. As done in the previous chapter, we compare and fit this methodology to our implementation of the Liu and Sajda [123] work, which corresponds to the state of the art.

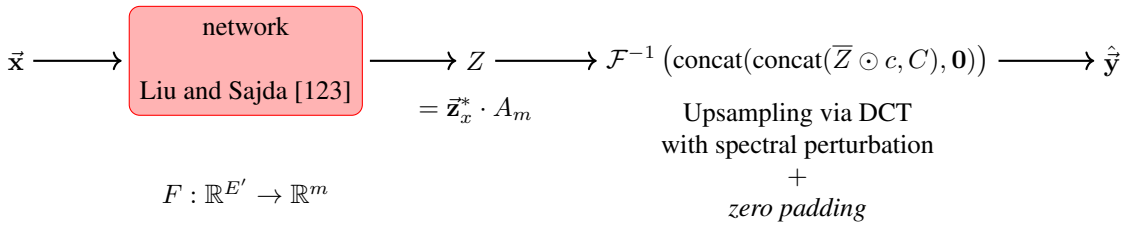


Figure 8.3: How this methodology fits into a network with an easy addition of DCT based layers.

8.4 Results

The results were gathered for two works that constitute the state-of-the-art in *EEG to fMRI synthesis*: Liu and Sajda [123] and Calhas and Henriques [24]. The baselines of the two models are:

- **original**: deterministic versions of the models proposed in both works;
- **proposed zero filling**: prediction of a lower resolution space with $m = 32 \times 32 \times 15$ followed by *zero filling*;
- **stochastic Wen et al. [136]**: reparametrization proposed by Wen et al. [136] with a direct application in the last layers of both works (one dimensional convolution (Liu and Sajda [123]) and an affine layer (Calhas and Henriques [24]));
- **proposed stochastic**: introduction of spectral random variables as illustrated in Figure 8.3.

Model	Liu and Sajda [123]		Calhas and Henriques [24]	
	RMSE	SSIM	RMSE	SSIM
original	0.6252 ± 0.0682	0.4821 ± 0.0000	0.4347 ± 0.0003	0.4302 ± 0.0005
proposed zero-filling	$\mathbf{0.3912 \pm 0.0001}$	$\mathbf{0.5034 \pm 0.0000}$	$\mathbf{0.4092 \pm 0.0000}$	0.4685 ± 0.0531
stochastic Wen et al. [136]	5.0010 ± 8.2006	0.4348 ± 0.0448	0.4544 ± 0.0007	0.4645 ± 0.0001
proposed stochastic	0.5096 ± 0.0385	0.4892 ± 0.0002	0.4097 ± 0.0001	$\mathbf{0.4727 \pm 0.0001}$

Table 8.1: The values reported were retrieved from five runs on the NODDI dataset for five different seeds, corresponding to the first five prime numbers. The first two rows refer to the original deterministic versions and the zero-filling variant for Liu and Sajda [123] and Calhas and Henriques [24]. The last two rows report the quantitative results for the stochastic models. In the latter, the first implements the reparametrization introduced by Wen et al. [136], for Liu and Sajda [123] and Calhas and Henriques [24]. The last row refers to the introduction of spectral random variables.

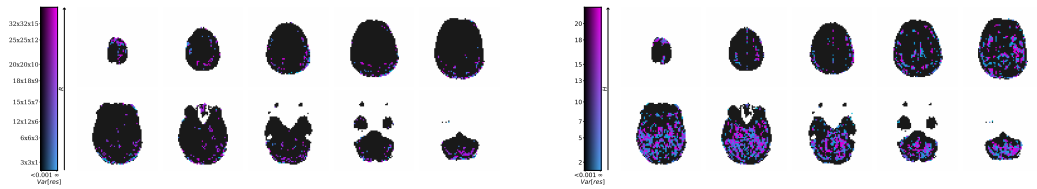
Model	Liu and Sajda [123]		Calhas and Henriques [24]	
	RMSE	SSIM	RMSE	SSIM
original	0.5501 ± 0.0008	0.4696 ± 0.0001	0.5590 ± 0.0001	0.4556 ± 0.0000
proposed zero-filling	0.4923 ± 0.0012	0.5056 ± 0.0003	0.5474 ± 0.0005	0.4703 ± 0.0000
stochastic Wen et al. [136]	6.6363 ± 8.7719	0.3935 ± 0.0017	0.5998 ± 0.0011	0.4329 ± 0.0005
proposed stochastic	$\mathbf{0.4358 \pm 0.0001}$	$\mathbf{0.5136 \pm 0.0001}$	$\mathbf{0.4679 \pm 0.0055}$	$\mathbf{0.4970 \pm 0.0010}$

Table 8.2: The values reported were retrieved from five runs on the CHUR-Xp2 dataset for five different seeds, similar to the setting of Table 8.1. The layout is also the same as Table 8.1.

As in the previous chapter, we report the results with respect to the root mean squared error (RMSE) and structural similarity index measures (SSIM) metrics. The results are shown in Tables 8.1 and 8.2 for the NODDI and CHUR-Xp2 datasets, respectively. In Table 8.1, RMSE improvements are observed for the zero-filling approach and the introduction of spectral r.v.s, both being superior to the deterministic state-of-the-art, defined by Calhas and Henriques [24]. Liu and Sajda [123], considering both deterministic and Bayesian (Wen et al. [136]) versions, were the worst baselines. In this context, we hypothesize that the

structure of the model does not handle well different seed states for a resting state data setting. Overall, RMSE suggests limited ability for uncertainty quantification, under the Wen et al. [136] reparametrization trick. On the other hand, the introduction of spectral r.v.s on the Calhas and Henriques [24] is competitive with zero-filling settings and is superior to the homologous version of Liu and Sajda [123]. According to the SSIM metric, the best approaches are the zero-filling for Liu and Sajda [123] and the introduction of spectral r.v.s for Calhas and Henriques [24]. We observe that the zero-filling approach for the model by Calhas and Henriques [24] shows higher deviation. These experiments also report the superiority of Liu and Sajda [123] against the aforementioned state-of-the-art, which are not in accordance with the claims made in Calhas and Henriques [24]. In contrast, the same does not hold under the reparametrization trick of Wen et al. [136]. In regards to Table 8.2, i.e. CHUR-Xp2 dataset, RMSE reports that the proposed introduction of spectral r.v.s showed the best results, for both Liu and Sajda [123] and Calhas and Henriques [24] models, with the first achieving the best performance. The zero-filling relaxation had the second best results, for the Liu and Sajda [123] it achieved much better results than its original version. The zero-filling Calhas and Henriques [24] model was also better than the original version. The SSIM metric, in contrast with the NODDI dataset, in the CHUR-Xp2 dataset it is in complete accordance with the observations made with the RMSE metric. Both metrics evince Liu and Sajda [123] with zero-filling and Calhas and Henriques [24] with spectral r.v.s, as the new state-of-the-art for EEG to fMRI synthesis task on the NODDI resting state dataset. As for the CHUR-Xp2 dataset, the introduction of spectral r.v.s outperformed every baseline, with the version of Liu and Sajda [123] achieving the best results quantitatively. This suggests that the proposed relaxation is appropriate for resting state settings, promoting a version of limited performance (original Liu and Sajda [123]) to an increase of as much as 37% in predictive power.

We now assess the **impact of the resolution and sinusoidal dimensions**, R and H respectively. The resolution dimension specifies how many coefficients are introduced in the frequency domain. The number of sinusoids specifies how many sinusoids are used to estimate one DCT spectral coefficient. Figure 8.4(a) illustrates the impact of the R dimension in the synthesis task. One can see that the residues did not deviate more than the default set threshold 0.001, meaning that the variation of this parameter was not reflected in the predictive power. In contrast, we found the H dimension to have a bigger impact in the synthesis task, as shown in Figure 8.4(b), with several more voxels being colored. For both analyses, the distribution of their values is uniform, with different values being placed across the brain region. Further, an analysis of

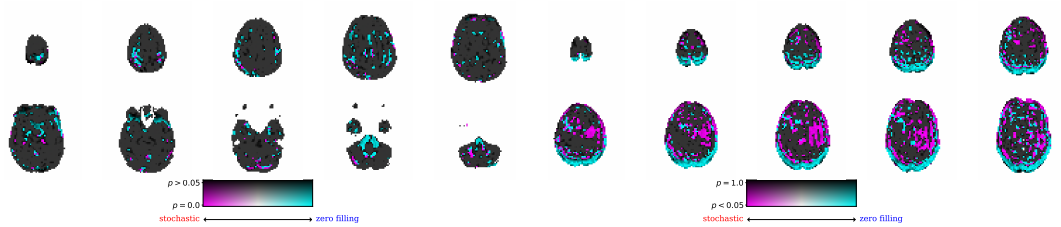


(a) By default the resolution goes from $\forall i \in \{1,2,3\}$: $m_i = \frac{M_i}{2}$ and R stochastic coefficients are introduced in the DCT domain. For this analysis, all of the parameters were fixed, with $m = 32 \times 32 \times 15$ and $H = 15$. Residues were computed for $R \in \{3\times3\times1, 6\times6\times3, 12\times12\times6, 15\times15\times7, 18\times18\times9, 20\times20\times10, 25\times25\times12, 32\times32\times15\}$.

(b) This tells us how important is the H parameter for the synthesis task. The parameters fixed to analyze the H parameter were $m = 32 \times 32 \times 15$ and $R = 32 \times 32 \times 15$. Residues were computed for $H \in \{2, 5, 7, 10, 13, 15, 18, 20\}$.

Figure 8.4: Plot of the residues, w.r.t. the NODDI dataset, of different values for the variables R and H, for the Calhas and Henriques [24] model.

the effect of introducing stochastic r.v.s in the DCT spectral domain and filling it with zeros is reported in Figure 8.5(a) for the Calhas and Henriques [24] model in the NODDI dataset. The same analysis is made for the CHUR-Xp2 dataset in Figure 8.5(b). The illustrative comparison of Liu and Sajda [123], in NODDI, does not support the superiority observed in the quantitative results, suggesting that indeed zero-filling is superior, but without statistical significance. Slightly more noticeable is the superiority of the zero-filling illustrated in Figure 8.5(a) for the thalamus region. Both figures report voxels with highlighted statistical significance differences between the approaches, found to be rather sparse across the brain region. For the CHUR-Xp2 dataset zero-filling shows statistical significance estimates in the occipital lobe. Specifically, Liu and Sajda [123] with the introduction of r.v.s showed statistical significance in regions along the brain, except the occipital lobe. From the baselines considered, two of them are able to quantify uncertainty,

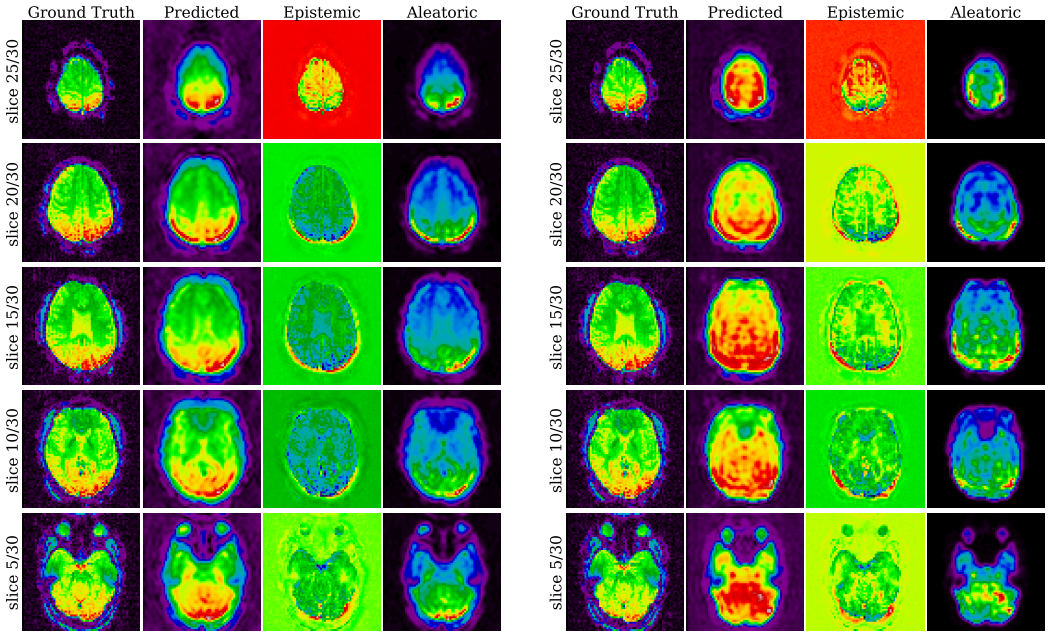


(a) Comparison on the Calhas and Henriques [24] model, for the NODDI dataset.

(b) Comparison on the Liu and Sajda [123] model, for the CHUR-Xp2 dataset.

Figure 8.5: Comparison between the zero filling procedure and the methodology introduced in this study. These figures provide a view, voxel by voxel, of the statistical differences on the significance of estimates produced with stochastic r.v.s and zero-filling the frequency space. Magenta voxels means stochastic r.v.s are superior for that voxel, whereas cyan voxels mean filling with zeros is better. White regions represent statistical significance but no superiority and black regions report no statistical significance.

due to their Bayesian nature, namely: *stochastic Wen et al. [136]* and *proposed stochastic*. However, the models that use the reparametrization trick introduced by Wen et al. [136] did not perform well, falling short to their deterministic versions. In contrast, introducing spectral r.v.s in Calhas and Henriques [24] model showed superiority against deterministic counterparts. The same approach for the Liu and Sajda [123] did not show as good **qualitative** results, as shown in Figure 8.6(b). In this section, we report on the ability of the stochastic variant of Calhas and Henriques [24] to quantify uncertainty. Figure 8.6 shows the *ground truth*, *predicted*, *epistemic* and *aleatoric uncertainty* side by side. There are two main takeaways from analyzing the figure: 1) the predicted volume shows less delineation relative to the ground truth, which means there is considerable difficulty in retrieving high resolution coefficients; 2) the epistemic uncertainty plot shows that there is uncertainty outside the brain region. Regarding the first point, we hypothesize that the coefficients being computed start the learning process by converging to zero, leaving to the lower resolutions the responsibility of fitting the data to minimize the objective (Equation 8.4). Indeed, the Karhunen–Loève transform (expansion) can be used to reduce the number of spectral coefficients to approximate images with high-resolution. Here, we make the bridge between this fact and the observed blur to conclude that the lower resolutions are given higher prevalence to approximate the ground truth. Following, the computed epistemic uncertainty leads us to believe that the neural network struggles to predict correctly the background of the fMRI volume, i.e. the outside of the brain regions. Figure 8.6(a) shows that there is a relatively high uncertainty present in those regions. The latter also explains the SSIM metrics computed and may motivate the use of segmentation techniques [137] to correct this issue. The



(a) Calhas and Henriques [24].

(b) Liu and Sajda [123].

Figure 8.6: Synthesized fMRI volume, corresponding to the output of introducing spectral r.v.s., for an instance of the NODDI dataset.

aleatoric uncertainty shows that there are higher degrees of uncertainty in the occipital lobe, which may be due to the recording protocol in the NODDI initiative considering an eyes-open setting.

8.5 Discussion

The number of sinusoids H impacts the synthesis task. In the previous section, we analyzed the impact of both the number of placed spectral r.v.s, R, and the number of sinusoids used to estimate each coefficient, H. The results suggest that H has more importance for the task, as is illustrated in Figure 8.4(b). The latter, may also suggest that a higher number of parameters allows the model to represent a higher number of possible combinations for the synthesized volume. Indeed, the applied attention schema allows the model to learn style signatures without having to apply multiple heads [138], that corresponds to a $b_i \times \cos(c_i^*)$, which can also be seen as a decision boundary (for style). The attention mechanism, applied in the H dimension, exerts the softmax activation, giving more importance to specific sinusoids. This means, that it is not precisely the sum of H sinudoids, but rather the fact that each sinusoid is learning a different style function [139], that is chosen by the attention mechanism based on the lower resolution coefficients, X. Grounded on the acquired results, we claim that the higher the number of sinusoids, the better the synthesized volume.

Zero-filling relaxation allows a better synthesis. The main contribution of this study is the integration of a DCT mechanism to upsample a low resolution volume to a higher resolution. Which allows the approximation to the target with a lower resolution. This type of relaxation is commonly used in compression algorithms [132] and has also been used in machine learning methods by Flynn et al. [140]. We differ from these works, because we do not predict patch by patch and instead predict a entire lower resolution volume. This is not seen as a pitfall, since the full resolution of fMRI is not retrievable by the characteristically low

EEG spatial resolution information [9]. In the experiments, two versions of the DCT approach are compared: the zero-filling and the introduction of spectral r.v.s. The first is hypothesized to relax a problem characterized by a high number of features to perform regression on the $\prod_i M'_i = 64 \times 64 \times 30 = 122880$ voxel space. All in the presence of a limited set of observations to learn from. We observed that this relaxation allowed both models, Liu and Sajda [123] and Calhas and Henriques [24], to have a better performance than their original versions. The zero-filling Liu and Sajda [123] model defines the state-of-the-art w.r.t. the quantitative metrics in resting state and motor imagery settings. Naturally, both models used less decoder parameters to predict the low resolution volume, decreasing the features by observations ratio, which in consequence partially tackles the curse of dimensionality [126]. Further, it is known that spectral smoothing allows a decrease of gradient variance [141], being another advantage of relaxing the problem. The second version of our approach, which yields the advantage of being able to quantify uncertainty, is generally competitive with the zero-filling one, surpassing the zero-filling in Calhas and Henriques [24] for the SSIM metric in resting state.

There are limits to the ability of quantifying uncertainty. The epistemic uncertanity, illustrated in Figure 8.6 shows background variance. This is a consequence of our approach, because it encompasses the estimation of spectral r.v.s for the whole image, i.e. the variability of the high coefficients affects both the brain region and the background. The same phenomenon does not affect aleatoric uncertainty, clearly showing that the uncertainty inherent in the data pertains to the brain region, with higher uncertainty in the occipital lobe. We claim that the spectral approach allows the best uncertainty quantification, yet is still limited when it comes to epistemic uncertainty as observed by the affected regions outside of the brain. Though, the use of segmentation mask can be placed to correct this problem. One candidate mask would be the 3D-UNet [137], the state-of-the-art in medical imaging segmentation. Further, aleatoric uncertainty could be used as information to assess which regions of the volume had more variation and better assess the prediction, similar to the approach done by Hemsley et al. [142]. Note that, there is disagreement between epistemic and aleatoric uncertainty on the occipital lobe, with the epistemic uncertainty reporting low values, whereas aleatoric uncertainty reports the highest levels of uncertainty in that region. The latter tells us that, aside from the uncertainty inherent in the data being high in the occipital lobe, there is agreeable significance on the predictions for this region.

8.6 Summary

- The neural network proposed considers spectral features at the latent space, with the decoder recovering the spectral coefficients and the sinusoids being the latent representation. This is directly compared to the discrete cosine transform, which is used to quantify uncertainty;

- Two uncertainty quantification methods are introduced: placing random variables in the affine decoder layer (spectral coefficients), and placing random variables in the appended DCT spectral domain of a low resolution synthesized fMRI volume;

- We showed the benefits of relaxing the complex *EEG to fMRI synthesis* task, generally characterized by a high number of targets with intricate spatiotemporal dependencies and by limited paired EEG-fMRI observations, allowing the learnt models to increase their predictive power;

- Our results suggest that appending the proposed DCT layers assists standard convolutional (Liu and Sajda [123]) and/or affine (Calhas and Henriques [24]) transformations using the reparametrization trick [130].

Chapter 9

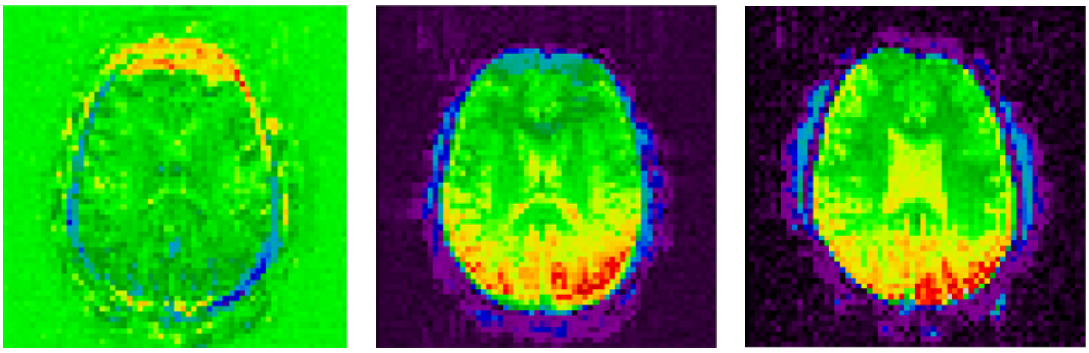
Discriminative insights

The methodological aspects of this thesis can be segmented in two main components: the feature extraction (encoder) and the regression task (decoder). For the encoder we focus on the number of layers, the sizes of the kernels, and the strides of each resnet block. For the decoder, the main focus is given to the equation that predicts the fMRI,

$$\hat{\vec{\mathbf{y}}} = A_M \cdot (cos(\omega \cdot \vec{\mathbf{z}}_x + \beta) \odot \vec{\mathbf{z}}_w)\,, \tag{9.1}$$

with the previous chapter focusing on changing the decoding matrix A_M. This chapter, on the other hand, focuses on the ability of separating in a sub domain interval $\forall i, j \in \{1, \ldots, S\} : \big(d = m * ||\vec{\mathbf{y}}_i - \vec{\mathbf{y}}_j||_1^1\big) \wedge \big(d = \omega \cdot \vec{\mathbf{z}}_{x_i} + \beta - (\omega \cdot \vec{\mathbf{z}}_{x_j} + \beta)\big) \wedge \big(\frac{d}{d} = cos(\omega \cdot \vec{\mathbf{z}}_{x_i} + \beta) - cos(\omega \cdot \vec{\mathbf{z}}_{x_j} + \beta)\big)$, where $\frac{d}{d} = 0, d = 0$ and $m \in]0, \pi]$ is a separation margin. Why do we look at this? The main goal of this thesis is to provide a lower cost and ambulatory diagnostic that provides an interpretability level equivalent to that of fMRI, through the cheaper EEG modality. Until now, we showed in chapter 7 the ability of several models (Liu and Sajda [123] and Calhas and Henriques [24]) to synthesize fMRI. However, this synthesized signal is yet to be tested in a classification setting, where **only EEG recordings** are available. None of the published (including Calhas and Henriques [24], Liu et al. [63], Liu and Sajda [123], Bricman et al. [143]) studies have shown how the synthesis is applied in a diagnostic setting, despite their acute motivation ground towards this end. We hypothesize that a discriminative synthesized fMRI modality is produced by first *learning the fMRI representation* and subsequently *learning the discriminative properties for target classes*. This is done by first learning the fMRI representation from a simultaneous EEG and fMRI dataset. Second, EEG recordings paired with pathology and control groups are considered to learn a classifier. The latter, allows the experimental setting to be as close to a real life health care application. Note that the introduction of new datasets challenges the learned fMRI representations, since differences can be present in:

- number of EEG channels;
- distribution of the EEG channels across the scalp;
- sampling frequency;
- recording duration;
- preprocessing techniques applied.



Training the parameters of the network along with the classifier

Training the parameters of the classifier with network fixed

Ground truth

Figure 9.1: When we minimize the cross entropy with respect to the parameters of the classifier along with the parameters of the neural network that performs synthesis (Calhas and Henriques [24] was used for this demonstration), the style of the fMRI is lost as is illustrated on the figure on the left. Note that, the performance of the classifier, given this view in a test set, was of 0.93 sensitivity, 1.0 specificity and 1.0 AUC. At the center, the synthesized fMRI is produced after only optimizing the parameters of the classifier. The latter, in terms of performance achieved 0.0 sensitivity, 1.0 specificity and 0.63 AUC. On the right, a ground truth example instance from the NODDI dataset is placed to serve as a reference, since the neural network is pretrained on the NODDI dataset before being trained for the classification task. In terms of quality, the center volume, which only optimizes the classifier, achieves the best quality when compared to the ground truth.

This heterogeneity is resembled between the EEG data of a simultaneous EEG and fMRI dataset, where EEG is distributed as $X_1 \sim \mathcal{X}_1$, and the EEG data of a EEG-only dataset, where $X_2 \sim \mathcal{X}_2$. The fact, that the distributions of these datasets are different $\mathcal{X}_1 \nsim \mathcal{X}_2$, disrupts the learned fMRI representation after computing the parameters of a classifier that minimize the cross entropy loss (for classification). Two additional problems are introduced:

- if we **do not** train the parameters of the synthesis model along with the parameters of the classifier the style of the fMRI is maintained but the data is not separated and the classifier is not able to separate the data;

- if the parameters of the synthesis and classification models **are** simultaneously trained, the style of the fMRI is lost, but the classifier is able to separate the data.

The latter is illustrated in Figure 9.1. There is balance between keeping the fMRI representation style and being able to classify the fMRI synthesized view. We know the problem is targetable since the optimization with the parameters of the neural network achieve good results, according to the metrics reported.

In this chapter, we provide insights on how to tackle the problem illustrated in Figure 9.1, and consequently assess the value of the synthesized fMRI for diagnosis. We start by defining the views (see section 9.1) that have been a foundation for this thesis: the raw EEG (see section 9.1.1) and the STFT views (see section 9.1.2). These views have been used for the synthesis of fMRI which itself provides an additional synthesized fMRI view (see section 9.1.3). After describing the selected views, we formulate the problem as a classification task for each view defined (see section 9.1.4). In addition, we discuss why a linear classifier is a suitable method to address the discriminative properties of each view. In section 9.2, we present a methodology that is hypothesized to enable the classification of an fMRI signal, while enabling F to preserve the fMRI style. Section 9.3 describes the experimental setting applied, which includes a

description of the data used, the validation schema applied and a description of *biclustering* [144] as an analysis tool for discriminative and interpretable pattern mining. Section 9.4 provides the results obatined from the experiments done. Following, section 9.5 gives a thorough interpretation of the results, relating it with previous studies that discriminate the pathology considered for the experiments. The final section 9.6 provides a summary of the chapter with the main takeaways.

9.1 How we view EEG for classification

Two major qualities are highlighted for neuroimaging-based diagnostics: *discrimination* and *interpretation* ability. We start by considering the view in which EEG is recorded, that is its *raw* form. Following, the time frequency *stft* representation of EEG is also considered an EEG view. The latter is known for its wide use in classification tasks [145–150]. The product of this thesis, the synthesized fMRI modality, is considered as a view of EEG, given its projection using F from the *stft* view.

Given these *views* of EEG, we want to assess the ability of each to classify data that is grouped (e.g. as healthy controls and a pathology group). In an optimal setting, we would evaluate all functions, $f_C^j : \mathbb{R}^{E_j} \rightarrow \mathbb{R}^c$, from the set of possible classifiers, $\mathcal{C}$. In a mathematical form this is reduced to

$$r_j = \int_{\mathcal{C}} \sum_{i}^{S} \mathbb{1} \left[f_C^j(v_j(\vec{\mathbf{x}}_i)) == y \right], \tag{9.2}$$

where r_j would denote the prediction power of a view j, $\vec{\mathbf{x}}_i \in \mathbb{R}^{E_i}$ is the EEG original (recording) representation with E_i denoting its structure (i.e. channels and recording duration), $v_j : \mathbb{R}^{E_i} \rightarrow \mathbb{R}^{E_j}$ being the view function that maps the original EEG view to the a specified one (e.g. for this thesis $v_j = F$), and $y \in \{0, 1\}^c$ being the ground truth for the number of classes, c, of the dataset. The question now becomes: *how can we estimate a feasible and faithful $\hat{r}_j$, that may not approximate the unknown r_j, for a view j?* We hypothesize that a linear classifier is a suitable f_C for this estimation. Previous works have shown how a linear classifier is able to capture knowledge without manipulating the feature representation and therefore enabling the quality evaluation in explainability methods [151]. We address this in a later section, first we provide a description of each view considered.

9.1.1 *raw* view

In its raw form, an EEG recording consists of a set of electrodes that contain the electrical activity present at the scalp. This activity has its source at the neurons, where the action potentials occur. Formally this view is defined as $v_0 = \{F_0, \theta_0\}$, where $F_0 = I \cdot \vec{\mathbf{x}}_0$ and $\theta_0 = I \in \mathbb{R}^{D_0}$. With EEG being a multivariate time series representation, researchers are able to study functional properties of the brain [50]. However, the application of this specific view is done mainly for epilepsy detection [152]. Structurally, this representation is defined as $\vec{\mathbf{x}}_0 \in \mathbb{R}^{E_0}$, with $E_0 = C \times T$. C stands for the number of channels and T defines the temporal dimension. It is worth noting in this view, that regarding explainability, one is limited to the channel-temporal features, characterized by noise and difficult medical interpretation [9]. In consequence, this view is unsuitable for diagnosis.

9.1.2 *stft* view

Research studies, along the years, have had a common trend of extracting time frequency features from the EEG signal. Several findings have been made using the frequency domain, from functional connectivity relations [153–155] to statistical significance of bands associated with pathologies [156, 157]. A time frequency representation of EEG can be achieved via a transform [158]. In this study we consider the short time Fourier transform (STFT) [159], whose kernel consists on sine and cosine waves with different shifts and frequencies. As such this view is defined as $v_1 = \{F_1, \theta_1\}$, where F_1 is the sum of sinusoids and θ_1 are the sinusoids themselves. Similarly to *raw*, this view is a multivariate time series representation, with the channel, C, and temporal, T, dimensions belonging to its structure. On top of that, each frequency is represented at different time steps and channels, with the dimension F. Hence, the function consists of $F_1 : \mathbb{R}^{E_0} \rightarrow \mathbb{R}^{E_1}$ and the *stft* view is structurally defined as $\vec{\mathbf{x}}_1 \in \mathbb{R}^{E_1}$, where $E_1 = C \times F \times T$. On the interpretation side, we consider it not just similar to the *raw* view, but also more limited due to the requirement of frequency domain knowledge.

9.1.3 *fmri* view

For the *fmri* view, we describe the top component (relative in Figure 7.4) of the neural network that performs the *EEG to fMRI synthesis*. Its input is the EEG time-frequency representation view, $\vec{\mathbf{x}}_1$. On the opposite side, the output is the *fmri* view, $\vec{\mathbf{x}}_2 \in \mathbb{R}^{M_1 \times M_2 \times M_3}$, characterized by M_1, M_2 and M_3, the referential axes dimensions. Zooming out of the neural network, we can describe this view as $v_2 = \{F_2, \theta_2\}$. $F_2 = F_1 \bigcup F$ denotes the union of two functions F_1, the short time Fourier transform view $v_1 = \{F_1, \theta_1\}$, and F, the neural network. Accordingly, the parameters $\theta_2 = \theta_1 \bigcup \theta_F$, where θ_F are the parameters of the neural network. All the parameters until the last layer are referred to as θ_E. This is the encoder of the neural network. Continuing, after T is processed by the encoder, it is then used to project random Fourier features, parametrized by ω and β. Before decoding, the sinudoids are multiplied with a learned latent style vector, $W \in \mathbb{R}^{L^*}$, as $cos(\omega \cdot \vec{\mathbf{z}}_x + \beta) \odot \vec{\mathbf{z}}_w$. The result is processed by an affine transformation, mapping it to the fMRI volume space $\mathbb{R}^{M_1 \times M_2 \times M_3} : E_2 = M_1 \times M_2 \times M_3$. With this, we formulate the function structure $F : \mathbb{R}^{E_1} \rightarrow \mathbb{R}^{E_2}$.

9.1.4 Problem description

Let $\mathcal{V} = \{v_1, \ldots, v_n\}$ be a set of different views, such that $\forall i \in \{1, \ldots, n\} : v_i = \{F_i, \theta_i\}$. A view v_i is characterized by a function structure, F_i, and its parameters, θ_i. Each projection $F_i : \mathbb{R}^{E_0} \rightarrow \mathbb{R}^{E_i}$ is performed from the original feature space, $\mathbb{R}^{E_0}$, to its view space, $\mathbb{R}^{E_i}$. The original view is defined as $v_0 = \{I \cdot \vec{\mathbf{x}}_0, I \in \mathbb{R}^{E_0}\}$, being I the identity matrix. Each instance, $\vec{\mathbf{x}}$, is paired with a label $y \in \{0, 1\}^C : \sum_j y_j = 1$, where C is the number of classes described in the data. A view, v_i, optimizes its parameters, θ_i, in order to minimize

$$\mathcal{L}_C(\vec{\mathbf{x}}_0, y) = -y \times \log(\sigma(W_f^i \cdot v_i(\vec{\mathbf{x}}_0) + b_f^i)), \tag{9.3}$$

where $W_f^i \in \mathbb{R}^{C \times D_i}$ and $b_f^i \in \mathbb{R}^C$ are the parameters of a linear classifier, referred to as $\theta_C^i = W_f^i \bigcup b_f^i$, using the softmax activation, σ. Then the objective, for each view i, is defined as

$$\text{argmin}_{\theta_C^i} \mathcal{L}_C(\vec{\mathbf{x}}_0, y). \tag{9.4}$$

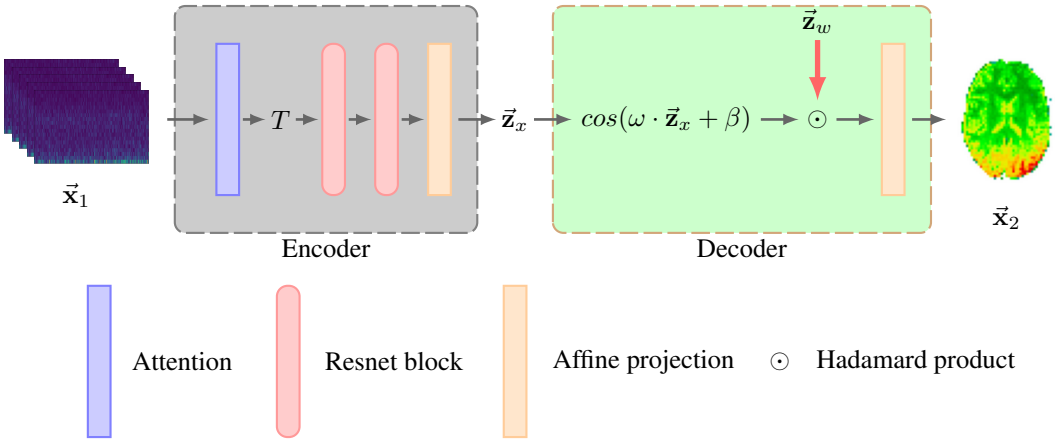


Figure 9.2: The neural architecture has two components: an Encoder (shaded in grey) and a Decoder (shaded in green). The input is the *stft* v_1 representation. The output is the synthesized *fmri*. The Encoder begins with a simple attention mechanism on the channels dimension of the *stft*. After, it is processed by two Resnet blocks and an affine layer. This produces the latent representation $\vec{\mathbf{z}}_x$. Following, comes the Decoder, which picks this representation and builds the cosine bases through the projection $\omega \cdot \vec{\mathbf{z}}_x + \beta$. The sinusoids are style induced with $cos(\omega \cdot \vec{\mathbf{z}}_x + \beta) \odot \vec{\mathbf{z}}_w$. W is a style fixed pretrained vector of an *fmri* representation, learned from a simultaneous EEG and fMRI dataset. Finally, an affine layer projects it to the *fmri* space.

9.1.5 Why a linear classifier?

In a nutshell, we are addressing the predictive power of three views: v_0, v_1 and v_2; while looking trivial, we showed through equation 9.2 that this problem is exponential. While, we can not say which one is truly the best, we can experiment given a simple classifier which one has the most informative features. We do not consider a non linear classifier, suuch as XGBoost, because a view is being evaluated on its discriminative and interpretability power. If a view is able to be classified using a linear classifier, then the interpretability of the feature space is high and therefore useful in diagnostic settings. As previously mentioned, Treviso and Martins [151] has shown how a linear classifier is suitable to evaluate views that should be easily interpreted by a human expert. In the latter, the views consisted of a subset of the original set of features, that were able to explain the prediction of a model.

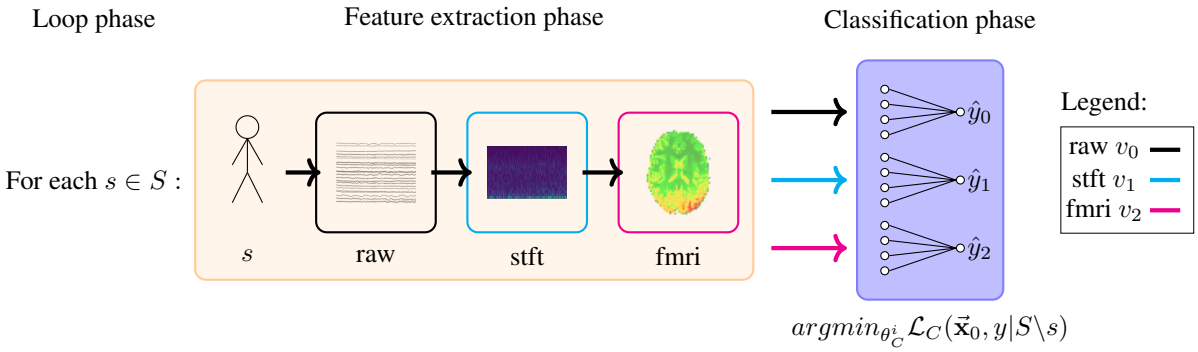


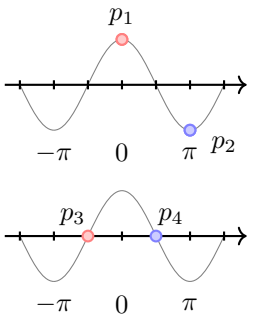
Figure 9.3: We do a leave-one-individual-out validation, where for each fold we either train a linear classifier with *raw*, *stft* or *fmri* representations. Each representation has its own validation. The arrows inside the feature extraction phase indicate dependency, that is: an *fmri* representation needs an *stft*; *stft* needs the *raw*; and the *raw*, of course, needs an individual's recording, denoted with a human figure. For each fold, $s \in S$, we train a linear classifier without s, $argmin_{\theta_C^i} \mathcal{L}_C(\vec{\mathbf{x}}_0, y|S\backslash s)$. For all individuals/folds the predictions are saved to compute the are under the curve (AUC) against the ground truth.

Consider a set of individuals S, where each individual, $s \in S$, is represented by the original EEG view $\vec{\mathbf{x}}_{0,s}$ and a target label $y_s \in \{0,1\}^c$, with c being the number of classes. Due to neuroimaging data high inter individual variability, as well as the single individual diagnostic happening in a real health care setting, a leave-one-individual-out validation schema is applied. This means that one trains a linear classifier parameters (along with the respective view parameters, in the case of v_2) with all the individuals except s and predict the target of individual s, $\hat{y}_s$, at test time. We consider the best view the one that more accurately predicts the ground truth $y_s, \forall s \in S$. This evaluation methodology is illustrated in Figure 9.3.

9.2 Learning to classify while synthesizing fMRI

We hypothesize that the reason, why the classification task either corrupts the fMRI style or is not able to classify (phenomena illustrated in Figure 9.1), is due to the fact that sinusoids project different domain points to the same image. Consequently, it destroys possible separation that is either made by the encoder of the Calhas and Henriques [24], $y|\vec{\mathbf{z}}_x$, or that already exists in the *stft* view, $y|\vec{\mathbf{x}}_1$. How can we maintain separability while keeping the learned fMRI style? We propose locking the learned A_M projection and stop the gradients $\nabla_{\theta^i_C} \mathcal{L}_C$ at the linear classifier. For the sake of separability, we introduce a regularization term to $\omega \cdot \vec{\mathbf{z}}_x + \beta$, which enables the encoder of the EEG representation to learn the separation. This regularization term is described in section 9.2.1. Consider that, for learning sake, the EEG modality contains information to separate the data, $\vec{\mathbf{x}}_1$, according to its labels, y. Though, if this representation has its distribution broken at the sinusoids, we can no longer use it. To avoid this, we propose a method that maintains the sinusoids in the neural architecture flow and manipulates the representation, so that the separation before the sinusoid, $y|\omega \cdot \vec{\mathbf{z}}_x + \beta$, is identically distributed to the representation after the sinusoids, $y|\vec{\mathbf{x}}_2$.

9.2.1 Sinusoid separation



$$p_1, p_2 \sim \mathcal{X}_1 : ||p_1 - p_2||_1 = \pi$$

$$p \sim \mathcal{X}_1 : \mathbb{E}[p] = \tfrac{\pi}{2} \wedge \text{Var}[p] = \tfrac{\pi}{2}$$

$$p_3, p_4 \sim \mathcal{X}_2 : ||p_3 - p_4||_1 = \pi$$

$$p \sim \mathcal{X}_2 : \mathbb{E}[p] = 0 \wedge \text{Var}[p] = \tfrac{\pi}{2}$$

Figure 9.4: Description of how two similarly distributed samples, taken from $\mathcal{X}_1$ and $\mathcal{X}_2$, can lead to different portions of the cosine function image, since a sinusoid is periodic. Two distributions $\mathcal{X}_1$ and $\mathcal{X}_2$, may be mapped to the same image (second/bottom example). This is why a cosine is a shift invariant function. However, there are intervals a shift can be made and it is not invariant. Such intervals take the form $\forall i \in \mathbb{Z} : [i\pi, (i+1)\pi]$.

In order to separate data along the projection of cosines, we have to operate in a sub-domain interval where the *cos* is not periodic. Layer normalization [160], with center in $\frac{\pi}{2}$ and standard deviation of $\frac{\pi}{2}$, maps most of the data to such an interval of the cosine. See Figure 9.4 for an illustration example. Nonetheless, the assurance that the projections are likely in a non periodic sub-domain of the cosine function, does not alone complete separation of the data. This is because the distribution $y|\omega \cdot \vec{\mathbf{z}}_x + \beta$ needs to be separated

accordingly. We propose pairwise learning with a modified contrastive loss [161] defined as

$$\mathcal{L}_D(p_1, p_2, y_p) = y_p \times D(p_1, p_2) + (1 - y_p) \times ||D(p_1, p_2) - m||_1, \tag{9.5}$$

where p_1, p_2 formulate a pair of two instances derived from their respective view $\vec{\mathbf{x}}_1$ and $y_p \in \{0, 1\}$ defines the pairwise label, being 0 when the label of p_1, p_2 mismatch and 1 when they match. $D(p_1, p_2)$ is a distance function, which we formulate as the $l1$ distance as $D(p_1, p_2) = ||p_1 - p_2||_1^1$. Setting $m \in [\epsilon, \pi] : \epsilon > 0$, together with the described layer normalization, allows separation of the data within a non periodic sub domain of the cosine. In Figure 9.5, the effects of this methodology are illustrated, for $m = \pi$, before and after the learning session.

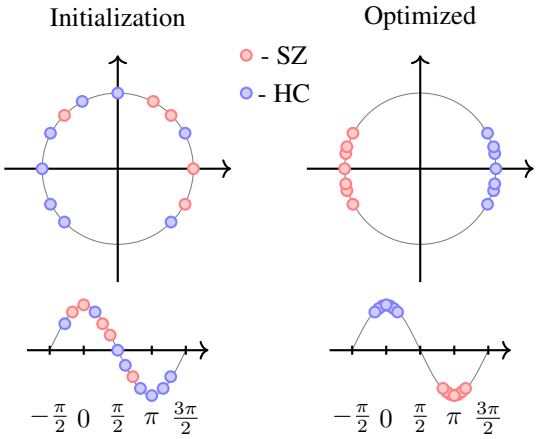


Figure 9.5: Normalization of data points inside the unit circle, using layer normalization, along with the optimization of a contrastive loss lead to correct separation of sinusoids. Data points belong to two classes, HC and SZ, that are separated after the minimization of $\mathcal{L}_D$. Because we separate false pairs, according to $(1 - y_p) \times ||D(p_1, p_2) - m||_1$, all points are placed within a shift variant interval of the cosine. The variance needed for classification.

To minimize the contrastive loss, we need to build pairs of instances. In order to avoid imbalanced data, we make a pairwise dataset, $\mathcal{D}_p = \{i, j\}^{|S|}$, such that pairs are chosen as $i, j \sim \mathcal{U}(1, \ldots, |S|)$ in a way that the number of positive pairs and negative pairs is equivalent, meaning $\sum 1[y_i = y_j] = \sum 1[y_i \neq y_j] = \frac{|S|}{2}, \forall p = \{i, j\} \in \mathcal{D}_p$. The loss that optimizes the parameters for the v_2 view is the addition of $\mathcal{L}_C$ for the two instances with $\mathcal{L}_D$ as

$$\mathcal{L}(\vec{\mathbf{x}}_{0,i}, \vec{\mathbf{x}}_{0,j}, y_i, y_j) = \mathcal{L}_C(v_2(\vec{\mathbf{x}}_{0,i})) + \mathcal{L}_C(v_2(\vec{\mathbf{x}}_{0,j})) + \mathcal{L}_D(\omega \cdot \vec{\mathbf{z}}_{x_i} + \beta, \omega \cdot \vec{\mathbf{z}}_{x_j} + \beta, 1[y_i = y_j]), \tag{9.6}$$

where $\vec{\mathbf{x}}_{0,i}$ and $\vec{\mathbf{x}}_{0,j}$ refer to the raw view of instance i and j, which formulate a pair of instances, $\vec{\mathbf{z}}_{x_i}$ refers to the latent representation of instance i.

9.3 Experimental setting

For the validation of the methodology, as well as the whole premise behind *EEG to fMRI synthesis*, we find schizophrenia to be a good use case. This is a neurological condition that affects a significant portion of the world population [162]. Yet, to this day, the diagnosis of schizophrenia generally involves several clinical tests, making it time consuming and exhausting for the patient. Diagnostic tests range from psycho-

logical symptoms to neuroimaging [163], molecular [164], and natural speech markers [165]. In particular, functional magnetic resonance imaging (fMRI) scans have shown potential for an automated diagnostic [163, 166–168], still their availability is limited. For instance, Ogbole et al. [10] report the density of MRI machines in Africa, claiming Nigeria to be the most critical country, where 0.3 MRI machines are made available for each one million people. This means, if you live in Nigeria and are being diagnosed for schizophrenia, you will need to wait in a 3.3 million people queue. As a consequence, the quality of these health care systems is low. And not only is MRI important for early diagnostics, but it also avoids unnecessary costs [11], such as surgery intervention in some cases. However, the reality is that MRI machines are expensive, and so are MRI sessions. How can alternative cheaper modalities be considered as a proxy to replace MRI machines? We make an attempt at answering this question through the validation of the proposed synthesis in a dataset with schizophrenic individuals and healthy controls.

Each view considered, v_0, v_1 and v_2, has advantages and disadvantages in different settings. Consider v_0 which is useful to diagnose epilepsy episodes [152]. However, its discriminative power of other pathologies is limited even when processed by models with a high level of feature engineering [147]. Falling short to other views [169]. In spite of that, one useful trick is to take the STFT of each channel and analyze the time frequency domain, v_1. There is a variety of studies that report statistical significance of frequency band that are correlated with schizophrenic individuals [156, 157]. On the other hand, v_2 is hypothesized to balance the interpretability and discriminability, lacked by v_0 and v_1. Though no previous studies have assessed the quality of EEG to fMRI synthesis in decision making settings, such as schizophrenia classification.

9.3.1 Validation

To validate the hypotheses drawn, each view, v_i, will do a validation process, as illustrated in Figure 9.3. To this end, the following is done in the validation process:

1. leave-one-individual-out cross validation (LOOCV);

2. cross validation (5 folds) hyperparameter optimization with 25 iterations for each fold of the LOOCV step;

3. particularly for fMRI synthesized views, v_2, a pretraining session is done, where F is optimized on simultaneous EEG and fMRI data.

The hyperparameters used to train F are in accordance with the original work [24]. For the cross validation (step 2.), Bayesian optimization [94] is performed and the hyperparameters subject to optimization are: $l1$ regularization constant $\in [1\text{e}-10, 2.0]$, learning rate $\in [1\text{e}-5, 1.0]$ and batch size $\in \{1, 2, 4, 8, 16, 32\}$. The learning session is fixed with 10 epochs and gradients are propagated using Adam optimizer [170]. Further, we also want to assess how v_2 behaves without the contribution of $\mathcal{L}_D$ and being optimized with the $\mathcal{L}_C$ alone.

9.3.2 Biclustering

We ran BicPAMS [144] to find cluster subspaces composed of a subset of rows (individuals) and columns (voxels) that discriminate a target (pathology or healthy). Two settings are considered: 1) clustering $\vec{\mathbf{x}}_2$

features with ground truth labels y; and 2) clustering $\vec{\mathbf{x}}_2$ features with its linear classifier predictions $\hat{y} = \sigma(W_f^2 \cdot \vec{\mathbf{x}}_2 + b_f^2)$. This two settings allow us to find discriminative and explainability patterns from the synthesized fMRI view.

Discriminative patterns provide us sets of individuals and features that support a certain target. This analysis is similar to classification, but instead of having a classifier, we have a set of association rules. In turn, *Explainability* patterns give us interpretable information about the decision made by the linear classifier. Although it does not provide discriminative patterns, it gives us association rules for the decision making process of the classifier. Jointly, these settings allow us to have a better comprehension from the synthesized fMRI view, potentially able to uncover advantages and pitfalls. An important analysis when working with data driven projections.

Biclusters were found in three resolutions: $5 \times 5 \times 3$; $10 \times 10 \times 5$; and $14 \times 14 \times 7$. These resolutions enable us to assess patterns at different granularities. In a $5 \times 5 \times 3$, clusters represent big regions of the brain, as big as entire lobes; a $10 \times 10 \times 5$ resolution can still retrieve regions of interest, but at a finer granularity; and $14 \times 14 \times 7$ goes even more detailed. Altogether, this analyses give us patterns, with statistical assurances for a target (schizophrenia). The analysis of different resolutions allows the retrieval of contiguous regions as well as sparsely located voxels. These biclusters can be interpreted as assessing which areas of the brain are associated with schizophrenia. In a sense, EEG, which is recorded at the scalp level, is projected to an fMRI, where sub-cortical activity patterns are retrieved. Therefore, EEG to fMRI Synthesis enriches the EEG modality with fMRI learned features [24]. This additional information increases interpretability, which we measure through a pattern discovery algorithm.

Parameter	$5 \times 5 \times 3$		$10 \times 10 \times 5$		$14 \times 14 \times 7$	
	y	$\hat{y}$	y	$\hat{y}$	y	$\hat{y}$
lift	1.3	1.3	1.3	1.2	1.3	1.2
# biclusters	100	100	100	100	100	100
# bins	3	3	5	5	5	5
min # voxels	1	1	10	10	15	15

Table 9.1: Parameters for the biclustering algorithm.

The parameters given to BicPAMS to search for biclusters in the different resolutions are shown in Table 9.1. Note that, for the lowest resolution $5 \times 5 \times 3$, the number of *columns*, which translates to the number of voxels is only 1. This is because 1 voxel in a $5 \times 5 \times 3$ corresponds to a big region in the original resolution, $64 \times 64 \times 30$. As the resolution increases, we also increased the number of voxels required to appear in a bicluster. The number of *voxels* pretains to the discretization of the data. It is important to not increase this parameter, as the *items-boundary* problem can arise [171]. This problem manifests when the number of *bins* is high and similar values are put in different bins. The number *biclusters* requires that all biclusters obtained are bigger than this value. The *lift* tells us how discriminative is the cluster for a target. In words, the *lift* is the ratio of how much an item occurs in the bicluster by how much it occurs in the dataset. Typically, *lift* $\gg 1$ indicates that the selected bicluster is able to discriminate the given target.

9.4 Results

The results for the classification experiment are shown in Table 9.2. The *stft* representation (shown as $\mathcal{L}_C(\vec{\mathbf{x}}_1, y)$) had the best performance with an AUC of 0.933, followed by *fmri* ($\mathcal{L}(\vec{\mathbf{x}}_2, y)$) with 0.765 and *raw* (shown as $\mathcal{L}_C(\vec{\mathbf{x}}_0, y)$) performed below random with 0.225. In terms of accuracy, all classifiers performed according to the AUC metric, except for the *raw* view that had an accuracy of 0.497. In fact, the *raw* representation did not handle well the data imbalance phenomena, because it classified most of the instances as healthy controls (0.860 specificity versus 0.037 sensitivity). The other views were less affected by this imbalance, with the difference between sensitivity and specificity being 0.104 and 0.066 for the *stft* and *fmri* views. The performance of the *raw* representation is low and shows that an EEG recording without preprocessing steps is not able to be applied in classification settings. On the other hand, the *stft* is capable of it, showing the time-frequency domain features are discriminative of schizophrenia. Nonetheless, we expected *fmri* to be closer to the performance of *stft*. An AUC of 0.765 shows it has a good prediction power, however, it was outperformed by its preceding representation, according to Figure 9.6. We found no

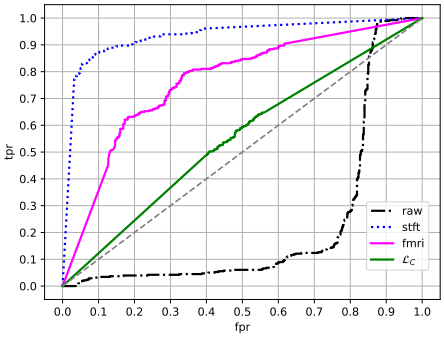


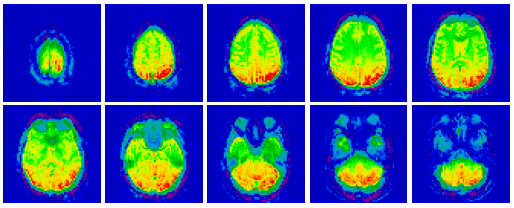
Figure 9.6: ROC curve plot of all the views considered for the EEG data of the Fribourg dataset.

statistical significance between *raw* and *fmri*, on the other hand *stft* outperformed *raw* and *fmri* with p-value of 0.04 (borderline) and 0.0004 (high statistical significance), respectively.

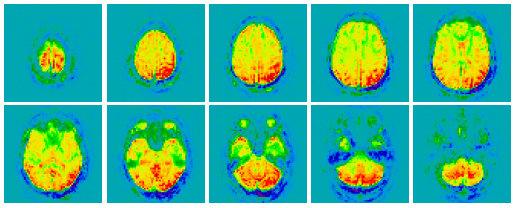
	$\mathcal{L}_C(\vec{\mathbf{x}}_0, y)$	$\mathcal{L}_C(\vec{\mathbf{x}}_1, y)$	$\mathcal{L}(\vec{\mathbf{x}}_2, y)$	$\mathcal{L}_C(\vec{\mathbf{x}}_2, y)$
AUC	0.225	**0.933**	0.765	0.551
Acc	0.497	**0.891**	0.703	0.545
Sens	0.037	**0.832**	0.666	0.524
Spe	0.860	**0.936**	0.733	0.563

Table 9.2: AUC, accuracy, sensitivity and specificity of a linear classifier with different views as input. These results refer to the Fribourg dataset. The first column $\mathcal{L}_C(\vec{\mathbf{x}}_0, y)$ refers to the *raw* representation; the second column $\mathcal{L}_C(\vec{\mathbf{x}}_1, y)$ refers to the *stft* representation; the third column $\mathcal{L}(\vec{\mathbf{x}}_2, y)$ refers to the *fmri* representation. The fourth and fifth column refer to additional analyses made to the *fmri* view: first $\mathcal{L}_C(\vec{\mathbf{x}}_2, y)$ refers to the *fmri* view, however it is optimized only with the negative log likelihood loss.

The additional analysis performed to the *fmri* representation (shown in the right part of Table 9.2) tell us the contrastive loss, $\mathcal{L}_D$, is necessary for the data to be separated, since without it the view only achieved an AUC of 0.551. In terms of synthesis quality, the synthesized fMRI were well defined, meaning that they appeared as fMRI volumes to the human eye. Though, there are ill defined predictions with activity present in the background of the volume. The latter may be due to each model at a fold, converging to different suboptimal parameters. Consequently, those parameters may lead to different distributions. Nevertheless,



(a) A good quality fMRI synthesis with no background activity reported.



(b) An fMRI synthesis where there is some activity present in the background.

Figure 9.7: Two fMRI predictions, from the F neural network after performing the minimization of $\mathcal{L}(\vec{\mathbf{x}}_2, y)$. In comparison with the fMRI synthesis, that had corruption given the minimization of the negative log likelihood with the A_M not fixed, previously shown in the left in Figure 9.1, our proposed methodology not only enables a good classification of the labes but is also good at synthesizing fMRI from EEG only data.

the success of the synthesis is encouraging, since it demonstrates the ability of $\mathcal{L}_D$, along with the proposed layer normalization, to maintain the style of the fMRI and at the same time separate the data. The proposed loss also demonstrated to work well in a joint training with an additional classification loss, $\mathcal{L}_C$. We were able to take advantage of a shift invariant function, *cosine*, and process a different EEG dataset by an encoder, that enabled the decoder to project to an fMRI with the learned distribution (the distribution of the NODDI fMRI). We provide an illustration of the synthesized fMRI in Figure 9.7. The latter shows that the synthesis is maintained while the whole network learns, along with the linear classifier, the targets. In terms of biclustering, we were able to find biclusters with the parameters for all the resolutions, using the ground truth and the predictions. From the gathered biclusters, we report the best biclusters according to the *lift*. All biclusters have statistical significance, an assurance of the BicPAMS algorithm. In contrast with the classification setting, the biclustering analysis shows us there are patterns with high discriminative power for schizophrenia. Suggesting that models with a better feature engineering would take advantage of these patterns.

9.5 Discussion

Castanho et al. [172] study the application of several biclustering algorithms in fMRI data to uncover statistically significant patterns. One of the algorithms studied was BicPAMS. The biclusters were of the type voxels by time, $E_2 \times T$. The authors claim this setup has the leverage of finding patterns that correlate/connect different brain regions over the temporal dimension, a.k.a. functional connectivity. Indeed, it is known that distant brain regions communicate between each other through neuronal pathways. This phenomena can be observed with similar frequencies present in distant EEG electrodes [173]. Note that, we do not consider the temporal dimension of fMRI in our pattern search made by BicPAMS. Thus, we can not make these functional connectivity claims from the synthesized fMRI. Nonetheless, clusters of the form individuals by voxels, $S \times E_2$, allow us to assess if there are *constant*[1] spatial patterns in the fMRI volume that can discriminate schizophrenia. **Outside of brain volume areas, such as background, are not discriminative.** We performed ablation experiments, to ensure no patterns were being found in regions where they were not supposed to exist, such as the background. For this, we collected all these regions of the synthesized volumes, and BicPAMS did not find biclusters with statistical assurances. This experiment

[1] Constant patterns are patterns that are equal for every individual.

rejected the hypothesis of discriminative information out of brain regions. **Yet, in in brain regions, BicPAMS found several biclusters**, all of them with lift greater than 1.38 for the schizophrenia class. This means, the synthesized fMRIs are able to represent schizophrenia related patterns and do not build patterns in healthy brains. The latter, is particularly encouraging, since healthy brains in this task should not have patterns present, suggesting that these have different distributions.

The synthesized cerebellum region is present in several biclusters associated with schizophrenia. In the biclusters of the ground truth labels, we found heterogeneous regions in the resolutions. Meaning, finer granularities uncover information that entire lobes, as a whole, do not. This suggests higher resolution volumes may contain relevant patterns that would not be discovered otherwise. To put in perspective, these biclusters in $5 \times 5 \times 3$ volumes were present in the parietal lobe, left temporal lobe and cerebellum. In $10 \times 10 \times 5$ resolution volumes patterns were found in parietal, occipital and left temporal lobes, as well as in the cerebellum region. And in $14 \times 14 \times 7$ biclusters had voxels present in parietal, occipital, frontal and left temporal lobes. Figure 9.8 illustrates the best biclusters found according to lift. This patterns go in accordance with previous findings reporting that prefrontal and temporal lobes are affected by schizophrenia [174]. On another note, Rahaman et al. [175] made a significant contribution on the application of biclustering in MRI data. They were able to find discriminative MRI patterns for schizophrenia patients. While MRI measures white matter, fMRI records blood supply levels. MRI is not fMRI. But we find it pertinent to relate this study with ours, since biclusters present in regions associated with schizophrenia (gyrus, brainstem) are relevant, independently of the modality. Also, comparing our results, of a synthesized fMRI modality, with an MRI, lets us assess if the synthesis is veracious in a spatial perspective. In their study, a biclustering algorithm was ran on individuals considering nine components (taken from a division of 30 regions of interest using independent component analysis). The biclusters found observed patterns in the gyrus and brainstem parts of the brain. Our different granularity experiments go in accordance with biclusters containing patterns in the cerebellum which is connected to the brainstem. However, we did not report any pattern in the gyrus region. Of course no major claim can be made about the spatial veracity of this inter study correlation, since both our view is synthesized and datasets are different. Additional analyses are needed to check if the activity synthesized in the different regions of the fMRI volume goes in accordance with the dynamics of a real fMRI.

In the explainability analysis, we found different biclusters at higher resolutions, but still present in the same regions. In $5 \times 5 \times 3$ resolution volumes, patterns were observed in parietal, occipital, left temporal lobes and cerebellum. At finer granularities, voxels in the frontal lobe were present in the biclusters retrieved for $10 \times 10 \times 5$ and $14 \times 14 \times 7$ resolutions. Areas uncovered in $5 \times 5 \times 3$ were also present at higher resolutions, except for the cerebellum at $14 \times 14 \times 7$. These biclusters all reported lifts above 1.2. Lifts in explainability were lower than the ground truth. In general high lift patterns were harder to find in this setting, and as a consequence the *minimum lift* parameter (see Table 9.1) had to be lowered (relative the ground truth) to relax the search.

The linear classifier decision making is correlated with the synthesized frontal lobe activity. Doing explainability on a leave-one-individual-out validation schema is difficult. We only have access to the model at the prediction time of the fold. In addition, different folds have different suboptimal parameters. Altogether, these are challenges that we tackled using a pattern mining tool. Pinto et al. [176] used the apriori algorithm to retrieve the explanations of predictions. The idea is that, by finding patterns of data

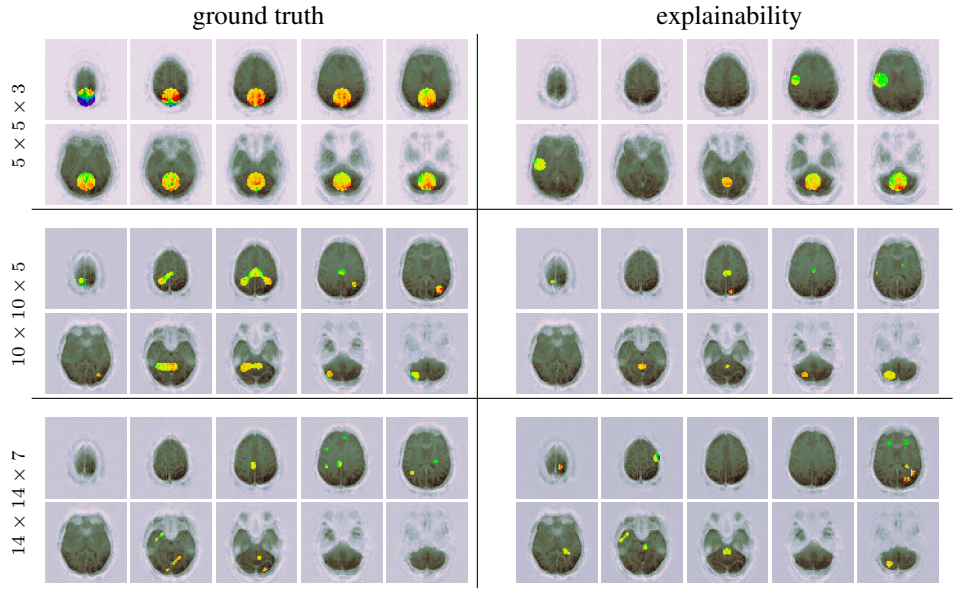


Figure 9.8: We analyzed resolutions $\in \{5 \times 5 \times 3, 10 \times 10 \times 5, 14 \times 14 \times 7\}$ and gathered the biclusters retrieved for the ground truth and predicted labels. Only the best biclusters (with the best *lift*) are shown in this figure for each setting.

that have statistical assurances for a target, we are explaining predictions. Note however, that the model may not be looking at those patterns to make its decision. What the explainability analysis tells us, is how the synthesized fMRI correlates with the predictions made by the linear classifier. We see them as biclusters that explain the predictions, with statistical assurances. In this setting, BicPAMS found different row sets, because the predictions differ from the ground truth. So, how can we view these biclusters? They discriminate the predictions, a difference seen in the different regions gathered by the biclusters. For instance, explainability biclusters reported the frontal lobe presence, while the ground truth ones did not report this region. Nonetheless, the frontal lobe is associated with problem solving and attention functions, which are recognized as impairments provoked by schizophrenia [177, 178]. This pathology affects the cognitive ability and signatures of the human brain [179]. There is extensive research on the different discriminative patterns that are able to identify this pathology and a lot of research is performed using MRI technologies.

The synthesized fMRI view is discriminative of schizophrenia. The statistical assurances (lift, support) were higher for the biclusters found in the ground truth analysis. No major comment is made for this observation, as they are different settings, that can not be compared. However, the high discriminative power of the ground truth biclusters for schizophrenia, show that the produced fMRI views have potential for schizophrenia diagnostics. The low AUC, 0.765, of the linear classifier shows us that it is not sufficient for a reliable application of this view. All in all, a linear classifier has no feature engineering properties, but BicPAMS gave us statistical assurances about the discriminative patterns retrieved. Showing us that there is information present in this view, potentially uncoverable by powerful models. We refer this for future work.

Frequency features are highly discriminative of schizophrenia, yet lack interpretability. The linear classifier showed us that it better assessed schizophrenia using the *stft* representation. It is very well known that frequency features are highly discriminative of this pathology [3, 156]. However, we were not able to

show the power of the *fmri* view, through this classification setup. Still, this makes sense. The distance from *stft* to the *fmri*, shown in Figure 9.2, inhibits the gradients of $\mathcal{L}_D$ w.r.t. θ_F to be significant at the top layers. These are the ones close to the *stft* representation. Since the number of epochs is fixed, we hypothesize that not enough time was given in the validation process to show that the *fmri* view is able to perform in comparison to *stft*. We hypothesize that finer regularization strategies, such as l_1-path-norm regularization [180], are needed to allow a faster convergence. Nonetheless, the superiority of *stft* features is due to it being better engineered for schizophrenia. In contrast, the *raw* representation lacks not only in interpretability, but also lacks this engineering property to allow a good performance of the linear classifier. We look at an *fmri* representation and see its potential applicability in a health care setting, since it has higher levels of interpretability and, as previously discussed, highly discriminative patterns. And although, *stft* had a better performance (see ROC curve in Figure 9.8), it still lacks in the interpretability level. It is essential that a clinical diagnostic be made of simple explanations. Not only has the doctor to understand, but also it is beneficial if the patient fully understands its diagnostic [181]. Explaining a diagnostic based on *stft* features is not tractable for people out of the EEG clinical scope. Not to say that the *fmri* representation is understandable by everyone, but it is easier since it is explained in the spatial (and temporal) domain. EEG is recognized to have low interpretability power [9], however it does not block its use in health care. Our main concern, is projecting this modality to a space where it can better be understood by a human, be it expert or not.

9.6 Summary

- We show that an EEG recording may be viewed in 3 different perspectives. These perspectives have each their advantages and disadvantages, with one of them having no reported traits (the fMRI view), observed in previous studies. These perspectives are evaluated in predictive power of pathologies and interpretability, with the goal of providing good explanations for a decision;

- The learned fMRI synthesis, when subject to a classification setting with EEG only data, loses the style previously learned from a simultaneous EEG and fMRI dataset. It either maintains the fMRI style and lacks in predictive power, or it loses the style and enables the separation of the data. We mitigate this problem with a novel methodology that locks the spectral coefficients (learned by the neural network) and learns a space that distributes the data identically before and after a cosine projection. Ultimately, this enables the application of a synthesized fMRI view in a diagnostic health care setting;

- Biclustering analyses showed us that the cerebellum region of the synthesized fMRI is associated with schizophrenic individuals and the frontal lobe region is associated with the model's decision of classifying an instance as schizophrenic. Both observations are in accordance with related work on schizophrenia;

Chapter 10

Future Research

EEG and fMRI have, in this dissertation, been seen as a natural pairing to advance knowledge in neuroscience. EEG varies rapidly in time while sacrificing spatial resolution, whereas fMRI excells in spatial resolution by varying at a slower rate. The relation between the two modalities has always been questioned and advances were made [64, 67, 182], though lacking to provide a strong answer on a still very open topic to the community. We studied the relationship between these two modalities, by building a foundational mapping from EEG to fMRI. The mapping function is dynamic, adaptable for different (paired) data sources, due to the nature of the underlying automated machine learning techniques employed. Further, the model operates in its own spectral space to generate an fMRI volume, which allows the explanation to communities outside of machine learning.

When this project started, Liu and Sajda [123] had already motivated the idea for this mapping function. We went further and pushed the limit of fMRI estimation from EEG, using a neural network composed of convolutional layers, Fourier features, and attention mechanisms. We included thorough reports, based on explainability methods, that answer why an fMRI prediction was made given a specific EEG. With the latter, we were able to compare our findings with related work on simultaneous EEG and fMRI data that reported haemodynamical correlations with EEG electrode links. Ultimately, validating our proposed methodology. Subsequently, we proposed a Bayesian frame to the synthesis task with the aim of assessing the degree of uncertainty the model had on a prediction and proposed a method that allows risk quantification, as well as is able to plug in the output layers of any neural based EEG to fMRI synthesis model. Until this point, we had a model that was well studied in its ability to synthesize fMRIs, relevant for answering pertinent questions regarding the efficacy of the underlying predictive mechanisms. Last but not least, the motivation, of synthesizing fMRI using EEG, was to potentially use our model in an health care setting where diagnostics are required. To test this, operating with EEG only data, we showed the ability of our model to extrapolate to a regime without fMRI. Further, the synthesized signal reported statistically significant discriminative power. The conducted research opened many new avenues for further exploration. These contributions enabled us to assess: the ability of a network to predict fMRI; the relationships of EEG electrodes that are related with haemodynamics; the degree to which the network is certain of a prediction; the ability of the network to extrapolate to a classification setting.

Temporal dimension of fMRI. Our work has many contributions that altogether built the state-of-the-art *EEG to fMRI synthesis* model. Despite having a sliding window that is able to produce temporal

variations of fMRI volumes, our methodology does not take into account the temporal dimension of the fMRI signal. All in all, the proposal was to give a model that is not expensive and is able to run on a day-to-day laptop, so that costs do not increase. Adding a dimension would increase the memory consumed, consequently needing better hardware available at higher costs.

Multi class classification setting. We showed how the synthesized fMRI signal can be used for binary classification. Many settings operate with more than two classes, making it a multi class classification problem. At the output of the network, we only need to treat the problem according to a Bernoulli distribution. On the other hand, at the sinusoids the solution is non trivial. The question becomes: *How could our methodology be adapted to a multi class classification?* This becomes necessary to handle outcomes with higher cardinalities, common in the neuropsychiatric domain. Considering Alzheimer's disease as an illustrative example, we want to discriminate between three groups: *healthy controls*, *mild congnitive disorder* and *Alzheimer's*.

Out of distribution data. In some cases, the EEG, that is forwarded to the network, has a considerably different distribution than the EEGs used to train the mapping function (e.g., differences in instrumentation, monitoring protocol, channel displacement). Given such an instance, the projected sinusoids, $cos(\omega \cdot \vec{\mathbf{z}}_x + \beta)$, may not match with the learned spectral fMRI coefficients, A_M. Hence, and ultimately, synthesizing an ill defined fMRI volume. To avoid such a scenario, we could prepare the network to handle such out of distribution data. One way this could be done is to add noise to the sinusoids, enabling the network to handle noisy shifts and produce a good fMRI estimate. Another solution, which may be also seen as another research direction, is to include markers such as demographic information in the input of our network. This would enable us to make a decision on data that was only seen before and reject an out of distribution instance. Physics informed machine learning is also an alternative that can correct outlier malfunctions of the network's predictions.

Open source contributions. As Python becomes a hub for scientific development, there is a need to provide open source software that facilitates EEG to fMRI synthesis. This allows third party scientific contributions from different laboratories to coexist and facilitates the integration in health care settings. With this aim, we leave the final product of this thesis published at the Python packaging index.[1]

Health care application. Besides collaboration with laboratories, another future direction is to validate and deploy the proposed pipeline in the clinical environment, by establishing partnerships with hospitals. We propose that such efforts are made, in order to apply our framework and provide open access diagnostics to any entity that needs it. We believe this type of work model allows more testing on what types of pathologies is our model able to address. As of now, we have tested it in schizophrenia, but what about neuro degenerative disorders, such as Parkinson and Alzheimer?

I hope this project has enticed researchers to explore this area, that ultimately aims at improving health for the vulnerable communities.

[1] EEG to fMRI synthesis package can be found at https://pypi.org/project/eeg-to-fmri/.

Bibliography

[1] A. Zanardi, J. Zilly, A. Aumiller, A. Censi, and E. Frazzoli. Wormhole learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7899–7905. IEEE, 2019.

[2] U. R. Acharya, S. V. Sree, and J. S. Suri. Automatic detection of epileptic eeg signals using higher order cumulant features. *International journal of neural systems*, 21(05):403–414, 2011.

[3] T. M. Itil. Qualitative and quantitative eeg findings in schizophrenia. *Schizophrenia bulletin*, 3(1): 61, 1977.

[4] C. Besthorn, H. Förstl, C. Geiger-Kabisch, H. Sattel, T. Gasser, and U. Schreiter-Gasser. Eeg coherence in alzheimer disease. *Electroencephalography and clinical neurophysiology*, 90(3):242–245, 1994.

[5] J. Taghia, W. Cai, S. Ryali, J. Kochalka, J. Nicholas, T. Chen, and V. Menon. Uncovering hidden brain state dynamics that regulate performance and decision-making during cognition. *Nature communications*, 9(1):1–19, 2018.

[6] M. A. Pisauro, E. Fouragnan, C. Retzler, and M. G. Philiastides. Neural correlates of evidence accumulation during value-based decisions revealed via simultaneous eeg-fmri. *Nature communications*, 8(1):1–9, 2017.

[7] H. Mohr, U. Wolfensteller, R. F. Betzel, B. Mišić, O. Sporns, J. Richiardi, and H. Ruge. Integration and segregation of large-scale brain networks during short-term task automatization. *Nature communications*, 7(1):1–12, 2016.

[8] I. Daly, D. Williams, F. Hwang, A. Kirke, E. R. Miranda, and S. J. Nasuto. Electroencephalography reflects the activity of sub-cortical brain regions during approach-withdrawal behaviour while listening to music. *Scientific reports*, 9(1):1–22, 2019.

[9] H. O. de Beeck and C. Nakatani. *Introduction to human neuroimaging*. Cambridge University Press, 2019.

[10] G. I. Ogbole, A. O. Adeyomoye, A. Badu-Peprah, Y. Mensah, and D. A. Nzeh. Survey of magnetic resonance imaging availability in west africa. *Pan African Medical Journal*, 30(1), 2018.

[11] E. J. van Beek, C. Kuhl, Y. Anzai, P. Desmond, R. L. Ehman, Q. Gong, G. Gold, V. Gulani, M. Hall-Craggs, T. Leiner, et al. Value of mri in medicine: More than just another test?, 2019.

[12] G. Bonmassar, K. Anami, J. Ives, and J. W. Belliveau. Visual evoked potential (vep) measured by simultaneous 64-channel eeg and 3t fmri. *Neuroreport*, 10(9):1893–1897, 1999.

[13] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

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

[15] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[16] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, pages 6571–6583, 2018.

[17] S. Chakraborty, K. S. Meel, and M. Y. Vardi. A scalable and nearly uniform generator of sat witnesses. In *International Conference on Computer Aided Verification*, pages 608–623. Springer, 2013.

[18] H. Liu, K. Simonyan, and Y. Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[19] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017.

[20] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv preprint arXiv:2006.10739*, 2020.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] D. Calhas, V. M. Manquinho, and I. Lynce. Automatic generation of neural architecture search spaces. In *Combining Learning and Reasoning: Programming Languages, Formalisms, and Representations*, 2022.

[23] D. Calhas and R. Henriques. fmri multiple missing values imputation regularized by a recurrent denoiser. In *International Conference on Artificial Intelligence in Medicine*, pages 25–35. Springer, 2021.

[24] D. Calhas and R. Henriques. Eeg to fmri synthesis benefits from attentional graphs of electrode relationships. *Machine Learning for Health Care Conference*, 2023.

[25] D. Calhas and R. Henriques. Eeg to fmri synthesis: Is deep learning a candidate? *International Conference on Information Systems Development*, 2023.

[26] D. Calhas and R. Henriques. Fitting regularized population dynamics with neural differential equations. In *The Symbiosis of Deep Learning and Differential Equations*, 2021.

[27] A. Krumholz, S. Wiebe, G. Gronseth, S. Shinnar, P. Levisohn, T. Ting, J. Hopp, P. Shafer, H. Morris, L. Seiden, et al. Practice parameter: Evaluating an apparent unprovoked first seizure in adults (an evidence-based review):[retired]: Report of the quality standards subcommittee of the american academy of neurology and the american epilepsy society. *Neurology*, 69(21):1996–2007, 2007.

[28] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, and H. Adeli. Deep convolutional neural network for the automated detection and diagnosis of seizure using eeg signals. *Computers in biology and medicine*, 100:270–278, 2018.

[29] S. L. Oh, Y. Hagiwara, U. Raghavendra, R. Yuvaraj, N. Arunkumar, M. Murugappan, and U. R. Acharya. A deep learning approach for parkinson's disease diagnosis from eeg signals. *Neural Computing and Applications*, pages 1–7, 2018.

[30] C. Ieracitano, N. Mammone, A. Hussain, and F. C. Morabito. A novel multi-modal machine learning based approach for automatic classification of eeg recordings in dementia. *Neural Networks*, 123: 176–190, 2020.

[31] L. A. Gemein, R. T. Schirrmeister, P. Chrabaszcz, D. Wilson, J. Boedecker, A. Schulze-Bonhage, F. Hutter, and T. Ball. Machine-learning-based diagnostics of eeg pathology. *NeuroImage*, 220: 117021, 2020.

[32] T. Song, W. Zheng, P. Song, and Z. Cui. Eeg emotion recognition using dynamical graph convolutional neural networks. *IEEE Transactions on Affective Computing*, 11(3):532–541, 2018.

[33] V. Gupta, M. D. Chopda, and R. B. Pachori. Cross-subject emotion recognition using flexible analytic wavelet transform from eeg signals. *IEEE Sensors Journal*, 19(6):2266–2274, 2018.

[34] X.-W. Wang, D. Nie, and B.-L. Lu. Emotional state classification from eeg data using machine learning approach. *Neurocomputing*, 129:94–106, 2014.

[35] C. Neuper, R. Scherer, M. Reiner, and G. Pfurtscheller. Imagery of motor actions: Differential effects of kinesthetic and visual–motor mode of imagery in single-trial eeg. *Cognitive brain research*, 25 (3):668–677, 2005.

[36] K.-R. Müller, M. Tangermann, G. Dornhege, M. Krauledat, G. Curio, and B. Blankertz. Machine learning for real-time single-trial eeg-analysis: from brain–computer interfacing to mental state monitoring. *Journal of neuroscience methods*, 167(1):82–90, 2008.

[37] X. An, D. Kuang, X. Guo, Y. Zhao, and L. He. A deep learning method for classification of eeg data based on motor imagery. In *International Conference on Intelligent Computing*, pages 203–210. Springer, 2014.

[38] S. Nakagome, T. P. Luu, Y. He, A. S. Ravindran, and J. L. Contreras-Vidal. An empirical comparison of neural networks and machine learning algorithms for eeg gait decoding. *Scientific reports*, 10(1): 1–17, 2020.

[39] E. A. Wan, R. Van Der Merwe, and S. Haykin. The unscented kalman filter. *Kalman filtering and neural networks*, 5(2007):221–280, 2001.

[40] S. U. Amin, M. Alsulaiman, G. Muhammad, M. A. Mekhtiche, and M. S. Hossain. Deep learning for eeg motor imagery classification based on multi-layer cnns feature fusion. *Future Generation computer systems*, 101:542–554, 2019.

[41] W. Wu, S. Nagarajan, and Z. Chen. Bayesian machine learning: Eeg\/meg signal processing measurements. *IEEE Signal Processing Magazine*, 33(1):14–36, 2015.

[42] M. Dai, D. Zheng, R. Na, S. Wang, and S. Zhang. Eeg classification of motor imagery using a novel deep learning framework. *Sensors*, 19(3):551, 2019.

[43] Y. Aghakhani, A. Bagshaw, C. Benar, C. Hawco, F. Andermann, F. Dubeau, and J. Gotman. fmri activation during spike and wave discharges in idiopathic generalized epilepsy. *Brain*, 127(5):1127–1144, 2004.

[44] W. Li, X. Lin, and X. Chen. Detecting alzheimer's disease based on 4d fmri: An exploration under deep learning framework. *Neurocomputing*, 388:280–287, 2020.

[45] S. H. Hojjati, A. Ebrahimzadeh, and A. Babajani-Feremi. Identification of the early stage of alzheimer's disease using structural mri and resting-state fmri. *Frontiers in neurology*, 10:904, 2019.

[46] Y. Kazemi and S. Houghten. A deep learning pipeline to classify different stages of alzheimer's disease from fmri data. In *2018 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–8. IEEE, 2018.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[48] S. Patz, D. Fovargue, K. Schregel, N. Nazari, M. Palotai, P. E. Barbone, B. Fabry, A. Hammers, S. Holm, S. Kozerke, et al. Imaging localized neuronal activity at fast time scales through biomechanics. *Science advances*, 5(4):eaav3816, 2019.

[49] R. L. Miller, A. Abrol, T. Adali, Y. Levin-Schwarz, and V. D. Calhoun. Resting-state fmri dynamics and null models: Perspectives, sampling variability, and simulations. *Frontiers in neuroscience*, 12: 551, 2018.

[50] Z. Huang, J. Zhang, J. Wu, G. A. Mashour, and A. G. Hudetz. Temporal circuit of macroscale dynamic brain activity supports human consciousness. *Science advances*, 6(11):eaaz0087, 2020.

[51] C. Chang and G. H. Glover. Time–frequency dynamics of resting-state brain connectivity measured with fmri. *Neuroimage*, 50(1):81–98, 2010.

[52] E. Tagliazucchi, P. Balenzuela, D. Fraiman, and D. R. Chialvo. Criticality in large-scale brain fmri dynamics unveiled by a novel point process analysis. *Frontiers in physiology*, 3:15, 2012.

[53] M. A. Casey. Music of the 7ts: Predicting and decoding multivoxel fmri responses with acoustic, schematic, and categorical music features. *Frontiers in psychology*, 8:1179, 2017.

[54] A. O. Nunez-Elizalde, A. G. Huth, and J. L. Gallant. Voxelwise encoding models with non-spherical multivariate normal priors. *Neuroimage*, 197:482–492, 2019.

[55] S. Radhakrishnan, C. Nutakki, and S. Diwakar. Mathematical modeling of fmri bold responses related nitric oxide production-consumption and in the cerebellum granule layer. *Procedia Computer Science*, 171:1606–1613, 2020.

[56] S. Jain and A. G. Huth. Incorporating context into language encoding models for fmri. In *Neurips*, pages 6629–6638, 2018.

[57] S. Jain, S. Mahto, J. S. Turek, V. A. Vo, A. LeBel, and A. G. Huth. Interpretable multi-timescale models for predicting fmri responses to continuous natural speech. *bioRxiv*, pages 2020–10, 2021.

[58] A. R. Vaidya, S. Jain, and A. G. Huth. Self-supervised models of audio effectively explain human cortical responses to speech. *arXiv preprint arXiv:2205.14252*, 2022.

[59] J. Peng, Y. Wang, R. Wang, W. Kong, and J. Zhang. Neural coupling mechanism in fmri hemodynamics. *Nonlinear Dynamics*, 103:883–895, 2021.

[60] M. Havlicek, D. Ivanov, A. Roebroeck, and K. Uludağ. Determining excitatory and inhibitory neuronal activity from multimodal fmri data using a generative hemodynamic model. *Frontiers in neuroscience*, 11:616, 2017.

[61] R. M. Cichy and A. Oliva. Am/eeg-fmri fusion primer: Resolving human brain responses in space and time. *Neuron*, 2020.

[62] D. W. Mann-Krzisnik and G. D. Mitsis. Modeling bold-fmri hemodynamics via multidimensional decomposition of electrophysiology data: A simulation study. In *International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 398–401. IEEE, 2020.

[63] X. Liu, L. Hong, and P. Sajda. Latent neural source recovery via transcoding of simultaneous eeg-fmri. *arXiv preprint arXiv:2010.02167*, 2020.

[64] C. Cury, P. Maurel, R. Gribonval, and C. Barillot. A sparse eeg-informed fmri model for hybrid eeg-fmri neurofeedback prediction. *Frontiers in neuroscience*, 13:1451, 2020.

[65] M. G. Philiastides, T. Tu, and P. Sajda. Inferring macroscale brain dynamics via fusion of simultaneous eeg-fmri. *Annual Review of Neuroscience*, 44, 2021.

[66] J. Wirsich, B. Ridley, P. Besson, V. Jirsa, C. Bénar, J.-P. Ranjeva, and M. Guye. Complementary contributions of concurrent eeg and fmri connectivity for predicting structural connectivity. *NeuroImage*, 161:251–260, 2017.

[67] G. V. Portnova, A. Tetereva, V. Balaev, M. Atanov, L. Skiteva, V. Ushakov, A. Ivanitsky, and O. Martynova. Correlation of bold signal with linear and nonlinear patterns of eeg in resting state eeg-informed fmri. *Frontiers in human neuroscience*, 11:654, 2018.

[68] Y. R. Tabar and U. Halici. A novel deep learning approach for classification of eeg motor imagery signals. *Journal of neural engineering*, 14(1):016003, 2016.

[69] D. Long, J. Wang, M. Xuan, Q. Gu, X. Xu, D. Kong, and M. Zhang. Automatic classification of early parkinson's disease with multi-modal mr imaging. *PloS one*, 7(11):e47714, 2012.

[70] A. Dehsarvi and S. L. Smith. Classification of resting-state fmri using evolutionary algorithms: Towards a brain imaging biomarker for parkinson's disease. *arXiv preprint arXiv:1910.05378*, 2019.

[71] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Twenty-second international joint conference on artificial intelligence*, 2011.

[72] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[73] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[74] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[75] A. Rahimi, B. Recht, et al. Random features for large-scale kernel machines. In *NIPS*, volume 3, page 5. Citeseer, 2007.

[76] B. Rieck, T. Yates, C. Bock, K. Borgwardt, G. Wolf, N. Turk-Browne, and S. Krishnaswamy. Uncovering the topology of time-varying fmri data using cubical persistence. *Advances in Neural Information Processing Systems*, 33, 2020.

[77] F. Deligianni, M. Centeno, D. W. Carmichael, and J. D. Clayden. Relating resting-state fmri and eeg whole-brain connectomes across frequency bands. *Frontiers in Neuroscience*, 2014.

[78] F. Deligianni, D. W. Carmichael, G. H. Zhang, C. A. Clark, and J. D. Clayden. Noddi and tensor-based microstructural indices as predictors of functional connectivity. *PLoS One*, 2016.

[79] H. H. Jasper. The ten-twenty electrode system of the international federation. *Electroencephalogr. Clin. Neurophysiol.*, 10:370–375, 1958.

[80] J. M. Walz, R. I. Goldman, M. Carapezza, J. Muraskin, T. R. Brown, and P. Sajda. Simultaneous eeg-fmri reveals temporal evolution of coupling between supramodal cortical attention networks and the brainstem. *Journal of Neuroscience*, 2013.

[81] J. M. Walz, R. I. Goldman, M. Carapezza, J. Muraskin, T. R. Brown, and P. Sajda. Simultaneous eeg–fmri reveals a temporal cascade of task-related and default-mode activations during a simple target detection task. *Neuroimage*, 2014.

[82] B. R. Conroy, J. M. Walz, and P. Sajda. Fast bootstrapping and permutation testing for assessing reproducibility and interpretability of multivariate fmri decoding models. *PloS one*, 2013.

[83] M. Pereira, N. Faivre, I. Iturrate, M. Wirthlin, L. Serafini, S. Martin, A. Desvachez, O. Blanke, D. V. de Ville, and J. del R. Millan. "simultaneous eeg-fmri for a speeded discrimination task with confidence", 2019.

[84] G. Lioi et al. Simultaneous mri-eeg during a motor imagery neurofeedback task: an open access brain imaging dataset for multi-modal data integration. *bioRxiv*, 2019.

[85] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[86] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[87] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7): e0130140, 2015.

[88] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209, 2019.

[89] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015.

[90] M. Böhle, F. Eitel, M. Weygandt, and K. Ritter. Layer-wise relevance propagation for explaining deep neural network decisions in mri-based alzheimer's disease classification. *Frontiers in aging neuroscience*, 11:194, 2019.

[91] G. M. Rojas, C. Alvarez, C. E. Montoya, M. de la Iglesia-Vayá, J. E. Cisternas, and M. Gálvez. Study of resting-state functional connectivity networks using eeg electrodes position as seed. *Frontiers in neuroscience*, 12:235, 2018.

[92] M. Claesen and B. De Moor. Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*, 2015.

[93] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13(1):281–305, 2012.

[94] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. *arXiv preprint arXiv:1206.2944*, 2012.

[95] V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[96] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9(4):611–629, 2018.

[97] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. M. Gambardella. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 342–347. IEEE, 2011.

[98] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[99] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018.

[100] B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

[101] L. Li and A. Talwalkar. Random search and reproducibility for neural architecture search. In *Uncertainty in Artificial Intelligence*, pages 367–377. PMLR, 2020.

[102] C. Ying, A. Klein, E. Christiansen, E. Real, K. Murphy, and F. Hutter. Nas-bench-101: Towards reproducible neural architecture search. In *International Conference on Machine Learning*, pages 7105–7114. PMLR, 2019.

[103] K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos, and E. Xing. Neural architecture search with bayesian optimisation and optimal transport. *arXiv preprint arXiv:1802.07191*, 2018.

[104] T. Elsken, J. H. Metzen, F. Hutter, et al. Neural architecture search: A survey. *J. Mach. Learn. Res.*, 20(55):1–21, 2019.

[105] A. Biere, M. Heule, and H. van Maaren. *Handbook of satisfiability*, volume 185. IOS press, 2009.

[106] C. P. Gomes, A. Sabharwal, and B. Selman. Near-uniform sampling of combinatorial spaces using xor constraints. In *Advances In Neural Information Processing Systems*, pages 481–488, 2007.

[107] R. S. Sukthanker, Z. Huang, S. Kumar, E. G. Endsjo, Y. Wu, and L. Van Gool. Neural architecture search of spd manifold networks. *arXiv preprint arXiv:2010.14535*, 2020.

[108] Y. Zhao, L. Wang, Y. Tian, R. Fonseca, and T. Guo. Few-shot neural architecture search. In *International Conference on Machine Learning*, pages 12707–12718. PMLR, 2021.

[109] N. Nayman, Y. Aflalo, A. Noy, and L. Zelnik-Manor. Hardcore-nas: Hard constrained differentiable neural architecture search. *arXiv preprint arXiv:2102.11646*, 2021.

[110] W. Grathwohl, E. Creager, S. K. S. Ghasemipour, and R. Zemel. Gradient-based optimization of neural network architecture, 2018.

[111] O. Roussel and V. M. Manquinho. Pseudo-boolean and cardinality constraints. *Handbook of satisfiability*, 185:695–733, 2009.

[112] L. M. de Moura and N. Bjørner. Z3: an efficient SMT solver. In C. R. Ramakrishnan and J. Rehof, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008. doi: 10.1007/978-3-540-78800-3_24. URL `https://doi.org/10.1007/978-3-540-78800-3_24`.

[113] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[114] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[115] Y. LeCun and C. Cortes. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 2010. URL `http://yann.lecun.com/exdb/mnist/`.

[116] Y. Guo, Y. Zheng, M. Tan, Q. Chen, J. Chen, P. Zhao, and J. Huang. Nat: Neural architecture transformer for accurate and compact architectures. *arXiv preprint arXiv:1910.14488*, 2019.

[117] C. Liao, K. Worsley, J.-B. Poline, J. Aston, G. Duncan, and A. Evans. Estimating the delay of the fmri response. *NeuroImage*, 16(3, Part A):593 – 606, 2002. ISSN 1053-8119. doi: https://doi.org/10.1006/nimg.2002.1096. URL `http://www.sciencedirect.com/science/article/pii/S1053811902910967`.

[118] J. Gu, L. Liu, P. Wang, and C. Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021.

[119] J. Pinto, S. Nunes, M. Bianciardi, A. Dias, L. M. Silveira, L. L. Wald, and P. Figueiredo. Improved 7 tesla resting-state fmri connectivity measurements by cluster-based modeling of respiratory volume and heart rate effects. *Neuroimage*, 153:262–272, 2017.

[120] Q. Yu, L. Wu, D. A. Bridwell, E. B. Erhardt, Y. Du, H. He, J. Chen, P. Liu, J. Sui, G. Pearlson, et al. Building an eeg-fmri multi-modal brain graph: a concurrent eeg-fmri study. *Frontiers in human neuroscience*, 10:476, 2016.

[121] L. Bréchet, D. Brunet, G. Birot, R. Gruetter, C. M. Michel, and J. Jorge. Capturing the spatiotemporal dynamics of self-generated, task-initiated thoughts with eeg and fmri. *Neuroimage*, 194:82–92, 2019.

[122] H. Laufs, A. Kleinschmidt, A. Beyerle, E. Eger, A. Salek-Haddadi, C. Preibisch, and K. Krakow. Eeg-correlated fmri of human alpha activity. *Neuroimage*, 19(4):1463–1476, 2003.

[123] X. Liu and P. Sajda. A convolutional neural network for transcoding simultaneously acquired eeg-fmri data. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 477–482. IEEE, 2019.

[124] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman. How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022):1279–1285, 2011.

[125] S. Maknojia, N. W. Churchill, T. A. Schweizer, and S. Graham. Resting state fmri: Going through the motions. *Frontiers in neuroscience*, 13:825, 2019.

[126] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[127] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

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

[129] D. P. Kroese, T. Brereton, T. Taimre, and Z. I. Botev. Why the monte carlo method is so important today. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(6):386–392, 2014.

[130] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[131] N. Ahmed et al. Discrete cosine transform. *IEEE transactions on Computers*, 1974.

[132] G. K. Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 1992.

[133] S. H. Khan et al. Regularization of deep neural networks with spectral dropout. *Neural Networks*, 2019.

[134] P. Magron et al. Bayesian anisotropic gaussian model for audio source separation. In *ICASSP*, 2018.

[135] T. Gerkmann. Bayesian estimation of clean speech spectral coefficients given a priori knowledge of the phase. *IEEE Transactions on Signal Processing*, 2014.

[136] Y. Wen et al. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*, 2018.

[137] Ö. Çiçek et al. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, 2016.

[138] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019.

[139] E. B. George et al. Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model. *IEEE transactions on speech and audio processing*, 1997.

[140] J. Flynn et al. Deepstereo: Learning to predict new views from the world's imagery. In *CVPR*, 2016.

[141] A. Pervez et al. Spectral smoothing unveils phase transitions in hierarchical variational autoencoders. In *ICML*, 2021.

[142] M. Hemsley et al. Deep generative model for synthetic-ct generation with uncertainty predictions. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2020.

[143] P. Bricman et al. Eeg2fmri: Cross-modal synthesis for functional neuroimaging. *Github IO*, 2021.

[144] R. Henriques, F. L. Ferreira, and S. C. Madeira. Bicpams: software for biological data analysis with pattern-based biclustering. *BMC bioinformatics*, 18:1–16, 2017.

[145] L. Chu, R. Qiu, H. Liu, Z. Ling, T. Zhang, and J. Wang. Individual recognition in schizophrenia using deep learning methods with random forest and voting classifiers: Insights from resting state eeg streams. *arXiv preprint arXiv:1707.03467*, 2017.

[146] C. A. T. Naira, C. Jos, et al. Classification of people who suffer schizophrenia and healthy people by eeg signals using deep learning. *International Journal of Advanced Computer Science and Applications*, 10(10), 2019.

[147] S. L. Oh, J. Vicnesh, E. J. Ciaccio, R. Yuvaraj, and U. R. Acharya. Deep convolutional neural network model for automated diagnosis of schizophrenia using eeg signals. *Applied Sciences*, 9(14): 2870, 2019.

[148] C.-R. Phang, F. Noman, H. Hussain, C.-M. Ting, and H. Ombao. A multi-domain connectome convolutional neural network for identifying schizophrenia from eeg connectivity patterns. *IEEE journal of biomedical and health informatics*, 24(5):1333–1343, 2019.

[149] D. Calhas, E. Romero, and R. Henriques. On the use of pairwise distance learning for brain signal classification with limited observations. *Artificial intelligence in medicine*, 105:101852, 2020.

[150] D. Ahmedt-Aristizabal, T. Fernando, S. Denman, J. E. Robinson, S. Sridharan, P. J. Johnston, K. R. Laurens, and C. Fookes. Identification of children at risk of schizophrenia via deep learning and eeg responses. *IEEE Journal of biomedical and health informatics*, 25(1):69–76, 2020.

[151] M. V. Treviso and A. F. Martins. The explanation game: Towards prediction explainability through sparse communication. *arXiv preprint arXiv:2004.13876*, 2020.

[152] W. O. Tatum IV. *Handbook of EEG interpretation*. Springer Publishing Company, 2021.

[153] A. A. Fingelkurts et al. Impaired functional connectivity at eeg alpha and theta frequency bands in major depression. *Human brain mapping*, 28(3):247–261, 2007.

[154] A. P. Burgess et al. Functional connectivity of gamma eeg activity is modulated at low frequency during conscious recollection. *International Journal of Psychophysiology*, 46(2):91–100, 2002.

[155] M. A. Albrecht et al. Time and frequency dependent changes in resting state eeg functional connectivity following lipopolysaccharide challenge in rats. *PLoS One*, 13(11):e0206985, 2018.

[156] B. J. Roach et al. Event-related eeg time-frequency analysis: an overview of measures and an analysis of early gamma band phase locking in schizophrenia. *Schizophrenia bulletin*, 34(5):907–926, 2008.

[157] A. T. Bates et al. Low-frequency eeg oscillations associated with information processing in schizophrenia. *Schizophrenia research*, 115(2-3):222–230, 2009.

[158] R. N. Bracewell et al. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York, 1986.

[159] E. Jacobsen et al. The sliding dft. *IEEE Signal Processing Magazine*, 20(2):74–80, 2003.

[160] J. L. Ba et al. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[161] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

[162] S. Saha et al. A systematic review of the prevalence of schizophrenia. *PLoS medicine*, 2(5):e141, 2005.

[163] C. Andreou and S. Borgwardt. Structural and functional imaging markers for susceptibility to psychosis. *Molecular psychiatry*, 25(11):2773–2785, 2020.

[164] B. M. Neale and P. Sklar. Genetic analysis of schizophrenia and bipolar disorder reveals polygenicity but also suggests new directions for molecular interrogation. *Current opinion in neurobiology*, 30: 131–138, 2015.

[165] C. M. Corcoran, V. A. Mittal, C. E. Bearden, R. E. Gur, K. Hitczenko, Z. Bilgrami, A. Savic, G. A. Cecchi, and P. Wolff. Language as a biomarker for psychosis: A natural language processing approach. *Schizophrenia research*, 226:158–166, 2020.

[166] J. Oh et al. Identifying schizophrenia using structural mri with a deep learning algorithm. *Frontiers in psychiatry*, 11:16, 2020.

[167] D. Velakoulis et al. Hippocampal and amygdala volumes according to psychosis stage and diagnosis: A magnetic resonance imaging study of chronic schizophrenia, first-episode psychosis, and ultra–high-risk individuals. *Archives of general psychiatry*, 63(2):139–149, 2006.

[168] U. K. Haukvik et al. Schizophrenia–what does structural mri show? *Tidsskrift for Den norske legeforening*, 2013.

[169] C. Barros et al. Advanced eeg-based learning approaches to predict schizophrenia: Promises and pitfalls. *Artificial intelligence in medicine*, 114:102039, 2021.

[170] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[171] R. Henriques and S. C. Madeira. Bicpam: Pattern-based biclustering for biomedical data analysis. *Algorithms for Molecular Biology*, 9(1):1–30, 2014.

[172] E. N. Castanho et al. Biclustering fmri time series: a comparative study. *BMC bioinformatics*, 23(1): 1–30, 2022.

[173] G. Pfurtscheller et al. Event-related eeg/meg synchronization and desynchronization: basic principles. *Clinical neurophysiology*, 110(11):1842–1857, 1999.

[174] R. Chin et al. Recognition of schizophrenia with regularized support vector machine and sequential region of interest selection using structural magnetic resonance imaging. *Scientific Reports*, 8(1): 1–10, 2018.

[175] M. A. Rahaman et al. N-bic: A method for multi-component and symptom biclustering of structural mri data: Application to schizophrenia. *IEEE Transactions on Biomedical Engineering*, 67(1):110–121, 2019.

[176] M. Pinto et al. Interpretable eeg seizure prediction using a multiobjective evolutionary algorithm. *Scientific reports*, 12(1):1–15, 2022.

[177] C. G. Wible et al. Prefrontal cortex, negative symptoms, and schizophrenia: an mri study. *Psychiatry Research: Neuroimaging*, 108(2):65–78, 2001.

[178] D. R. Weinberger et al. The frontal lobes and schizophrenia. *The Journal of neuropsychiatry and clinical neurosciences*, 1994.

[179] J. Sui et al. Multimodal neuromarkers in schizophrenia via cognition-guided mri fusion. *Nature communications*, 9(1):1–14, 2018.

[180] B. Neyshabur et al. Path-sgd: Path-normalized optimization in deep neural networks. *NIPS*, 28, 2015.

[181] A. K. Yadav et al. Patients understanding of their diagnosis and treatment plans during discharge in emergency ward in a tertiary care centre: a qualitative study. *JNMA: Journal of the Nepal Medical Association*, 57(219):357, 2019.

[182] R. Abreu, J. Jorge, A. Leal, T. Koenig, and P. Figueiredo. Eeg microstates predict concurrent fmri dynamic functional connectivity states. *Brain topography*, 34(1):41–55, 2021.

[183] G. Van Rossum and F. L. Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

Chapter 11

PhD Research Activities

Publications

- Calhas, David and Romero, Enrique and Henriques, Rui, *On the use of pairwise distance learning for brain signal classification with limited observations*, in Artificial Intelligence in Medicine by Elsevier, 2020;
- Calhas, David and Manquinho, Vasco M and Lynce, Ines, *Automatic Generation of Neural Architecture Search Spaces*, in Association for the Advancement of Artificial Intelligence (AAAI) Workshop Combining Learning and Reasoning: Programming Languages, Formalisms, and Representations, 2022;
- Calhas, David and Henriques, Rui, *EEG to fMRI Synthesis Benefits from Attentional Graphs of Electrode Relationships*, in Machine Learning for Health Care (MLHC) Conference, 2023;
- Calhas, David and Henriques, Rui, *fMRI Multiple Missing Values Imputation Regularized by a Recurrent Denoiser*, in Artificial Intelligence in Medicine: 19th International Conference on Artificial Intelligence in Medicine by Springer, 2021;
- Calhas, David and Henriques, Rui, *Fitting Regularized Population Dynamics with Neural Differential Equations*, in Advances in Neural Information Processing Systems (NeurIPS) Workshop The Symbiosis of Deep Learning and Differential Equations, 2021;
- Calhas, David, *EEG-to-fMRI: Neuroimaging Cross Modal Synthesis in Python*, in Scipy Conference, 2023;
- Calhas, David and Henriques, Rui, *EEG to fMRI Synthesis: Is Deep Learning a candidate?*, in International Conference on Information Systems Development (ISD), 2023.

Communications

Talks

- *EEG to fMRI Synthesis*, PhD track at Symposium on Intelligent Data Analysis;

- *Fitting Regularized Population Dynamics with Neural Differential Equations*, at Advances in Neural Information Processing Systems Workshop The Symbiosis of Deep Learning and Differential Equations;
- *EEG to fMRI Synthesis: Quantifying Uncertainty*, Machine Learning in Science Workshop, `https://workshopmachinelearning.weebly.com/`;
- *EEG to fMRI Synthesis: Extrapolation for Diagnostic Settings*, ESR Talks by INESC-ID;
- *EEG to fMRI Synthesis Benefits from Attentional Graphs of Electrode Relationships*, in Machine Learning for Health Care Conference;
- *EEG to fMRI Synthesis: Is Deep Learning a candidate?*, in International Conference on Information Systems Development.

Posters

- *EEG to fMRI Synthesis*, PhD track at Symposium on Intelligent Data Analysis;
- *Fitting Regularized Population Dynamics with Neural Differential Equations*, at Advances in Neural Information Processing Systems Workshop The Symbiosis of Deep Learning and Differential Equations;
- *Automatic Generation of Neural Architecture Search Spaces*, at Association for the Advancement of Artificial Intelligence Workshop Combining Learning and Reasoning: Programming Languages, Formalisms, and Representations;
- *EEG-to-fMRI: Neuroimaging Cross Modal Synthesis in Python*, at Scipy Conference.

Scientific Meetings

- Meeting with Mathis Fleury and Neil Mehta, from LaSEEB, ISR, IST;
- Jules Padova from Synthesia, `https://youtu.be/NWYks4_lh6k`;
- 829 PhD club weekly meetings.

Articles under revision

- Calhas, David and Henriques, Rui, *EEG to fMRI Synthesis for Medical Decision Support: A Case Study on Schizophrenia Diagnosis*, submitted in Computer Methods and Programs in Biomedicine;
- Calhas, David and Henriques, Rui, *Bayesian Spectral Coefficients to Quantify Uncertainty in Neuroimaging Synthesis*, submitted in Journal of Biomedical and Health Informatics.

Appendix A

Evaluation the MNIST Dataset

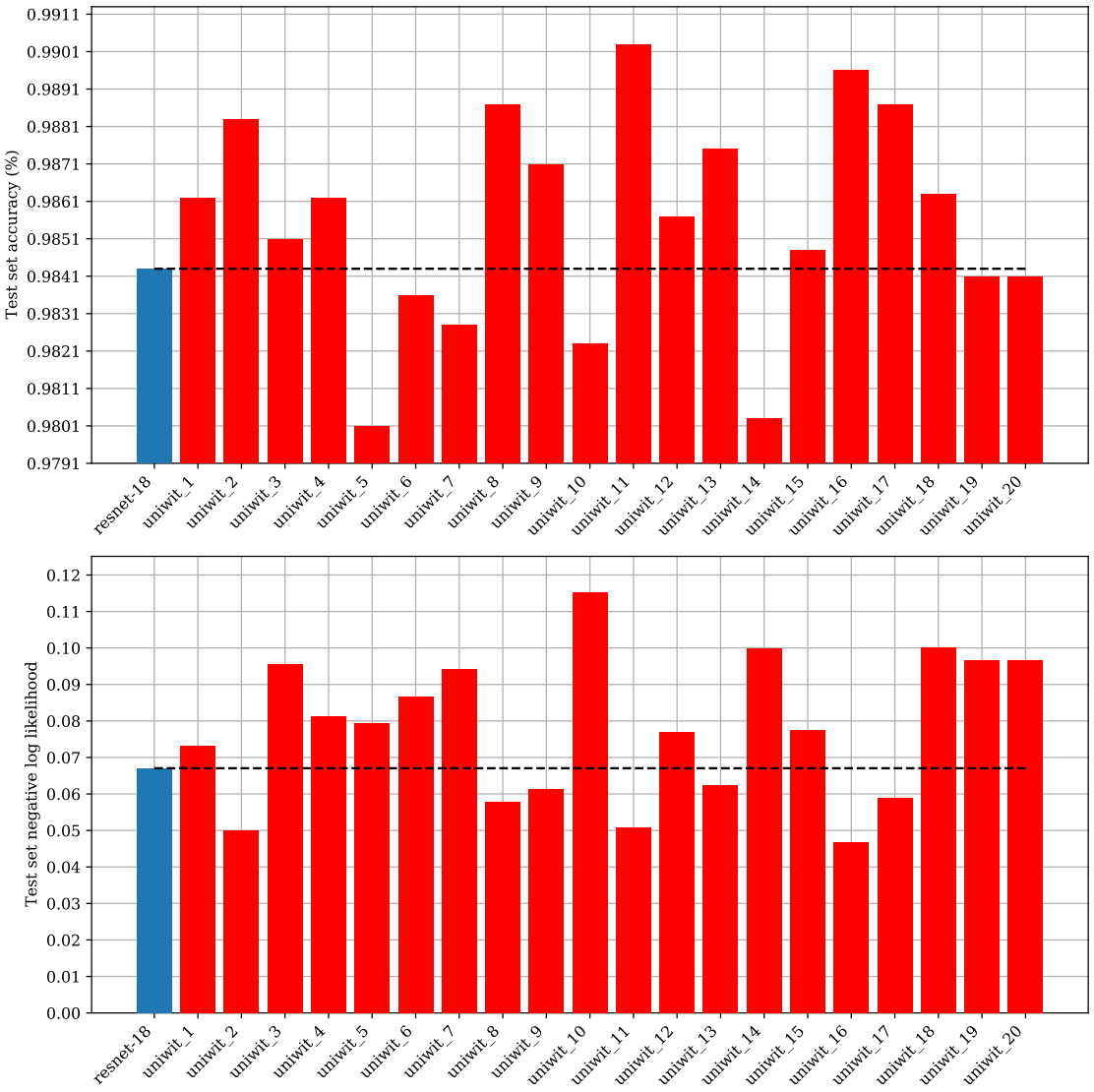


Figure A.1: The top figure shows the accuracy achieved in the MNIST test set, by each network. In blue (most left) we have the resnet and the rest of the bars represent different generated instances by the *Limited_Uniform-S* approach. On the bottom figure, the negative log likelihood computed between the ground truth and the predicted softmax logits is shown. The *uniwit_11* achieved the best accuracy of all the generated architectures, with 0.9903 accuracy, and also outperformed the Resnet by $+0.0060$. The *uniwit_5* was the architecture with worst performance, with 0.9801 accuracy. The generated networks had 0.9856 ± 0.0028 accuracy and the Resnet achieved 0.9843.

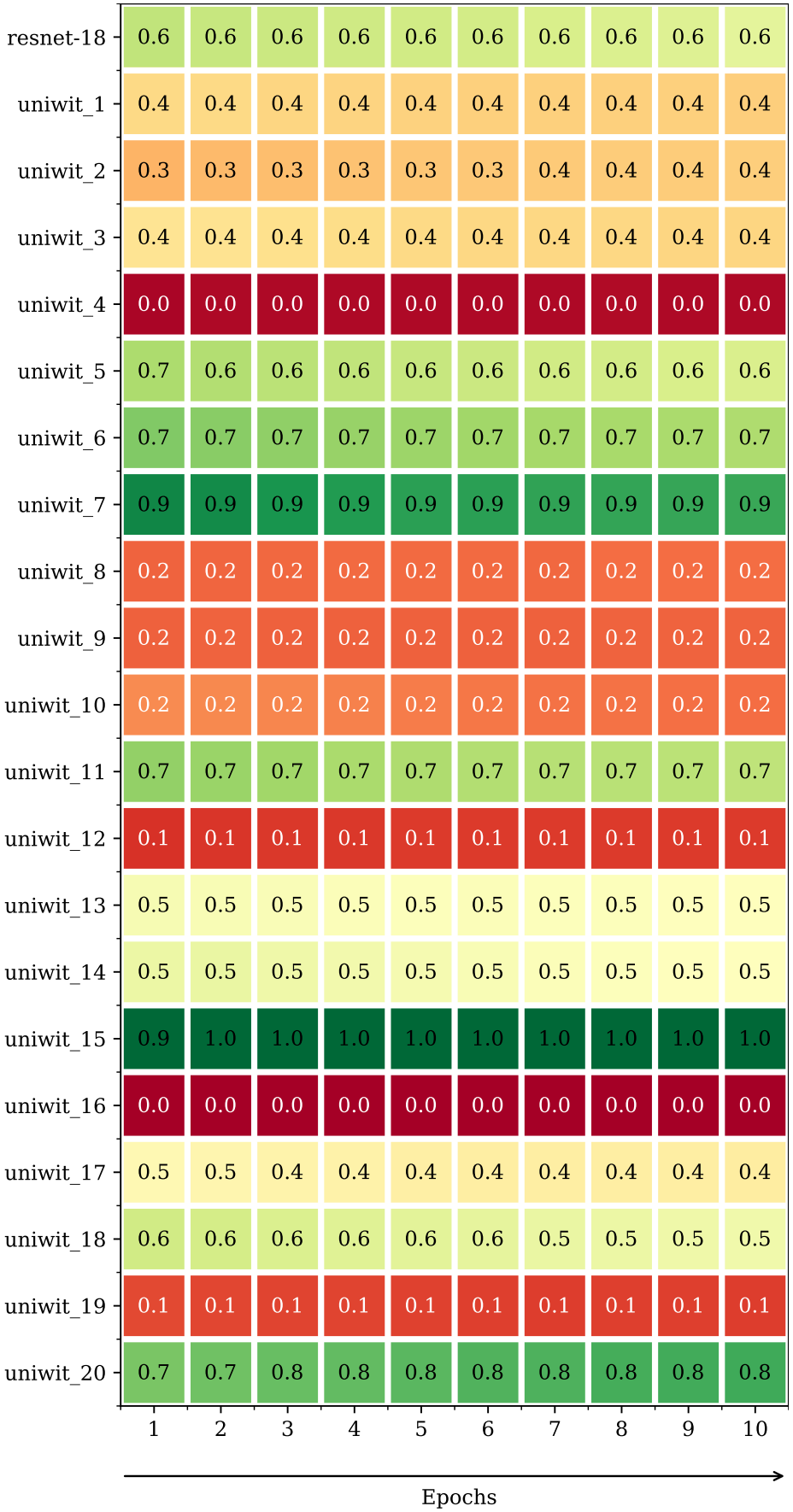


Figure A.2: A comprehensive evolution of the importance given to each network, by analyzing the weights, α, defined in Section 6.5.2. The values in this figure refer to the *Limited_Uniform-20* generated space and are normalized.

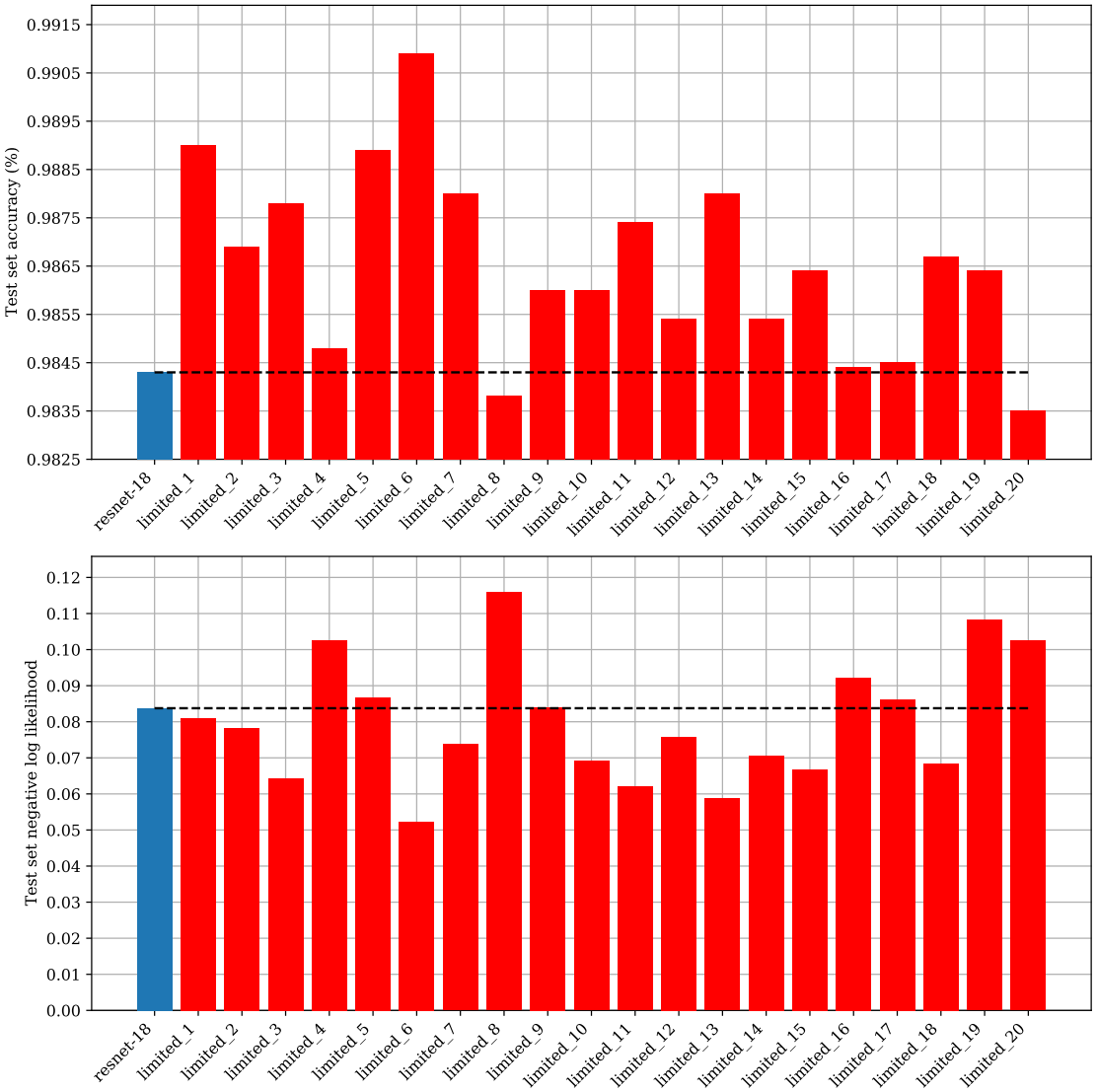


Figure A.3: The top figure shows the accuracy achieved in the MNIST test set, by each network. In blue (most left) we have the resnet and the rest of the bars represent different generated instances by the *Limited-S* approach. On the bottom figure, the negative log likelihood computed between the ground truth and the predicted softmax logits is shown. The *limited_6* achieved the best accuracy of all the generated architectures, with 0.9909 accuracy, and also outperformed the Resnet by $+0.0066$. The *limited_20* was the architecture with worst performance, with 0.9835 accuracy. The generated networks had 0.9865 ± 0.0019.

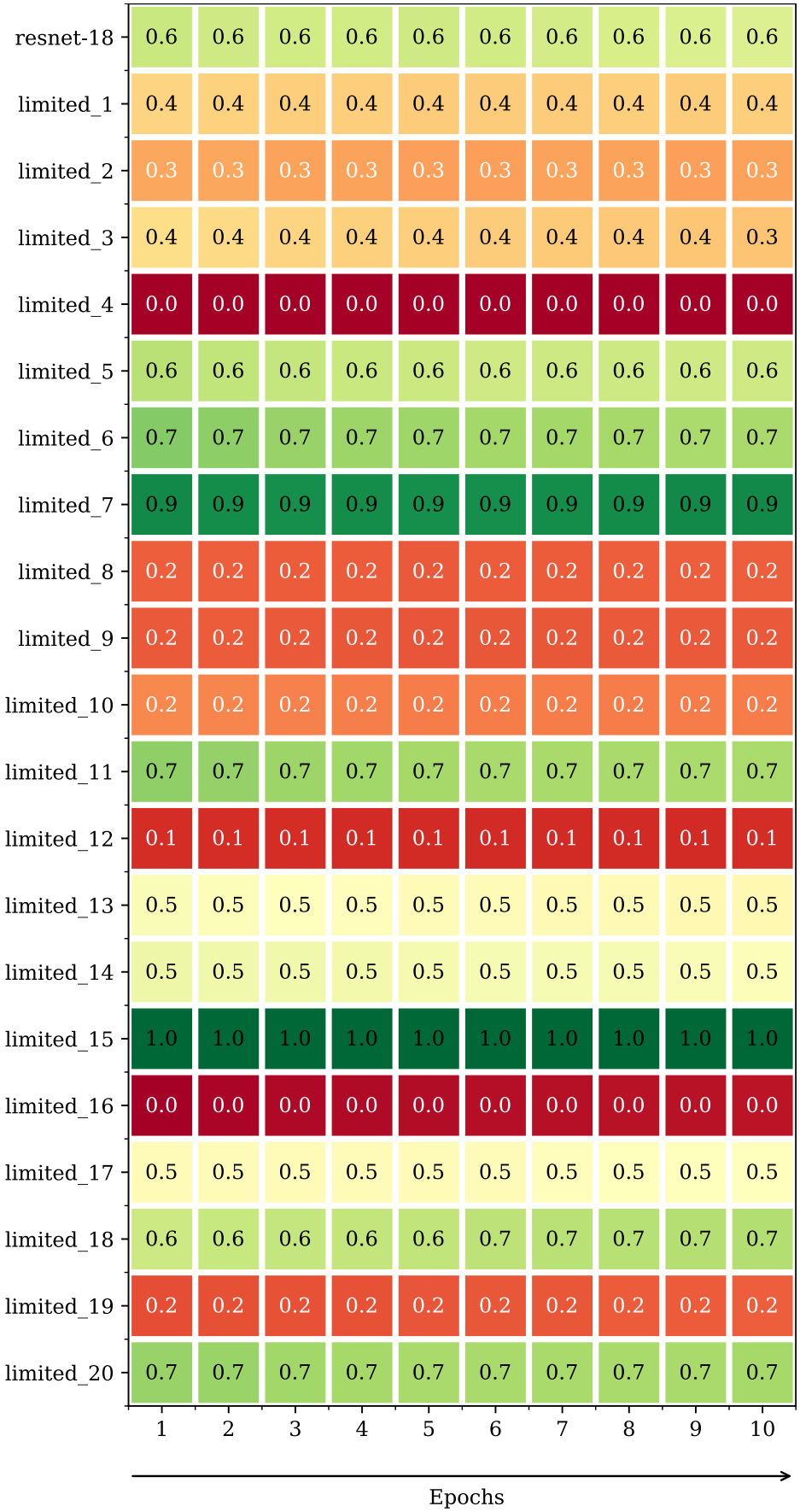


Figure A.4: A comprehensive evolution of the importance given to each network, by analyzing the weights, α, defined in Section 6.5.2. The values in this figure refer to the *Limited-20* generated space and are normalized.

Appendix B

AutoNAS Implementation

In this appendix, a detailed description of the implementation of the methodology presented in Chapter 6 is given. The methodology consists on two major steps:

1. Generation of neural arcthiectures (NAs);
2. DARTS [18] search to discover the best NA.

First, the reader is directed to the public Github repository[1], which contains the Python [183] implementation. The structure of this appendix contains a description of the generation of NAs (Section B.1) , following the DARTS algorithm implementation is described (Section B.2).

B.1 Generation of Neural Architectures - Z3-Python

We start by initializing a variable *net_restrictions* to true. Following the blocker variables, referent to the pseudo boolean optimization [111] schema introduced in Section 6.3, are setup.

```
net_restrictions = True

for l in range(self.u_bound_layers-1):
        net_restrictions = z3.And(net_restrictions, \\
                                  z3.Implies(z3.Not(self.blockers[l]), \\
                                             z3.Not(self.blockers[l+1])))
```

Above is encoded $\forall i \in \{n, \ldots, N\} : \neg x_i \implies \neg x_{i+1}$ and the following piece of code encodes $\forall i \in \{n, \ldots, N\} : x_i \implies x_{i-1}$. In addition, x_n is set to true, since we want at least n layers to N.

```
for l in range(1, self.u_bound_layers):
        net_restrictions = z3.And(net_restrictions, \\
                                  z3.Implies(self.blockers[l], \\
                                             self.blockers[l-1]))
```

[1] https://github.com/DCalhas/auto_nas_space

```
net_restrictions = z3.And(net_restrictions, \\
                          self.blockers[self.l_b

                          ound_layers-1])
```

For each layer, the kernel and stride sizes, $k_l^{(k)}$ and $s_l^{(k)}$, respectively, are given restrictions, such that $k_l^{(k)} > 0 \wedge s_l^{(k)} > 0$ and the stride can not be greater than the kernel, $k_l^{(k)} > s_l^{(k)}$.

```
for l in range(self.u_bound_layers):

    for d in range(self.D):
        net_restrictions = z3.And(net_restrictions, \\
                          self.kernel[l*self.D+d].__to_int__() - \\
                          self.stride[l*self.D+d].__to_int__() > 0)
        net_restrictions = z3.And(net_restrictions, \\
                          self.kernel[l*self.D+d].__to_int__() > 0)
        net_restrictions = z3.And(net_restrictions, \\
                          self.stride[l*self.D+d].__to_int__() > 0)
```

The *__to_int__()* is a method that converts a binary value to integers. This is done since the Uniwit [17] operates in the SAT domain, therefore integer numbers are represented by bits, being each bit a boolean variable.

Now, one needs to specify the formula of the arithmetic of convolutions for each dimension of each layer. In accordance, each layer has to be treated as either a hidden or output layer, as specified in Section 6.3, defined by H_h and H_o.

```
for l in range(self.u_bound_layers):
    new_layer_input_shape = ()

    hidden_layer = True
    output_layer = True

    layer_restrictions = True
```

For each dimension, the output of a layer, $O_l = 1 + \frac{I_l - k_l^{(k)}}{s_l^{(k)}}$, is first specified.

```
for d in range(self.D):

    if(l < self.u_bound_layers-1):
        #hidden layer
        out_l = 1+\\
        (layer_input_shape[d] - \\
```

```
self.kernel[l*self.D+d].__to_int__())/\\
self.stride[l*self.D+d].__to_int__()
```

Then, the output of the next layer, $l+1$, is also computed, given by $O_{l+1} = 1 + \frac{O_l - k_{l+1}^{(k)}}{s_{l+1}^{(k)}}$.

```
out_next_l = 1+\\
(out_l - self.kernel[(l+1)*self.D+d].__to_int__())/\\
self.stride[(l+1)*self.D+d].__to_int__()
```

Following, the input of the next layer is computed with its output, O_{l+1}, as $I_{l+1} = (O_{l+1} - 1) \times s_{l+1}^{(k)} + k_{l+1}^{(k)}$.

```
in_next_l = (out_next_l-1)*\\
self.stride[(l+1)*self.D+d].__to_int__() +\\
self.kernel[(l+1)*self.D+d].__to_int__()
```

This process corresponds to saying that $I_l = O_{l-1}$. Continuing, the layer can now be treated as a hidden layer or an output layer. It is a hidden layer if $O_l = I_{l+1}$ and an output if $O = O_l$.

```
hidden_layer = z3.And(hidden_layer, out_l == in_next_l)

output_layer = z3.And(output_layer, \\
(self.output_shape[d] - 1)*\\
self.stride[l*self.D+d].__to_int__() +\\
self.kernel[l*self.D+d].__to_int__() == \\
layer_input_shape[d])

new_layer_input_shape += (in_next_l,)
```

The blockers are now used to represent the number of layers specified, being hidden or output.

```
hidden_layer = z3.Or(hidden_layer, \\
z3.Or(z3.Not(self.blockers[l]), z3.Not(self.blockers[l+1])))

output_layer = z3.Or(output_layer, \\
z3.Or(z3.Not(self.blockers[l]), self.blockers[l+1]))

net_restrictions = z3.And(net_restrictions, \\
z3.And(hidden_layer, output_layer))
```

This is done for all dimensions of all layers. The process is less complex for the final layer, where one just has to account for it being an output layer. As of now the user has the formula F encoded. In extension, the Uniwit algorithm is also coded, but since it is not proposed in this work, the implementation is not documented, please refer to the code for its implementation.

B.2 DARTS over generated NAs - Python

Once the formula is encoded, one is able to generate a set of architectures, please refer to the github repository (beginning of the appendix) for the specific command to generate NAs. With a space of NAs defined, one is only left with the optimization algorithm DARTS. The algorithm is not proposed by us, but since the computational resource limitations forced us to implement a specific version of it, it will be described here.

The Resnet-18 along with a batch size of 256 needs approximately 4GB of GPU dedicated to the process. Since the hardware, the experiences were ran on, has a total of 7GB of GPU memory, one can only fit one network at a time. With this said, the computational graph built by tensorflow [89] can not be preserved therefore the gradients $\nabla_{w_i}\mathcal{L}(\hat{y}, y)$, which correspond to the final softmax weighted prediction of all networks $\hat{y}$, can not be computed for each network. Instead, the prediction of each network $a_i(\vec{x})$ is used to compute the gradients of each network, a_i, with $\nabla_{w_i}\mathcal{L}(a_i(\vec{x}), y)$, and the softmax layer is given $\nabla_{\alpha}\mathcal{L}(\hat{y}, y)$. In terms of code implementation, this means that there is a parent process that launches child processes one at a time (so that all GPU memory is not occupied), first launching all the networks.

The learning loop corresponds to launching processes for each batch computation.

```
for epoch in range(epochs):

    for batch in range(1,n_batches+1):
        #do not train with last batch, save weights if needed
        if(batch+1==n_batches+1):
            if(save_weights):
                process_utils.launch_process(\\
                process_utils.save_weights, \\
                (epoch, na_path, save_weights_path))
            continue
```

The predictions of all networks need to be stored in an array maintained by the parent process, *o_predictions*.

```
flattened_predictions = Manager().Array('d',\\
        range(batch_size*x_dim*y_dim*z_dim))
o_predictions = np.zeros((networks,\\
                          batch_size,\\
                          x_dim,y_dim,z_dim,1),\\
                          dtype=np.float32)
```

And a process is launched for each network, which is responsible for making the predictions, $a_i(\vec{x})$, and storing them in a shared array. Additionally, the gradients, $\nabla_{w_i}\mathcal{L}(a_i(\vec{x}), y)$, are also propagated in that process. When the process finsihes, the predictions are saved in another array.

```
for network in range(networks):
        process_utils.launch_process(\\
            process_utils.batch_prediction, \\
                (flattened_predictions, batch_path, \\
```

```
                batch, epoch, network, na_path, \\
                batch_size, learning_rate, \\
                memory_limit, seed))

        o_predictions[network]=\\
                np.array(flattened_predictions).\\
                reshape((batch_size,x_dim,y_dim,z_dim,1))
```

And in the end, the process with the softmax layer is launched to make the final prediction and propagate the gradients, $\nabla_\alpha \mathcal{L}(\hat{y}, y)$, to that layer.

```
process_utils.launch_process(\\
    process_utils.continuous_training, \\
        (o_predictions, batch_path, \\
        batch, learning_rate, \\
        epoch, na_path, \\
        memory_limit, seed))
```

The graph state of each network or, in other words, the context that tensorflow maintains during the training of each network is saved and in the next batch, tensorflow recognizes the previous training steps, so that the random generator continues in its previous state. The latter is important to ensure convergence of the algorithm.