Machine learning for bi-modal EEG-fMRI neurofeedback : EEG electrodes localization and fMRI NF scores prediction

Caroline Pinte

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Informatique*

Par
Caroline PINTE

« Machine learning for bi-modal EEG-fMRI neurofeedback: EEG electrodes localization and fMRI NF scores prediction »

Thèse présentée et soutenue à Rennes, le 29/11/2024
Unité de recherche : Univ Rennes, CNRS, Inria, Inserm, IRISA UMR 6074, EMPENN — ERL U 1228, F-35000 Rennes, France.

Rapporteurs avant soutenance :

Patrícia FIGUEIREDO	Professeure, Instituto Superior Técnico, Lisboa, Portugal
Fabien LOTTE	Directeur de recherche, Inria, Bordeaux, France

Composition du Jury :

Président.e :	Anatole LECUYER	Directeur de recherche, Inria, Rennes, France
Examinateurs :	Patrícia FIGUEIREDO	Professeure, Instituto Superior Técnico, Lisboa, Portugal
	Fabien LOTTE	Directeur de recherche, Inria, Bordeaux, France
	Frédéric GROUILLER	Directeur de recherche, CIBM MRI HUG, Université de Genève, Suisse
Dir. de thèse :	Pierre MAUREL	Professeur, Université de Rennes, France
Co-dir. de thèse :	Claire CURY	Chargée de recherche, Inria, Rennes, France

Remerciements

Je souhaite commencer par remercier Claire et Pierre pour leur incroyable soutien tout au long de ma thèse. Merci de m'avoir guidée, encouragée et rassurée avec patience et bienveillance jusqu'au dernier jour de ce parcours.

Je tiens également à exprimer ma reconnaissance aux membres du jury : Patrícia Figueiredo, Fabien Lotte, Frédéric Grouiller et Anatole Lecuyer. Votre bienveillance et votre intérêt ont rendu cette journée de soutenance très agréable et enrichissante. Merci pour votre présence et vos retours constructifs.

Un grand merci à toute l'équipe Empenn pour ce cadre de travail si agréable. À toutes celles et ceux que j'ai eu la chance de croiser au cours de ces trois années, merci pour votre accueil, vos conseils, et ces instants partagés autour d'un café ou d'un déjeuner. Je pense ici notamment à Manu, Fanny, Malo, Camille, Francesca, Michael, Quentin, Benoît, Anne, Élise, Isabelle, Armelle, Burhan, François, Camille, Benjamin, Cédric, Julie, Boris, Pierre-Henri, Julien, Arthur, Florent, Alexandre, Gwendal, mais aussi à celles et ceux qui ont quitté l'équipe avant moi : Élodie, Agustina, Jean-Charles, Thomas, Raphaël, Lou, Xavier, Mathis, Olivier, Giulia ... Un soutien tout particulier aux doctorants : Carla, Seb, Constance, Ricky, Marie, Carlo, Nolwenn, Youenn, Valentine, Melvin, Mathys, Adèle, Mathilde, Grégoire, Youwan... et à tous ceux qui s'engageront dans cette voie dans les années à venir. Je vous souhaite à toutes et tous de parvenir au bout de vos thèses respectives le plus sereinement possible, et j'espère être présente pour célébrer les prochaines soutenances !

Merci également aux autres membres du laboratoire Inria-IRISA que j'ai eu l'occasion de côtoyer. Les échanges inter-équipes ont été enrichissants et inspirants, tant sur le plan scientifique que personnel.

Je remercie aussi mon équipe d'accueil au Japon, l'équipe DecNef, pour cette expérience inoubliable. Merci à Aurelio, Mor, Hugo, Egor, Reiko, Miho-san, Guill-san, Nakamura-san, Akiko-san, ainsi que Reinmar de l'équipe DBI. Votre accueil et vos conseils ont fait de ce séjour une période d'une richesse incroyable.

Un immense merci à ma famille pour leur présence et leurs encouragements. À mes parents, Gilles et Gaëlle, et à mon frère Alexandre, merci pour tout. Merci à ma grand-

mère Suzanne, ainsi qu'à toute la famille : Noëlie, Gildas, Gwénola, Manu, Rozenn, Jean-Jacques, Jeanne, Vincent, Katell, Gérard, Laurence, Alexia, Carla, et tous les autres qui m'ont adressé de gentils messages. Merci également à la famille Cusson, Philippe, Marie-Cécile, Thomas, Annabelle et Youen, pour leur accueil chaleureux.

Enfin, merci à mes amis pour leur soutien sans faille. Merci à Aziliz, Lisa et Agathe pour nos appels, nos moments de jeu et de détente à distance. Merci à Enora, Nolwenn, Ludivine, Romane, Pauline, Cylia, et Manon pour leur amitié constante. Un remerciement spécial au cours de danse du lundi soir pour avoir célébré avec moi le rendu de la thèse !

Et pour finir, merci à Marin d'avoir toujours été là pour moi durant toutes ces années, j'espère pouvoir passer encore beaucoup de temps à tes côtés.

Table of Contents

List of Figures

List of Tables

Acronyms

AI	Artificial Intelligence
BCI	Brain-Computer Interface
BOLD	Blood Oxygenation Level Dependent
CNN	Convolutional Neural Network
EA	Euclidean-space Alignment
EEG	Electroencephalography
ERD	Event-Related Desynchronization
FNN	Feedforward Neural Networks
fMRI	Functional Magnetic Resonance Imaging
GPU	Graphics Processing Unit
ICP	Iterative Closest Point
LSTM	Long Short-Term Memory
M1	Primary Motor Area
MI	Motor Imagery
MR	Magnetic Resonance
MSE	Mean Squared Error
NF	Neurofeedback
NIRS	Near-InfraRed Spectroscopy
NN	Neural Network
PE	Position Error
PETRA	Pointwise Encoding Time reduction with Radial Acquisition
PPV	Positive Predictive Value
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
ROI	Region Of Interest
SGD	Stochastic Gradient Descent
SMA	Supplementary Motor Area
t-SNE	t-distributed Stochastic Neighbor Embedding
UTE	Ultrashort Echo Time

Résumé en français

Ce résumé en français présente le contexte dans lequel s'inscrivent les recherches menées au cours de la thèse, les questionnements que les différentes contributions ont adressé, ainsi qu'une vue d'ensemble des chapitres qui composent ce document.

Contexte

Comment mesurer l'activité cérébrale ?

Mesurer l'activité cérébrale est essentiel pour de nombreuses raisons : mieux comprendre le fonctionnement du cerveau, diagnostiquer des dysfonctionnements, et développer des solutions thérapeutiques. Il existe plusieurs techniques, aussi connues sous le nom de modalités, pour mesurer l'activité cérébrale, chacune avec des objectifs spécifiques et des contraintes associées. Les données utilisées au cours de cette thèse proviennent des modalités appelées électroencéphalographie (EEG) et imagerie par résonance magnétique fonctionnelle (IRMf).

L'EEG est une technique qui mesure les variations de l'activité électrique générée par les neurones à l'aide d'électrodes. Celles-ci sont placées à la surface du cuir chevelu et sont généralement accompagnées d'un gel conducteur qui réduit l'impédance, c'est-à-dire la résistance au passage d'un courant électrique, le tout étant maintenu par un bonnet souvent légèrement élastique. C'est donc une modalité dite non invasive.

L'IRMf est une technique de neuro-imagerie utilisée pour mesurer l'activité cérébrale en détectant les changements dans le flux sanguin. Contrairement à l'EEG, qui enregistre l'activité électrique à la surface du cuir chevelu, l'IRMf utilise le signal BOLD (de l'anglais *blood oxygen level dependent*) pour en déduire l'activité neuronale. Cette modalité est également non invasive.

Comme toutes les modalités, l'EEG et l'IRMf ont leurs avantages et leurs inconvénients. Concernant l'EEG, son principal intérêt est son excellente résolution temporelle. C'est également une modalité peu coûteuse, silencieuse et mobile, puisque l'équipement peut être facilement déplacé. En revanche, l'EEG est très sujette au bruit et a une résolution spatiale assez faible.

Concernant l'IRMf, elle dispose de l'une des meilleures résolutions spatiales de toutes les modalités de neuro-imagerie, même pour l'étude des régions profondes du cerveau, sans nécessiter une exposition à des radiations ionisantes. En revanche, la technique est très coûteuse et peut être contraignante pour les participants, puisqu'elle nécessite que le sujet soit allongé dans un espace confiné et bruyant. De plus, sa résolution temporelle est inférieure à celle de l'EEG.

Ainsi, puisque ces deux modalités ont des forces et des faiblesses complémentaires, l'utilisation simultanée de l'EEG avec l'IRMf ouvre la porte à une localisation plus précise de l'activité cérébrale dans le temps et dans l'espace.

Qu'est-ce que le neurofeedback ?

Le neurofeedback est une technique qui présente au participant un score en temps réel, que l'on appelle feedback, indiquant si son activité cérébrale correspond à l'activité souhaitée, dans le but de réguler ou de réhabiliter l'activité neuronale. Il existe de nombreux protocoles pour une grande variété d'applications. L'idée générale comprend trois grandes étapes qui forment une boucle : l'acquisition par une ou plusieurs modalités, le traitement des données dépendant du protocole, et le feedback pouvant être présenté de manière visuelle, mais aussi auditive ou encore haptique.

La modalité utilisée historiquement pour l'acquisition de l'activité cérébrale en neurofeedback est l'EEG, bien que l'IRMf soit également très utilisée dans le domaine. Néanmoins, puisque l'EEG et l'IRMf présentent une complémentarité dans leurs résolutions temporelles et spatiales, le domaine du neurofeedback peut également bénéficier de la bi-modalité EEG-IRMf.

Qu'est-ce qu'un réseau de neurones artificiels ?

Un réseau de neurones artificiels est constitué d'unités de calcul appelées neurones. Au départ, l'idée était de s'inspirer du fonctionnement des neurones qui composent le cerveau, d'où le terme « neurone » pour désigner une unité de calcul. Cependant, le concept a depuis été étendu avec des améliorations qui vont au-delà de son inspiration biologique. Dans le réseau, les neurones sont connectés par des arêtes, qui relient les neurones d'une couche à une autre. Chaque arête est associée à un poids, une valeur numérique qui pondère les informations transmises. Les poids sont cruciaux car ils représentent la capacité du réseau à apprendre et à faire des prédictions précises. En effet, ces réseaux de neurones artificiels font partie de la grande famille des méthodes d'apprentissage automatique. Le

cadre général de l'apprentissage automatique peut être divisé en deux étapes : la phase d'apprentissage, où ces poids sont appris grâce au passage de nombreuses données dans le réseau, et la phase de test, où le réseau a terminé son apprentissage, dispose désormais de poids fixes, et peut ainsi être utilisé pour faire des prédictions sur de nouvelles données.

Comment segmenter une image ?

La segmentation d'image est une tâche faisant partie du domaine de la vision par ordinateur, qui permet de classer les pixels d'une image dans différentes régions correspondant à divers objets ou instances d'objets. Chaque pixel est donc affecté à une classe, ce qui permet de diviser l'image en zones d'intérêt distinctes.

Parmi les nombreuses méthodes permettant de réaliser cette tâche, les méthodes d'apprentissage automatique comme les réseaux de neurones artificiels permettent d'effectuer des segmentations dites automatiques, qui ne nécessitent pas d'interactions manuelles de la part de l'utilisateur. Cependant, elles reposent généralement sur une approche d'apprentissage supervisé qui nécessite un ensemble de données d'apprentissage avec des étiquettes représentant le résultat souhaité, connu sous le nom de vérité terrain.

La catégorie de réseaux de neurones artificiels la plus utilisée pour les tâches de segmentation d'image est celle des réseaux de neurones convolutifs (CNN en anglais) [1], dont les couches convolutives permettent d'apprendre et de détecter les caractéristiques spatiales des images. Cette catégorie comprend l'architecture U-Net [2], un type de réseau qui permet d'obtenir de très bonnes performances de segmentation avec un faible nombre d'images d'entraînement. En effet, cette architecture fut initialement conçue pour l'imagerie biomédicale, un domaine où le nombre d'échantillons disponibles est souvent limité. Elle permet la segmentation d'images, mais également de volumes 3D, comme par exemple les volumes IRM qui nous intéressent ici.

Comment prédire une valeur à partir d'une série temporelle ?

La régression de séries temporelles est un autre type de tâche d'apprentissage automatique. L'objectif est de prédire une valeur continue sur la base de données dites temporelles ou séquentielles. Contrairement aux tâches de régression standard, où les valeurs sont généralement supposées être indépendantes, la régression de séries temporelles prend justement en compte cette relation temporelle entre les points.

De nombreux types de réseaux de neurones artificiels peuvent être utilisés pour effectuer cette tâche. L'un des plus importants est l'architecture LSTM [3] (de l'anglais

long short-term memory). Les réseaux LSTM comprennent des cellules dites de mémoire à long terme qui peuvent maintenir et ajuster l'information sur de longues périodes. Une autre manière d'appréhender cette tâche est d'utiliser les réseaux de neurones convolutifs que nous avons évoqués précédemment. Conçus à l'origine pour les images, ils ont également été adaptés aux séries temporelles, sous le nom de réseaux de neurones convolutifs unidimensionnels (1D CNN en anglais).

Objectifs

Cette thèse a pour but d'adresser et d'explorer les questions suivantes :

➔ **Comment obtenir automatiquement et avec précision la position de chaque électrode d'un dispositif EEG dans un volume IRM ?**

La localisation des sources en EEG implique la résolution d'un problème inverse sensible à plusieurs paramètres, notamment la position des électrodes EEG sur le cuir chevelu. En effet, la précision des estimations des coordonnées des électrodes a un impact direct sur la localisation des sources EEG. Les erreurs de position peuvent entraîner des imprécisions dans leur estimation. Ce problème est d'autant plus présent lors d'études où plusieurs sessions, et donc plusieurs installations de dispositifs EEG, sont nécessaires. Cependant, dans le contexte d'acquisitions simultanées EEG-IRM, on dispose d'un élément utile pour répondre à ce problème : le volume IRM lui-même. En effet, en utilisant une séquence IRM particulière appelée UTE (de l'anglais *ultrashort echo-time*) permettant de visualiser les électrodes à l'intérieur du volume IRM, on dispose de suffisamment d'informations pour procéder à une segmentation de volume afin de retrouver la position des électrodes. La contribution relative à cette question est présentée au chapitre 3.

➔ **Est-il possible de prédire des scores neurofeedback IRMf uniquement à partir de signaux EEG via une modélisation sur plusieurs sujets ?**

Dans le contexte du neurofeedback, l'acquisition de l'activité cérébrale se fait traditionnellement avec un dispositif EEG, qui est relativement peu coûteux. Plus récemment, les acquisitions simultanées EEG-IRMf sont également employées, notamment pour leur complémentarité en termes de résolutions temporelles et spatiales. Cependant, l'utilisation de l'IRM impose certaines contraintes. C'est une modalité très coûteuse, et qui demande au participant de rester immobile dans un espace confiné et bruyant. Ainsi, parvenir à

prédire les scores neurofeedback IRMf à partir des signaux EEG permettrait de réduire l'usage contraignant de l'IRM tout en conservant ses avantages. Cette question a commencé à être explorée avec le développement de modèles individuels [4]. Cependant, le développement d'un modèle global permettrait une application plus accessible et pratique dans un contexte clinique. Un modèle dit global est un modèle entraîné à partir de données provenant de différents sujets, et est donc immédiatement applicable à un nouveau participant. De plus, à plus long terme, un modèle global pourrait aider à investiguer la question des liens entre les deux modalités. La contribution relative à ces questionnements est présentée au chapitre 4.

➔ **La réduction de la variabilité inter-sujet peut-elle améliorer les performances de ces modèles de prédiction des scores neurofeedback IRMf ?**

Cette question est un prolongement du questionnement précédent. Parmi les possibilités d'amélioration des performances des modèles cherchant à prédire les scores neurofeedback IRMf à partir des signaux EEG de manière globale, réduire la variabilité entre les données provenant de différents sujets est une piste intéressante. En effet, les distributions de données EEG peuvent présenter des caractéristiques exagérément différentes d'un sujet à l'autre, ce qui peut affecter les performances des modèles cherchant à apprendre une relation globale entre les entrées et les sorties, ainsi qu'à généraliser à de nouveaux sujets. La contribution relative à cette question est présentée au chapitre 5.

Organisation du manuscrit

Partie I : La première partie présente le contexte dans lequel s'inscrit cette thèse. Elle est constituée de deux chapitres.

Chapitre 1 : Ce premier chapitre présente les rôles distincts mais complémentaires de l'électroencéphalographie (EEG) et de l'imagerie par résonance magnétique fonctionnelle (IRMf) dans l'étude de l'activité cérébrale. Chaque modalité a ses propres avantages : l'EEG offre une haute résolution temporelle, tandis que l'IRMf offre une haute résolution spatiale. Utilisées conjointement, ces modalités permettent d'obtenir une vision plus complète des fonctions cérébrales. Le chapitre introduit également le neurofeedback (NF), en détaillant comment le processus, sous forme de boucle, permet aux participants de moduler leur activité cérébrale grâce à un retour d'information appelé feedback. Le

potentiel de la combinaison de l'EEG et de l'IRMf dans le neurofeedback est également abordé. Cet aperçu permet de mieux comprendre le contexte et les données utilisées dans les contributions de la thèse.

Chapitre 2 : Ce chapitre présente les principes généraux des réseaux de neurones artificiels, en particulier la façon dont ils mettent à jour leurs poids par rétropropagation. Puis, il introduit deux domaines étudiés en apprentissage automatique : la segmentation d'images et la régression de séries temporelles. Les méthodes de segmentation d'images sont très nombreuses, allant de la segmentation manuelle, qui nécessite beaucoup de temps et d'expertise, à la segmentation automatique utilisant des techniques d'apprentissage profond. Ce chapitre propose un aperçu détaillé des couches qui constituent un réseau neuronal convolutif (CNN en anglais), suivi d'un examen approfondi de l'architecture U-Net dans ses formes 2D et 3D, qui sont largement reconnues dans le domaine biomédical. Enfin, on y retrouve l'évolution historique des techniques de régression des séries temporelles, se concentrant ensuite sur les réseaux LSTM (de l'anglais *long short-term memory*) et les réseaux de neurones convolutifs unidimensionnels (CNN 1D). Ce chapitre apporte ainsi une meilleure compréhension des concepts et méthodes utilisés dans les contributions de cette thèse.

Partie II : La deuxième partie détaille les contributions de cette thèse, qui adressent les questionnements évoqués ci-dessus. Elle comporte trois chapitres.

Chapitre 3 : Dans ce chapitre, nous présentons une nouvelle méthode de détection et d'étiquetage des électrodes EEG dans un volume IRM acquis à l'aide d'une séquence spécifique dite UTE (de l'anglais *ultrashort echo-time*), appelée PETRA (de l'anglais *pointwise encoding time reduction with radial acquisition*). La méthode, dont une illustration est fournie ci-dessous (figure 1) et dans le chapitre en question, consiste à entraîner un modèle sur un ensemble de données d'entraînement avec les vérités terrains associées, puis à utiliser ce modèle pour obtenir des cartes de segmentation, et enfin à appliquer une étape de raffinement basée sur la méthode ICP (de l'anglais *iterative closest point*) pour améliorer les détections et leur étiquetage. Cette méthode, permettant d'obtenir des segmentations entièrement automatiques, est facile à mettre en œuvre, nécessite très peu d'étapes et produit d'excellents résultats.

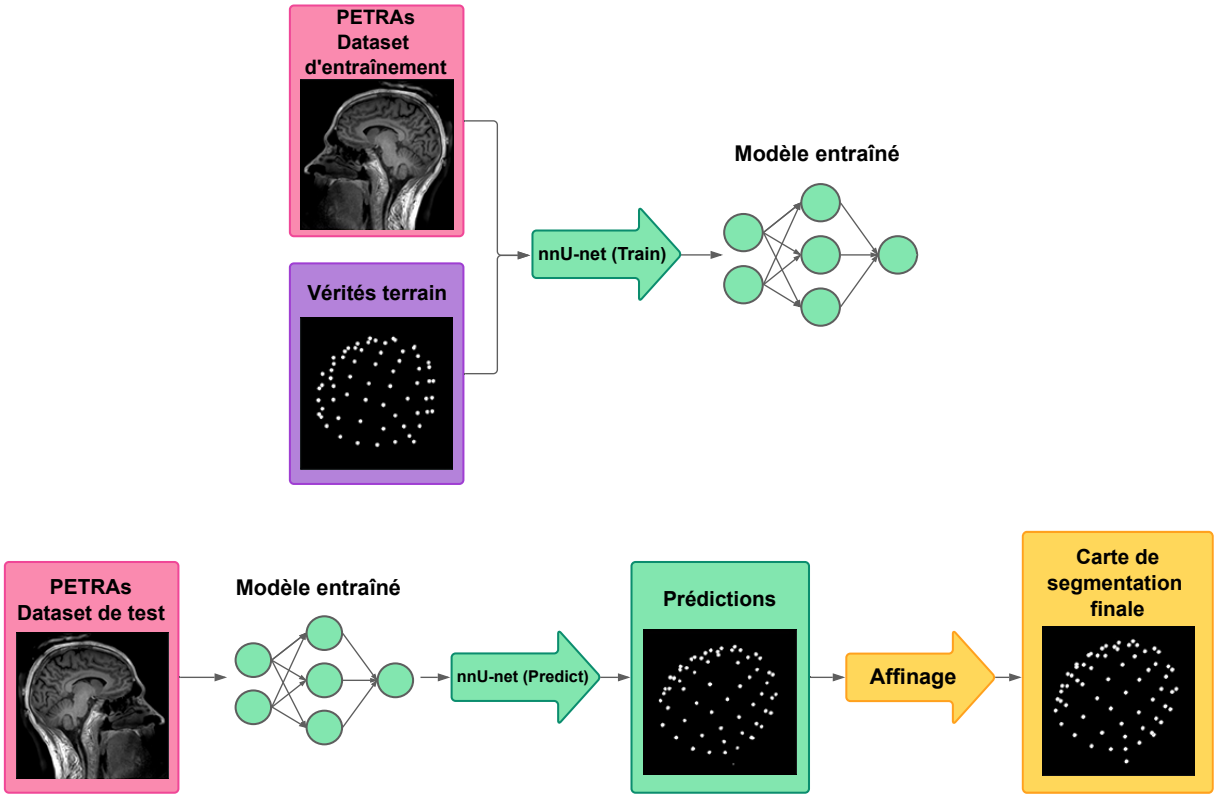


Figure 1 – **Vue d'ensemble de la méthode présentée au chapitre 3.**

Chapitre 4 : Ce chapitre explore la prédiction des scores neurofeedback IRMf à partir des signaux EEG par un modèle global, dans un contexte illustré (figure 2) ci-dessous. Afin d'étudier différents types de modèles, nous présentons une méthode de recherche d'architecture de modèle basée sur un algorithme génétique. Cette méthode flexible peut être facilement adaptée à différents types de réseaux de neurones. Nous utilisons cet algorithme génétique pour étudier quatre configurations, comprenant deux types de réseaux de neurones : LSTM et CNN 1D, ainsi que deux manières de préparer les données : l'une basée sur des caractéristiques extraites du signal et l'autre basée sur le signal brut. Après une analyse poussée des résultats, c'est l'approche CNN 1D avec caractéristiques extraites qui présente des performances légèrement supérieures en termes d'erreur quadratique moyenne (MSE en anglais) par rapport aux autres architectures testées. Cependant, ces performances ne sont pas suffisamment satisfaisantes, ce qui amène d'autres questionnements et pistes d'améliorations.

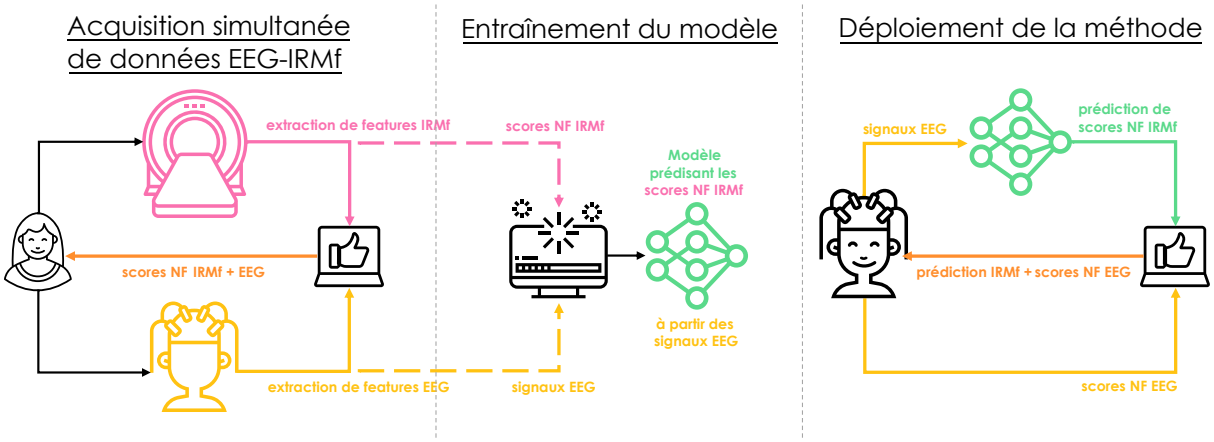


Figure 2 – **Vue d'ensemble du contexte de la méthode présentée au chapitre 4.**

Chapitre 5 : Ce dernier chapitre représente un premier pas vers la réduction de la variabilité inter-sujet dans le cadre des modèles globaux de prédiction des scores neurofeedback IRMf à partir des signaux EEG. Pour ce faire, nous appliquons à nos signaux EEG une méthode appelée alignement dans l'espace euclidien (de l'anglais *Euclidean space alignment (EA)*), qui aligne les données de chaque sujet dans le but de réduire les différences individuelles et de rendre ainsi les données plus comparables pour les tâches d'apprentissage automatique. Nous observons les effets de cet alignement à l'aide de la projection t-SNE (de l'anglais *t-distributed stochastic neighbor embedding*), une technique non linéaire de réduction de la dimensionnalité qui regroupe les points similaires et sépare les points dissemblables afin de visualiser les différences sous-jacentes dans les distributions. Avec cet outil, nous observons une certaine réduction de la variabilité entre les sujets après l'alignement, bien que cela reste imparfait. Enfin, nous évaluons l'impact de cet alignement en utilisant la configuration la plus performante du chapitre précédent, en appliquant exactement la même méthode qu'au chapitre 4, ce qui révèle que l'application de la méthode EA n'améliore pas les performances des modèles de prédiction des scores NF IRMf. Malgré cela, la piste reste prometteuse et demande une exploration plus poussée pour progresser vers un modèle global.

General introduction

This general introduction presents the context in which the research conducted during the thesis takes place, the questions that the various contributions addressed, as well as an overview of the chapters that constitute this document.

Context

How to measure brain activity?

Measuring brain activity is essential for many reasons: to better understand brain function, diagnose dysfunctions, and develop therapeutic solutions. Several techniques, also known as modalities, can be used for measuring brain activity, each with specific advantages and associated constraints. The data used in this thesis come from modalities known as electroencephalography (EEG) and functional magnetic resonance imaging (fMRI).

EEG is a technique that measures variations in the electrical activity generated by neurons using electrodes. These electrodes are placed on the scalp and are typically associated with a conductive gel that reduces impedance (i.e., resistance to the passage of electrical current), all of which is held in place by a slightly elastic cap. Thus, it is a so-called non-invasive modality.

fMRI is a neuroimaging technique used to measure brain activity by detecting changes in blood flow. Unlike EEG, which records electrical activity on the scalp, fMRI uses the BOLD (blood oxygen level dependent) signal to infer neuronal activity. This modality is also non-invasive.

Like all modalities, EEG and fMRI have their advantages and disadvantages. The main advantage of EEG is its excellent temporal resolution. It is also an low-cost, quiet, and mobile modality, as the equipment can be easily transported. However, EEG is very prone to noise and has relatively low spatial resolution.

Regarding fMRI, it offers one of the best spatial resolutions among all neuroimaging modalities, even for studying deep brain regions, without requiring exposure to ionizing

radiation. However, the technique is very expensive and can be restrictive for participants, as it requires the subject to lie in a confined and noisy space. Additionally, its temporal resolution is lower than that of EEG.

Therefore, since these two modalities have complementary strengths and weaknesses, the simultaneous use of EEG with fMRI paves the way for more precise localization of brain activity in both time and space.

What is neurofeedback?

Neurofeedback is a technique that provides the participant with a real-time score, called feedback, indicating whether their brain activity corresponds to the desired activity, with the goal of regulating or rehabilitating neuronal activity. There are many protocols for a wide variety of applications. The general idea involves three main steps forming a loop: acquisition through one or more modalities, data processing depending on the protocol, and feedback, which can be presented visually, as well as auditorily or haptically.

The modality historically used for acquiring brain activity in neurofeedback is EEG, although fMRI is also widely used in the field. However, since EEG and fMRI have complementary temporal and spatial resolutions, the field of neurofeedback can also benefit from the EEG-fMRI bi-modality.

What is an artificial neural network?

An artificial neural network consists of computing units called neurons. Initially, the idea was to to draw inspiration from the functioning of the neurons that compose the brain, hence the term "neuron" to designate a computing unit. However, the concept has since been extended with improvements that go beyond its biological inspiration. In the network, neurons are connected by edges that link the neurons from one layer to the other. Each edge is associated with a weight, a numerical value that ponders the information transmitted. Weights are crucial because they represent the network's ability to learn and make accurate predictions. Indeed, these artificial neural networks are part of the broader family of machine learning methods. The general framework of machine learning can be divided into two phases: the learning phase, where these weights are learned by giving large amounts of data to the network, and the testing phase, where the network has completed its learning, has now fixed weights, and can thus be used to make predictions on new data.

How to segment an image?

Image segmentation is a task within the field of computer vision, which involves classifying the pixels of an image into different regions corresponding to various objects or instances of objects. Each pixel is therefore assigned to a class, allowing the image to be divided into distinct areas of interest.

Among the many methods available for performing this task, machine learning methods such as artificial neural networks enable the so-called automatic segmentations, which do not require manual interactions from the user. However, they generally rely on a supervised learning approach that requires a training dataset with labels representing the desired outcome, known as the ground truth.

The category of artificial neural networks most commonly used for image segmentation tasks is convolutional neural networks (CNNs) [1], whose convolutional layers allow for learning and detecting spatial features within images. This category includes the U-Net architecture [2], a type of network that provides excellent segmentation performance even with a small number of training images. Indeed, this architecture was initially designed for biomedical imaging, a field where the number of available samples is often limited. It enables the segmentation of both images and 3D volumes, such as the MRI volumes that interest us here.

How to predict a value from a time series?

Time series regression is another type of machine learning task. The goal is to predict a continuous value based on data called temporal or sequential data. Unlike standard regression tasks, where values are generally assumed to be independent, time series regression takes into account the temporal relationship between the points.

Many types of artificial neural networks can be used to perform this task. One of the most important is the LSTM (long short-term memory) architecture [3]. LSTM networks include so-called long-term memory cells that can maintain and adjust information over long periods. Another way of approaching this task is to use the convolutional neural networks (CNNs) we mentioned earlier. Originally designed for images, they have also been adapted to time series, under the name of one-dimensional convolutional neural networks (1D CNNs).

Objectives

This thesis aims to address and explore the following questions:

➔ **How to automatically and accurately obtain the position of each electrode of an EEG device in an MRI volume?**

Source localization in EEG involves solving an inverse problem that is sensitive to several parameters, notably the position of EEG electrodes on the scalp. Indeed, the accuracy of electrode coordinate estimates has a direct impact on EEG source localization. Positional errors can lead to inaccuracies in their estimation. This problem is even more significant in studies where multiple sessions, and thus multiple installations of EEG devices, are required. However, in the context of simultaneous EEG-MRI acquisitions, the MRI volume itself is a useful tool for tackling this problem. By using a specific MRI sequence called ultrashort echo-time (UTE), which allows visualization of the electrodes within the MRI volume, sufficient information is available to perform volume segmentation in order to determine the position of the electrodes. The contribution related to this question is presented in Chapter 3.

➔ **Is it possible to predict fMRI neurofeedback scores solely from EEG signals through modeling across multiple subjects?**

In the context of neurofeedback, brain activity is traditionally recorded using an EEG device, which is relatively inexpensive. More recently, simultaneous EEG-fMRI acquisitions have also been employed, especially for their complementarity in terms of temporal and spatial resolutions. However, using MRI comes with certain constraints. It is a very costly modality and requires the participant to remain still in a confined and noisy space. Therefore, being able to predict fMRI neurofeedback scores from EEG signals would reduce the need for the constraining use of MRI while retaining its advantages. This question has already started to be explored with the development of individual models [4]. However, the development of a global model would offer a more accessible and practical application in a clinical setting. A global model is trained on data from different subjects and can therefore be immediately applied to a new participant. Additionally, in the long term, a global model could help investigate the relationship between the two modalities. The contribution related to these questions is presented in Chapter 4.

➔ **Can reducing inter-subject variability improve the performance of these fMRI neurofeedback scores prediction models?**

This question is an extension of the previous one. Among the possibilities for improving the performance of models aiming to predict fMRI neurofeedback scores from EEG signals in a global manner, reducing the variability between data from different subjects is an interesting direction. Indeed, EEG data distributions can exhibit significantly different characteristics from one subject to another, which can affect the performance of models trying to learn a global relationship between inputs and outputs, as well as generalizing to new subjects. The contribution related to this question is presented in Chapter 5.

Organization of the manuscript

Part I: The first part presents the context of this thesis. It consists of two chapters.

Chapter 1: This first chapter presents the distinct but complementary roles of electroencephalography (EEG) and functional magnetic resonance imaging (fMRI) in the study of brain activity. Each modality has its own advantages: EEG offers high temporal resolution, while fMRI offers high spatial resolution. Used simultaneously, these modalities provide a more comprehensive view of brain functions. The chapter also introduces neurofeedback (NF), detailing how the loop-based process allows participants to modulate their brain activity through feedback. The potential of combining EEG and fMRI in neurofeedback is also discussed. This overview provides a better understanding of the context and data involved in the thesis contributions.

Chapter 2: This chapter presents the general principles of artificial neural networks, focusing in particular on the way they update their weights through backpropagation. It then introduces two topics studied in machine learning: image segmentation and time series regression. Image segmentation methods are numerous, ranging from manual segmentation, which requires a considerable amount of time and expertise, to automatic segmentation using deep learning techniques. This chapter provides a detailed overview of the layers that constitute a convolutional neural network (CNN), followed by an in-depth look at the U-Net architecture in its 2D and 3D forms, which are widely recognized in the biomedical field. Finally, the historical evolution of time series regression techniques is reviewed, with a focus on LSTM (long short-term memory) networks and

one-dimensional convolutional neural networks (1D CNNs). This chapter thus provides a better understanding of the concepts and methods used in the thesis contributions.

Part II: The second part details the contributions of this thesis, which address the questions mentioned above. It includes three chapters.

Chapter 3: In this chapter, we present a new method for detecting and labeling EEG electrodes in an MRI volume acquired using a specific type of ultrashort echo-time (UTE) sequence called pointwise encoding time reduction with radial acquisition (PETRA). The method, illustrated below (Figure 3) and within the chapter, involves training a model on a training dataset with associated ground truths, then using this model to obtain segmentation maps, and finally applying a refinement step based on the iterative closest point (ICP) method to improve detections and their labeling. This method, which enables fully automatic segmentation, is easy to implement, requires very few steps and produces excellent results.

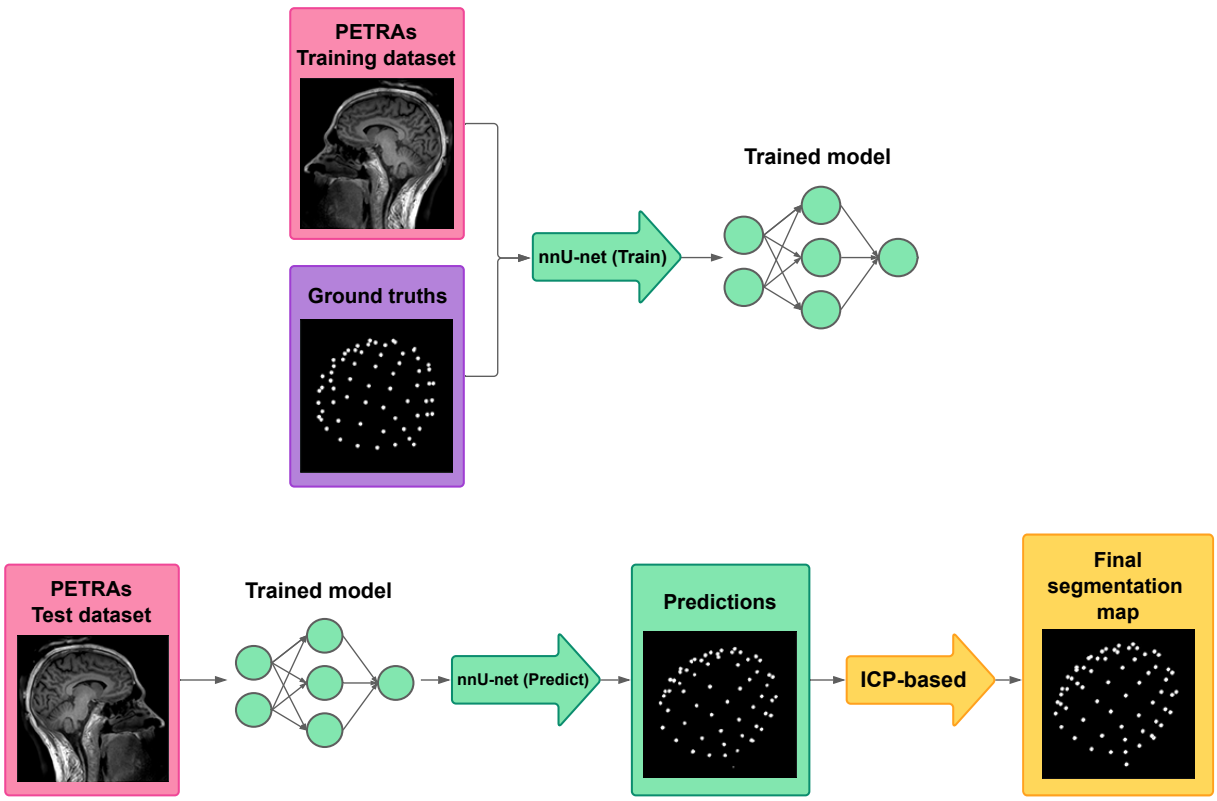


Figure 3 – **Overview of the method presented in Chapter 3.**

Chapter 4: This chapter explores the prediction of fMRI neurofeedback scores from EEG signals with a global model, within a context illustrated below (Figure 4) and in the chapter. In order to investigate different types of models, we present a model architecture search method based on a genetic algorithm. This flexible method can be easily adapted to different types of neural networks. We use this genetic algorithm to study four configurations, including two types of neural networks: LSTM and 1D CNN, as well as two ways of formatting the data: one based on features extracted from the signals and the other based on the raw signals. After a thorough analysis of the results, the 1D CNN approach with extracted features slightly outperforms the other architectures in terms of mean squared error (MSE). However, these results are not satisfactory yet, leading to further questions and opportunities for improvement.

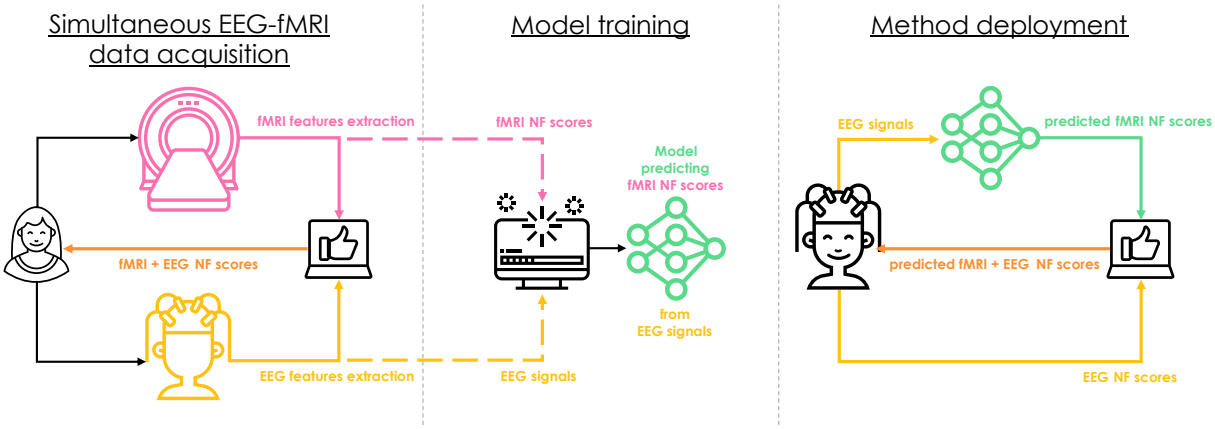


Figure 4 – **Overview of the context of the method presented in Chapter 4.**

Chapter 5: This final chapter represents a first step towards reducing inter-subject variability within the context of global models for predicting fMRI neurofeedback scores from EEG signals. To achieve this, we apply to our EEG signals a method called Euclidean space alignment (EA), which aligns each subject's data with the purpose of reducing individual differences, making the data more comparable for machine learning tasks. We observe the effects of this alignment using t-distributed stochastic neighbor embedding (t-SNE) projection, a non-linear dimensionality reduction technique that groups similar points and separates dissimilar ones to visualize underlying differences in distributions. Using this tool, we observe some reduction in the variability between subjects after alignment, although it remains imperfect. Finally, we evaluate the impact of this alignment

using the best-performing configuration from the previous chapter, by applying exactly the same method as in Chapter 4, which reveals that applying the EA method does not improve the performance of fMRI NF score prediction models. Despite this, the direction remains promising and requires further exploration to progress towards a global model.

Part I

Context

Chapter 1

Bi-modal EEG-fMRI neurofeedback

In this chapter, I introduce the context of my research as well as the type of data used in the contributions. I present the electroencephalography (EEG) and functional magnetic resonance imaging (fMRI) modalities, their simultaneous acquisition, and the use of bi-modal EEG-fMRI in neurofeedback (NF).

1.1 Introduction

This first chapter provides an introduction to the context of my PhD research. While this thesis is in the field of computer science, its applications are medical, specifically in the context of the human brain. Human brain activity can be measured through several imaging modalities, among which electroencephalography (EEG) and functional magnetic resonance imaging (fMRI) are the most popular, although they measure very different aspects of brain function. This chapter aims to introduce these modalities by highlighting their similarities and differences, as these differences represent the main challenge of this PhD. The first contribution of this thesis, covered in Chapter 3, involves simultaneous EEG-fMRI acquisitions, where the goal is to accurately find the coordinates of EEG electrodes within MRI volumes. The second and third contributions, presented in Chapters 4 and 5, are set in the context of bi-modal EEG-fMRI neurofeedback (NF), where the objective is to predict fMRI NF scores from EEG signals only.

The chapter begins by outlining the history, mechanisms, and applications of the two main imaging modalities used to measure brain activity over time: EEG and fMRI. We will then discuss the interest of the simultaneous acquisition of these modalities. Finally, we will introduce the general concept of neurofeedback, concluding with a presentation of bi-modal EEG-fMRI neurofeedback.

1.2 Electroencephalography (EEG)

Electroencephalography (EEG) is a technique that measures variations in the electrical activity generated by neurons using electrodes. Depicted as a graph known as an electroencephalogram, as shown in Figure 1.1, the acquisition of this activity via EEG is a non-invasive process.

Figure 1.1 – **Illustration of an electroencephalogram.** Generalized 3 Hz spike and wave discharges in a child with childhood absence epilepsy. Figure and caption provided by Der Lange from the Wikimedia Commons, published under the following license: Creative Commons Attribution-Share Alike 2.0 Generic license.

The history of electroencephalography dates back to the late 19th century [5]. Its origins are generally credited to the British scientist and physician Richard Caton, who

was apparently the first to discover electrical currents in the brain in 1875, based on his work on animals [6]. However, it was not until 1924 that German neurologist Hans Berger succeeded in amplifying the electrical signals from a young patient's neuronal activity and producing a trace of it. Berger also introduced the term "electroencephalogram" to describe these electrical signals.

The electrical activity recorded by EEG comes mainly from pyramidal neurons. The electrical potentials generated by the neurons are picked up by small sensors, which are most often silver chloride (AgCl) electrodes. There are many different types of headsets, but generally speaking, the electrodes are placed to the surface of the scalp using conductive gel that reduces impedance (i.e., resistance to the passage of an electric current), all held in place by a cap that is often slightly elastic. EEG equipment generally have 32 or 64 electrodes, all having positions and labels defined by an international system. The template for the 64-electrode configuration is shown in Figure 1.2. Finally, since EEG signals are of low amplitude, they need to be amplified thousands of times using an amplification device [5].

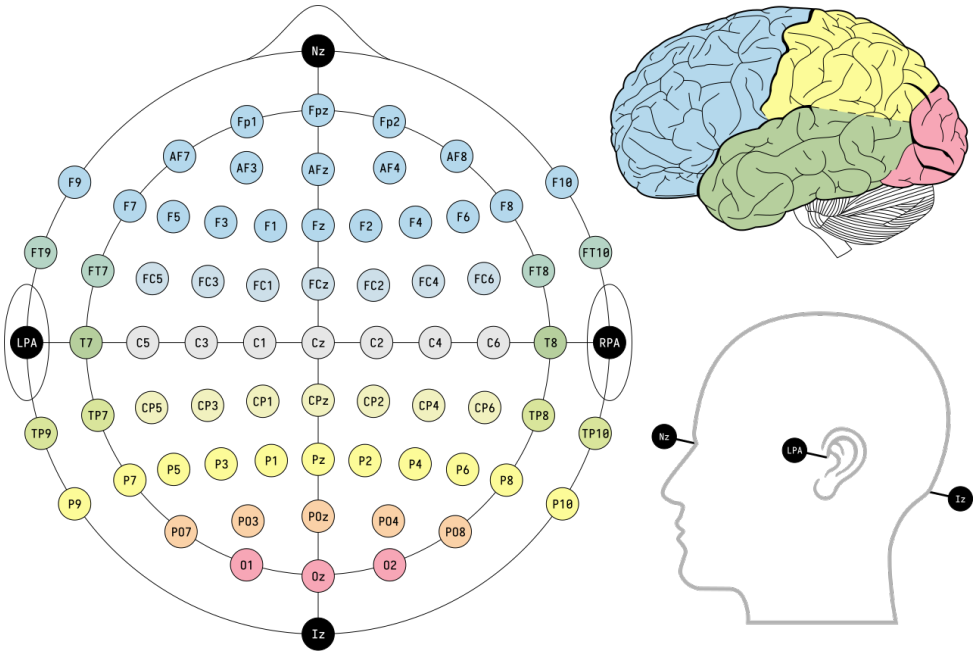


Figure 1.2 – **EEG electrode standard positions for 64 channels.** The electrode sites are colour-coded according to the lobes of the brain which their labels (F, C, P, O, and T) represent. The head indicates the location of the fiducials: the nasion, the (left) pre-auricular point, and the inion. Figure and caption provided by Laurens R. Krol from the Wikimedia Commons, published under the following license: Creative Commons CC0 1.0 Universal Public Domain Dedication.

EEG applications include not only the study of brain function in healthy individuals, but also the diagnosis of certain conditions that alter cerebral electrical activity, such as epilepsy and sleep disorders. An electroencephalogram can be recorded while a patient is awake or asleep, either sitting or lying down. With EEG, the two main areas of interest are rhythmic brain electrical activity (also popularly called brain waves) and event-related potentials [7]. Brain waves are spontaneous signals that can be measured without external stimulation. They can be used to classify sleep patterns and identify abnormal neuronal activity associated with conditions such as epilepsy. These brain rhythms are categorized by their frequency, measured in Hertz (Hz). According to [7], the five basic brain waves are delta ($0.5-4$ Hz), theta ($4-8$ Hz), alpha ($8-12$ Hz), beta ($12-35$ Hz), and gamma (greater than 35 Hz). To give a rather generic idea, delta waves are generally associated with deep and dreamless sleep, theta waves with light sleep, alpha waves with relaxed awake state, beta waves with active awake state, and gamma waves with more intense focus. Unlike brain rhythms, event-related potentials represent the brain's response to an external stimulus (e.g., visual stimulation) or an internal event (e.g., motor imagery). They are valuable for studying both typical and atypical cognitive processes. Since the noise detected by the electrodes is usually much stronger than the low-amplitude EEG signal of interest, it is necessary to record the EEG signal associated with the same activity or repeated stimulation multiple times. By averaging the signals from these repetitions, it is possible to isolate the event-related potentials.

Finally, like all modalities, EEG has its advantages and disadvantages. Its primary strength is its good temporal resolution (see Figure 1.4 from section 1.4). EEG is also one of the least expensive and most mobile modalities, as the equipment can be easily moved. Additionally, it is non-invasive, silent, and does not require a specific body position like MRI, though staying still is usually requested to minimize noise. However, in contrast, EEG is highly prone to noise and has a somewhat low spatial resolution (see Figure 1.4 from section 1.4), notably due to the ill-posed inverse problem of source reconstruction.

1.3 Functional Magnetic Resonance Imaging (fMRI)

Functional magnetic resonance imaging (fMRI) is a neuroimaging technique used to measure brain activity by detecting changes in blood flow. An example of the images produced is shown in Figure 1.3. Unlike EEG, which records electrical activity on the

scalp, fMRI relies on the BOLD (Blood Oxygenation Level Dependent) signal to infer neuronal activity. This modality is also non-invasive.

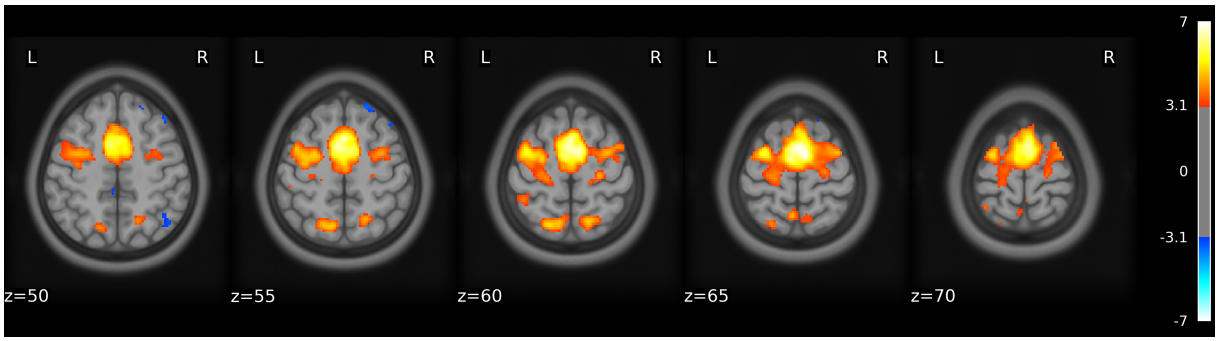


Figure 1.3 – **Illustration of fMRI activation maps.** The brain regions with positive activation values are highlighted in yellow and negative activation values in blue, indicating areas of increased and decreased neural activity, respectively. Figure generated by Quentin Duché upon request, whom I thank.

The history of fMRI dates back to the early 1990s [8]. It builds upon the development of magnetic resonance imaging (MRI) in the 1970s, a medical imaging technique that provides two or three-dimensional views of the body's interior in a non-invasive manner. It is also allegedly based on the work of Angelo Mosso, dating back to 1884 (as presented in [9]), who invented the "human circulation balance", a device that could apparently measure the redistribution of blood during emotional and intellectual activity. In 1990, Seiji Ogawa and his colleagues introduced blood oxygenation level dependent (BOLD) contrast imaging [10], an indirect measure of brain activity through blood oxygenation levels. This discovery was pivotal, leading to the first functional MRI scans in humans in 1991, conducted independently by two groups led by Seiji Ogawa and Kenneth K. Kwong [11, 12].

This technique works by measuring the BOLD signal, which reflects changes in blood oxygenation that occur in response to neuronal activity. As indicated in this comprehensive review [13], the process of obtaining fMRI data involves immensely complex physical, biophysical, and engineering procedures. To put it as simply as possible, when a specific region of the brain becomes more active, it consumes more oxygen, leading to an influx of oxygenated blood into that area. fMRI can detect these changes because oxygenated and deoxygenated hemoglobin have different magnetic properties, which affect the MRI signal. The resulting images are then processed to create detailed maps of brain activity.

These maps are often superimposed on anatomical MRIs to provide an overview of both structure and function [14].

The fMRI modality has many applications, an overview of which is provided in [15]. It is frequently used in cognitive neuroscience research to explore brain functions related to language, sensory perception, memory, and emotion. In terms of clinical applications, fMRI is also used to study and monitor conditions such as multiple sclerosis, epilepsy, psychiatric disorders, and cerebrovascular diseases. Additionally, in neurosurgery, fMRI is used for pre-surgical planning in patients with conditions such as brain tumors or epilepsy. Beyond this summary, there are many more contexts where fMRI is used, such as simultaneous acquisitions with other modalities or brain-computer interface (BCI) protocols, which we will discuss in more detail in the following sections.

One of the main advantages of fMRI is its excellent spatial resolution, which is among the best of all neuroimaging modalities (see Figure 1.4 from section 1.4), even for deep brain regions. It is a non-invasive modality that does not require exposure to ionizing radiation. On the other hand, the technique is very costly and can be restrictive for participants, since it requires the subject to lie in a confined and noisy space. In addition, its temporal resolution is lower than that of EEG (see Figure 1.4 from section 1.4), as we will discuss in the next section.

1.4 Simultaneous EEG-fMRI acquisitions

As mentioned in the previous two sections, the EEG and fMRI modalities have different and complementary strengths [16]. Indeed, EEG measures brain activity directly from the electrical activation of neurons, providing a temporal resolution on a millisecond scale. However, it has limited spatial resolution, mainly due to the fact that the electrodes used to measure activity are placed at different positions on the scalp, making the localization of the source an ill-posed inverse problem. In contrast, fMRI offers detailed images of cerebral structures even deep into the brain, allowing for the localization of brain activity with millimeter precision. On the other hand, given the sampling rate of image acquisition ($\approx$ 1-3 seconds) and the temporal lag between neuronal activation and the BOLD measurement, due to the so-called hemodynamic response, its temporal resolution is lower, typically on the order of seconds. Therefore, since EEG offers high temporal resolution

and fMRI provides high spatial resolution (see Figure 1.4), the idea is that combining the two modalities allows for accurate localization of brain activity in both time and space.

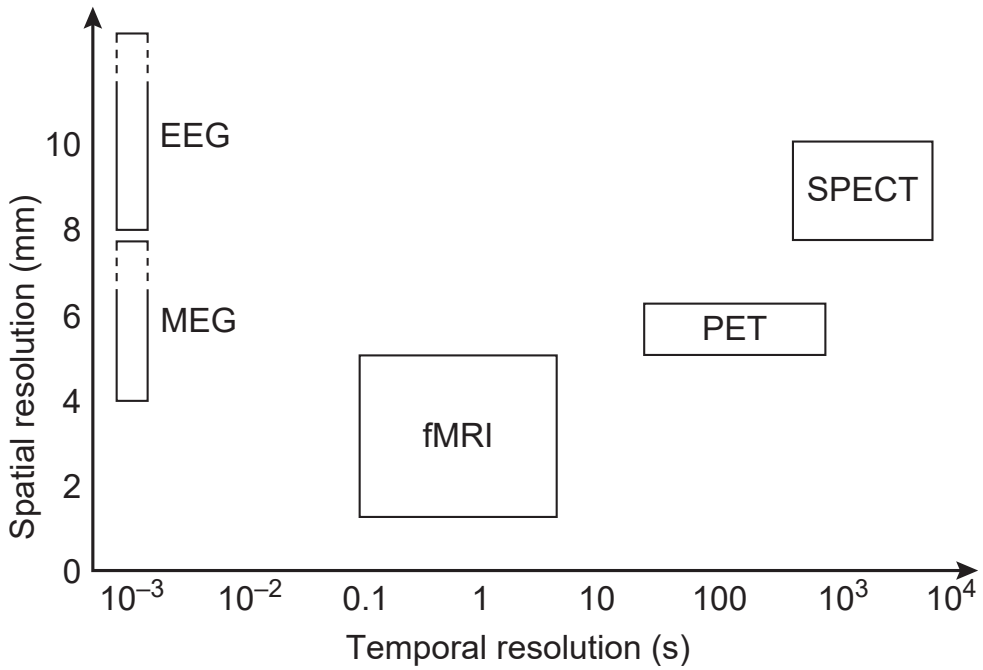


Figure 1.4 – **Approximation of the resolution in time and space of the most commonly employed functional neuroimaging techniques.** Figure and title provided by Melanie Boly et al. in the chapter "Functional neuroimaging techniques" [16]. For information, the other modalities mentioned and not presented in this chapter are magnetoencephalography (MEG), a modality that records magnetic fields generated by brain electrical currents using magnetometers, positron emission tomography (PET), an imaging technique that uses radioactive tracers to visualize changes in metabolic processes, and single-photon emission computed tomography (SPECT), a nuclear medicine tomographic imaging technique that detects gamma rays from injected radioactive substances.

In this article [17], Helmut Laufs recounts the history of EEG-fMRI integration. In summary, the idea of EEG-fMRI integration was driven by clinical need. Its development was guided by the desire of epileptologists who aimed to localize the electrical sources of epileptic discharges in cases where the EEG modality alone was insufficient. John Ives and his colleagues were the first to record EEG inside an MRI scanner in 1993 [18]. A few years later, the same team demonstrated the first epileptic discharges correlating with BOLD signal changes [19]. Significant engineering advancements [20] and careful consideration of patient safety since the very early years [21] enabled the development of this technique.

Prior to looking at the various integration methodologies, it is important to address a crucial step: pre-processing. This review [22] provides a detailed presentation: for EEG,

pre-processing mainly consists in correcting the gradient artifact and pulse artifact [23], as well as other MR-related motion artifacts linked to the environment including the Helium cooling pump vibrations [24] and the patient ventilation system [25]. Additionally, EEG-specific artifacts unrelated to the MR environment, such as face muscle activity, eye movements, blinks, and bad channels, must also be corrected [26]. For fMRI, pre-processing mainly involves correcting image artifacts caused by the presence of EEG equipment inside the MR scanner [27]. In addition, the fMRI signal can also be contaminated by physiological noise of non-neuronal origin, typically arterial pulsation and respiration [28, 29].

Simultaneous EEG-fMRI acquisitions are used in a wide range of applications. One way to categorize them is to make a distinction between fMRI-informed EEG, which generally corresponds to methods that use fMRI spatial information to improve the localization of EEG signal sources, and EEG-informed fMRI, which generally corresponds to methods that use EEG data to guide or model fMRI information, such as the BOLD signal. Since the contribution presented in Chapter 4 falls into the latter category, we will focus on it. The previously mentioned review [22] focused on EEG-informed fMRI methods, classifying them into univariate and multivariate categories. In univariate methods, BOLD changes are predicted using temporal or spectral features extracted from one or a few time courses, which must of course be representative of the phenomenon under study. On the other hand, multivariate methods use multiple EEG channels to calculate spatial correlation features or functional connectivity measures.

1.5 Bi-modal EEG-fMRI neurofeedback (NF)

Neurofeedback is a technique that provides participants with a score in real-time, aiming to indicate whether their brain activity corresponds to the desired activity, with the goal of regulating or rehabilitating neuronal activity [30]. There are numerous protocols for a variety of applications, which is why the description of this technique can be quite general. As shown in Figure 1.5, the neurofeedback process can be represented as a loop consisting of three main steps: acquisition, data processing, and feedback display. Firstly, acquisition involves measuring brain activity. To do so, the EEG modality has traditionally been used, but many protocols now also utilize fMRI [31] or fNIRS [32]. Secondly, the data processing step is used to compute a score representing the brain activity under study. This generally involves pre-processing, followed by feature extraction (depending on the

experimental paradigm), and finally a classification step to obtain a score that reflects the brain activity of interest. Thirdly, this score is presented to the participant in the form of a feedback, which is typically visual but can also be auditory or haptic. The participant thus learns to modulate their brain activity through repeated iterations of the loop.

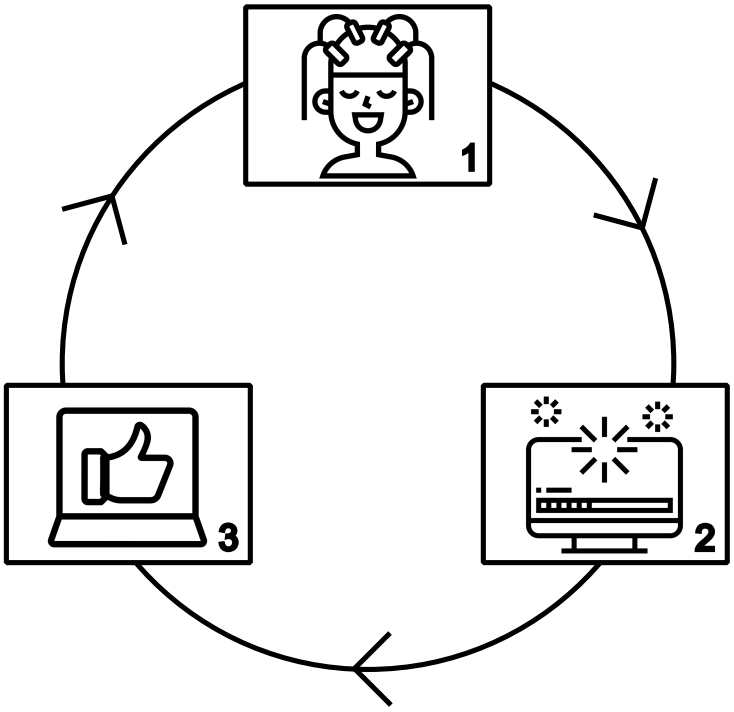


Figure 1.5 – **The neurofeedback loop.** The participant learns to modulate their brain activity in relation to the feedback. (1) Brain activity is measured with on or multiple modalities. (2) Pre-processing, feature extraction and classification are applied on the data. (3) A brain activity score is computed and displayed as the feedback.

It turns out that this loop is also reminiscent of protocols known as brain-computer interfaces (BCI). Indeed, as Lorraine Perronnet discussed in her thesis [33], the terms BCI and neurofeedback, although referring to a very similar methodology, have been employed in parallel, BCI more commonly in the engineering field and neurofeedback in the medical field. It is generally considered that the main difference between the two terms lies in the objective, where BCI protocols seek to achieve external control, such as commanding a computer or a robotic limb, whereas neurofeedback aims to generate a so-called internal control by regulating brain activity, often for therapeutic purposes. Depending on definitions, neurofeedback can also be considered a sub-field of BCI, under the emerging term "restorative BCI".

In terms of applications, also covered in great detail in [33], we can consider two main categories: applications for performance optimization and applications for therapeutic purposes. In the first category, goals include enhancements linked to attention, memory, mental rotation, sports performance, creativity, meditation, and more [34, 35]. In the second category, we can encounter conditions such as attention deficit hyperactivity disorder (ADHD), epilepsy, depression, anxiety, post-traumatic stress disorder (PTSD), addiction, or recovery after stroke [36]. The most common clinical application is ADHD, which has been studied in the most depth [37]. However, it should be noted that the majority of other applications still remain at the research stage and are not yet widely used in clinical practice. In cases like depression or PTSD, further research has been carried out, leading to the use of the fMRI modality to target the amygdala in the context of depression [38], or to the development of new methodologies such as decoded neurofeedback in the context of PTSD, which provides feedback without the participant's conscious awareness of the training target [39].

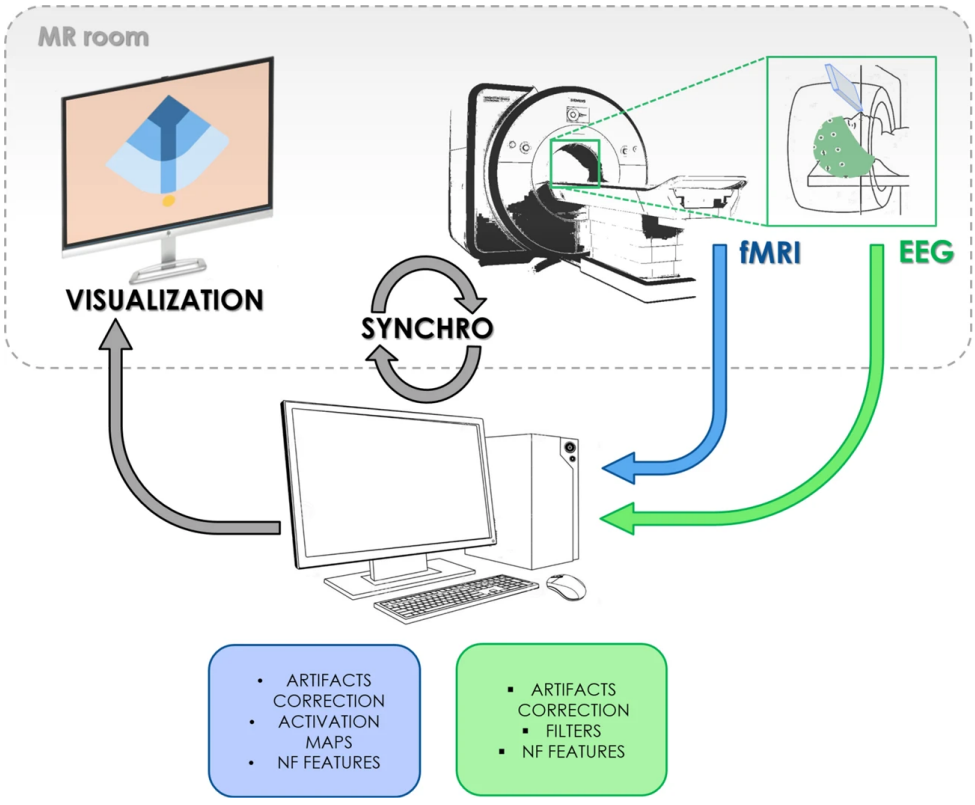


Figure 1.6 – **Schematic visualisation of the bi-modal EEG-fMRI NF platform.** Figure provided by Giulia Lioi et al. in the article "Simultaneous EEG-fMRI during a neurofeedback task, a brain imaging dataset for multimodal data integration" [40].

For several of these applications, the traditional EEG modality is not always sufficient to meet expectations. Therefore, for the same reasons of complementary temporal and spatial resolutions discussed in the previous section, the field of neurofeedback can also benefit from bi-modal EEG-fMRI. The first proof-of-concept of bi-modal EEG-fMRI neurofeedback was demonstrated in 2014 by Vadim Zotev and his colleagues [41], who hypothesized that such an approach could be more effective than unimodal approaches. Then, in 2017, Marsel Mano and his colleagues set up a bi-modal EEG-fMRI neurofeedback platform in Rennes, detailing its functioning in detail in [42]. A schematic illustration of the platform is shown in Figure 1.6. Using this technology, Lorraine Perronnet and her colleagues were able to show that bi-modal EEG-fMRI neurofeedback could provide more specific and engaging training in a motor imagery context [43]. It is within this context, and with these data, that the contributions presented in Chapters 4 and 5 are developed. Therefore, further details will be provided in these chapters.

1.6 Conclusion

In this chapter, I have presented the distinct yet complementary roles of electroencephalography (EEG) and functional magnetic resonance imaging (fMRI) in the study of brain activity. Each modality has its own strengths and limitations: EEG provides high temporal resolution, while fMRI offers high spatial resolution. When used together, these modalities give a more comprehensive view of brain function. Then, I introduced neurofeedback (NF), detailing how the loop process allows participants to modulate their brain activity through real-time feedback. Finally, we looked at the potential of combining EEG and fMRI in neurofeedback to create a more effective and engaging training process. This overview provides a better understanding of the data used in the contributions and the context in which they were produced. In the next chapter, we will get closer to the core of my contributions by exploring the computer science aspect of the context, presenting the machine learning tasks and techniques applied in these works.

Chapter highlights

- Electroencephalography (EEG) is a technique that measures variations in the electrical activity generated by neurons using electrodes.
- Functional magnetic resonance imaging (fMRI) is a neuroimaging technique used to measure brain activity by detecting changes in blood flow.
- Since EEG offers high temporal resolution and fMRI provides high spatial resolution, combining the two modalities allows for accurate localization of brain activity in both time and space.
- Neurofeedback (NF) is a technique that provides participants with a score in real-time, indicating whether their brain activity corresponds to the desired activity, with the aim of regulating or rehabilitating neuronal activity.
- Using bi-modal EEG-fMRI neurofeedback can provide more specific and engaging training.

Chapter 2

Machine learning for image segmentation and time series regression

In this context chapter, I present the machine learning tasks and techniques used in the contributions. For image segmentation, I detail the U-Net architecture, which belongs to the convolutional neural network (CNN) family. For time series regression, I present two types of neural networks: long short-term memory (LSTM) networks and one-dimensional convolutional neural networks (1D CNNs).

2.1 Introduction

Over the last few years, the general public has been increasingly hearing about a concept: "artificial intelligence". This term is commonly associated with large-scale text, image, and video generation models, which have recently become available for personal use. At first, artificial intelligence appears to be a revolutionary novelty, sparking both excitement and concern. There is often confusion between the terms artificial intelligence, machine learning, and deep learning. Therefore, I wanted to start by defining these terms clearly.

Artificial intelligence (AI) is a very broad concept, which could be qualified as philosophical [44], defined by the desire to create machines that mimic human behavior and intelligence. From this definition, it can be considered that this idea has always been a part of human history [45]. Machine learning, on the other hand, is a more recent and much more practical field, considered a subset of the AI concept. It encompasses all methods that enable algorithms to learn from data and generalize to unseen data, learning

through experience to perform specific tasks. The term was popularized in 1959 by Arthur Samuel [46], making it rather recent history. Finally, deep learning is a subset of machine learning. The term "deep" refers to the use of multiple layers in neural networks and was introduced in 1986 by Rina Dechter [47]. Although the boundary between machine learning and deep learning can be somewhat blurred, the general rule is that artificial neural networks with many layers are typically considered deep learning. Since the methods presented as contributions to this thesis are based on different types of neural networks (not necessarily deep), it is essential to outline the general principles of these networks here.

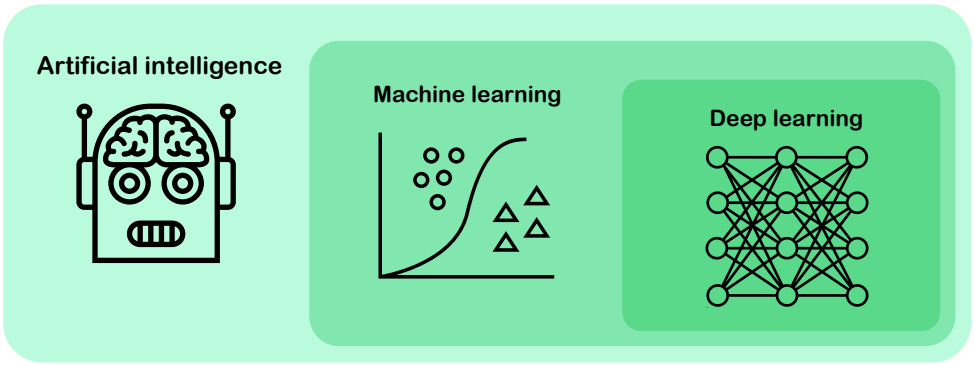


Figure 2.1 – **Links between the concepts of AI, machine learning, and deep learning.** Artificial intelligence (AI) is a concept that encompasses machine learning methods, known for learning patterns from data, which in turn include the more complex deep learning methods, such as artificial neural networks.

An artificial neural network consists of computational units called **neurons**. The initial idea was to draw inspiration from the functioning of neurons in the human brain, hence the term "neuron" for a computing unit, but the concept has since been extended with improvements that go beyond its biological inspiration. They receive inputs, process them via an **activation function**, and pass the output to the next **layer**. The activation function usually introduces non-linearity into the model, enabling it to learn complex patterns. The most common activation functions are sigmoid, rectified linear unit (ReLU), and hyperbolic tangent function (tanh), each with different properties that affect network performance and convergence [48]. Neurons are connected by **edges**, which link neurons from one layer to another, carrying signals through the network. Each edge is associated with a **weight**, a numerical value that adjusts the strength of the transmitted signal. Weights are crucial because they embody the network's ability to learn and make accurate predictions.

The general machine learning framework can be divided into two stages: the learning phase, where these weights are adjusted using the backpropagation technique described in the following paragraph, and the testing phase, where the trained model with fixed weights is used to make predictions. Typically, the available data is divided into multiple datasets: the training dataset, used for model training, and the test dataset, used to assess the model's performance and ability to generalize to unseen data. In some cases, a validation dataset is also created to evaluate models during the training phase, helping to select the best one for testing.

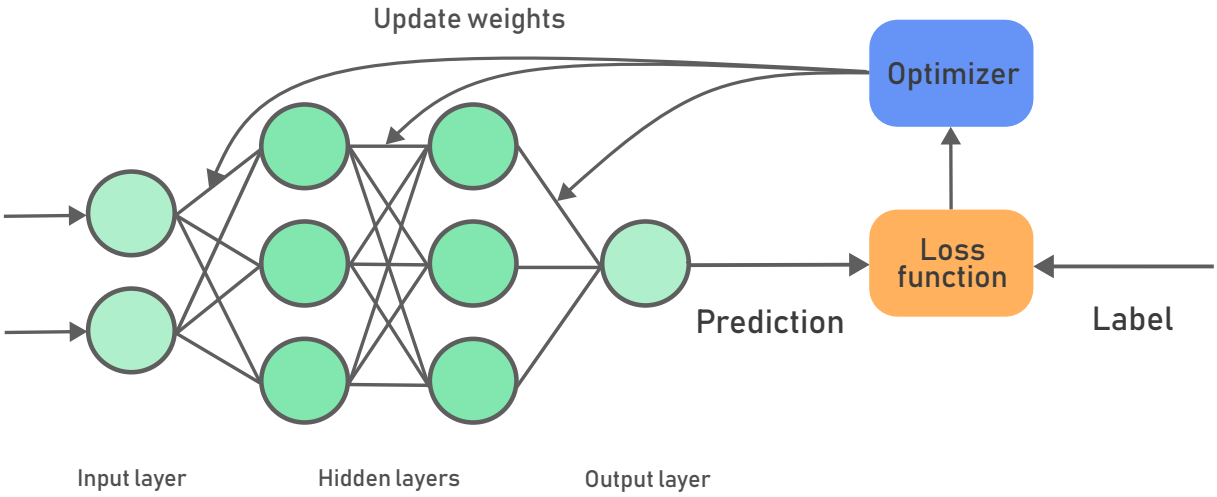


Figure 2.2 – **Principle of backpropagation in a neural network.** The neurons are represented by green circles, and the edges containing the weights are depicted by gray lines. The forward pass is used to obtain the prediction, which is compared with the label using the loss function. During the backward pass, the optimizer updates the weights. This iterative process allows the neural network to adjust its weights over time.

So, when training a neural network, weight values are learned through backpropagation, a method introduced by Frank Rosenblatt in 1962 [49], implemented by Seppo Linnainmaa in 1970 [50] and applied to neural networks by Paul Werbos in 1982 [51]. Backpropagation is an iterative algorithm consisting of two main phases: forward pass and backward pass. During the forward pass, input data are passed through the network, layer by layer, to produce an output called **prediction**. This output is compared with its corresponding **label** using a **loss function** to quantify the error between the predicted output and the true value. Next comes the backward pass, where the error is propagated backward through the network. To do this, gradients of the loss function for each network

weight are calculated, indicating the extent to which each weight contributes to the overall error. An **optimizer**, such as Stochastic Gradient Descent (SGD) or Adam, uses these gradients to update the weights, typically by subtracting a fraction of the gradient from the current weight. This fraction is determined by a value called the **learning rate**. The forward and backward pass process is repeated iteratively, allowing the network to learn by continuously adjusting its weights to minimize the loss function, thus improving its performance over time. This process is illustrated in Figure 2.2.

Another important aspect of machine learning is the selection of hyperparameter values. These are the values chosen by the human user before training begins. They are distinct from weights, which are sometimes called network parameters. Hyperparameters can usually be placed into two categories: training hyperparameters and architecture hyperparameters. Training hyperparameters, such as the number of **epochs**, **batch size**, and the optimizer along with its learning rate, control the learning process. Some regularization techniques can be applied during training, with their values also considered hyperparameters, to prevent **overfitting**, which is a phenomenon where a model learns the training data too well, capturing noise and details that do not generalize to new, unseen data. Techniques like **dropout** randomly omit neurons during training, with the rate (a hyperparameter to fix) corresponding to the probability of a neuron being dropped. Another technique is **kernel regularization**, which applies L1 or L2 regularization penalties to the weight matrix of a layer during training to discourage large weight values. Next, architecture hyperparameters, such as the number of layers, the number of neurons in each layer, and those specific to particular types of neural networks, define the model before training. Even today, most of them are chosen arbitrarily, sometimes thanks to prior expertise, but often without any concrete justification for the choices made other than empirical exploration. In addition, in cases where users do not have sufficient expertise, it becomes very difficult to achieve a high-performance network architecture as well as to justify the choices made by manually designing a network. This is why the field of hyperparameter optimization is on the rise. As indicated in this review [52], there is a wide range of methods for selecting model hyperparameters, from arbitrary choices to fully automated approaches. We will explore this area in greater detail on several occasions in the following sections of this thesis.

The fields of machine learning and deep learning contain a vast amount of vocabulary used to describe data, model architectures, and training processes. To help readers easily

return to this section to find the information they need, I have created a vocabulary table (Table 2.1) that defines the terms used here and in the following sections.

Neuron	A computing unit that can receive, process, and send signals to other connected neurons.
Layer	A group of neurons that operate together at the same depth in the network.
Edge	A connection between neurons that is associated with a weight.
Weight	A numerical value, also called a network parameter, that adjusts the strength of the transmitted signal.
Prediction	The output of the neural network model.
Label	The real expected value that the model should predict, to be compared with the prediction.
Activation function	A function that processes signals inside neurons, introducing non-linearity into the model.
Loss function	A function that compares predictions to labels.
Optimizer	An algorithm that updates the weights of the model to minimize the loss function.
Learning rate	A parameter that influences the speed and stability of weight updates.
Batch size	The number of input samples processed before the model's weights are updated. The entire training dataset is divided into batches, and weights are updated after each batch is processed.
Epoch	One complete pass through the entire training dataset during the learning process. Each epoch consists of multiple iterations based on the batch size.

Overfitting	A phenomenon occurring when a model learns the training data too well, capturing noise and details that do not generalize to new, unseen data, leading to high performance on the training set but poor performance on the test set.
Dropout	A regularization technique where randomly selected neurons are "dropped out", or omitted, during training to prevent overfitting. The rate correspond to the probability of a neuron being dropped out.
Kernel regularizer	A regularization technique, typically using methods like L1 or L2, applied to the weights matrix of a layer during training to penalize large or complex weight values, helping to prevent overfitting.

Table 2.1 – **Machine learning vocabulary.**

In the remainder of this chapter, we will present two broad categories of tasks performed by the neural networks used in our contributions: image segmentation, which is used in Chapter 3, and time series regression, which is discussed in Chapters 4 and 5.

2.2 Image segmentation

2.2.1 General overview

In computer vision, segmentation is a technique used to classify pixels into different regions corresponding to various objects in an image. Each pixel is assigned to a class, enabling the image to be divided into distinct areas of interest. The result of this process is called a segmentation map. There are many applications for segmentation, summarized non-exhaustively in this review [53], including video surveillance, autonomous vehicles, satellite imaging, and agricultural monitoring. One of the most significant fields of application is medicine, with numerous uses across radiography [54], computed tomography (CT) [55], ultrasound [56], and magnetic resonance imaging (MRI) [57], helping with tasks

such as surgical planning [58], tumor localization [59], or the study of brain anatomical structures [60].

Segmentation techniques can be categorized as manual, semi-manual, and automatic. As explained in [61], manual segmentation involves precisely drawing the boundary between the area of interest and the rest of the image to accurately annotate each pixel. This process demands considerable time and expertise. Typically, only a limited amount of data is created in this manner, and there is often variability between different raters. Semi-manual segmentations, on the other hand, require some user interaction before automated computation, such as providing an initial seed point for seeded region growing (SRG) [62] or selecting an approximate initial region of interest (ROI) for level set-based active contour model [63]. Finally, automatic segmentation is designed to require no user interaction. However, they usually rely on a supervised learning approach that requires a training dataset with labels representing the desired outcome, known as the ground truth, for comparison during learning. Therefore, it is generally necessary to obtain manual segmentation that is as reliable as possible to serve as the ground truth. Despite this constraint, deep learning-based automatic segmentation methods currently represent the state of the art [64, 65].

One of the most widely used neural networks for segmentation tasks is the U-Net [2], which belongs to the broader category of convolutional neural networks (CNNs). In fact, the contribution presented in Chapter 3 uses an automatic method based on a U-Net model. So, we will first review the main principles of CNNs, followed by a detailed examination of the U-Net architecture.

2.2.2 Convolutional Neural Network (CNN)

CNNs are a class of neural networks mainly used in computer vision due to their ability to find patterns and spatial hierarchies of features within images. Although many researchers have contributed to the evolution of CNNs, their contemporary implementation can be credited to Yann Le Cun [1], who drew inspiration from the neocognitron framework proposed by Kunihiko Fukushima [66].

Like all neural networks, CNNs have an input layer, one or more hidden layers, and an output layer. Their distinctive feature is that these hidden layers comprise **convolutional**

layers, generally followed by **pooling layers**, with one or more **dense layers** at the very end.

The core principle of CNNs resides in those convolutional layers, which conduct convolution operations to learn spatial hierarchies of features from input images. The first layers of the network typically detect basic features such as horizontal and vertical edges, while subsequent layers extract increasingly complex features such as objects or faces. Typically, the convolution operation involves sliding a **filter**, which is a small matrix of weights (learned during training) with its size referred to as the **kernel size**, over the input image and computing the dot product between the filter and the overlapping image region. This produces a **feature map** that highlights the presence of specific features at different locations in the input image and contributes to the input of the next layer.

Following a convolutional layer, there is usually a pooling layer used to reduce the spatial dimensions (width and height) of the feature maps. This helps in lowering the computational cost of the network and mitigating overfitting. Pooling achieves this by down-sampling or summarizing the information from the feature maps. A pooling window is a small fixed-size region that slides across the feature map. The two most popular approaches are max pooling, which only keeps the maximum value from each region (also called patch) of the feature map covered by the pooling window, and average pooling, which takes the average value.

Finally, the dense layer, also known as the fully connected (FC) layer, integrates these features to make the final predictions. Before feeding the data into a dense layer, the output of the final convolutional or pooling layer, which is typically a multi-dimensional tensor, is flattened into a one-dimensional vector. The final dense layer often uses an activation function like softmax, a function that converts a vector of values into a probability distribution, where each value represents the probability of that class relative to the others (for multi-class classification) or sigmoid, a function that maps values to a range between 0 and 1, interpreting them as probabilities or binary decisions (for binary classification). An example of a CNN architecture summarizing the role of these layers is presented in Figure 2.3.

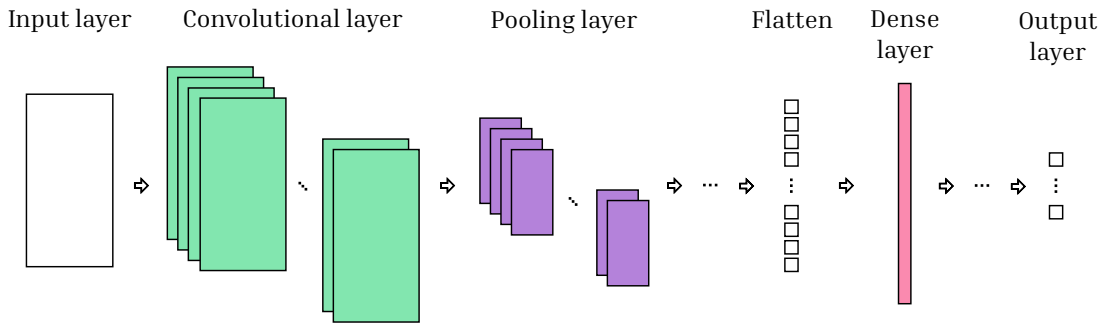


Figure 2.3 – **A typical CNN architecture.** Convolutional layers are typically followed by pooling layers, forming blocks that can be repeated to extract increasingly complex features. The output of these layers is then flattened and passed through one or more dense layers before reaching the final output. This output may represent a continuous value in regression tasks or multiple class probabilities in classification tasks. The color coding applied to the layers, which will be reused later, does not carry any specific meaning in this context.

Similarly, due to the extensive vocabulary specific to the CNN architecture, a summary table (Table 2.2) is provided below.

Convolutional layer	A layer that detects local patterns or features in the input image by applying convolution operations.
Filter	A small matrix of weights, also known as kernel, that slides over the image, computing a dot product between the filter weights and the image pixels covered by the filter at each position.
Kernel size	The size of the sliding filter, typically much smaller than the input image.
Feature map	The output of the convolution operation, also called an activation map. It represents the presence of specific features detected by the filter across different spatial locations of the input image.
Pooling layer	A layer that reduces the spatial dimensions of the feature maps.

Dense layer	A layer, also known as a fully connected layer, where each neuron is connected to every neuron in the preceding layer.
Spatial dropout	A variation of dropout applied to convolutional layers, where entire feature maps are dropped out during training to prevent overfitting and encourage the model to learn more robust features.

Table 2.2 – **Convolutional neural network (CNN) vocabulary.**

2.2.3 U-Net architecture

The U-Net architecture was proposed by Olaf Ronneberger, Philipp Fischer, and Thomas Brox in 2015 [2], drawing inspiration from the fully convolutional network (FCN) [67], which relies solely on convolutional, pooling, and upsampling layers without the use of dense layers. U-Net improves upon this type of CNN, achieving good segmentation performances with a very low number of training images (approximately 30 per application, according to the authors). This network was initially designed for biomedical imaging, a field where the number of available samples is often limited due to cost and time constraints. However, many variants have since been proposed [68], and the U-Net architecture has been adopted in a wide range of domains due to its simplicity and high performance [69, 70].

The architecture is shown in Figure 2.4, which I borrowed from the original article [2]. We can see that it consists of two paths: the contracting path, also called the encoder, and the expanding path, also called the decoder. The expanding path is more or less symmetric to the contracting path, yielding a U-shaped architecture, which gave its name to the U-Net.

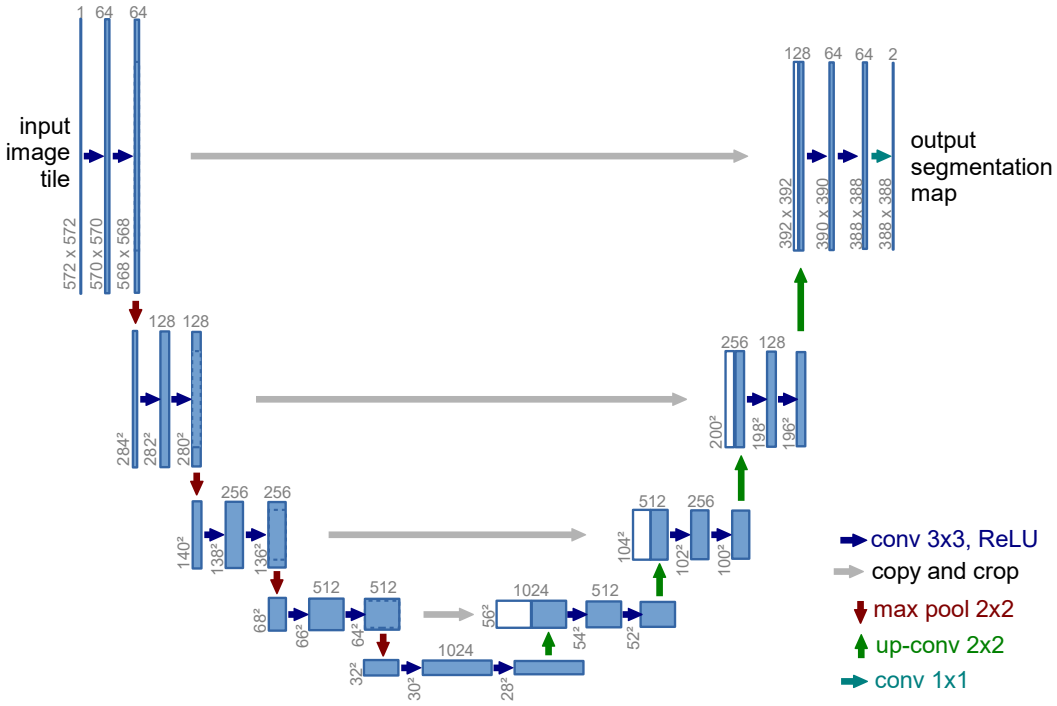


Figure 2.4 – **A typical U-Net architecture.** U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. Figure and caption provided by Olaf Ronneberger et al. in the article "U-Net: Convolutional Networks for Biomedical Image Segmentation" [2].

Firstly, the contracting path follows the typical structure of a convolutional network. It consists of several blocks, each containing two convolutional layers (each having 64 filters) in a row with a 3×3 kernel size, followed by a rectified linear unit (ReLU) activation function. After each block of convolutional layers, a 2×2 max-pooling layer is applied to downsample the feature maps, reducing the spatial dimensions by a factor of 2. As we move deeper into the contracting path, the number of feature channels increases, doubling after each downsampling step. This allows the network to learn increasingly abstract and larger context features. We can also say that on this path, the model learns the most important contextual information of the image (the "what"), but loses the notion of location within the image (the "where").

Secondly, the expanding path is responsible for upsampling and reconstructing the image to its original resolution while maintaining precise localization. Each block in the expanding path begins with a resizing of the feature map followed by a 2x2 convolution ("up-convolution") that increases the spatial dimensions of the feature maps. This step

effectively reverses the downsampling done in the contracting path. Then, the upsampled feature maps are concatenated (and cropped due to the loss of pixels at the edges during convolution) with the corresponding feature maps from the contracting path. This provides fine-grained information from earlier layers, which is crucial for precise localization. After concatenation, two 3×3 convolutional layers followed by ReLU are applied to refine the upsampled features. As we move through the expanding path, the number of feature channels decreases, typically halving after each upsampling step, reversing the process of the contracting path. This path thus combines the "what" obtained previously with the "where", allowing precise segmentation in terms of both context and location.

A year later, Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, along with two of the original U-Net authors, proposed the 3D U-Net, which is applicable for 3D image segmentation [71]. The principle remains very similar, with the main difference being that operations such as convolutions and max poolings are replaced by their 3D counterparts. Additionally, 3D U-Net uses only three downsampling operations instead of four and adds a batch normalization layer after each convolutional layer. Batch normalization is a technique used to improve the training of deep neural networks by normalizing the inputs of a layer, thereby accelerating convergence and stabilizing the learning process. The 3D U-Net architecture, taken from the referenced article, is shown in Figure 2.5.

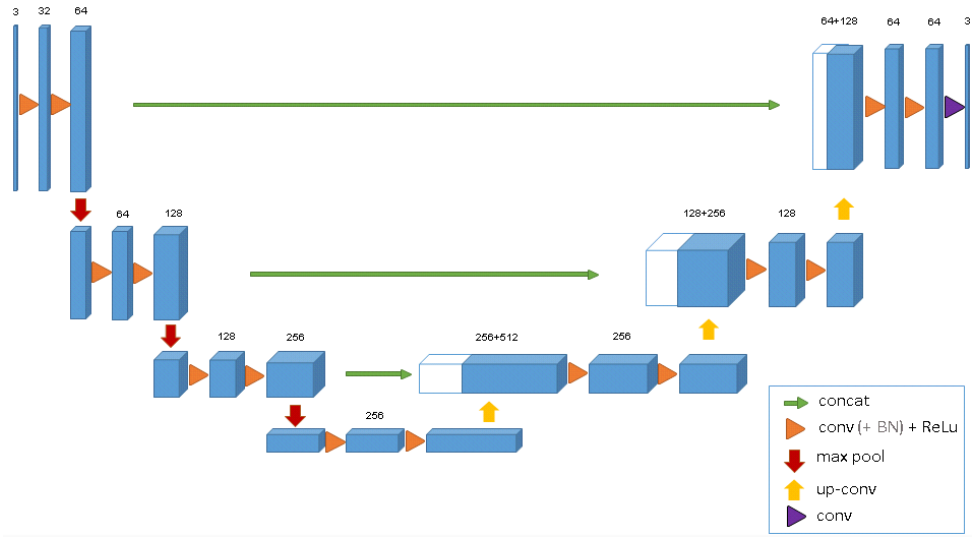


Figure 2.5 – **A typical 3D U-Net architecture.** Blue boxes represent feature maps. The number of channels is denoted above each feature map. Figure and caption provided by Özgün Çiçek et al. in the article "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation" [71].

In many applications, selecting an appropriate set of hyperparameters is crucial for achieving the best performance. Fortunately, for U-Net, a framework called nnU-Net ("no new net", not yet another architecture, but a framework) was proposed back in 2018 [72] and subsequently developed [73, 74]. This framework was used to optimize the hyperparameters of the 3D U-Net model used in the contribution presented in Chapter 3, which is why I will introduce it here. It is a tool that can automate the choice of hyperparameters for training a model on any given dataset across different segmentation tasks. This tool is especially useful because segmentation tasks, especially in the biomedical field, have seen numerous variations of neural network architectures proposed over the past few years, but the authors of [73] demonstrate that slight improvements in network design rarely improve performance significantly, whereas the choice of hyperparameters is crucial. In fact, this framework, using a basic U-Net architecture, outperformed most specialized deep learning pipelines in 19 international competitions and 49 segmentation tasks, demonstrating its efficiency and adaptability.

The framework design can be explained using the illustration presented in Figure 2.6. Starting with the training data, the framework extracts "data fingerprints", which are key properties of the dataset such as image size, image spacing (i.e., the physical size of the voxels), intensity distribution, image modality (from metadata), and the number of classes. This information is then combined with heuristic rules, which the authors refer to as domain knowledge. For example, domain knowledge indicates that in some datasets, voxel spacing is heterogeneous, so the framework must check and resample all images to the same target spacing if needed. The combination of these two pieces of information results in the "rule-based parameters", which include the necessary hyperparameters to adapt the framework to a particular new dataset, such as batch size, patch size, image normalization and resampling, and even architecture configuration. These are associated with the "fixed parameters", the default values of design choices made independently of the data. These choices, made by the authors, pertain to the base architecture, the training and testing schedule, the optimizer with its learning rate, the loss function, and data augmentation. These parameters represent assumptions about the best network topology based on ten biomedical image datasets. They help derive the "pipeline fingerprint", which is the design of the segmentation algorithm based on domain knowledge and specifically adapted to the provided dataset. The user can then choose between different network training options, such as 2D, 3D, or 3D cascade, with or without cross-validation. It is possible to train all possible types, each for five cross-validation folds, and then use the

technique of ensembling to achieve the best possible performance, though this comes at the cost of a very long training time. Finally, the "empirical parameters" allow for post-processing if needed and ensembling if multiple trainings were conducted, to choose or combine the best ensembles. Once trained, the model can be used to make predictions from the test dataset. In summary, this framework allows for efficient adaptation to any dataset by primarily focusing on the choice of hyperparameters.

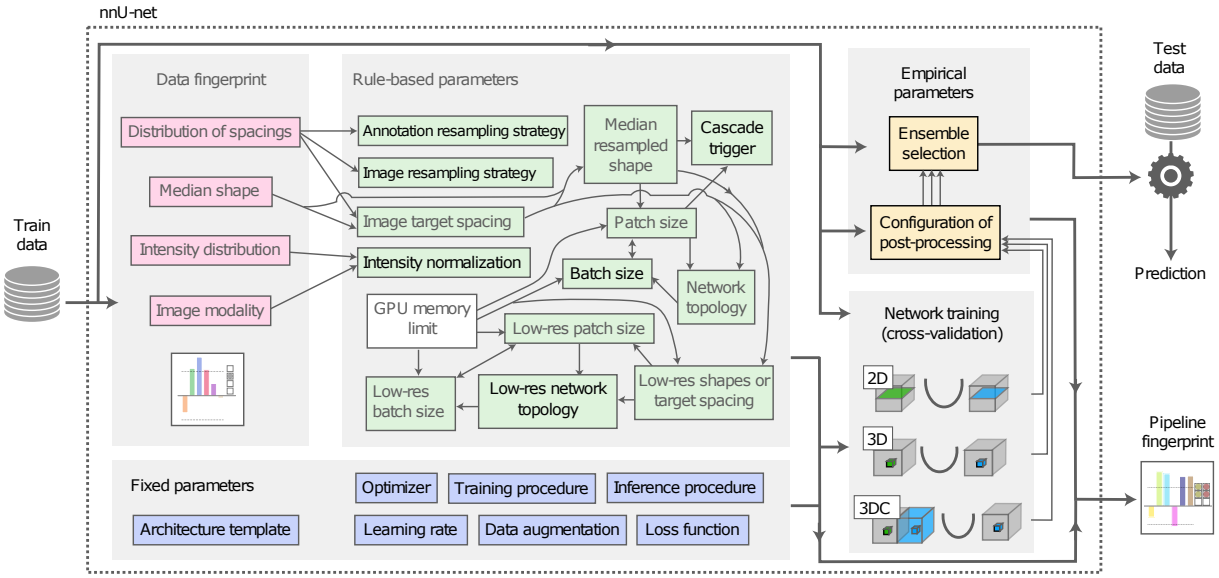


Figure 2.6 – **The nnU-Net framework.** Proposed automated method configuration for deep learning-based biomedical image segmentation. Given a new segmentation task, dataset properties are extracted in the form of a 'dataset fingerprint' (pink). A set of heuristic rules models parameter interdependencies (shown as thin arrows) and operates on this fingerprint to infer the data-dependent 'rule-based parameters' (green) of the pipeline. These are complemented by 'fixed parameters' (blue), which are predefined and do not require adaptation. Up to three configurations are trained in a five-fold cross-validation. Finally, nnU-Net automatically performs empirical selection of the optimal ensemble of these models and determines whether post-processing is required ('empirical parameters', yellow). Figure and caption provided by Fabian Isensee et al. in the article "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation" [74].

2.3 Time series regression

2.3.1 General overview

Time series regression is a type of machine learning task where the objective is to predict a continuous value based on sequential data points. Unlike standard regression tasks, where data points are typically assumed to be independent, time series regression

considers the temporal relationship between the data points. This approach is widely used in fields that require forecasting future values, such as finance, sales, weather prediction, and environmental monitoring, as noted in [75]. Additionally, it has applications in healthcare, such as predicting disease progression [76], patient vitals [77], and the effects of medications over time [78].

Several types of neural networks have been developed for time series regression tasks. One of the earliest approaches involved using feedforward neural networks (FNNs) by flattening the data and using lagged values as features. However, this method did not handle sequential data effectively because FNNs lack memory of previous inputs. To address this limitation, recurrent neural networks (RNNs) were introduced in the 1980s. While the concept of RNNs has a long history, it is often attributed to John Hopfield's work [79]. RNNs introduced loops that allowed past information to persist, making them well-suited for sequential data. However, they suffered from the vanishing gradient problem [80], which made it difficult to learn long-term dependencies. To overcome this issue, Hochreiter and Schmidhuber [3] presented the long short-term memory (LSTM) architecture in 1997 as an improved version of traditional RNNs. LSTMs include a memory cell that can maintain information over long periods and gates that control the flow of information. Additionally, gated recurrent units (GRUs) were introduced in 2014 by Kyunghyun Cho et al. [81] as a simplified version of LSTMs. GRUs are comparable to LSTMs in terms of performance but are sometimes preferred for problems requiring faster computation [82]. In a different direction, convolutional neural networks (CNNs), originally designed for image data, were also adapted for time series by applying 1D convolutions along the time axis, allowing the network to capture local temporal patterns. Finally, transformers, introduced in 2017 by Ashish Vaswani et al. [83], revolutionized sequential data modeling in the natural language processing domain. They rely on self-attention mechanisms to capture dependencies across different time steps without relying on sequential processing like RNNs. However, the price is that they require even larger datasets than LSTMs in order to learn effectively [84].

As we opted to use LSTM and CNN architectures in our contribution presented in Chapter 4, we will provide a detailed explanation of their functioning in the following sections.

2.3.2 Long Short-Term Memory (LSTM) Network

Long short-term memory (LSTM) networks are a specialized type of recurrent neural network (RNN) designed to model sequential data while addressing the vanishing gradient problem, which is a limitation of traditional RNNs. LSTMs are capable of learning long-term dependencies in data, making them suitable for tasks such as natural language processing, speech recognition, and time series prediction, as discussed in this review [85].

As a neural network, LSTMs consist of an input layer, some hidden layers, and an output layer. The core component of the network is the LSTM layer. The network can have a single LSTM layer or multiple stacked LSTM layers to capture different levels of temporal dependencies. Following the LSTM layers, there are typically one or more dense layers.

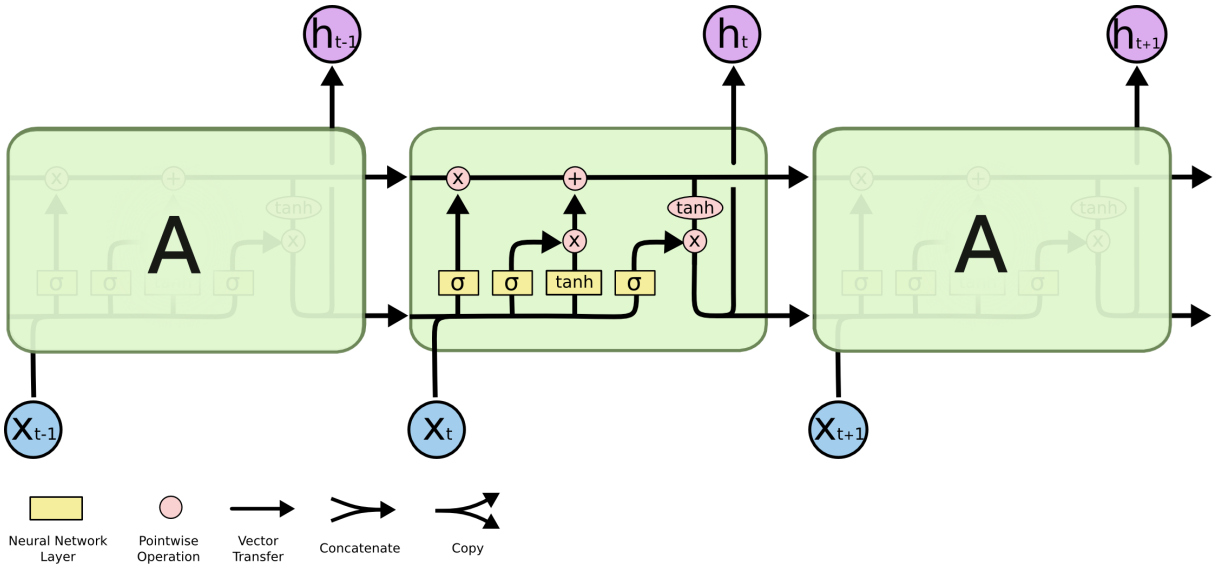


Figure 2.7 – **The LSTM unit.** Figure from Christopher Olah's "Understanding LSTM Networks" [86].

Each LSTM layer comprises multiple LSTM **units** (or cells), as shown in Figure 2.7. This illustration, along with the next three figures, was borrowed from the work of Christopher Olah [86] with the author's permission. The LSTM unit is the fundamental building block of the LSTM network, with several key terms that need to be defined here. To begin with, the **hidden state** is the output of the LSTM unit at each time step and serves as input to subsequent layers. It represents the short-term memory of the network and is

influenced by the **cell state**. The cell state is the long-term memory of the LSTM unit, carrying information across time steps and being modified through gates. LSTM units use three primary gates to regulate the flow of information.

First, the **forget gate** (illustrated in Figure 2.8) controls what information should be discarded from the current cell state. It takes the current input x_t and the previous hidden state h_{t-1}, processes them through a sigmoid activation function, and produces values between 0 and 1 for each element in the cell state C_{t-1}. The closer to 1, the more information is retained, and the closer to 0, the more information is discarded.

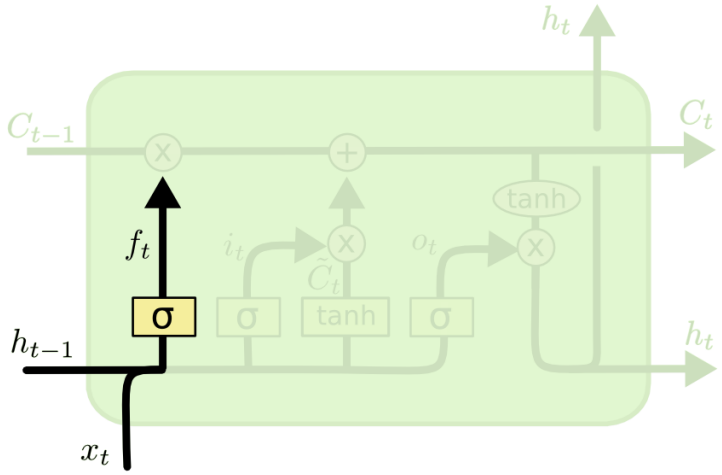


Figure 2.8 – **The forget gate of an LSTM unit.** Figure from Christopher Olah's "Understanding LSTM Networks" [86].

Second, the **input gate** (Figure 2.9) determines what new information should be added to the cell state. It controls the extent to which the new candidate cell state is integrated into the current cell state. Using the same inputs x_t and h_{t-1}, a sigmoid layer decides which values to update. A tanh layer then generates a vector of new candidate values, which could be added to the state. The tanh function ensures that the candidate values range between -1 and 1, which helps to moderate the effect of new information. At that point, the cell state C_t is updated by combining the previous cell state C_{t-1} (adjusted by the forget gate) with the new candidate cell state (weighted by the input gate).

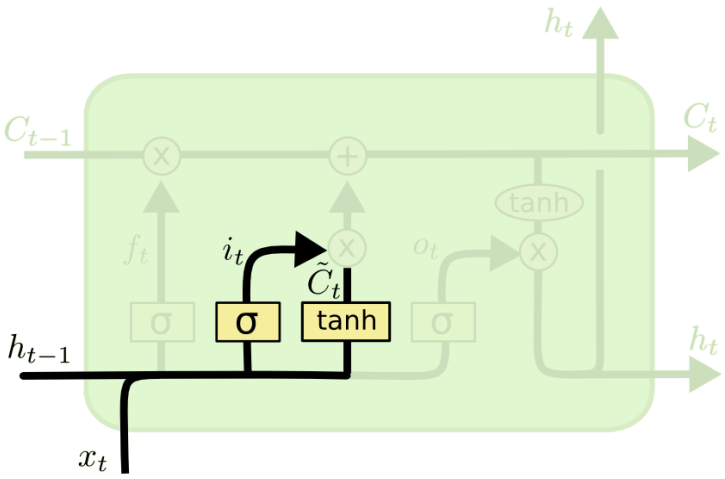


Figure 2.9 – **The input gate of an LSTM unit.** Figure from Christopher Olah's "Understanding LSTM Networks" [86].

Third, the **output gate** (Figure 2.10) determines the next hidden state h_t, based on the updated cell state C_t. The role of the tanh function is to compress the cell state values to a range of -1 to 1, while the sigmoid function impacts the contribution of the current cell state to the final output. This hidden state is then passed to the next time step and can also serve as output at the current time step.

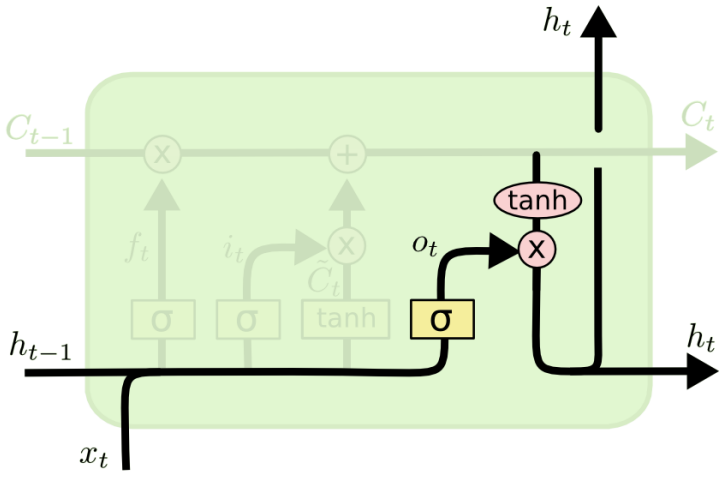


Figure 2.10 – **The output gate of an LSTM unit.** Figure from Christopher Olah's "Understanding LSTM Networks" [86].

A vocabulary table (Table 2.3) is provided below.

LSTM unit	A cell, also called a node, that manages long-term dependencies in sequential data using gates to control the information in its internal memory.
Hidden state	The output of the LSTM unit at each time step, which is derived from the cell state.
Cell state	The long-term memory of the LSTM unit that carries information across time steps and is regulated by gates.
Forget gate	The gate that controls the flow of information by deciding what to discard from the cell state.
Input gate	The gate that determines how much new information to add to the cell state.
Output gate	The gate that decides what part of the cell state to output as the hidden state.

Table 2.3 – **Long short-term memory (LSTM) networks vocabulary.**

2.3.3 One-Dimensional Convolutional Neural Network (1D CNN)

A one-dimensional convolutional neural network (1D CNN) is a type of CNN specifically designed to process sequential data, such as time series. Unlike LSTMs, which are explicitly designed to capture long-term dependencies, 1D CNNs do not have a built-in mechanism for long-term memory. However, their simpler architecture generally allows for quicker training times and often requires less data. They have also been widely used for tasks such as speech recognition, time series classification and regression [87].

To understand 1D CNNs, we need to adapt the key concepts of convolutional layers to one-dimensional data. So, the input data is typically a sequence of values. For example, in time series data, the input might be a series of measurements over time. If the data has multiple features (channels), the input might be represented as a 2D array. While 2D CNNs operate on 2D grids (such as images), 1D CNNs apply convolutions along a single axis, typically time or sequence index, as illustrated in Figure 2.11. The filter slides

along a single dimension to capture local temporal patterns and features from the input sequence. The kernel size corresponds to the length of the sliding window in the temporal dimension, operating across all channels. Similarly, pooling layers are replaced by their 1D counterparts that slide along one dimension. Afterward, flattening and dense layers are applied in the same manner as in 2D CNNs.

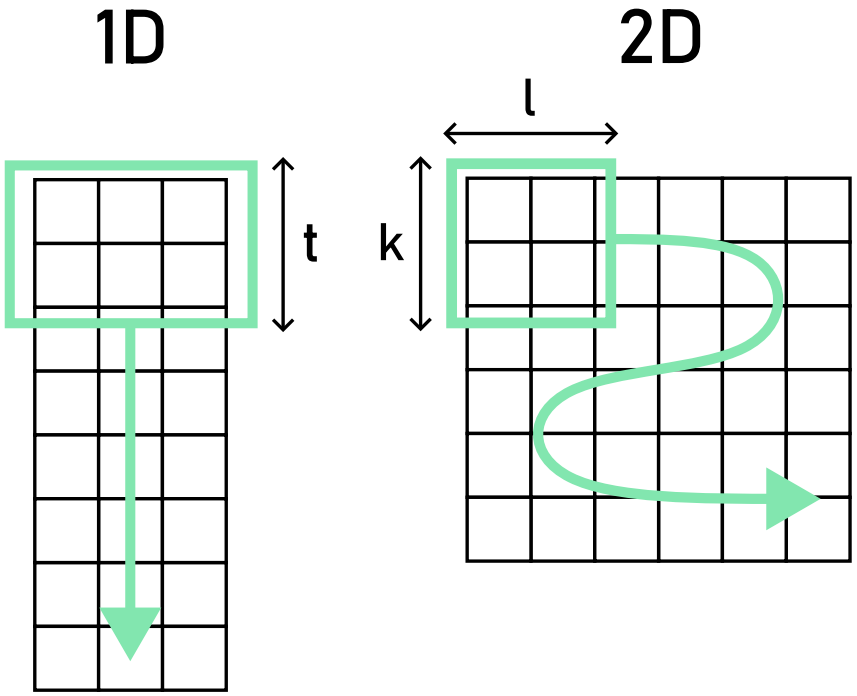


Figure 2.11 – **Comparison between 1D and 2D convolution.** In 1D convolution, the filter slides along the temporal axis and across all channels. The kernel size, defined by the user, is represented here as $t = 2$ and represents the number of timesteps to be considered. In 2D convolution, the filter moves along both spatial dimensions. The user selects the kernel sizes k and l, although the kernel is often square-shaped, as shown here with $k = l = 2$.

2.4 Conclusion

In this chapter, I have presented the foundational concepts necessary to understand the contributions of this thesis. I began by explaining the general principles of neural networks, particularly how they update their weights through backpropagation. Then, I explored the objectives and methods of image segmentation, highlighting the evolution from manual segmentation, which requires significant time and expertise, to automatic segmentation using deep learning techniques. I provided a detailed overview of the layers that constitute a convolutional neural network (CNN), followed by an in-depth look at the U-Net architecture in both its 2D and 3D forms, which are widely recognized in the biomedical field. Finally, I reviewed the historical development of techniques for time series regression, focusing afterward on long short-term memory (LSTM) networks and one-dimensional convolutional neural networks (1D CNNs). Through this chapter, I hope to have provided a greater understanding of these concepts and methods, which I applied in my contributions presented in the next part.

Chapter highlights

- A neural network is a model that learns its parameters, known as weights, from data through a process called backpropagation.
- Hyperparameters are user-defined values that determine the architecture and training framework of a neural network.
- Convolutional neural networks (CNNs) are a class of neural networks primarily used in computer vision due to their ability to detect patterns and spatial hierarchies of features within images.
- The U-Net architecture is a sub-type of CNNs that is highly popular for 2D and 3D biomedical image segmentation because it can achieve good performance with a limited number of training images, thanks to its U-shaped design. This architecture is the one chosen for our contribution presented in Chapter 3.
- Long short-term memory (LSTM) networks are a specialized type of recurrent neural network (RNN) designed to learn long-term dependencies in sequential data while addressing the limitations of traditional RNNs.
- One-dimensional convolutional neural networks (1D CNNs) are a sub-type of CNNs designed specifically to process sequential data. Their simpler architecture, which do not have a built-in mechanism for long-term memory, generally allows for quicker training times and often requires less data to achieve good performance. Both LSTMs and 1D CNNs were used in our contribution presented in Chapter 4.

Part II

Contributions

Chapter 3

Localization of EEG electrodes within MRI acquisitions

In this chapter, we propose a new, fully automatic method for retrieving the coordinates and labels of EEG electrodes in an ultra-short echo-time (UTE) MR volume. This method combines a U-Net deep learning model for segmentation with the ICP algorithm for refinement. This work [88] has been published in Frontiers in Neurology: `https://doi.org/10.3389/fneur.2021.644278`.

3.1 Introduction

As presented in Chapter 1, functional magnetic resonance imaging (fMRI) is a technique that allows to visualize brain activity by detecting hemodynamic variations. It is a non-invasive method widely used for studying brain function [89]. Electroencephalography (EEG), another non-invasive method, measures the brain's electrical activity using electrodes placed on the scalp. It is widely employed for diagnosing brain disorders and studying neurophysiological activity [90]. These two techniques are complementary in studying many neurological disorders. Indeed, fMRI offers excellent spatial resolution, in the order of a millimeter, but has lower temporal resolution, in the order of a second. Conversely, EEG has high temporal resolution (milliseconds) but lower spatial resolution [91]. Source localization in EEG involves solving an inverse problem sensitive to several parameters [92], one of the main ones being the forward head model used. Another important parameter is the 3D position of the EEG electrodes on the scalp [93], as the accuracy of the estimated coordinates of the electrodes impacts the localization of the EEG sources. In fact, position errors lead to inaccuracies in the estimation of the EEG inverse solution [94]. This issue is even more significant in studies involving simultaneous EEG-fMRI

acquisitions, where multiple sessions and thus multiple EEG cap installations may be required. To fully benefit from these mixed acquisitions, optimal registration between EEG and MRI data is essential. Therefore, obtaining the EEG electrode positions reliably and accurately is crucial.

Several methods have been proposed to address this question [95]. To begin with, there are semi-automated methods that require manual measurements [96], which are time-consuming and subject to human error. Additionally, some methods require extra materials, such as electromagnetic or ultrasound digitizers [97, 98]. In the context of simultaneous EEG-fMRI acquisitions, methods utilizing MR localization of electrodes are also available. In that case, although a measurement system external to the EEG (the MRI) is used, MR-compatible EEG systems are designed to be as invisible as possible on most MRI sequences. Consequently, some methods still require manual measurements [99], while others necessitate special equipment [100, 101]. More recent studies have proposed using an ultrashort echo-time (UTE) sequence, in which the electrodes are more visible [102, 103]. This type of sequence [104, 105] allows visualization of tissues with very short T2 and T2$\star$, such as cortical bone, tendons, and ligaments, and has the side effect of enabling imaging of MR-compatible electrodes. The introduction of these new sequences opens the door to more automatic methods that are easily usable in clinical practice. Indeed, no additional equipment is required, and the additional acquisition time is quite short (a few minutes, see section 3.2.2), which does not overly burden the corresponding EEG-fMRI studies. In a preceding work of our lab [106], the authors proposed a fully automated method based on a segmentation step followed by a Hough transform to select the positions of MR-compatible electrodes in an MRI volume using the UTE sequence. This method does not require any additional hardware and is fully automatic, but can be sensitive to scalp segmentation errors. Thus, our aim here is to retain the advantages of this method, namely generalization and automation, while simplifying the process by minimizing preliminary steps. In this work, we also use a type of UTE sequence to create an automatic method and explore the contribution of machine learning to the electrode detection task.

So, we propose a new approach that combines deep learning segmentation and template-based registration. Our method begins by training a model to detect the position of the electrodes in an MRI volume. This model is based on the U-Net neural network, a fully convolutional neural network known for achieving accurate

segmentations [107]. As mentioned above, we use a type of UTE sequence: the PETRA (Pointwise Encoding Time Reduction with Radial Acquisition) sequence [108], which is gradually becoming the new standard for UTE sequences. Finally, we apply the iterative closest point (ICP) algorithm [109] to consider the geometrical constraints after the deep learning phase and to label the electrodes accurately.

3.2 Materials

3.2.1 Data

Prior to my arrival in the Empenn team, a set of 60 PETRA volumes was acquired, along with corresponding T1 images, from 20 different participants. Each participant had between 2 and 5 images acquired in different sessions, which implies a new positioning of the EEG cap each time. These images varied between two quality levels: 30k and 60k spokes. A spoke refers to a radial line of data collected in k-space during radial sampling, which contributes to the reconstruction of the final PETRA image. Higher spoke count leads to better resolution and more accurate representation of structures. The volumes were divided into two datasets: one for training the segmentation model and one for testing its performance. We separated the data by assigning 12 subjects to the training dataset and 8 subjects to the test dataset, resulting in 37 training volumes and 23 test volumes. A comparison between a T1 image and a PETRA image is shown in Figure 3.1.

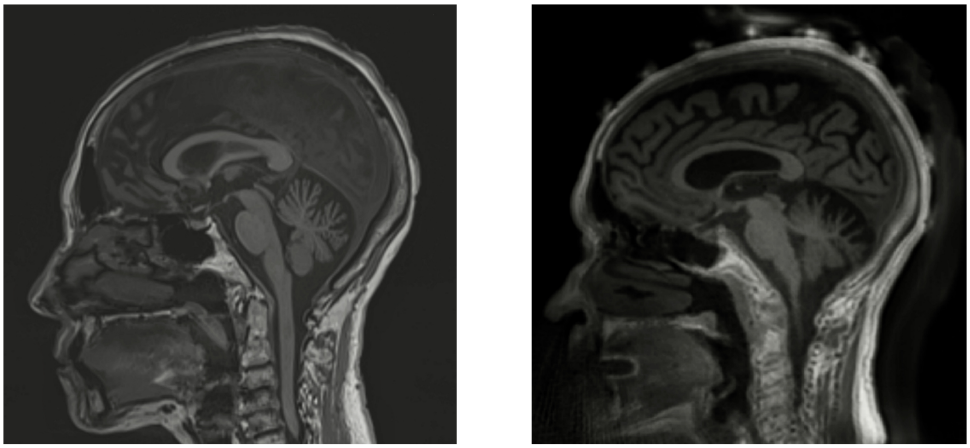


Figure 3.1 – **2D visualization of MR images.** On the left is a T1 MR image, and on the right is the corresponding PETRA image. The PETRA image demonstrates enhanced visibility of the EEG electrodes compared to the T1 image.

3.2.2 Equipment

EEG signals were acquired using an MR-compatible 64-channel cap (Brain Products, Gilching, Germany) with a circumference between 56 and 58 cm. The cap featured 64 Ag/AgCl electrodes placed in accordance with the extended international 10–20 EEG system, including one additional ground electrode at AFz. Two 32-channel MR-compatible amplifiers (actiCHamp, Brain Products, Gilching, Germany) were used. The electrodes were attached to small cups with a diameter of 10 mm and a height of 4 mm, and inserted into the cap with gel. Particular attention was given to reducing electrode impedance and positioning the electrodes according to standard fiducial points. A picture is provided in Figure 3.2.

MRI was performed with a 3T Prisma Siemens scanner running VE11C with a 64-channel head coil (Siemens Healthineers, Erlangen, Germany), a picture of which is also provided in Figure 3.2. PETRA acquisitions were obtained using echo-planar imaging (EPI) with the following parameters: repetition time TR1/TR2 = 3.61 ms/2 250 ms, inversion time TI1/TI2 = 1 300/500 ms, echo time TE = 0.07 ms, flip angle = 6°, field of view (FOV) = 300×300 mm^2, voxel size = $0.9 \times 0.9 \times 0.9$ mm^3, matrix size = 320×105, with 60 000 and 30 000 spokes. The acquisition lasted 6 minutes for the 60K quality and 3 minutes for the 30K quality. Consequently, the PETRA images we used had a size of $320 \times 320 \times 320$ mm and a voxel spacing of 0.9375×0.9375 mm. Additionally, a 1 mm isotropic 3D T1 MPRAGE structural scan was acquired (shown in Figure 3.1).

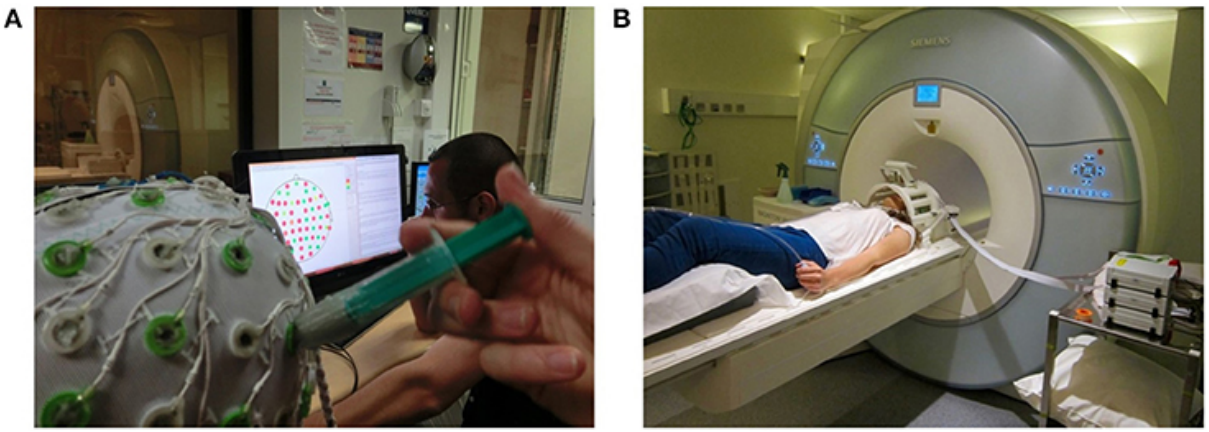


Figure 3.2 – **Pictures of the equipment from the bi-modal EEG-MRI platform.** (A) MR-compatible 64-channel EEG cap from Brain Products. (B) 3T MRI scanner from Siemens. Figure provided by Marsel Mano et al. in the article "How to build a hybrid neurofeedback platform combining EEG and fMRI" [42].

3.3 Methods

Our method consists of two main steps: the first is based on a deep learning segmentation, and the second is based on template registration. Figure 3.3 provides an overview of the method's principles. We will begin by describing the process of training a segmentation model, covering data preparation and neural network training. Next, we will detail our approach for detecting and labeling EEG electrodes in MR images, explaining how to utilize the trained model and the template registration step to obtain the electrode coordinates.

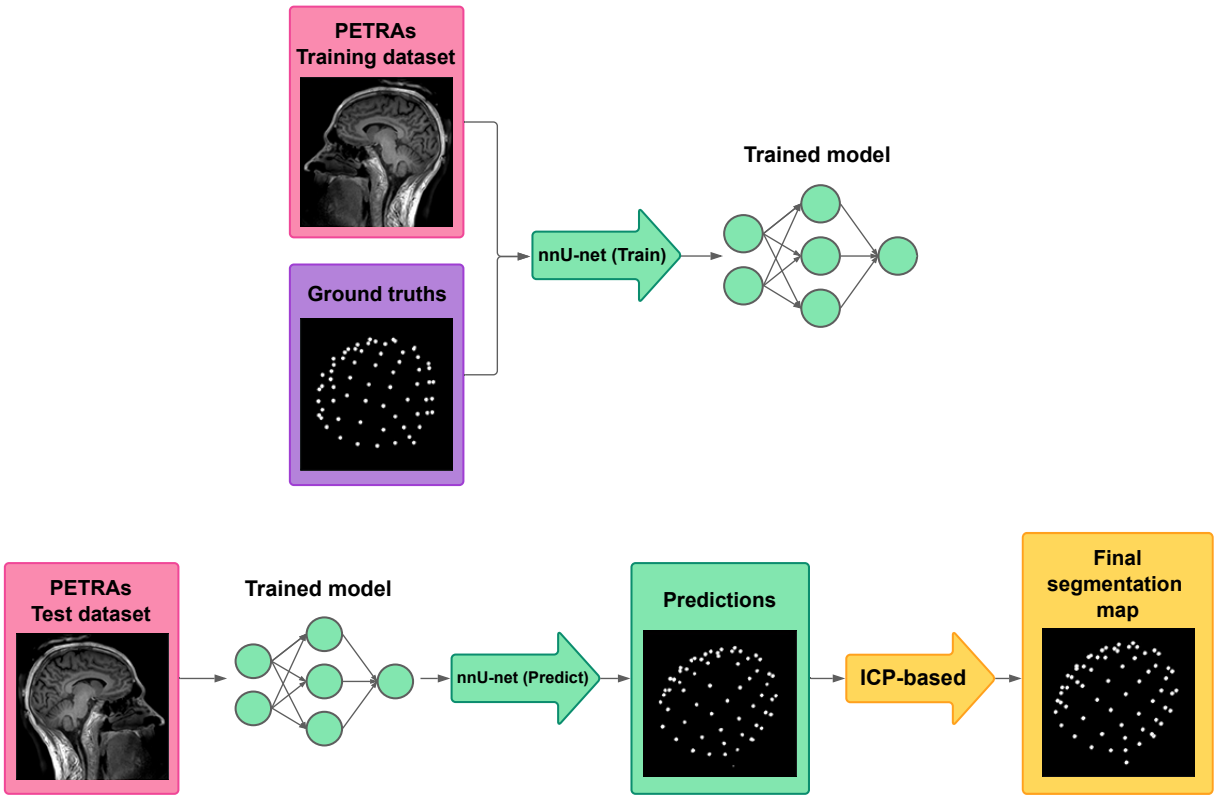


Figure 3.3 – **Overview of the electrode detection framework.** The framework consists of the training process (top) and then the model prediction and registration-based refinement step (bottom). Using the training dataset and the corresponding labeled ground truths, the U-net model is trained with the nnU-Net framework. Next, our method involves taking an image that the model has not previously seen and generating a predicted segmentation map of the electrodes. Finally, template-based adjustments are made to obtain the final labeled segmentation map.

3.3.1 Ground truth estimation

To train and evaluate our model, we computed ground truth segmentation on the PETRA volumes by manually indicating the positions of the electrodes. The ground truths here are segmentation maps of the same size and characteristics as the PETRA images, with segmented spheres representing the 65 EEG cap electrodes visible on the scalp. Each electrode is assigned a different value or "label" (corresponding to the same channel for all subjects), with a value of 0 for the background. To ease the manual creation of these ground truths, an automatic scalp segmentation mask was first estimated. Since T1 images have higher quality than PETRA images in the scalp area, this mask was obtained by first registering the T1 image to the corresponding PETRA image and then segmenting the scalp on the registered T1 image using the FSL library [110]. These two inputs allowed the use of a Matlab implementation developed by Butler [111] of a method proposed by de Munck et al. [99], which displays a so-called "pancake" view of the scalp. This colorimetric 2D projection of the scalp region eases the manual selection of electrode positions. As a result, a 3D labeled segmentation of each PETRA volume was created, illustrated in Figure 3.4.

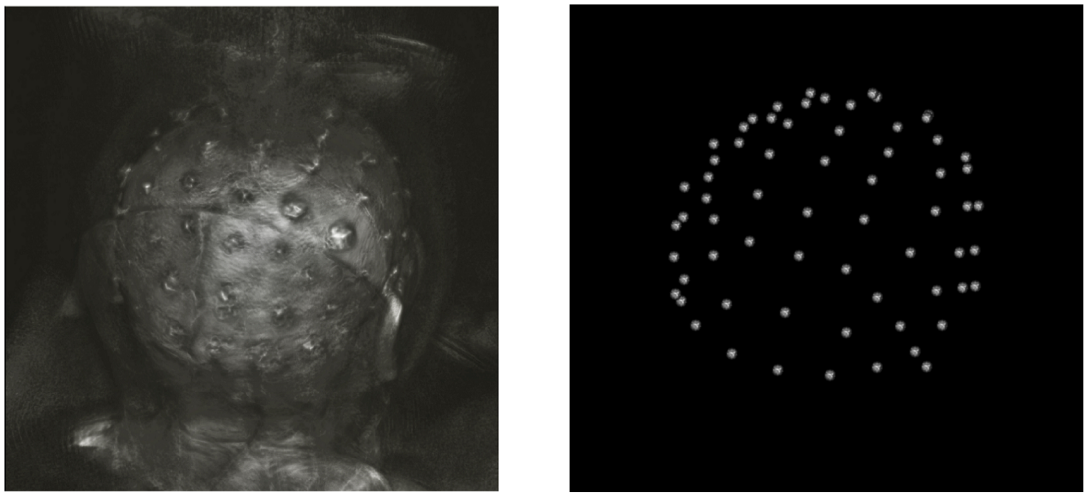


Figure 3.4 – **3D visualization of input image and ground truth.** On the left is a 3D rendering of a PETRA image used as input, and on the right is the corresponding segmentation map serving as the ground truth. The segmentation map shows the precise locations of the 65 EEG cap electrodes, represented as segmented spheres.

3.3.2 Training framework

The training dataset consists of 37 PETRA images and their associated ground truths, as described above. We used the nnU-Net framework [112] presented in Chapter 2, a tool that automates the choice of hyperparameters for training a model from any dataset and for any segmentation task. This is especially useful given the numerous variations of neural network architectures proposed for segmentation. The authors of [112] demonstrated that slight design improvements hardly enhance performance, while the choice of hyperparameters seems to be crucial. Indeed, using this framework with a basic U-Net architecture outperformed most specialized deep learning pipelines in 19 international competitions and 49 segmentation tasks, demonstrating its efficiency and adaptability. Among the different types of neural networks available, we chose the 3D U-Net [113], in which operations such as convolution and max pooling are replaced by their 3D counterparts. Once the neural network architecture was chosen, the framework automatically estimated the best hyperparameters from the provided dataset. Our model was trained over 1 000 epochs using a batch size of 250, with a loss function that is the sum of cross-entropy and Dice loss, and with a Stochastic Gradient Descent (SGD) optimizer. The patch size was $128 \times 128 \times 128$, and the default data augmentation scheme provided by nnU-Net was used. It is important to note that we used an earlier version of the framework, which is still available at `https://github.com/MIC-DKFZ/nnUNet/tree/nnunetv1`. This version predates the release of the article from which Figure 2.6 from Chapter 2 is taken, as well as the current version, which is available at `https://github.com/MIC-DKFZ/nnUNet`.

3.3.3 Model predictions and template-based refinement

Once the model is trained, PETRA images from the test dataset can be provided as input for the model to perform predictions. The nnU-Net framework uses a sliding window approach for making predictions, employing the same patch size used during training, with each step overlapping half of the patch. To enhance performance, avoid artifacts, and ensure high-quality segmentation, several strategies were employed: Gaussian importance weighting was used to reduce edge problems and stitching artifacts, and test-time augmentation, which is data augmentation for test datasets, was applied by generating slightly modified images (rotations, scaling, deformations, mirroring) from the test image and averaging the detection made on them. Although this test-time augmentation step is time-consuming, we will compare the results obtained with and without it in section 3.4.

The U-net model can, of course, take into account spatial information. However, it has more difficulty incorporating the strong geometrical constraint of our problem: the electrodes are all placed on a cap, certainly slightly elastic, but the distances between electrodes remain relatively constant. To take advantage of this geometric constraint, we propose a second step to refine the predictions provided by the neural network. The main objectives of this second step are to ensure the number of detection is exactly 65 and to correctly label the electrodes. We begin by registering the n detections (n is not necessarily equal to 65) to a template of the EEG cap using the iterative closest point (ICP) [109] algorithm. Figure 3.5 illustrates the principles of this step. The template used here is obtained by averaging the coordinates of 12 manually obtained ground truth point clouds taken from the training set. We use one per subject to account for head shape variability. This step involves registering these two point clouds, namely the prediction from the U-net model and the template, using the ICP algorithm with similarity transformation (rotation, translation, scaling). First, each point in the moving set is associated with the nearest point in the fixed set. Then, the geometric transformation that minimizes the distance between these pairs of corresponding points is estimated. This transformation is applied, and the process is iterated until convergence. Then, by comparing the distance between the prediction points and the template points, a refinement of the detection is carried out. First, each prediction point is associated with its closest template point, and for each template point, only the closest prediction point is kept. As a result of this sub-step, a maximum of 65 predicted positions are conserved. Since only the predictions closest to the template were kept, outliers may have been removed from our initial detections, which is likely to improve the registration. Consequently, a new ICP is then performed. Finally, using this improved registration, and in the case where fewer than 65 predictions were kept, the missing positions are added as follows: each template point not associated with any prediction position is added to the final result. Thus, our final result contains exactly 65 detections, each associated with a point on the template, which provides us with a label.

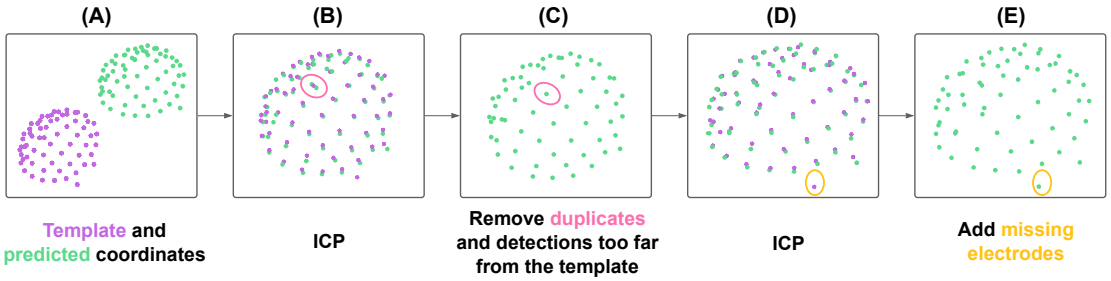


Figure 3.5 – **Description of the registration-based refinement step.** (A) In green: the prediction points from the U-net model; in purple: the template obtained by averaging the training set. (B) A first ICP is performed to register the two point clouds. (C) For each template point, only the closest detection is kept. (D) Then, a second ICP is performed, and the number of detection is now less than or equal to 65. (E) Finally, the points in the template not associated with any predictions are added to our final result, which therefore contains exactly 65 detections.

3.3.4 Evaluation on the test dataset

To evaluate the proposed method using the test dataset, we compared the detected electrodes to the ground truth coordinates obtained manually. We computed the connected components for the two images and determined the position of their centers. For each prediction point, we calculated its distance to the nearest ground truth point. Here, this distance is called a position error (PE). A prediction with an error greater than 10 mm, corresponding to the diameter of an electrode cup, is considered a wrong detection (false positive). Since we systematically consider the nearest ground truth electrode, we do not consider the labeling when estimating the position error. The quality of the final labeling, as well as that of the intermediate labeling, will be evaluated separately by simply comparing the predicted labels with the template labels. Since the number of detection is exactly 65, the number of false positives is automatically equal to the number of missing points (false negatives).

3.3.5 Evaluation of the robustness of the method on a different UTE sequence

In order to evaluate the robustness of the method and to compare our results with those of the previous work [106], we also applied it to images acquired using a different UTE sequence, as described in the mentioned article. First, we directly used the model trained

on the PETRA images to study the generalizability of the trained model to another MR sequence. Then, we trained a new model on the different UTE database, which contains fewer images. This allowed us to investigate the importance of the number of data in the training set and to compare our results to the previously introduced method.

3.4 Results

3.4.1 Predictions on the test dataset with test-time augmentation

The results are assessed by measuring the position error, as described in section 3.3.4, for all volumes in the test dataset. Table 3.1 presents the average results for all subjects in the PETRA test dataset. This test set consists of 23 volumes from 8 different subjects not included in the training dataset, with sampling resolutions of either 30k or 60k spokes.

	After segmentation	**After refinement**
Mean PE (mm)	2.12	2.24
Std PE (mm)	1.50	1.37
Max PE (mm)	8.84	7.99
Mean number of false positives	0.30	0.22
Mean number of true positives	65.0	64.8
PPV (%)	**99.5**	**99.7**

Table 3.1 – **Predictions on the test dataset with test-time augmentation.** Rows 1, 2, and 3: mean, standard deviation, and maximum values of position error (PE). Rows 4 and 5: mean number of false positives (PE>10 mm) and true positives (PE≤10 mm). Row 6: positive predictive value (PPV). First column: intermediate results after the deep learning segmentation step. Second column: final results after the registration-based refinement step.

	After segmentation	**After refinement**
Mean number of mislabeled electrodes	1.87	0
Maximum number of mislabeled electrodes	11	0

Table 3.2 – **Labels for the predictions on the test dataset with test-time augmentation.** Rows 1 and 2: Mean and maximum labeling errors among the **true positives**. First column: intermediate results after the deep learning segmentation step. Second column: final results after the registration-based refinement step.

The average position error is 2.24 mm, and is to be compared to the diameter of one electrode cup, which is 10 mm. The number of good (true positive) and wrong (false positive) detections was also assessed, using a 10 mm distance as the threshold. As shown in the table, after the deep learning-based step, the number of detections was too high on average, but it was corrected after the registration step. This correction resulted in a better positive predictive value (PPV), defined as the percentage of detections that are true positives relative to the total number of detections. The average total number of detections after the first step is 65.3 (mean number of false positives (0.3) + mean number of true positives (65.0)), which is higher than the actual number of electrodes (65). This was expected since the neural network used does not incorporate any constraint on the number of detections. The output of this first step is a simple volume where each voxel has a label indicating whether it is considered to belong to the background or to a specific electrode. Note that, in this case, two detections associated with the same electrode can count as two good detections, as long as their distance to the ground truth electrode is less than 10 mm. After our registration-based refinement step, the final number of detections is, as expected, exactly equal to 65 (mean number of false positives (0.22) + mean number of true positives (64.78)). Twenty-three volumes were processed, corresponding to a total of 1 495 electrodes, out of which 1 490 were correctly detected and 5 were missed. These missing electrodes often correspond to those located behind the ears, which caused a few outliers in the output. These outliers are reflected in the value of the average maximum error, 8.84 mm. A slight increase in the mean PE after registration is noted. Indeed, the refinement step usually allows for the recovery of some missing electrodes in the intermediate detections provided by the neural network. These new electrodes are therefore provided by the registered template. Although often considered as true positives because they are close enough to the ground truth, they are sometimes slightly less accurate than the true positive deep learning-based detections, causing this relative increase in the mean PE. However, it can be noted that this increase in mean PE comes with a decrease in the standard deviation of position error.

Finally, regarding labeling, 100% of the electrodes were correctly labeled in our final results. As shown in Table 3.2, this was not the case after the deep learning-based step. This explains our choice of ICP for the registration step: we cannot always rely on the labeling of intermediate results. Indeed, the number of labeling errors can be as many as 11 in a volume. In fact, these observed errors often correspond to a simple offset in labeling: an electrode is incorrectly labeled and all its neighbors are then likely to be contaminated

by this error. Therefore, we decided to disregard the labeling information provided by the neural network and rely solely on the ICP result for this purpose. It may seem odd to include the electrode labels in the ground truth for the training step instead of using a binary map, since the labels resulting from the model segmentation are discarded and the labels from the refinement step are used instead. However, our experiments have shown that training a neural network with labeled ground truth improves detection results in terms of position error compared to a situation where the ground truths are simple binary maps. In particular, when 65 different labels are provided during training, the network is more likely to detect a number close to 65 during the test phase.

3.4.2 Faster predictions on the test dataset without test-time augmentation

For each new PETRA image provided, the method presented above allows us to make predictions in about 7 minutes on our GPU, with almost all of this time being used by the first step, which is based on the neural network. In fact, the ICP-based refinement step runs in just a few seconds. Therefore, we explored the possibility of reducing the computing time required by the framework to obtain a prediction. To this end, we removed the test-time augmentation, mentioned in section 3.3.3. The prediction time of an image was then significantly reduced to about 2 minutes. Table 3.3 presents the results of this faster detection pipeline.

	After segmentation	**After refinement**
Mean PE (mm)	6.78	2.23
Std PE (mm)	25.4	1.40
Max PE (mm)	168.7	8.20
Mean number of false positives	2.57	0.13
Mean number of true positives	65.1	64.9
PPV (%)	**96.3**	**99.8**

Table 3.3 – **Faster predictions on the test dataset without test-time augmentation.** Rows 1, 2, and 3: mean, standard deviation, and maximum values of position error (PE). Rows 4 and 5: mean number of false positives (PE>10 mm) and true positives (PE≤10 mm). Row 6: positive predictive value (PPV). First column: intermediate results after the deep learning segmentation step. Second column: final results after the registration-based refinement step.

	After segmentation	**After refinement**
Mean number of mislabeled electrodes	3.2	0
Maximum number of mislabeled electrodes	13	0

Table 3.4 – **Labels for the faster predictions on the test dataset without test-time augmentation.** Rows 1 and 2: Mean and maximum labeling errors among the true positives. First column: intermediate results after the deep learning segmentation step. Second column: final results after the registration-based refinement step.

All of the indicators for intermediate results, after the deep learning-based step alone, show that they are clearly worse with this accelerated version: there is a significant increase in position error (mean, standard deviation, and maximum values) and an increase in the total number of detections. However, the associated detections contain enough valuable information so that the robustness provided by our refinement step allows us to finally obtain results as good as in the version with test-time augmentation. Counter-intuitively, some metric values are even slightly better. However, a statistical paired t-test showed that none of these changes were significant ($p > 0.5$ for all comparisons).

Finally, as in the version with test-time augmentation, the labeling contained some errors in the intermediate results but is completely accurate in our final results, even with this faster version, as shown in Table 3.4. The second step, already important for improving the results in the previous version, turns out to be crucial when we want to accelerate the processing by the neural network, allowing us to obtain similar results.

3.4.3 Predictions on a different UTE sequence to evaluate robustness

In order to evaluate the robustness of our method, we challenged it by testing it on a dataset from another MRI sequence, the original UTE one [106], with test-time augmentation. Eleven subjects were included in this new study. A 60k-spokes acquisition was performed for all subjects, and a 30k-spokes image was acquired for seven of them. First, the previous model, trained using the PETRA images, was used to detect the electrode positions on these 18 new images acquired with a different UTE sequence. Results are shown in Table 3.5.

	After segmentation	**After refinement**
Mean PE (mm)	1.81	2.47
Std PE (mm)	1.67	1.64
Max PE (mm)	11.06	9.36
Mean number of false positives	0.33	0.72
Mean number of true positives	56.4	64.22
PPV (%)	**99.4**	**98.89**

Table 3.5 – **Predictions on a different UTE sequence using the previous model trained on PETRA images.** Rows 1, 2, and 3: mean, standard deviation, and maximum values of position error (PE). Rows 4 and 5: mean number of false positives (PE>10 mm) and true positives (PE≤10 mm). Row 6: positive predictive value (PPV). First column: intermediate results after the deep learning segmentation step. Second column: final results after the registration-based refinement step.

As expected, the detections estimated by the neural network were not as good as in the previous case. The average number of electrodes provided was lower than 57. However, and very interestingly, these electrodes were mostly true detections. For this reason, and as can be seen in the table, the ICP-based registration step was able to retrieve almost all missing electrodes, once again leading to excellent performance results. Moreover, all the detected electrodes were once again well-labeled in the final results: there was no mislabeling among the true positives. Our registration-based refinement step thus brings robustness to the method, limits the risk of overfitting, and improves its generalizability.

Finally, to compare our results to those in [106], we trained a new neural network using only this different UTE sequence, applied the refinement step, and evaluated the resulting performance. From the previously described UTE dataset, we built two groups: 9 MR volumes in the training set and 9 volumes in the test set, ensuring that no subjects were present in both sets. Table 3.6 shows the corresponding results.

	After segmentation	After refinement
Mean PE (mm)	1.70	2.42
Std PE (mm)	1.24	1.29
Max PE (mm)	8.02	8.19
Mean number of false positives	0.56	0.44
Mean number of true positives	60.0	64.6
PPV (%)	**99.1**	**99.3**

Table 3.6 – **Predictions on a different UTE sequence using a new model trained on images acquired with the same UTE sequence.** Rows 1, 2, and 3: mean, standard deviation, and maximum values of position error (PE). Rows 4 and 5: mean number of false positives (PE>10 mm) and true positives (PE≤10 mm). Row 6: positive predictive value (PPV). First column: intermediate results after the deep learning segmentation step. Second column: final results after the registration-based refinement step.

Training the model using the same type of images as in the tests slightly improved the quality of the detections compared to using the model trained on PETRA images. Despite the smaller group size, our results are now better than those reported in [106]. For example, the mean PPV is now 99.3%, whereas it was between 88% and 94% for 30k and 60k spokes images, respectively. Once again, all the true positive detected electrodes were well-labeled.

3.5 Discussion

We have introduced a new fully automatic method for the detection of EEG electrodes in an MRI volume in the context of simultaneous EEG-MRI acquisition. This technique is easy to set up and use, and it provides accurate and reliable results. Once the model has been trained, the method requires only the acquisition of a PETRA volume after the installation of the EEG headset. No additional equipment is needed, and the PETRA volume can be acquired in a few minutes. The majority of the computation time is spent on the model prediction, which can be accelerated to about 2 minutes. This deep learning step is crucial to the proposed method. However, as the results have shown, the second registration-based step not only improves the final results but also makes them more robust to potential outliers.

It is well-known that deep learning models are highly dependent on the quality and representativeness of the training data. Our preliminary investigations using a different

UTE sequence suggest that the method can be generalized to other types of images, even when using the model trained on the initial data, thanks to the robustness provided by the registration step. Another interesting question is how the method behaves when the number of electrodes differs between the training and testing phases. We hope that the robustness of the second ICP-based step will allow for accurate detection if the same sequence and the same type of electrodes are used, but this needs to be verified with further investigation. Finally, this method has been tested on one type of EEG cap (Brain Products), but it is applicable to any detection problem involving elements on the scalp. It will be interesting to test it on other EEG headsets as well as on other systems, such as the near-infrared spectroscopy (NIRS) modality, which uses optodes placed on the scalp.

It is also worth noting that our second study, using the original UTE sequence, had a smaller sample size, which is probably more consistent with a typical simultaneous EEG-MRI study. Eleven subjects were involved, corresponding to 18 volumes, with only 9 used in the training phase. Despite the smaller amount of data, the results (Table 3.6) were only slightly less accurate than those obtained with a larger sample (Table 3.1).

3.6 Data, code, and model availability

— Data: Regrettably, the data used in this work are not publicly accessible.

— Code: The code developed for this research is available on Gitlab Inria at `https://gitlab.inria.fr/cpinte/deep-learning-based-localization-of-eeg-electrodes-within-mri-acquisitions`. Additionally, the code has been archived with Software Heritage to ensure long-term preservation at `https://archive.softwareheritage.org/swh:1:dir:fab927068a756d20b8925b754d2b7d0fb6e5d534;origin=https://gitlab.inria.fr/cpinte/deep-learning-based-localization-of-eeg-electrodes-within-mri-acquisitions.git;visit=swh:1:snp:bc32d1c87bfb525865326ecced86ed6061585f8f;anchor=swh:1:rev:a07402a3687127fed3f4eacd8bfe51d616601a9d`. All implementations were made on an Nvidia Quadro M6000 24GB GPU, which was the most powerful graphics processing unit in 2016 according to NVIDIA Corporation. Training takes between 1 and 2 weeks, depending on the number of processes running on the available GPU. As is typical in deep neural network methods, the prediction on the test

dataset is much faster. The presented method predicts a segmentation map from a PETRA image in about 7 minutes on the aforementioned GPU.

— Model: Since the data used to train the models are not open, sharing these weights introduces a theoretical risk of data reconstruction. There is an ongoing debate about how much information model weights retain about the original training data. Given these concerns, we feel compelled to prioritize privacy by not sharing the weights. If the question is resolved in favor of sharing the models, we will add them to the Gitlab repository mentioned above.

3.7 Conclusion

We presented a new method for the detection and labeling of EEG electrodes in an MR volume acquired using the PETRA sequence. The method involves training a model on a training dataset and associated ground truths, using this model to obtain a segmentation map, and then applying an ICP-based refinement step to improve the detections and their labeling. This fully automatic method is easy to implement, requires very few steps, and provides excellent results. For all these reasons, we strongly believe that it can be very useful for all protocols involving simultaneous EEG-fMRI acquisitions. In particular, when EEG source localization is planned, accurate information on the position of the electrodes is a definite advantage.

Chapter highlights

- We presented a method to automatically retrieve the coordinates and labels of EEG electrodes within ultrashort echo-time (UTE) MRI acquisitions.
- This method consists of two steps: first, segmentation of the images using a U-net neural network, and second, refinement and labeling of the detections using registration by ICP.
- The results show an average detection accuracy of 99.7% with an average positional error of 2.24 mm, and 100% accuracy in labeling.
- Faster predictions can be made, reducing the time from 7 minutes to 2 minutes, without loss of accuracy, thanks to the refinement step that provides robustness.
- The method can be extended to other types of UTE sequences and works well even on smaller dataset sizes.

Chapter 4

Prediction of fMRI neurofeedback scores from EEG signals

In this chapter, we explore the possibility of predicting functional magnetic resonance imaging (fMRI) neurofeedback (NF) scores from electroencephalography (EEG) signals, with the goal of reducing the reliance on MRI in the context of bi-modal neurofeedback. To select the architecture of the models, we introduce a hyper-parameter search method based on a genetic algorithm, which is applicable to different categories of neural networks. We then compare the results obtained by using LSTM and 1D CNN models. This work has been submitted to MELBA in November 2024.

4.1 Introduction

As presented in Chapter 1, neurofeedback (NF) is a non-invasive therapeutic technique that uses real-time monitoring of brain patterns to provide individuals with NF scores, feeding back information about their brain activity [114]. The primary goal of neurofeedback in clinical settings is to enable individuals to self-regulate their brain function, leading to improvements in cognitive, emotional and behavioral functioning across various health conditions, such as motor recovery after stroke [115, 116]. Acquisitions are typically made through non-invasive modalities such as electroencephalography (EEG) [117] or functional magnetic resonance imaging (fMRI) [118, 119].

EEG provides a direct measurement of real-time electrical potential changes in the brain using electrodes placed on the scalp. This equipment is known for its portability and affordability. While it offers excellent temporal resolution, operating within the millisecond range, its spatial resolution is limited to the centimeter range [120], notably due to the ill-posed inverse problem of source localization.

fMRI indirectly estimates brain activity by measuring variations in the blood oxygenation level-dependent (BOLD) signal, reflecting neurovascular activity. This activity generally occurs a few seconds after neural activation measured by EEG and is referred to as the hemodynamic response. In contrast to EEG, fMRI is non-portable and much more costly. While it offers better spatial resolution than EEG, in the millimeter range, its temporal resolution is inferior, typically in the second range.

The complementary nature of these resolutions quickly motivated the combination of EEG and fMRI. Despite the ongoing challenges in processing and integration, which persist to this day, simultaneous EEG-fMRI acquisitions offer multi-modal non-invasive measurements of brain activity applicable in many contexts [22], including neurofeedback [41, 42, 121]. Several studies have investigated the relationship between EEG signals and BOLD activity [122, 123, 124, 125]. However, the correlations identified do not consistently establish a link between the two modalities, as the results are highly dependent on the task, brain region, and frequency bands considered.

In the context of motor imagery neurofeedback, the use of simultaneous EEG-fMRI acquisitions has resulted in higher and more specific activation compared to EEG neurofeedback alone, as demonstrated in [43, 126]. However, as the use of MRI is very costly and burdensome for the participant, we aim to minimize its usage while maintaining the quality of the sessions. Thus, our goal is to develop a model capable of predicting fMRI NF scores from EEG signals alone, in order to enhance EEG NF scores during unimodal EEG neurofeedback sessions. This objective has already been investigated in a previous study of our lab [4], where a sparse regression model was proposed to predict fMRI NF scores from EEG signals during motor imagery tasks, showing encouraging results in the development of individualized models for each participant. Now, to take a step towards real application in clinical settings, we seek to create a single global model, applicable to all participants. As our problem falls into the category of time series regression, we have at our disposal two widely used classes of machine learning models for such tasks: recurrent neural networks (RNNs) and convolutional neural networks (CNNs).

As described in Chapter 2, RNNs [79] take into account past inputs through a feedback loop that incorporates both the current input and information from the previous input. This so-called short-term sequential memory is stored in the network's hidden state, which is updated with each new input, enabling it to capture the context and patterns of the sequence based on prior inputs. However, the original architecture had shortcomings,

notably the vanishing gradient problem [127]. To overcome this issue and better retain long-term dependencies in the data, the long short-term memory (LSTM) [128] architecture was introduced as an improved version of the RNN. RNN-LSTMs introduce a cell state with gates to retain important information over long sequences. However, despite their benefits, LSTMs are notoriously challenging to deploy effectively. The difficulty arises from the complexity of the architecture, leading to a greater risk of over-fitting, the requirement for a large training dataset, and the necessity of making numerous decisions concerning architecture and training hyperparameters [129]. Consequently, performance can vary significantly based on these factors [130].

CNNs [1] are a class of neural networks mainly used in computer vision due to their ability to find patterns and spatial hierarchies of features within images. The core principle of CNNs resides in convolution layers, which employ sets of learnable filters that slide across the input image and conduct convolution operations to extract spatial features. The first layers of the network detect basic features such as horizontal and vertical edges, while subsequent layers extract increasingly complex features such as objects or faces. Additionally, CNNs incorporate pooling layers to reduce the spatial dimensions of the feature maps produced, and fully connected layers to integrate the high-level features learned to perform classification or regression tasks. In the context of time series analysis, a specialized variant known as the one-dimensional convolutional neural network, or 1D CNN, can be employed. This type of CNN is designed for processing sequential data, such as time series datasets. It is worth noting that, contrary to LSTMs, CNNs are not explicitly designed to capture long-term dependencies. However, due to their simpler architecture, they typically offer quicker training times, often require less data for training, and can exhibit more stable performance [131, 132, 133].

When using neural networks, one of the most important issues is the design of the architecture. Most of them are designed manually, sometimes thanks to prior expertise, but often without any concrete justification for the choices made other than empirical exploration. In addition, in cases where users do not have sufficient expertise, for example on a problem like ours that has not yet been widely investigated, it becomes very difficult to achieve a high-performance network architecture as well as to justify the choices made by manually designing a network. One way of addressing this challenge is to adopt a well-known method, called the genetic algorithm, and apply it to the particular case of neural network architecture search. The genetic algorithm was introduced in 1975 by John

Holland and his collaborators, gaining popularity in the 1990s, as indicated by the reissued work [134]. It is an evolutionary algorithm inspired by the process of natural selection and genetic evolution and is used to solve all kinds of optimization problems. The general idea is to borrow from natural selection the concepts of reproduction, crossover, and mutation to iteratively evolve a population of potential solutions toward better solutions over the course of successive generations.

One of the many optimization problems that benefit from this approach is the search for neural network hyperparameters. Hyperparameters are values set by the user prior to the model training, distinct from parameters known as weights that are updated during the model learning process. Examples of hyperparameters include those controlling the learning process, such as the learning rate and batch size, or those defining the model architecture, such as the number of hidden layers and the dropout rate. The idea of applying the genetic algorithm approach to automatically set hyperparameter values was quickly investigated [135], and then specialized for various types of networks and tasks, such as regression with feed-forward neural networks [136], image classification with CNNs [137, 138], and natural language processing task with LSTMs [139]. Regarding the time series regression task, which is our focus in this work, genetic algorithms have occasionally been applied to some types of neural networks, including time-delay neural networks (TDNNs) [140], LSTMs [141, 142], and 1D CNNs [143]. However, while all these works use the general principle of genetic algorithms, they tailor their approaches to the specific network type chosen, and consequently, the hyperparameters to be optimized. Often, these choices are influenced by the application context. Given that our problem is still relatively unexplored, we had no preconceived ideas about the best model type to use, necessitating a more general approach. Therefore, we decided to implement a genetic architecture search algorithm for our time series regression task without specializing in any single network type, allowing us the flexibility to explore several possibilities, such as LSTMs and CNNs.

In this work, we introduce a genetic algorithm approach for searching architecture hyperparameters, designed to be applicable to various types of neural networks. We chose to apply it to LSTMs and CNNs in order to study the prediction of fMRI NF scores from EEG signals alone, using a global model approach, as opposed to subject-specific models. The goal of this approach is to reduce the reliance on the costly fMRI modality in a bi-modal neurofeedback context.

4.2 Materials

The data presented here were acquired, processed, and made available prior to my arrival in the Empenn team. The pseudonymized data are available in BIDS format on the OpenNeuro platform: `https://openneuro.org/datasets/ds002338`. OpenNeuro is an open science database dedicated to storing datasets from human brain imaging research studies, providing a free and open platform for sharing data.

This dataset consists of simultaneous EEG-fMRI acquisitions performed during a motor imagery neurofeedback task. It is described in [40] as the XP2 protocol and received approval by the Institutional Review Board. This section offers information about the participants, the experimental protocol, equipment, offline processing steps, and concludes with specifics regarding the computation of the NF scores used in our study. I want to clarify that all details provided in this section (4.2) are not a result of my own work, and are presented here only due to their relevance to the subsequent sections.

4.2.1 Participants and protocol

We used data collected from 15 healthy subjects included in the XP2 protocol, described below. The original study [144] involved 17 subjects, but two individuals (identified as sub-xp202 and sub-xp203) were excluded from our analysis due to a lack of BOLD data. All participants were right-handed and had never taken part in a neurofeedback session. Each participant provided signed informed consent, including consent for the publication of their anonymized data.

This protocol, illustrated in Figure 4.1, comprises a single session, featuring three motor imagery (MI) neurofeedback runs. Initially, a MIpre run, which is a run where the subject engages in motor imagery without receiving any feedback, was used to identify the region of interest (ROI) for fMRI processing. Then, three neurofeedback runs were performed with a one-minute break between each run. A single run consists of eight blocks alternating between 20 seconds of rest with eyes open and 20 seconds of motor imagery task involving the right hand, with visual feedback. The session concluded with a MIpost block without feedback to evaluate the participant's performance. Finally, within this protocol, seven subjects received unidimensional (1D) feedback, while the remaining eight were provided with bidimensional (2D) feedback. While a previous study conducted in the team in 2020 [126] showed that participant behavior differs between 1D and 2D

feedback, we included all 15 subjects in our analysis, following the approach in [4]. This was done to ensure an adequate amount of data given the already limited subject pool, under the assumption that this inclusion would not significantly impact the results.

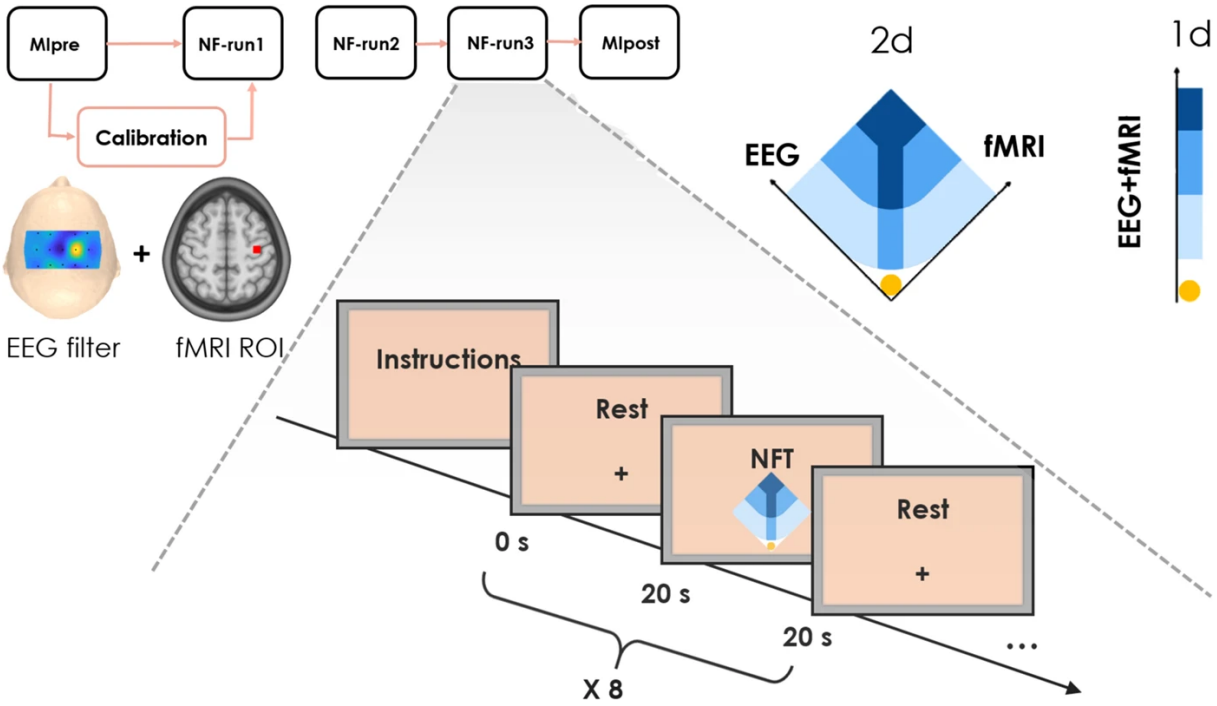


Figure 4.1 – **Experimental protocol XP2.** Figure provided by Giulia Lioi et al. in the article "Simultaneous EEG-fMRI during a neurofeedback task, a brain imaging dataset for multimodal data integration" [40].

4.2.2 Equipment

Data were obtained through a hybrid EEG-fMRI neurofeedback setup located at the Neurinfo platform (Rennes University Hospital, France), with detailed specifications provided in [42]. This platform facilitates EEG-fMRI acquisition, online processing, EEG-fMRI NF scores computation over time, and synchronisation before sending the visual feedback.

EEG data was recorded using a 64-channel extended international 10–20 EEG system solution from Brain Products (Brain Products GmbH, Gilching, Germany), which is MR-compatible. The signal was sampled at a rate of 5 kHz and a resolution of 0.5 µV, with FCz used as the reference electrode and AFz as the ground electrode.

fMRI acquisitions were conducted using a 3T Verio MRI running VB17 (Siemens Healthineers, Erlangen, Germany) and equipped with a 12-channel receiver head coil. The acquisitions were carried out using echo-planar imaging (EPI) and covered the upper half of the brain with the following parameters: TR = 1s, TE = 23ms, resolution: 2×2×4 mm^3, number of 4-mm slices: 16, no slice gap.

4.2.3 Offline processing

Detailed information about preprocessing can be found in [40], leading to a signal sampled at 200 Hz for the EEG modality. As for fMRI acquisitions, MIpre runs mentioned earlier were used in each session to conduct a first-level general linear model (GLM) analysis. The resulting activation maps, voxel-wise family-wise error corrected at $p < 0.05$, were used to define two regions of interest (ROIs), each measuring $9 \times 9 \times 3$ voxels, centered around the maximum activation in the primary motor area (M1) and the supplementary motor area (SMA), respectively.

4.2.4 NF scores computation

Since the NF scores shown to subjects in the original study [144] were not retained in a reusable format, the authors of [40] recalculated these scores for both modalities. The EEG NF scores were computed as a measure of event-related desynchronization (ERD), following the formula:

$$NF_{EEG}(t) = \frac{BP_{C3}(rest) - BP_{C3}(t)}{BP_{C3}(rest)}$$

where $BP_{C3}(t)$ represents the power in the 8–30 Hz frequency band of a Laplacian around C3 at time t and $BP_{C3}(rest)$ denotes the average power in the 8–30 Hz frequency band over the resting block preceding the neurofeedback training. $NF_{EEG}(t)$ quantifies the desynchronization occurring during motor imagery in relation to the baseline at rest. The EEG NF scores were converted into visual feedback every 250 ms, resulting in 1280 NF scores per run.

The fMRI NF scores were then calculated for M1 and SMA areas separately according

to the following formula:

$$NF_{fMRI}(t) = \frac{B_{ROI}(t)}{B_{ROI}(rest)} - \frac{B_{BG}(t)}{B_{BG}(rest)}$$

where $B_{ROI}(t)$ represents the fMRI signal in the ROI (M1 or SMA) selected during the calibration step at time t, divided by the corresponding signal averaged across the last 6 seconds of the preceding rest block. $B_{BG}(t)$ denotes the BOLD signal in a background lower slice, included to normalize by global BOLD signal changes. The fMRI NF scores were converted into visual feedback every second, resulting in 320 NF scores per run.

4.3 Methods

Our approach focuses on predicting fMRI neurofeedback scores from EEG signals. As illustrated in Figure 4.2, we operate in the context of bi-modal EEG-fMRI neurofeedback sessions. The long-term objective is to deploy this method during future neurofeedback sessions, where the model provides fMRI NF predictions without MRI acquisitions, to reduce its associated costs. The purpose of this section is to explain the construction of such a model. Firstly, we will describe the creation of the supervised learning dataset. Then, we will detail the search for the model architecture using a genetic algorithm. Finally, we will conclude by discussing the training and performance evaluation of the model.

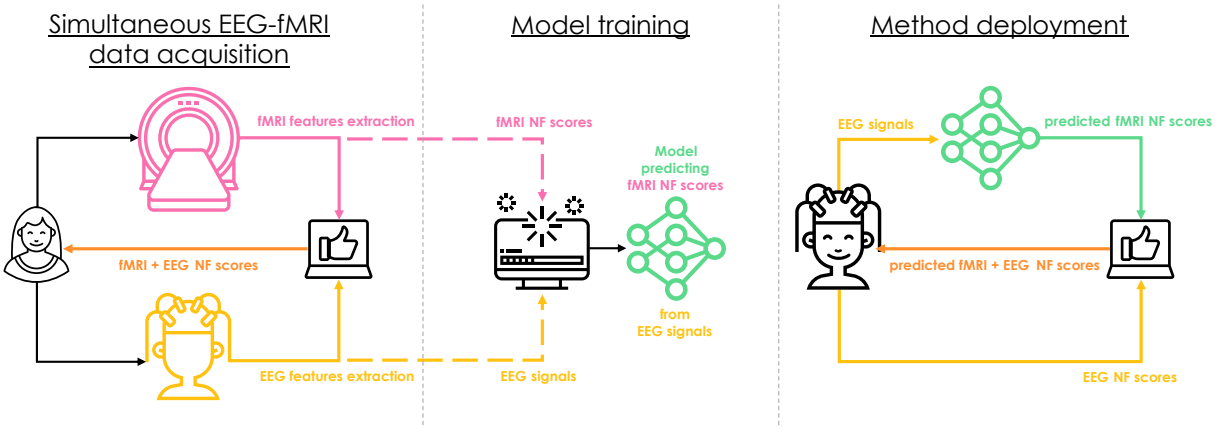


Figure 4.2 – **Illustration summarizing our approach and objectives for predicting fMRI neurofeedback scores from EEG signals.** Firstly, simultaneous EEG-fMRI data are acquired during neurofeedback sessions. This data is used to build a model capable of predicting fMRI NF scores from EEG signals alone. Our ultimate goal is to use the previous model learned from the bi-modal sessions to enhance the EEG unimodal sessions by proposing an improved NF score that incorporates the model fMRI predictions.

4.3.1 Formatting of the dataset

We have decided to set up two ways of generating our dataset in order to compare both approaches. Firstly, we created supervised learning samples directly from the raw signals by associating relevant parts of the EEG signal with the corresponding fMRI NF scores, which is the outcome we aim to predict. Alternatively, we extracted features from the raw EEG signals, potentially facilitating model learning, and similarly associated relevant segments with the corresponding fMRI NF scores.

4.3.1.1 From raw signals to supervised learning samples

Initially, we had raw EEG signals acquired with 64 channels at a sampling rate of 200 Hz over 320 seconds per run, resulting in 64,000 points per channel and per run. The first decision we made involved dimensionality reduction, selecting signals only from 25 channels referred to as motor electrodes ('F3', 'F4', 'C3', 'C4', 'Fz', 'Cz', 'FC1', 'FC2', 'CP1', 'CP2', 'FC5', 'FC6', 'CP5', 'CP6', 'F1', 'F2', 'C1', 'C2', 'FC3', 'FC4', 'CP3', 'CP4', 'C5', 'C6', 'CPz'), chosen for their proximity to the motor area under study (see Figure 4.3).

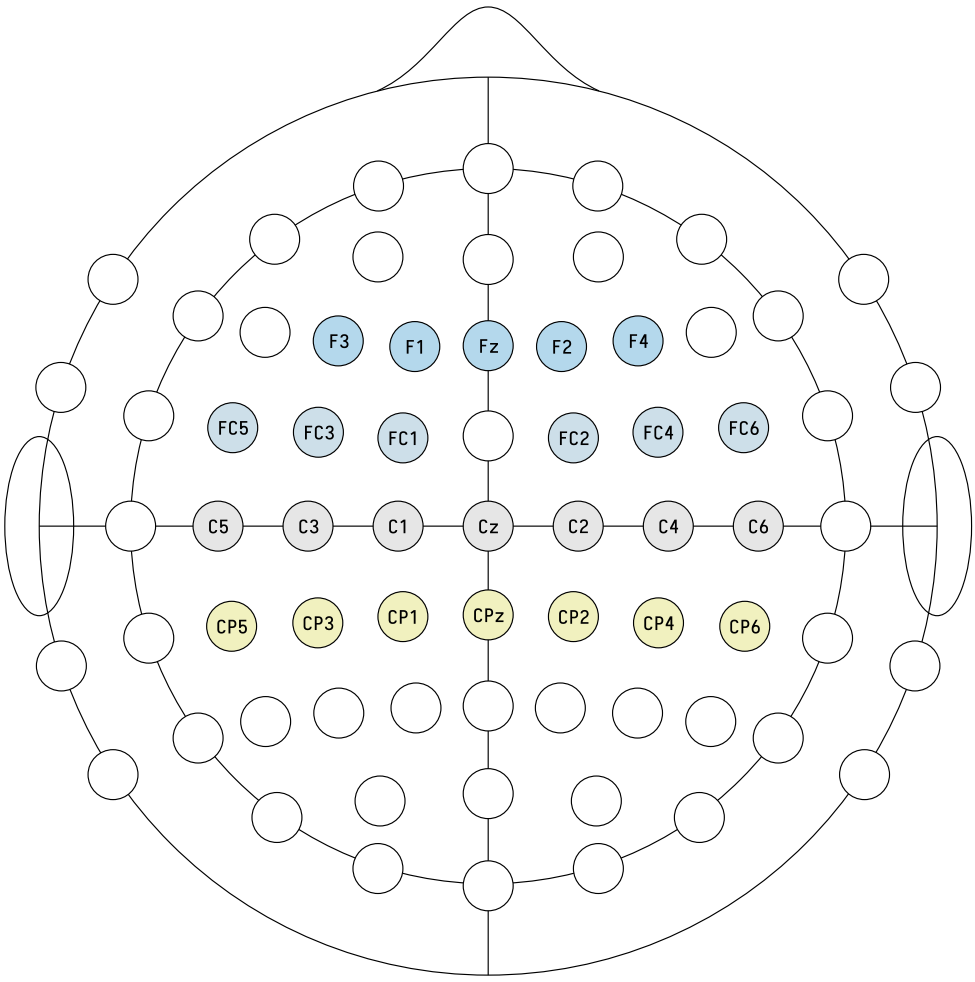


Figure 4.3 – **Channels selected for the creation of the samples.** The colored electrodes on the diagram correspond to the channels selected for creating our samples due to their proximity to the motor area under study. The electrode corresponding to the empty space between Cz and Fz, known as FCz, was not selected here as it was used as the reference electrode, as described in section 4.2.2.

To facilitate reproducibility of the study, I want to give extra details about where to find the fMRI NF scores that we used. The derivatives of our previously described data available on OpenNeuro include NF scores calculated for both the primary motor cortex area (M1) and the supplementary motor area (SMA). M1 is primarily responsible for the execution of voluntary movements, whereas SMA is involved in the planning and coordination of complex movements. Additionally, for each of these NF targets, both raw NF scores and smoothed NF scores are provided. Raw NF scores were calculated according to the formula presented in section 4.2.4, whereas the smoothed version of the NF scores was computed over the preceding three volumes. For this study, we decided to use the

raw NF scores calculated for the M1 area as the outcome we aim to predict.

We initially had 320 fMRI NF scores per run, representing 1 NF score per second. For the following steps, we increased this number to 1280 NF scores per run by linear interpolation, resulting in 4 NF scores per second. The reasoning behind this comes from the fact that the EEG NF scores have a size of 1280 per run. Therefore, having the same size for both fMRI and EEG NF scores will prove useful for the analysis in the results section 4.4. Furthermore, this idea of harmonizing towards 1280, rather than 320, serves as a form of data augmentation where we artificially increase the number of samples to be given to the model during training. We then proceeded with a calibration step. This involved retrieving the fMRI NF scores from the three runs of the same subject, calculating the 70th percentile for these three runs, dividing the scores by this value, and finally clipping the scores to be within the range of 0 to 1. This corresponds most closely to the real scores shown to the subjects during neurofeedback. In the following, this outcome we aim to predict will be referred to as true fMRI NF scores, as opposed to our predicted fMRI NF scores.

Next, we proceeded to create the supervised learning samples by associating each true fMRI NF score with a corresponding segment of the EEG data. Since fMRI NF scores are derived from the BOLD signal in the motor ROI, as well as from the signal averaged across the last 6 seconds of the previous rest block as explained in section 4.2.4, we decided to select the equivalent of 6 seconds of the EEG signal preceding the corresponding fMRI NF score, along with the last 6 seconds of the preceding rest block. Consequently, this selection does not allow for the creation of samples for the first 6 seconds, resulting in the exclusion of the initial 24 fMRI NF scores out of the total 1280 per run. This process is illustrated in Figure 4.4.

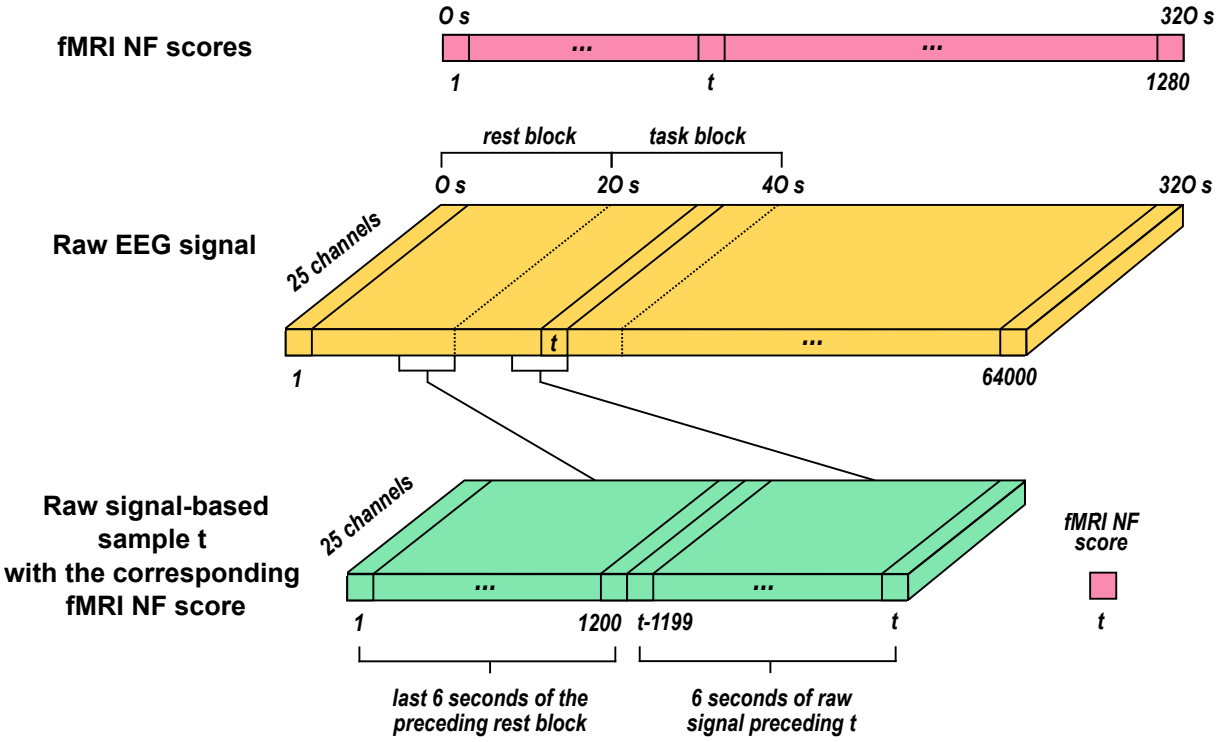


Figure 4.4 – **From raw signals to supervised learning samples.** The sample corresponding to the fMRI NF score at time t is created by combining the last 6 seconds of raw EEG signal from the preceding rest block with the 6 seconds of the raw EEG signal preceding t.

4.3.1.2 From extracted features to supervised learning samples

In this approach, we initially had the same materials at our disposal: the raw EEG signals from the 25 selected channels and the calibrated fMRI NF scores. The additional step involves extracting features from the EEG signal instead of using the raw signal. We chose to extract the bandpower in the alpha range (8-12 Hz) and the beta range (12-30 Hz) over a 2-second window with a shift of 0.05 seconds. In the previous approach, we used the channels as features, resulting in 25 features per time point. Here, we calculate both alpha and beta power bands for each of the 25 channels, resulting in a total of 50 features per time point. In the same manner, we then associated each fMRI NF score with the corresponding segment of the EEG data. As the previous approach, the segment consisted of the equivalent of 6 seconds of EEG bandpowers preceding the corresponding fMRI NF score, along with the last 6 seconds of bandpowers from the preceding rest block. In the end, this approach does not allow for the creation of samples for the first 6 seconds

either. Additionally, 2 more seconds are excluded due to the feature extraction window size, resulting in the exclusion of the initial 32 fMRI NF scores out of the total 1280 per run. This process is illustrated in Figure 4.5.

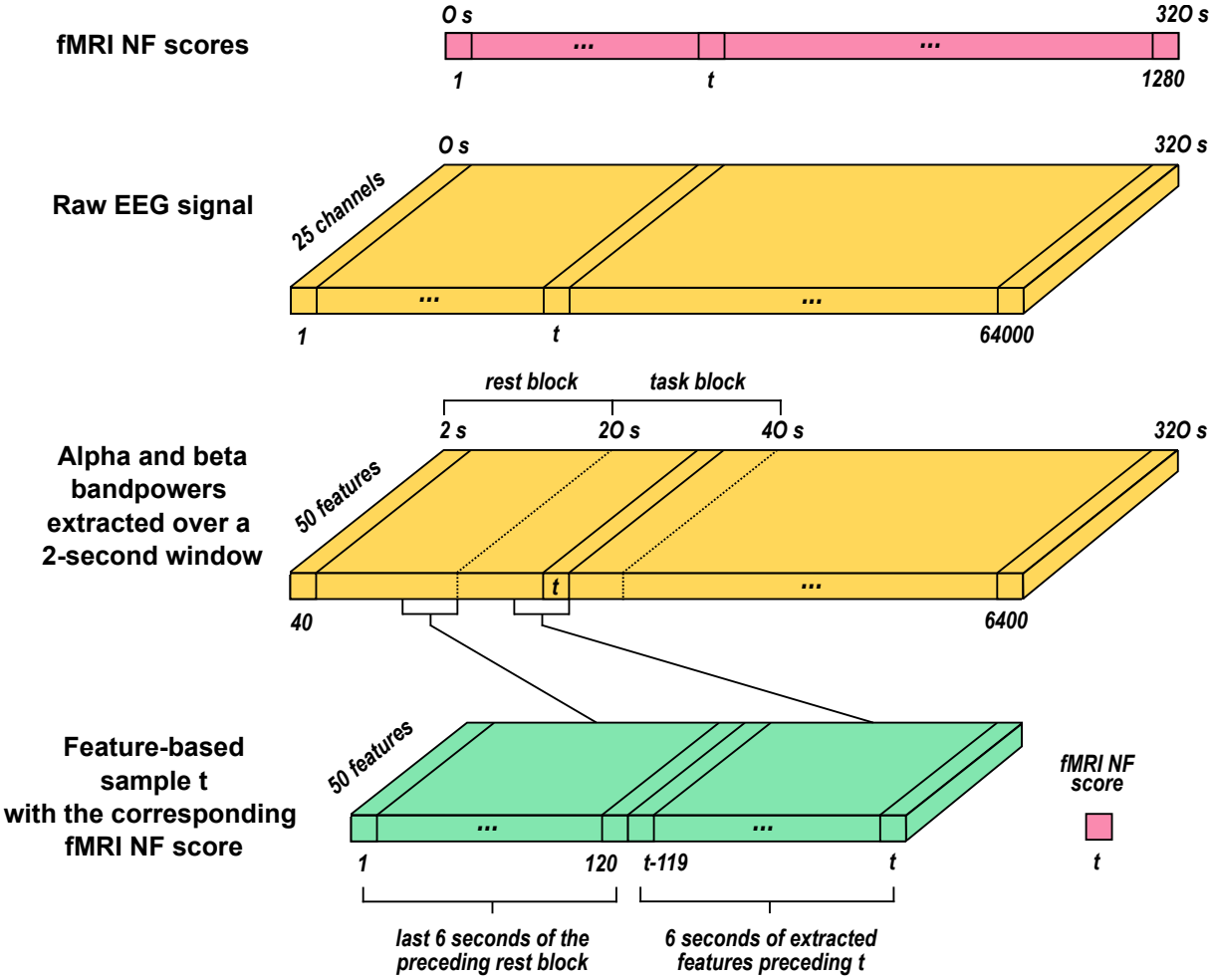


Figure 4.5 – **From extracted features to supervised learning samples.** Features are extracted by computing alpha (8-12 Hz) and beta (12-30 Hz) bandpowers over a 2-second window with a shift of 0.05 seconds. The sample corresponding to the fMRI NF score at time t is created by combining the last 6 seconds of extracted EEG features from the preceding rest block with the 6 seconds of extracted EEG features preceding t.

4.3.2 Genetic search for neural network architecture

We proposed a genetic algorithm framework with the aim of making it adaptable to several types of network architectures, such as CNNs or LSTMs. The population in the genetic search consists of individuals. An individual is a set of architecture hyperparameter values, which differs according to the type of network chosen. These hyperparameter values are used to build a model architecture, which is then evaluated to score the individual and ultimately select the best architecture. The full training and testing of the model based on the chosen network took place at a later and independent stage.

4.3.2.1 Genetic search algorithm overview

A genetic algorithm involves three main components: the initialization of the population, the scoring of individuals, and the evolution of the population. Our genetic search, outlined in Algorithm 1, takes into account a pre-selected set of hyperparameter values that depends on the type of architecture we aim to optimize. Additionally, it needs the data described above, a fixed number of individuals constituting the population, and finally, the number of generations to be processed. Detailed specifications are provided in section 4.3.2.5.

The genetic search begins by initializing the individuals, each corresponding to a set of hyperparameters which values are taken at random from the pre-selection, as described in Algorithm 2. For each generation, the algorithm iterates through the population, training a neural network model with hyperparameters defined by the current individual. The performance, referred to as score, of each individual is then assessed using the mean squared error (MSE) metric, as outlined in Algorithm 3. After evaluating all individuals in a generation, the population undergoes evolution, which involves processes such as the selection of the best individuals in this generation as parents, crossover to create offspring, and mutation of some hyperparameter values also known as genes, as detailed in Algorithm 4. This evolution creates a new population for the next generation. The loop continues until we reach the desired number of generations, and at the end of the process, the algorithm returns the individual with the best performance from the last generation.

Algorithm 1: Genetic search

Input: A pre-selected set of hyperparameter values to search for, the data, the number of individuals in the population, the number of generations.

Output: The best individual.

```
P_0 ← Initialize the individuals in the population as described in Algorithm 2.
for each generation n do
    for each individual i do
        Score_i ← Evaluate the individual's performance, as described in
         Algorithm 3.
    end
    if generation < number of generations then
        P_n ← Evolve the population, as described in Algorithm 4.
    end
end
Return the individual that has the best score out of the last generation.
```

4.3.2.2 Initialization of the population

For the initialization step, we need the set of hyperparameters and their pre-selected set of values, as well as the desired number of individuals within a population.

The initialization algorithm consists of assigning randomly those values to each newly created individual. In this manner, it iterates over each individual in the population and, for each individual, iterates over each hyperparameter. During this process, it selects a value at random for each hyperparameter from a pre-defined set of possible values. This random creation process provides some diversity in the initial population, which is the starting point of a broad exploration of the hyperparameters space. Once values for all hyperparameters are chosen for each individual, the algorithm returns the population for the first generation.

Algorithm 2: Initialization of the population

Input: The pre-selected set of hyperparameter values to search for, the number of individuals in the population.

Output: A population.

```
for each individual i do
    for each hyperparameter h do
        i_h ← Select a value at random from the pre-selection.
    end
end
Return the population consisting of the desired number of individuals.
```

4.3.2.3 Scoring of individuals

During this crucial step, the individuals taken as input undergo training and scoring using the same training, early stopping, and scoring datasets. These datasets consist of 13 subjects for training, 1 subject for early stopping, and 1 subject for scoring.

The method begins by creating a neural network architecture based on the hyperparameter values specified in the individual. Then, it proceeds to train it on the training dataset, using the early stopping dataset to mitigate underfitting and overfitting. Once training is done, the model is assessed on the scoring dataset by generating predictions and computing the mean squared error (MSE) with true fMRI NF scores. The mean of these errors, called a model score, serves as a quantitative measure of the individual's performance, reflecting how well the neural network, designed with the specified hyperparameters from the individual, can accurately predict the desired output. The lower the score, the better the individual's performance.

Algorithm 3: Scoring of individuals

Input: An individual i, the training dataset, the early stopping dataset, the scoring dataset.

Output: A score which represents the performance of the individual.

1 $Model_i \leftarrow$ Initialize the neural network architecture with hyperparameter values taken from individual i.
2 $Model_i \leftarrow$ Train the model with the training and early stopping datasets.
3 $Score_i \leftarrow$ Evaluate the model with the scoring dataset.
4 **Return** the MSE between fMRI NF predictions and true fMRI NF scores over the scoring dataset.

4.3.2.4 Evolution of the population

Finally, for the last major step, we once again need the set of hyperparameters, as well as the previous population whose individuals have been trained and evaluated.

The evolution process involves selecting the top n individuals from the previous population as parents. To introduce diversity, i new individuals are randomly created and added to the list of parents. Specifications are provided in section 4.3.2.5. These parents

are then included in the new population. To maintain a consistent number of individuals across generations, the population is supplemented with offspring generated from these parents. To create an offspring, two parents are chosen randomly, and a crossover process occurs, where the value of each hyperparameter is randomly selected from either parent. A small chance of mutation is then introduced, meaning there is a probability p that one hyperparameter value is replaced by a new one chosen at random from a pre-selected set of values. This combination of selection, crossover, and mutation aims to create a new population with a mix of well-performing individuals from the previous generation and potentially new individuals that explore different regions of the hyperparameter space. The algorithm concludes by returning the updated population, ready for the next generation of the genetic search.

Algorithm 4: Evolution of the population

Input: The pre-selected set of hyperparameter values to search for, the previous population.

Output: Updated population after performing selection, crossover, and mutation operations on the previous population.

1 $Parents \leftarrow$ Select the best individuals.
2 $Parents \leftarrow$ Add new individuals, randomly created in the same manner as outlined in Algorithm 2.
3 $Population \leftarrow$ Add parents to the new population.
4 **for** *each remaining space in the population* **do**
5 $\quad Offspring \leftarrow$ Take two distinct parents at random, then perform crossover by randomly selecting the value of each hyperparameter from one of the two parents.
6 $\quad Offspring \leftarrow$ Mutate with a probability p one hyperparameter value of the offspring.
7 $\quad Population \leftarrow$ Add the offspring to the new population.
8 **end**
9 **Return** the new population.

4.3.2.5 Implementation details

To begin with, we will describe the choices made for the inputs to the aforementioned algorithms. Firstly, an important step is to define the search space from which the genetic algorithm will select values to create individuals. The search space varies depending on the type of architecture being searched for.

- LSTM:
  - Number of layers: [1, 2, 3]
  - Number of nodes: [1, 2, 3, 4, 5, 10, 20, 30, 40, 50]
  - Number of neurons in the dense layer: [32, 64, 128, 256, 512]
  - Dropout: [0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]
  - Kernel regularizers value: [0.0005, 0.001, 0.005, 0.01]
- CNN:
  - Number of convolutional layers: [2, 3, 4]
  - Number of filters in the first layer: [16, 32, 64, 128]
  - Kernels size: [3, 9, 25, 65, 95, 125]
  - Number of neurons in the dense layer: [64, 128, 256, 512]
  - Spatial dropout: [0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]
  - Dropout: [0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]
  - Kernel regularizers value: [0.0005, 0.001, 0.005, 0.01]

These values are referred to as the pre-selection. For both architecture types, we excluded values that we empirically considered either too low or too high while maintaining a wide amplitude, such as dropout values of 0.1 and 0.9, and regularizer values below 0.0005 and above 0.01. Additionally, considering hardware limitations, we opted for reasonable values. For LSTMs, this included a maximum of 3 layers, a maximum of 50 nodes per layer, and a maximum of 512 neurons in the dense layer. For CNNs, we opted for a maximum of 4 convolutional layers, a maximum of 128 filters in the first layer, a kernel size no greater than 125, and a maximum of 512 neurons in the dense layer. It's important to note that the architecture search space was restricted to configurations achievable with the chosen pre-selected hyperparameters. LSTM models were constrained to have the same number of nodes for all layers. For CNNs, we chose a consistent pattern for the number of filters, doubling the number from the preceding layer. This rule applied starting with the first layer, which was the only layer under consideration in the search process.

Regarding the rest of the genetic search algorithm inputs, we reiterate that the datasets used in the genetic search comprised the three runs of 13 subjects for the training dataset, the three runs of one subject for the early stopping dataset, and the three runs of one subject for the scoring dataset. The creation of these samples has been detailed in section 4.3.1. Finally, we set the number of individuals in the population to 10, and the number of generations to 10. All input values were chosen expecting a balance between having ample research depth and accommodating hardware limitations.

Secondly, regarding the training hyperparameters, we fixed the maximum number of epochs at 300 and implemented early stopping with a patience of 10 and restoration of the best weights. The batch size was set to 32, the training loss function used was mean squared error, and the Adam optimizer was used with an initial learning rate of 1e-05.

Finally, regarding the choices made for the evolution of the population, given that we fixed the number of individuals in the population to 10, we chose to keep the top 2 individuals as parents and introduce 1 randomly created individual into the parents' list. Consequently, to achieve the final population size of 10 individuals, 7 offspring were generated, each with a 20% probability of mutation occurring in a randomly selected hyperparameter value.

4.3.3 Post-genetic model for fMRI NF scores prediction

4.3.3.1 Post-genetic model training

Once the best architecture design was selected, we proceeded with the final phase: model training. From the preceding step, we retained only the architecture hyperparameters of the individual with the best performance. We did not retain the weights of the trained models, as genetic search and model training were designed to be separate phases. For training hyperparameters, we used similar values to those employed during the genetic search phase. Early stopping was implemented with a patience of 10 and restoration of the best weights, the batch size was set to 32, the training loss function used was mean squared error, and the Adam optimizer was used with an initial learning rate of 1e-05. However, we fixed the maximum number of epochs at 500 to allow more freedom for the models to converge. This phase was less time-consuming than the genetic search phase since we only had to train 15 models, as explained in the next section, compared to the 100 models (i.e., 10 generations with 10 models each) in the genetic search phase.

4.3.3.2 Post-genetic model evaluation

To evaluate our method, we used the selected neural network architecture and conducted 15 training processes on different data arrangements, resulting in 15 different models trained on the same architecture. Notably, the partition used during the genetic search corresponds to the 15th fold of the model evaluation conducted here.

Fold ID	**Test**	**Early stopping**	**Training**
Fold 1	sub-xp201	sub-xp204	all 13 others
Fold 2	sub-xp204	sub-xp205	all 13 others
Fold 3	sub-xp205	sub-xp206	all 13 others
Fold 4	sub-xp206	sub-xp207	all 13 others
Fold 5	sub-xp207	sub-xp210	all 13 others
Fold 6	sub-xp210	sub-xp211	all 13 others
Fold 7	sub-xp211	sub-xp213	all 13 others
Fold 8	sub-xp213	sub-xp216	all 13 others
Fold 9	sub-xp216	sub-xp217	all 13 others
Fold 10	sub-xp217	sub-xp218	all 13 others
Fold 11	sub-xp218	sub-xp219	all 13 others
Fold 12	sub-xp219	sub-xp220	all 13 others
Fold 13	sub-xp220	sub-xp221	all 13 others
Fold 14	sub-xp221	sub-xp222	all 13 others
Fold 15	sub-xp222	sub-xp201	all 13 others

Table 4.1 – **Details of the composition of the 15 folds used.** Each fold consists of three datasets: training, early stopping, and test. The three runs from the same subject are always grouped together within the same dataset.

Our data consists of 15 subjects, each with 3 NF runs of 1256 samples for the raw signal-based approach and 1248 samples for the extracted features-based approach. Using these subjects, we performed 15 different permutations, referred to as folds, in a leave-one-subject-out cross validation manner. Each fold corresponds to a different partitioning where one subject is used to test the model and evaluate its performance, another one to apply an early stopping strategy to avoid overfitting, and the rest to learn the model weights. They are respectively referred to as the test, early stopping, and training datasets.

So, in fold i, the test dataset consisted of the 3 runs of subject i, the early stopping dataset of the 3 runs of subject $i+1$, and the training dataset of the 3 runs of all the other subjects. The detailed list of permutations is provided in Table 4.1. Such an approach facilitated the training and evaluation of 15 models built with the same architecture, enhancing the robustness of our method's performance assessment.

Finally, to assess the performance of these models, we predicted the fMRI NF scores for each sample of each test run and compared them with true fMRI NF scores using mean squared error (MSE). The average error for these test runs represent the performance of each of the 15 trained models, and the average of these performances represents the overall performance of our method. Then, to assess the applicability of our method in real-life conditions, as presented in the deployment section of Figure 4.2, we proceed to the next step. Our ultimate goal is to generate NF scores that integrate both our fMRI NF predictions and EEG NF scores, aiming to approximate true bi-modal EEG-fMRI NF scores without fMRI acquisitions. Therefore, we combined both our predictions and true fMRI NF scores with the calibrated EEG NF scores (presented in section 4.2.4 and calibrated in the same way as fMRI NF scores in section 4.3.1). This combination is achieved by computing the mean between the fMRI NF scores and the EEG NF scores. Since both have been calibrated between 0 and 1, the combined scores also fall within this range. Then, we calculated the error between the fMRI NF predictions averaged with EEG NF scores and the true bi-modal EEG-fMRI NF scores. It is worth noting that the resulting average errors will be lower than the MSE between predicted and true fMRI NF scores since EEG NF scores are added on both sides. To be exact, this operation is equivalent to dividing those MSEs by 4. The purpose of this step is to obtain an estimate of the quality of this new virtually bi-modal NF score. Since our fMRI NF predictions are intended to enhance EEG NF scores, their combination should exhibit a lower error with the true EEG-fMRI NF scores than EEG-NF scores alone for us to validate our method.

4.4 Results

4.4.1 Results overview

As the results of this contribution are substantial and quite detailed, we begin with an overview to clarify the structure of this section. We will present the results of the models created using two types of neural network architectures: LSTM and 1D CNN. These models were trained using two different data preparation approaches described in section 4.3.1: extracted features-based samples and raw signal-based samples. We start by summarizing and illustrating the performances corresponding to the fMRI NF predictions of these four configurations in Table 4.2 and Figure 4.6. Then, we will examine the final results, which incorporate EEG NF scores, in Table 4.3.

To begin with, Table 4.2 allows us to assess the performance of the four configurations using two metrics: mean squared error (MSE) and Pearson's correlation. MSE is the main metric used here, while Pearson's correlation offers additional insights into the shape of the predictions. The values presented here correspond to the error and correlation between the fMRI NF predictions and the true fMRI NF scores, representing model performance directly at the output.

In summary, the mean MSEs between fMRI NF predictions and true fMRI NF scores across the four configurations are fairly similar, but it is the CNN architecture with extracted features samples that stands out with the lowest value (0.1586). Regarding Pearson's correlations, the values are overall very low. While correlation isn't necessarily the best metric in this case, given that both predictions and true scores can feature frequent spikes and fluctuations, it serves as a useful secondary metric. It allows us to evaluate the appearance of the predictions, which is an important aspect when judging the quality of the results, as we will see in the following. We note that the two CNN configurations perform slightly better in this regard, with correlations around 0.1, while the LSTM configurations show correlations close to zero.

Configuration	**Mean MSE between predicted and true fMRI NF scores**	**Mean correlation between predicted and true fMRI NF scores**
LSTM extracted features samples	0.1673 (±0.0177)	0.0418 (±0.1114)
LSTM raw signal samples	0.1707 (±0.0560)	-0.0162 (±0.0572)
CNN extracted features samples	0.1586 (±0.0212)	0.1022 (±0.1956)
CNN raw signal samples	0.1692 (±0.0299)	0.1151 (±0.2283)

Table 4.2 – **Comparison of fMRI NF predictions across all configurations.** First column: Mean MSE between fMRI NF predictions versus true fMRI NF scores (corresponding to the pink (left) boxplot in the following figures). Second column: Mean Pearson's correlation between fMRI NF predictions versus true fMRI NF scores.

Then, we summarize the performance of our four configurations side by side in Figure 4.6. At first glance, the differences between the configurations are not obvious. We begin by comparing the neural network types: for LSTMs, the difference between the extracted features approach and the raw signal approach is not significant. However, for CNNs, the extracted features approach performs significantly better than the raw signal approach. Next, we compare network types within each approach: for the raw signal approach, the difference between LSTM and CNN is not significant, while for the extracted features approach, CNN significantly outperforms LSTM. Further details will be provided in the following sections.

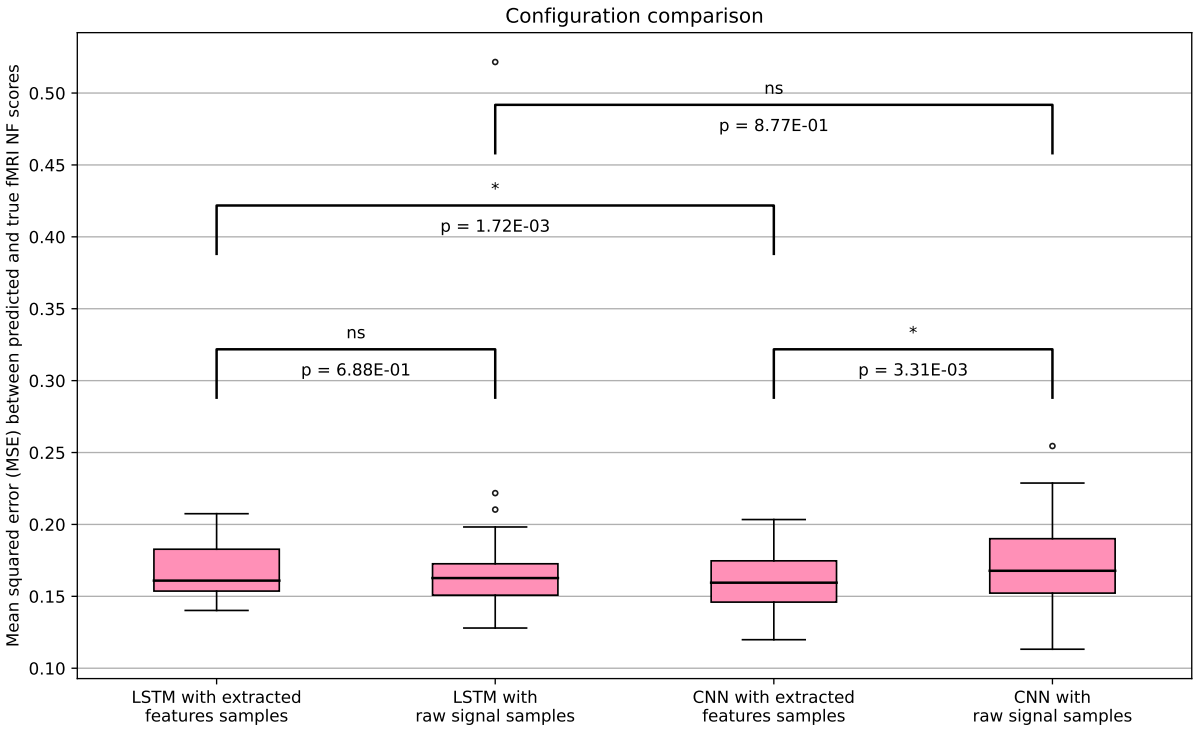


Figure 4.6 – **Results for all test subjects across all folds using the mean squared error (MSE) metric for all configurations.** Each boxplot represent the MSE between the predictions of fMRI NF scores directly from the models and true fMRI NF scores. The p-value from a paired t-test is displayed for key comparisons (*: significant, ns: not significant).

Next, Table 4.3 presents what we refer to as the final results, compared against two baselines (n°1 and n°2) to be surpassed, which allow us to evaluate the success of the experiments. The principle behind these final results, described in section 4.3.3.2, consists of combining both our predictions and true fMRI NF scores with the corresponding EEG NF scores, following the goal of approximating true bi-modal EEG-fMRI NF scores without requiring fMRI acquisitions. The values presented here thus correspond to the mean MSE between fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores. These values do not provide any additional information compared with the Table 4.2 (as mentioned earlier, this operation is equivalent to dividing those values by 4), but they are necessary for comparison with the two baselines.

The first baseline (noted baseline n°1), also introduced in section 4.3.3.2, corresponds to the mean MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores. The purpose of this comparison is to verify if our fMRI NF predictions are actually enhancing

EEG NF scores. Ideally, the combination should yield a lower error when compared to the true EEG-fMRI NF scores than using EEG NF scores alone.

The second baseline (noted baseline n°2) represents the mean MSE between the mean of true fMRI scores averaged with the EEG NF scores versus true bi-modal EEG-fMRI NF scores. The idea is to mimic a perfectly centered flat prediction. In fact, the calculation is close to the variance of true fMRI NF scores (with the combination with EEG NF scores), but we have chosen to explain it this way to connect it with what we are trying to verify. Indeed, the purpose of this second comparison is to verify if our fMRI NF predictions are more accurate than simply predicting the mean of the true fMRI scores. The story of why we decided to look at our results with baseline n°2 will be told in the following.

So, let's analyse Table 4.3. We first observe that baseline n°1 and n°2 values differ slightly between the extracted features and raw signal categories. Theoretically, these values should be identical. This difference can be explained by the dataset formatting process, as presented in section 4.3.1. The number of samples created is not exactly the same (with 8 fewer samples per run for the extracted features approach), meaning that the true fMRI NF scores are also of slightly different sizes. This explains the minor variation between the means.

When comparing our final results with baseline n°1, we find that all configurations outperform it, symbolizing that adding our fMRI NF predictions to EEG NF scores brings us significantly closer to the true bi-modal EEG-fMRI NF scores compared to using EEG NF scores alone. This is excellent news, as it suggests that our models effectively enhance EEG neurofeedback by incorporating fMRI predictions. However, as we will see in section 4.4.3, the LSTM configuration using raw signal-based samples unexpectedly produced flat-looking predictions, often centered around the mean of the true fMRI NF scores. Even more surprisingly, the mean MSE between these predictions and the true scores was very close to that of the LSTM models using extracted features-based samples, which produced predictions with greater amplitude. This observation raised the question of whether merely surpassing baseline n°1 is sufficient for our method to be considered satisfactory.

Therefore, we turn our attention to baseline n°2. Here, we observe that none of the four configurations manage to outperform it. As we will see in the next sections, the LSTM with raw signal approach and the CNN with extracted features approach have

lower performance than baseline n°2, but not significantly so (with a p-value of 0.0882 and 0.2039 respectively (details provided in their respective sections), which exceeds the 0.05 threshold, indicating that we cannot reject the null hypothesis of identical average scores). The comparison with baseline n°1 showed us that we were indeed predicting something interesting from the EEG signals alone, despite considerable differences between the two modalities. However, these predictions were in fact no better than simply predicting the average of the true fMRI NF scores. With models as complex as those in the deep learning category, one might expect to produce predictions more accurate than simply predicting the average of the true fMRI NF scores, which is why these results are ultimately unsatisfactory.

Configuration	**Mean MSE for final results**	**Baseline n°1 (MSE)**	**Baseline n°2 (MSE)**
LSTM extracted features samples	0.0418 (±0.0044)	0.0829 (±0.0198)	0.0384 (±0.0037)
LSTM raw signal samples	0.0427 (±0.0140)	0.0831 (±0.0196)	0.0388 (±0.0037)
CNN extracted features samples	0.0397 (±0.0053)	0.0829 (±0.0198)	0.0384 (±0.0037)
CNN raw signal samples	0.0423 (±0.0075)	0.0831 (±0.0196)	0.0388 (±0.0037)

Table 4.3 – **Comparison of final results across all configurations.** First column: Mean MSE between fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the yellow (center left) boxplot in the following figures). Second column: Mean MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores (corresponding to the purple (center right) boxplot in the following figures), referred to as baseline n°1. Third column: Mean MSE between the mean of true fMRI scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the blue (right) boxplot in the following figures), referred to as baseline n°2.

Despite the extensive modeling performed, the results are thus not satisfactory. Since baseline n°2 was not surpassed, we consider that these global models (designed to be applicable to all participants, as opposed to individualized models) cannot be used for now in clinical settings to predict fMRI NF scores from EEG signals. Nevertheless, we will take the time to thoroughly analyze them, in order to derive valuable insights for future research on this topic. In the next four sections, we will examine and discuss the results of

each configuration presented in Tables 4.2 and 4.3. For each section, we will provide: (1) an illustration of the architecture identified through genetic search, (2) learning curves for the 15 trained folds, (3) overall performance represented by boxplots using the mean squared error (MSE) metric, (4) prediction examples showcasing the highest and lowest errors across all folds, and (5) an analysis figure of the relationship between the quality of predictions and the quality of the EEG input signals.

4.4.2 LSTM model with extracted features samples as inputs

This section presents the results of applying our method to the LSTM network type using extracted features-based input data. After running the genetic algorithm for hyperparameter optimization, the architecture hyperparameters found are as follows: one LSTM layer with 40 units, followed by a dense layer with 256 neurons. Regularization was applied to both the LSTM and dense layers with a value of 0.001, and the dropout rate for the dense layer was set at 0.3. An illustration of this network is provided in Figure 4.7.

The use of only one LSTM layer, combined with a relatively large number of units, suggests a focus on capturing simple patterns in the input sequence. The dense layer, with its substantial number of neurons, contains the majority of the model's weights, and could be prone to overfitting. However, the regularization applied, while moderate, helps mitigate this risk by penalizing large weights. Additionally, the modest dropout rate in the dense layer further reduces the likelihood of overfitting by randomly setting a fraction of the layer's output units to zero during training. The relatively low regularization and dropout values indicate that significant overfitting mitigation was not necessary, which is typical for one-layer models like this one. This suggests that during the genetic search, this simpler network might have identified trends in the data better than bigger models, possibly because the input features were pre-extracted, simplifying the task. This interpretation will be contrasted with the results from raw signal-based samples in the next section.

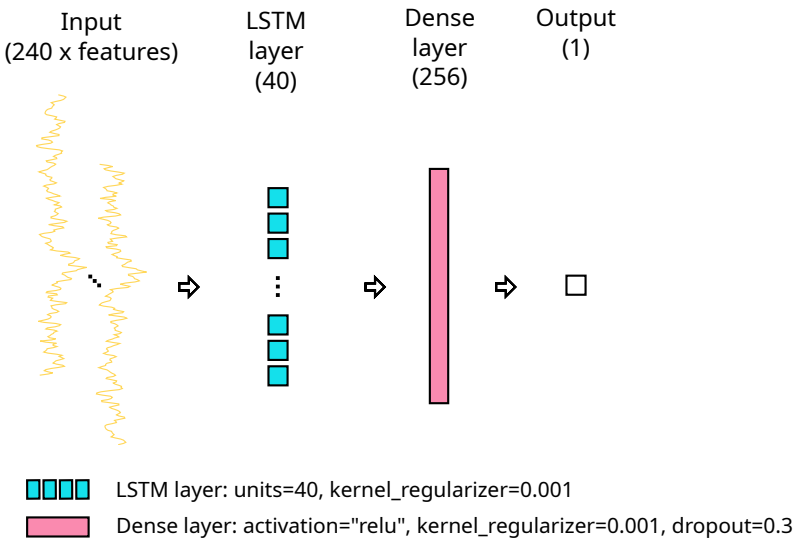


Figure 4.7 – **Architecture of the LSTM found through the genetic search, using extracted features samples as input.**

Then, as described in section 4.3.3.2, we trained 15 different models using the selected architecture by permuting the data within the training, early stopping, and testing datasets to achieve a more robust evaluation of our method. Figure 4.8 displays the 15 learning curves, depicting the loss calculated on both the training and early stopping datasets. The learning curve generated from the training dataset provides insight into how well the model learns, while the curve derived from the early stopping dataset, which is not used for updating weights, assesses the model's generalization ability. This validation curve allows us to halt the learning process before overfitting occurs, thanks to the early stopping mechanism.

The figure shows a general trend of good convergence between the training and validation loss curves across the 15 folds. However, in some instances, the validation loss is noisier and does not decrease as much as the training loss. This can be attributed to several factors. Firstly, the fact that the validation loss is higher than the training loss may suggest that, while the model fits the training data well, it faces some challenges in generalizing to unseen data in the validation set. It typically indicates early signs of overfitting. Secondly, the noise in the validation loss could be due to variability in the data distribution between the training and validation datasets. Since the data is permuted across different folds, some validation subsets might include more challenging samples. This case demonstrates the importance of early stopping, which stops training before the model diverges too far from the equilibrium.

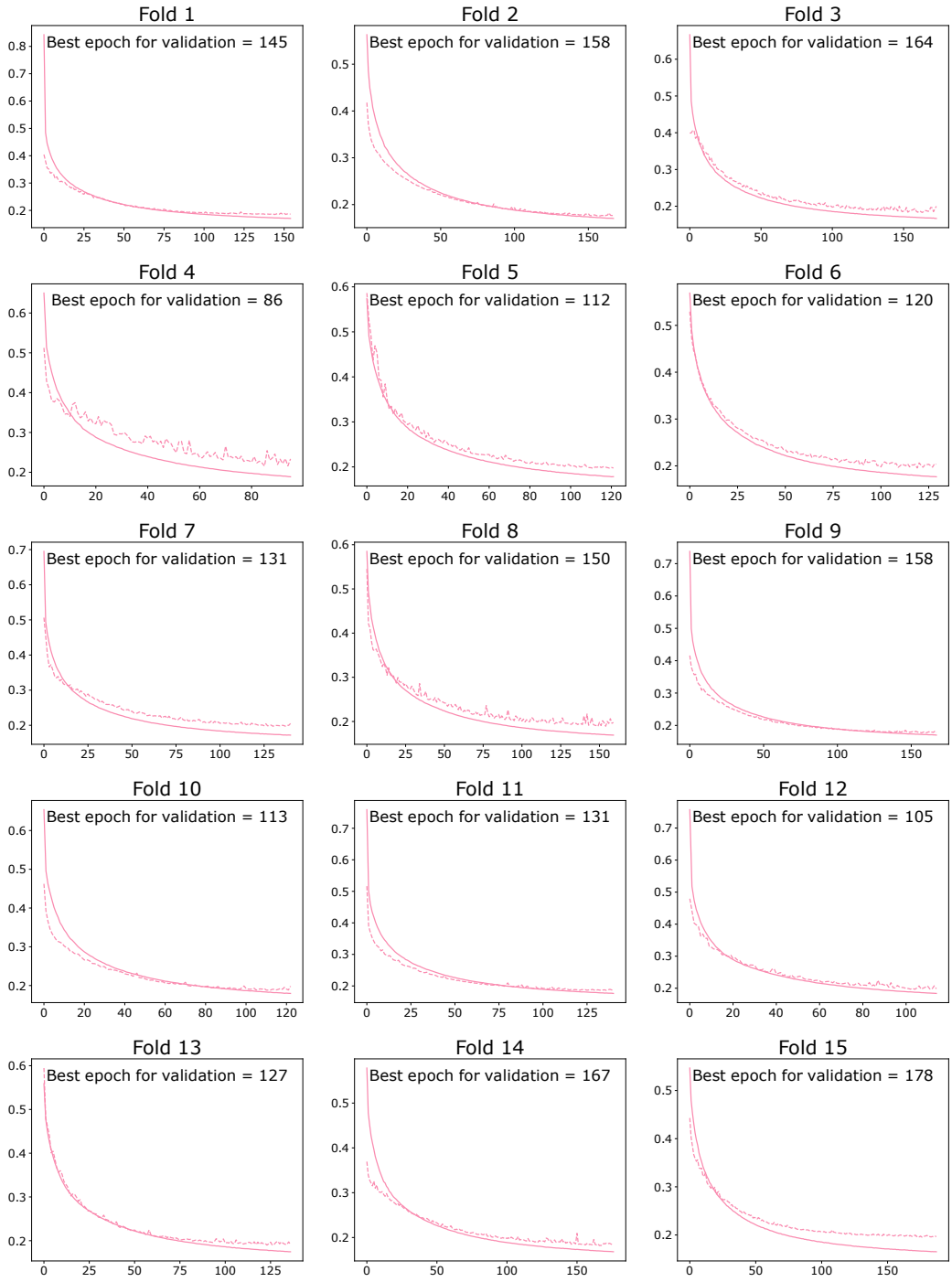


Figure 4.8 – **Learning curves for the 15 folds of the LSTM approach with extracted features samples as input.** The training loss is shown as a solid line, while the validation loss is displayed as a dotted line. The abscissa shows the number of epochs, and the ordinate shows the loss value (MSE). The best epoch in terms of early stopping (i.e., validation) loss is indicated. As an early stopping strategy with patience and restoration of the best weights is employed, this number indicates the number of epochs for which each model was trained.

Now, we present the comprehensive results following the complete application of our method. As previously explained, we consider the predictions from all 15 trained models on their respective test subjects. Figure 4.9 shows four key comparisons using the mean squared error (MSE) metric. Firstly, we directly compare the predicted fMRI NF scores to the true fMRI NF scores (corresponding to the pink (left) boxplot), providing a direct evaluation of model performance. Secondly, we compare our fMRI NF predictions averaged with EEG NF scores to the true bi-modal EEG-fMRI NF scores (corresponding to the yellow (center left) boxplot). While adding the same EEG NF scores on both sides naturally reduces the MSE, it allows us to evaluate the predictions in a context more relevant to our goal of improving unimodal EEG neurofeedback sessions. To evaluate this final result, we include baseline n°1 (corresponding to the purple (center right) boxplot), representing the MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores, which has the purpose of verifying if our predictions averaged with EEG NF scores align more closely with the true bi-modal EEG-fMRI NF scores than the EEG NF scores alone. Lastly, we provide baseline n°2 (corresponding to the blue (right) boxplot), representing the MSE between the mean of true fMRI scores averaged with EEG NF scores versus the true bi-modal EEG-fMRI NF scores, which has the purpose of verifying if our fMRI NF predictions are more accurate than simply predicting the mean of the true fMRI scores.

Firstly, we look at the predictions from the models: the mean MSE between predicted fMRI NF scores and true fMRI NF scores (pink boxplot) is $0.1673(\pm 0.0177)$. Secondly, our final results: the mean MSE between fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (yellow boxplot) is $0.0418(\pm 0.0044)$. These MSE values are not interpretable on their own, which is why we compare the final results with baseline n°1 and n°2. For baseline n°1 (purple boxplot), the mean MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores is $0.0829(\pm 0.0198)$. By having a significantly ($p = 4.12e-18 < 0.05$ using a paired t-test) lower MSE when the predictions are averaged with EEG NF scores compared to EEG-only NF scores, the goal of enhancing EEG neurofeedback with predicted fMRI NF scores is reached. What is predicted, even if not perfect, seems to carry information that was not in the EEG NF scores alone. However, for baseline n°2 (blue boxplot), the mean MSE between the mean of true fMRI scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores is $0.0384(\pm 0.0037)$. Our predictions averaged with EEG NF scores are thus significantly ($p = 2.07e-4 < 0.05$ using a paired t-test) less accurate than the mean of true fMRI scores averaged with EEG NF scores.

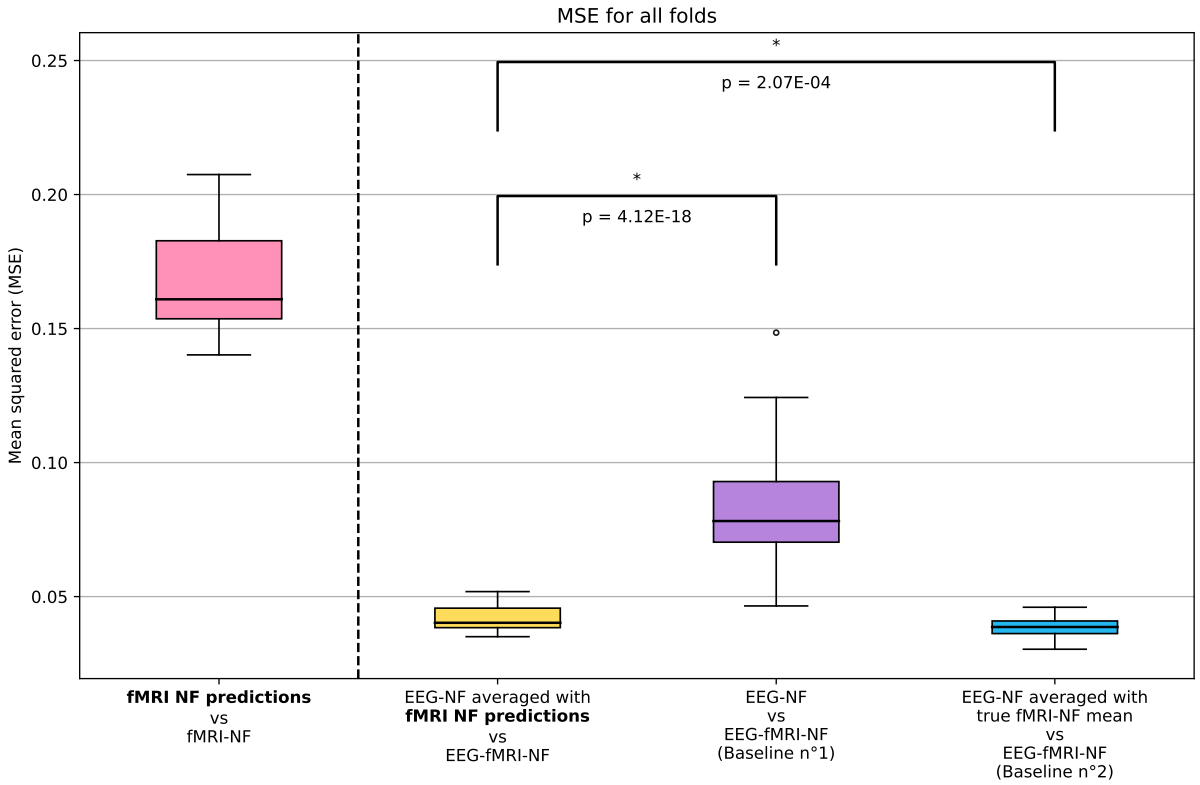


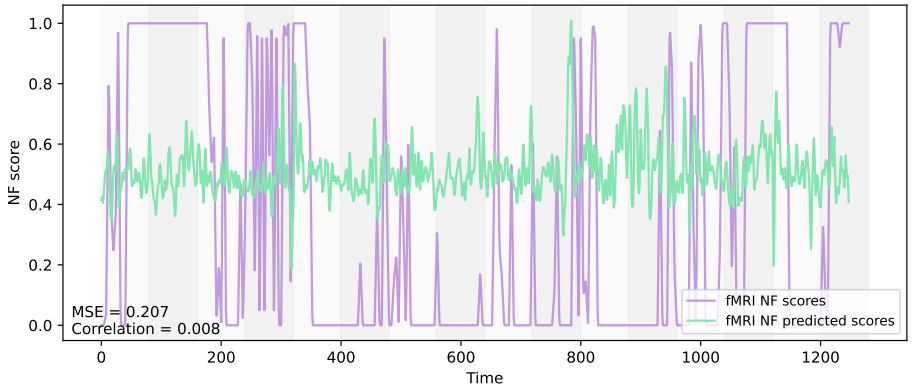
Figure 4.9 – **Results for all test subjects across all folds using the mean squared error (MSE) metric for the LSTM with extracted features samples as input.** From left to right: MSE between the predictions of fMRI NF scores directly from the models and true fMRI NF scores (pink). MSE between fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (yellow). MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores, representing baseline n°1 (purple). MSE between the mean of true fMRI NF scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores, representing baseline n°2 (blue).

To better comprehend the results, we provide examples of predictions made using these models. We selected predictions from subjects with the highest and lowest mean squared error (MSE) in comparison with true fMRI NF scores across all folds. Figure 4.10 illustrates the comparison between the predicted and true fMRI NF scores for sub-xp213 run 2, which represents the highest error, and sub-xp218 run 3, which represents the lowest error.

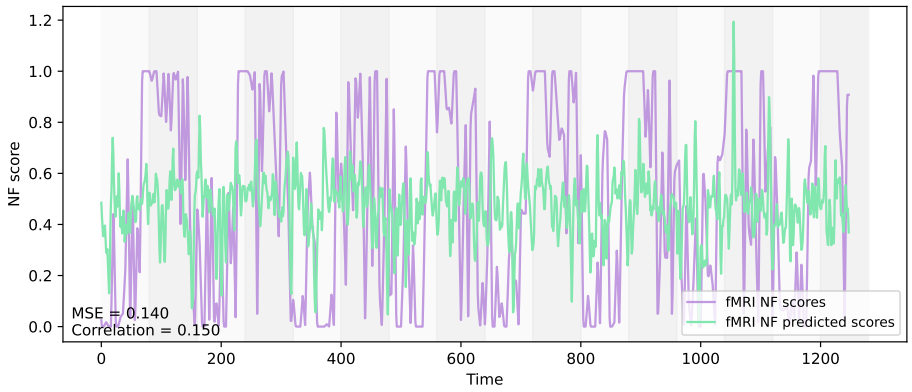
In the example with the highest MSE of 0.207 (a), it is noticeable that the true fMRI NF scores exhibit an irregular pattern, not quite following the expected rest/task alternation. The predictions are spiky but centered. It appears that the model struggled to accurately predict these scores. The irregularity of the true fMRI NF scores may have

contributed to the model's difficulty in generalizing to this run, leading to this high MSE value.

In contrast, for the lowest MSE example (b), the true fMRI NF scores follow more clearly the expected rest/task trend. The model's predictions, though still very spiky, seem to follow a bit better the true fMRI NF scores, resulting in a lower MSE of 0.140. We could think that the closer adherence to the expected trend in the true fMRI NF scores made it easier for the model to generalize to this run, leading to a more accurate performance.



(a) Prediction for sub-xp213 run 2



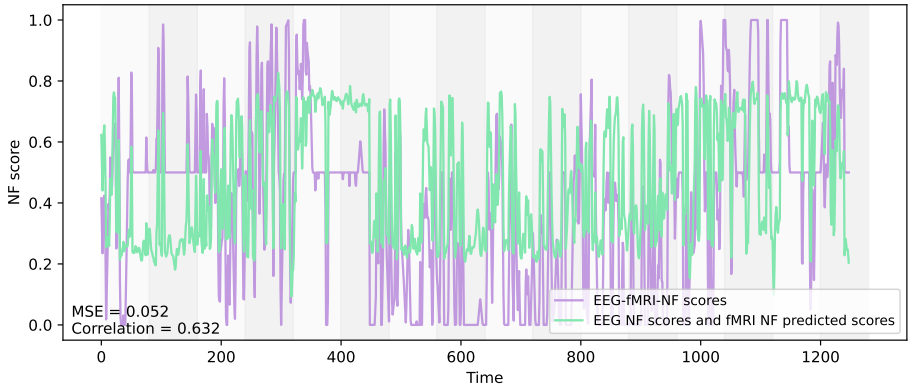
(b) Prediction for sub-xp218 run 3

Figure 4.10 – **Examples of predictions made using an LSTM model with extracted features samples as input.** The green lines represent the model predictions, while the purple lines represent true fMRI NF scores. (a) illustrates the comparison between the prediction and the true scores for sub-xp213 run 2, representing the highest error across all folds. (b) illustrates the comparison between the prediction and the true scores for sub-xp218 run 3, representing the lowest error across all folds.

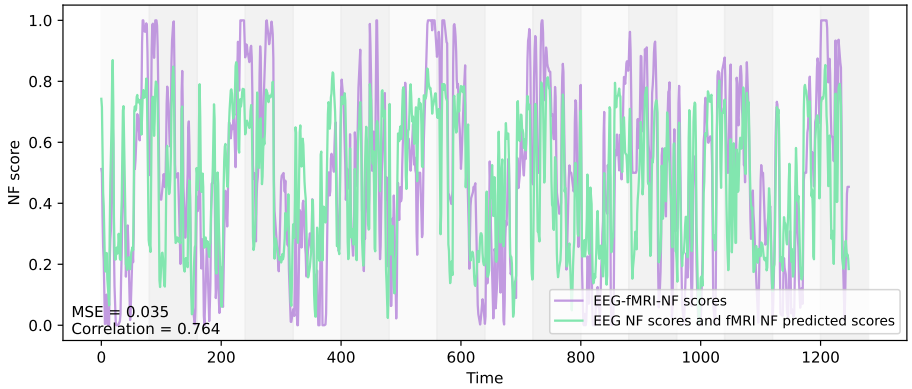
As described in section 4.3.3.2, we combined the EEG NF scores to the fMRI NF predictions to compare them with the true bi-modal EEG-fMRI NF scores, in order to get closer to the practical use of this method. To continue the illustration, Figure 4.11 presents the predictions for the same subjects used in Figure 4.10, averaged with the calibrated EEG NF scores, for comparison with the true bi-modal EEG-fMRI NF scores.

For the highest MSE example (a), the true bi-modal EEG-fMRI NF scores exhibit an irregular pattern, deviating significantly from the expected rest/task alternation. The presence of a few plateaus at 0.5 suggests that while one modality had an NF score of 1, the other had an NF score of 0, which is not ideal. This discrepancy, where EEG NF scores sometimes oppose the true fMRI NF scores, could indicate a lower quality of EEG signals. We will explore this possibility with the next figure. The model's predictions in this case somewhat follow the true scores, mainly due to the addition of EEG NF scores on both sides, but exhibit clear offsets and numerous spikes, leading to a mean MSE of 0.052.

For the lowest MSE example (b) with a value of 0.035, the true bi-modal EEG-fMRI NF scores follow more closely the expected rest/task trend, though they remain somewhat spiky, but to a reasonable degree. Here, the model's predictions align more closely with the true scores, capturing both the overall trend and amplitude more effectively. This better alignment could suggest that when the true bi-modal scores have a clearer rest/task pattern, the model is better at generalizing and producing accurate predictions. However, this is not exactly our goal, as we aim for the model to predict accurate fMRI NF scores regardless of whether the participant performed well during the rest and task blocks or struggled more. The next figure will investigate this idea further.



(a) Prediction for sub-xp213 run 2



(b) Prediction for sub-xp218 run 3

Figure 4.11 – **Examples of final results made using an LSTM model with extracted features samples as input.** The green lines represent the model's fMRI NF predictions averaged with EEG NF scores, while the purple lines represent the true bi-modal EEG-fMRI NF scores.

So, to further analyze our results, we aim to understand why our method performs differently across subjects. To address this question, we hypothesize that the quality of the EEG signal might be a significant factor contributing to these differences. Other sources of variation might include the participant's affinity for one modality over the other during the bi-modal session (e.g., if the participant is less responsive to fMRI, predicting random scores becomes challenging). We focus on the quality of the EEG signal since it serves as the input for our models. We assume that: if a participant has EEG NF scores that follow the expected rest/task trend, it is because the participant is responding well to neurofeedback and that the EEG signal captures well this information, and therefore is of

good quality. So, to evaluate the quality of EEG NF scores, reflecting the quality of our input EEG signals, we used the t-statistic measure between the task and rest values of these scores. A high t-statistic indicates that NF scores during task blocks are significantly higher than those during rest blocks, suggesting a strong neurofeedback response from the participant and good signal quality. Conversely, a t-statistic value below zero means that the participant responded better to neurofeedback during rest than during the task, implying that the EEG signal might not be of high quality and/or that the participant did not understand the instructions (e.g., they might be thinking of the task during rest). Figure 4.12 shows the performance of our fMRI NF predictions against the t-statistic calculated on the EEG NF scores of the corresponding run for each fold (i.e., 1 fold is 1 subject with 3 runs tested).

Firstly, we look at the t-statistic between the task and rest blocks of the EEG NF scores (abscissa). This analysis will be valid for all four sections, as the same values of EEG NF scores are used each time. We observe that all test runs across all folds have t-statistics ranging from approximately -10 to 25. Although it's difficult to define an exact threshold at which a run is considered to have a "good" rest/task trend in general, we can at least observe that the vast majority of runs have a positive t-statistic. However, a substantial number are still close to zero, indicating a more lukewarm neurofeedback response. Finally, sub-xp211 appears to be a clear outlier, with very negative t-statistics for 2 of its runs and a t-statistic close to 0 for the last run.

Then, regarding the MSE between predicted and true fMRI NF scores (ordinate), we observe that values range from approximately 0.14 to 0.21. The lowest MSE example run that we observed earlier, from sub-xp218, indeed corresponds to a better t-statistic value (t-stat ≈ 15) than the highest MSE run (t-stat ≈ 5). A trend could be observed in the $0-15$ t-statistic range, where higher MSEs are closer to a t-statistic value of 0 and lower MSEs are associated with increasing t-statistics. However, the correlation coefficient value of -0.085 does not allow us to conclude that a higher t-statistic is systematically linked to a lower MSE. In fact, some runs with t-statistics in the 20-25 range have some of the highest MSEs. Therefore, in this section, it appears that the quality of the EEG NF scores cannot be considered as a factor contributing to the difference of performance between runs.

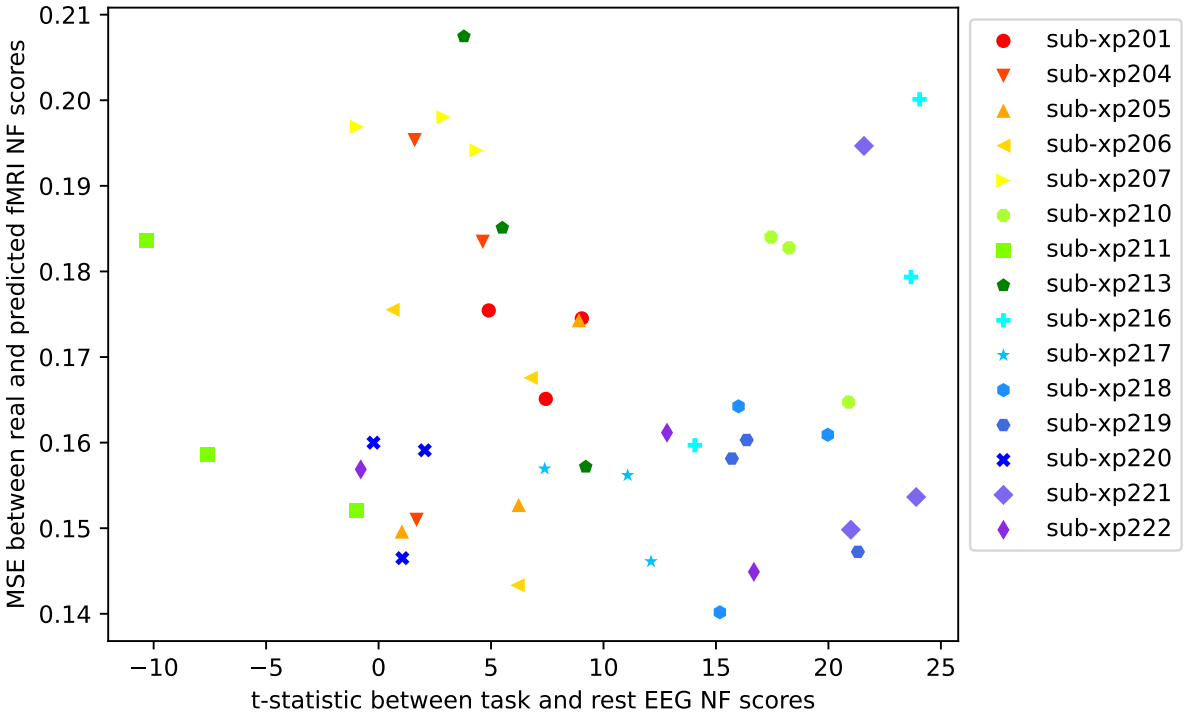


Figure 4.12 – **Analysis for the LSTM approach with extracted features samples as input.** Mean squared error (MSE) between fMRI NF predictions and true fMRI NF scores, in contrast to the t-statistic between task and rest blocks of the EEG NF scores which indirectly represents the quality of the EEG signal, for all test subjects across all folds. Each test subject with its 3 runs is represented by 3 points of different shape and color. The correlation coefficient between the two variables is −0.085.

4.4.3 LSTM model with raw signal samples as inputs

This section presents the result of our method applied to the LSTM network type with raw signal-based input data. After running the genetic algorithm to search for architecture hyperparameters, the architecture found includes three LSTM layers with 4 units each, followed by a dense layer containing 512 neurons. Kernel regularizers applied to the LSTM and dense layers were set to 0.01. Finally, the dropout rate for the dense layer was set at 0.2. An illustration of this network is available in Figure 4.13.

The architecture identified by the genetic algorithm is notably more complex than the one used for extracted features inputs. The presence of three LSTM layers provides a deeper network structure designed to capture more intricate patterns within the raw signal inputs. However, having only 4 units per layer might be considered a bit too small. The

dense layer, containing 512 neurons, offers significant capacity for processing the output of the LSTM layers. While this substantial number of neurons contributes to a more powerful model, it also increases the risk of overfitting. To mitigate this risk, the regularization applied to both the LSTM and dense layers is relatively high. However, the dropout rate in the dense layer is surprisingly modest. The similar performances shown in Table 4.2 for the raw signal and the extracted features approach suggests that although this model is more complex, it may face greater challenges in extracting meaningful patterns directly from the raw signal inputs. It could also indicate that the model is working harder to achieve a similar level of performance as the simpler model that uses extracted features inputs.

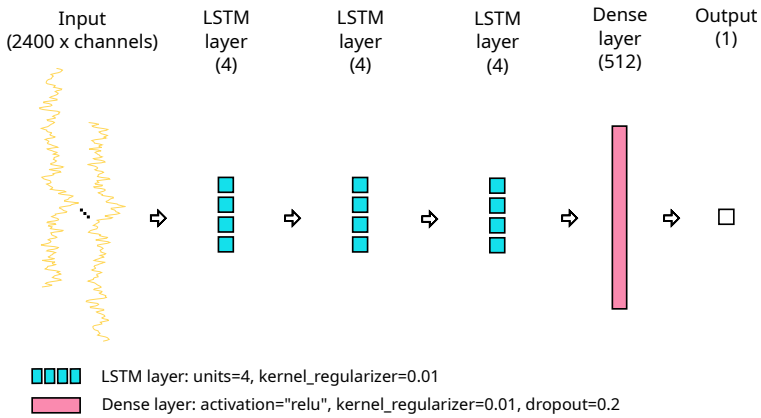


Figure 4.13 – **Architecture of the LSTM found through the genetic search, using raw signal samples as input.**

Figure 4.14 shows the learning curves for the LSTM models trained on raw signal inputs, based on the same evaluation method described previously. As before, these curves illustrate both the training and validation losses across 15 models, providing insights into the model's learning and generalization.

The curves here present an interesting observation: in some instances, the validation loss is unexpectedly lower than the training loss. This behavior, though less common, can be attributed to several factors. Firstly, it might suggest that the regularization techniques, such as dropout or kernel regularizers, are having a strong impact on the training process. Since regularization is only applied during training, it could result in a higher training loss compared to the validation. Secondly, as before, some of the validation subsets may contain samples that are inherently easier to predict than those in the training data. Overall, this may indicate that the data is complex and challenging to model.

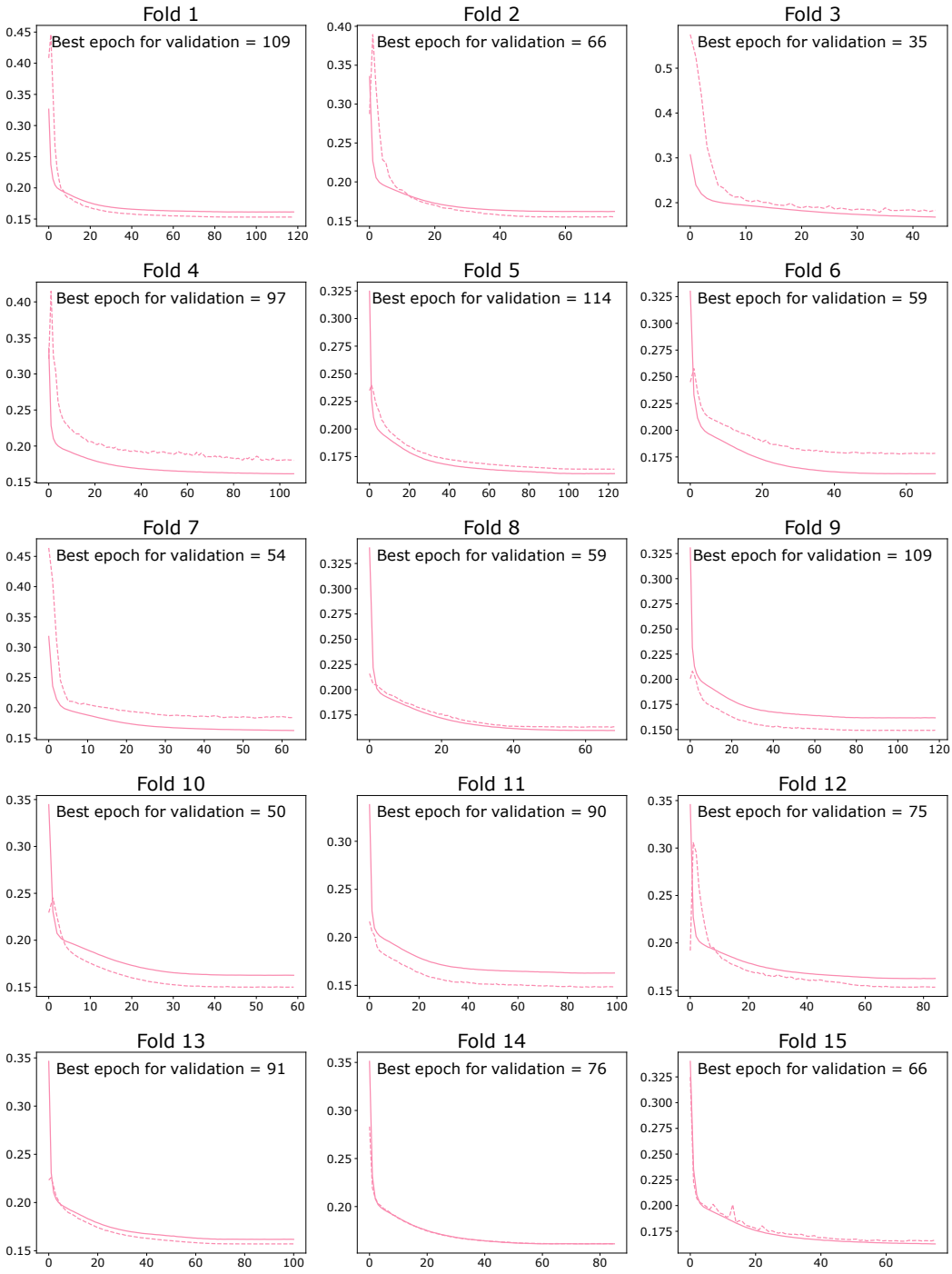


Figure 4.14 – **Learning curves for the 15 folds of the LSTM approach with raw signal samples as input.** The training loss is shown as a solid line, while the validation loss is displayed as a dotted line. The abscissa shows the number of epochs, and the ordinate shows the loss value (MSE). The best epoch in terms of early stopping (i.e., validation) loss is indicated. As an early stopping strategy with patience and restoration of the best weights is employed, this number indicates the number of epochs for which each model was trained.

Same as the previous section, Figure 4.15 presents four key comparisons using mean squared error (MSE), based on predictions from all 15 trained models on their respective test subjects. It includes predicted fMRI NF scores versus true fMRI NF scores (corresponding to the pink (left) boxplot), fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the yellow (center left) boxplot), EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the purple (center right) boxplot, also referred to as baseline n°1), and true fMRI NF scores means averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the blue (right) boxplot, also referred to as baseline n°2).

Firstly, we look at the predictions from the models: the mean MSE between predicted fMRI NF scores and true fMRI NF scores (pink boxplot) is 0.1707($\pm$0.0560). Secondly, our final results: the mean MSE between fMRI NF predictions averaged with EEG NF scores and true bi-modal EEG-fMRI NF scores (yellow boxplot) is 0.0427($\pm$0.0140). Thirdly, for baseline n°1: the mean MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores (purple boxplot) is 0.0831($\pm$0.0196). Despite slightly higher MSE values for predictions and final results compared to the extracted features approach, the predictions averaged with EEG NF scores still significantly ($p = 4.17e-15 < 0.05$ using a paired t-test) outperform baseline n°1. Fourthly, for baseline n°2 (blue boxplot), the mean MSE between the mean of true fMRI scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores is 0.0388($\pm$0.0037). Here, as the p-value computed using a paired t-test is $p = 0.0882 > 0.05$, we cannot reject the null hypothesis of identical average scores, meaning that our predictions averaged with EEG NF scores have a similar accuracy as the mean of true fMRI scores averaged with EEG NF scores.

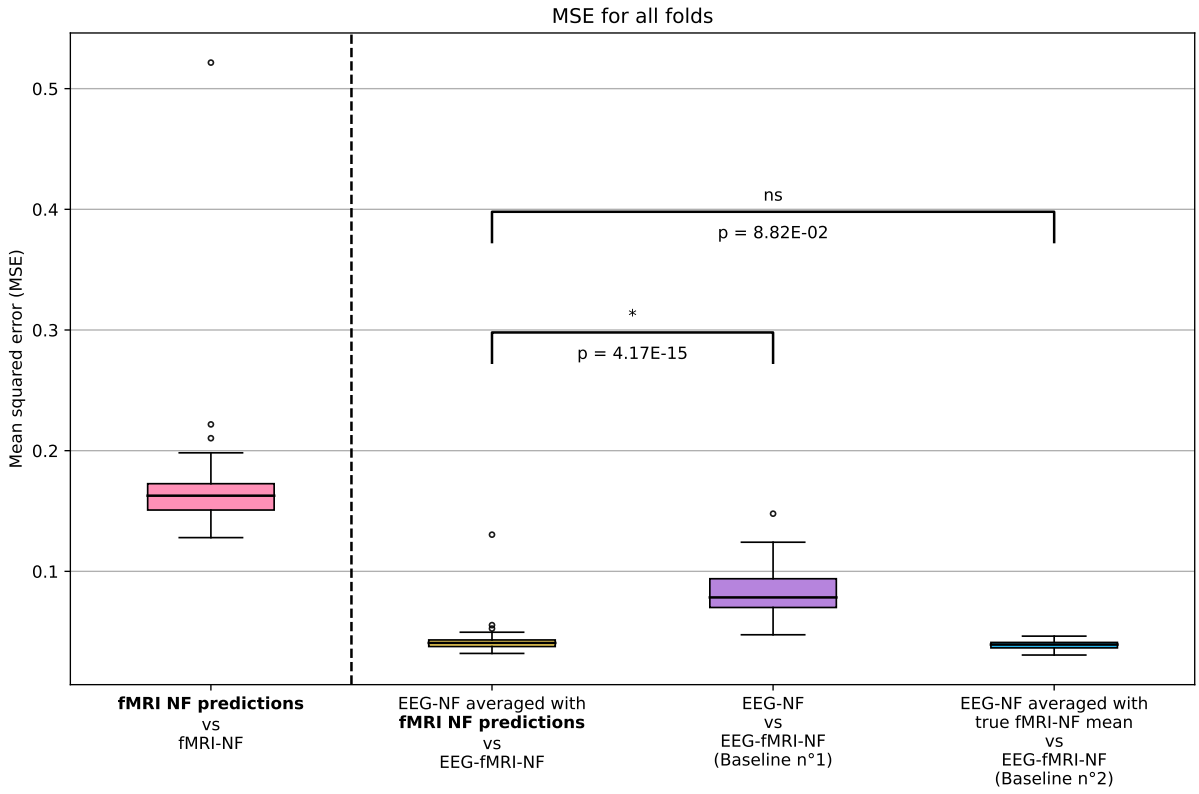
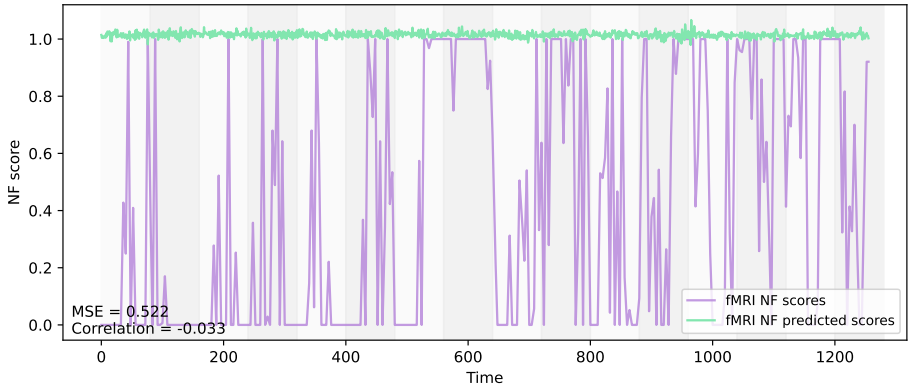


Figure 4.15 – **Results for all test subjects across all folds using the mean squared error (MSE) metric for the LSTM with raw signal samples as input.** From left to right: MSE between the predictions of fMRI NF scores directly from the models and true fMRI NF scores (pink). MSE between fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (yellow). MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores, representing baseline n°1 (purple). MSE between the mean of true fMRI NF scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores, representing baseline n°2 (blue).

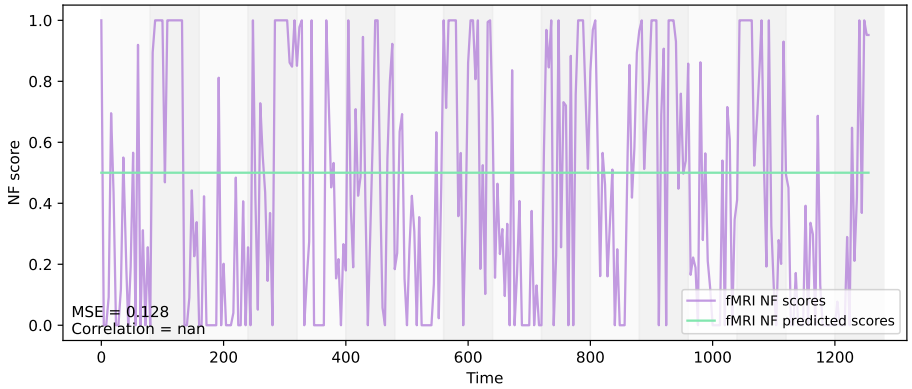
To better comprehend the results, we provide examples of predictions made using these models. We selected predictions from subjects with the highest and lowest mean squared error (MSE) in comparison with true fMRI NF scores across all folds. Figure 4.16 illustrates the comparison between the predicted and true fMRI NF scores for sub-xp206 run 3, which represents the highest error, and sub-xp204 run 3, which represents the lowest error.

For the highest MSE example (a), the true fMRI NF scores exhibit an irregular pattern, similarly to the previous section. Very surprisingly, the model's predictions are almost flat, with only small variations, and a noticeable offset from the mean of the true scores. This result has a mean MSE of 0.522, marking it as a clear outlier.

For the lowest MSE example (b) with a value of 0.128, although the true fMRI NF scores approximately follow the expected rest/task trend at certain points, they remain quite spiky. Even more surprisingly, the model's predictions are also flat, but they are more accurately centered around the mean of the true scores. This alignment with the mean suggests some basic understanding of the overall trend. However, the fact that the average MSE over all folds with this approach is very similar to the extracted features approach, despite the very different-looking results, raises questions. Indeed, it is disturbing that these flat predictions, which "play it safe", have the same average error as previous predictions that attempted high and low predicted values. To tell the story, it was this observation that gave us the idea of looking at the results with baseline n°2 presented in section 4.4.1.



(a) Prediction for sub-xp206 run 3



(b) Prediction for sub-xp204 run 3

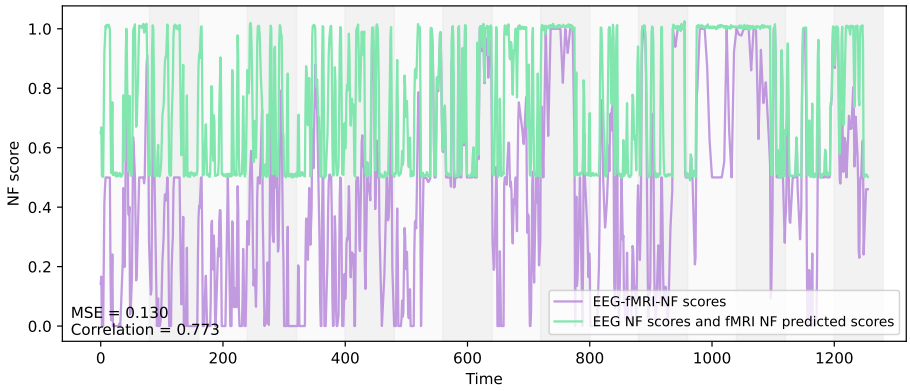
Figure 4.16 – **Examples of predictions made using an LSTM model with raw signal samples as input.** The green lines represent the model predictions, while the purple lines represent true fMRI NF scores. (a) illustrates the comparison between the prediction and the true scores for sub-xp206 run 3, representing the highest error across all folds. (b) illustrates the comparison between the prediction and the true scores for sub-xp204 run 3, representing the lowest error across all folds.

As described in section 4.3.3.2, we combined the EEG NF scores to the fMRI NF predictions to compare them with the true bi-modal EEG-fMRI NF scores, in order to get closer to the practical use of this method. To continue the illustration, Figure 4.17 presents the predictions for the same subjects used in Figure 4.16, averaged with the calibrated EEG NF scores, for comparison with the true bi-modal EEG-fMRI NF scores.

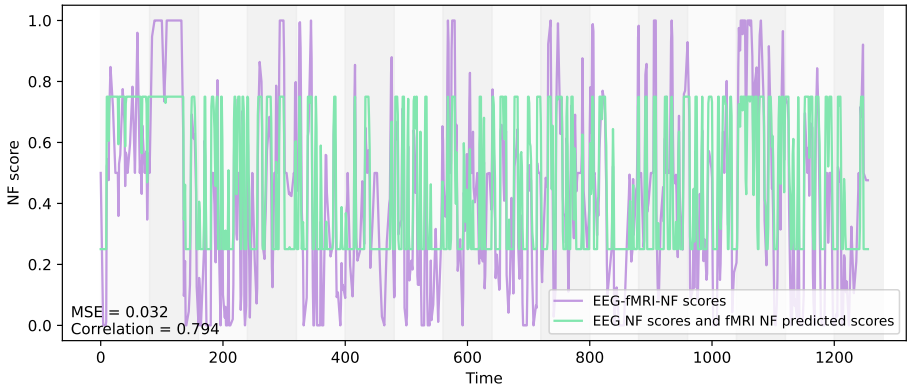
For the highest MSE example (a), the true bi-modal EEG-fMRI NF scores also do not exhibit a clear rest/task trend. We can see very clearly that the final result here represents the EEG NF scores averaged with the flat fMRI prediction. This outcome shows the trend

of the EEG NF scores along with the significant offset from the fMRI NF prediction. This leads to poor alignment with the true scores, resulting in the section's highest mean MSE of 0.130.

For the lowest MSE example (b), the true bi-modal EEG-fMRI NF scores follow the rest/task trend slightly better. In this case, it is visible again that the final result is the EEG NF scores averaged with a flat fMRI NF prediction. However, here, the flat fMRI prediction does not exhibit a significant offset, resulting in an outcome that is better-centered around the true scores. Although the overall result initially appears to lack the desired accuracy and amplitude, it has a mean MSE of 0.032, which is very similar to the previous section's lowest MSE of 0.035.



(a) Prediction for sub-xp206 run 3



(b) Prediction for sub-xp204 run 3

Figure 4.17 – **Examples of final results made using an LSTM model with raw signal samples as input.** The green lines represent the model's fMRI NF predictions averaged with EEG NF scores, while the purple lines represent the true bi-modal EEG-fMRI NF scores.

Finally, we investigate why our method performs differently across subjects. As previously mentioned, a high t-statistic indicates that NF scores during task blocks are significantly higher than those during rest blocks, suggesting a strong neurofeedback response from the participant and good signal quality. Conversely, a t-statistic value below zero indicates that the participant responded better to neurofeedback during rest than during the task, which could imply lower EEG signal quality and/or a misunderstanding of the instructions. Figure 4.18 shows the performance of fMRI NF predictions compared to the t-statistic calculated on the EEG NF scores of the corresponding run for each fold (i.e., 1 fold is 1 subject with 3 runs tested).

The results regarding the t-statistic between the task and rest blocks of the EEG NF scores are identical to those from the previous section, as the same values of EEG NF scores are used. In summary, the vast majority of runs have a positive t-statistic, although a significant number are close to zero, and sub-xp211 stands out as a clear outlier.

Now, regarding the MSE between predicted and true fMRI NF scores, we observe a surprising cluster between approximately 0.12 and 0.23, with a single outlier at 0.52 corresponding to the run shown in the example figure above. As seen in the examples, these LSTM models using raw signal-based samples tend to result in flat predictions that are often centered around the mean of the true fMRI NF scores. This suggests that the model may struggle to predict the variations in the true fMRI NF scores. Instead, it appears to default to safer, less variable predictions, which could be a consequence of the model's difficulty in extracting meaningful information from the raw signal-based EEG inputs. One possible explanation is that the LSTM architecture identified through the genetic search, especially the use of only 4 units per layer, might not be sufficient to handle the variability present in raw EEG signals. This limitation could lead to overly generalized predictions. The next question, then, is why the genetic search resulted in such an architecture. As we saw earlier, the overall performance in terms of MSE is almost the same as in the previous section, indicating that this "safe" approach is as effective as the previous models, which had more variation in their predictions. So, during the genetic search, this architecture which produces those flat predictions was likely selected as a parent due to its good mean MSE, and then was "safe" enough so that any other architecture attempting more variations could not outperform it.

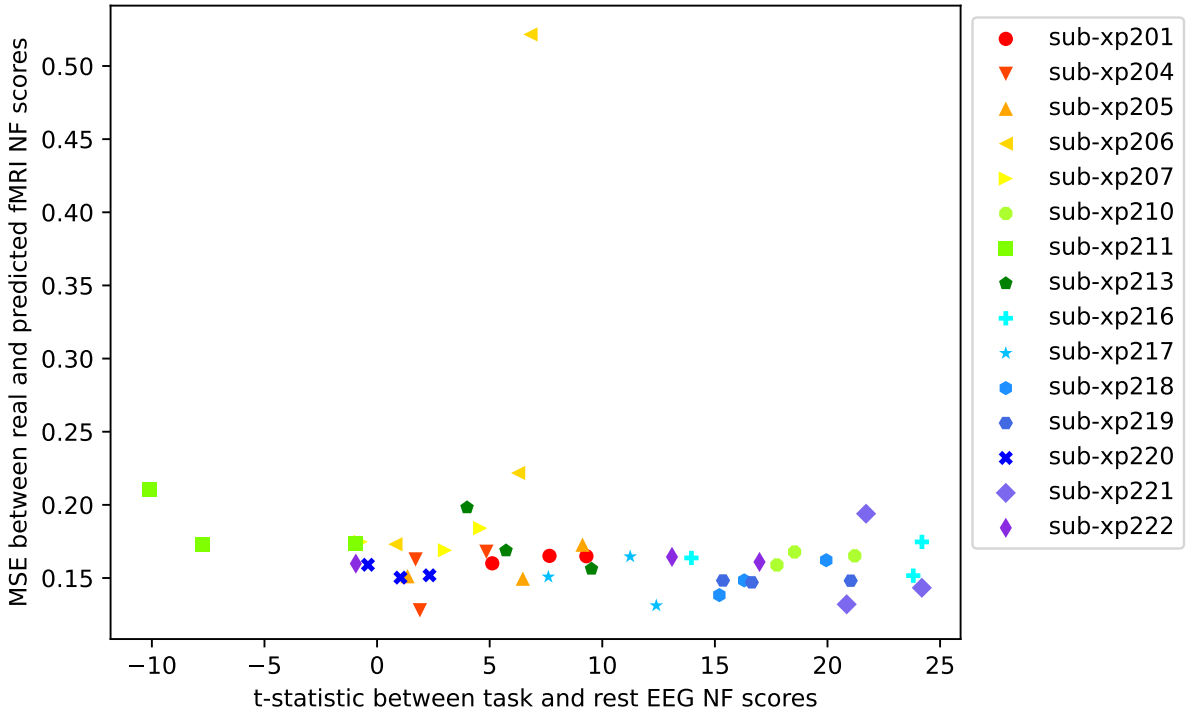


Figure 4.18 – **Analysis for the LSTM approach with raw signal samples as input.** Mean squared error (MSE) between fMRI NF predictions and true fMRI NF scores, in contrast to the t-statistic between task and rest blocks of the EEG NF scores which indirectly represents the quality of the EEG signal, for all test subjects across all folds. Each test subject with its 3 runs is represented by 3 points of different shape and color. The correlation coefficient between the two variables is −0.145.

4.4.4 CNN model with extracted features samples as inputs

In this section, we present the results of our method applied to the 1D CNN type with extracted features-based input data. After running the genetic algorithm, the architecture hyperparameters found are as follows: three convolutional layers, with the first having 32 filters (subsequent layers doubling that number from the preceding one), all with a kernel size of 3. The dense layer contains 64 neurons. Kernel regularizers (applied to the convolutional and dense layers) were set to 0.001. Moreover, a spatial dropout rate of 0.8 was employed for the convolutional layers, while the dropout rate for the dense layer was set to 0.2. An illustration of this network is provided in Figure 4.19.

The use of three convolutional layers, which is not the maximum allowed by the genetic algorithm, along with only 32 filters in the first layer and only 64 neurons in the dense

layer, allows us to consider that this model is relatively small. The regularization applied is fairly low, suggesting that the model is either small enough not to require significant regularization or that the dropouts are sufficient to mitigate overfitting. And indeed, the spatial dropout rate applied to the convolutional layers is very high, which likely forces the network to learn more robust patterns. However, the dropout applied to the dense layer is surprisingly low, further indicating that the model might not be as prone to overfitting as larger models might be. This reflects a similar tendency observed in the LSTM approach, where the use of extracted features simplifies the network's task. Consequently, the genetic search selects smaller models, which appear well-suited to this kind of input data.

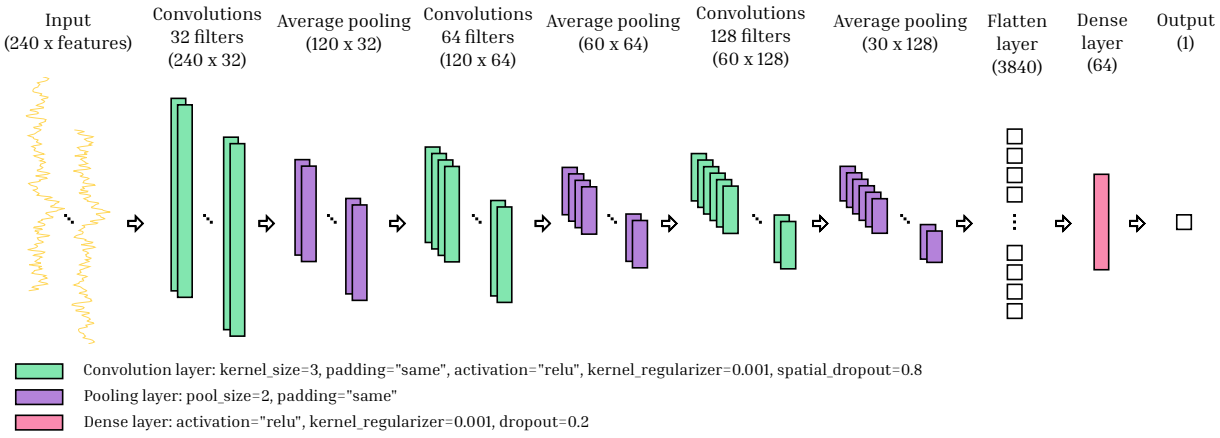


Figure 4.19 – **Architecture of the CNN found through the genetic search, using extracted features samples as input.**

Figure 4.20 presents the training and validation loss curves for the 15 CNN models trained on extracted features-based samples, as described in the preceding sections.

Both loss curves demonstrate good convergence overall, indicating that the model is learning effectively from the data without significant overfitting. The close alignment of the training and validation losses suggests that the model maintains good generalization ability, adapting well to unseen data. Similarly to the LSTM approach with extracted features samples, the validation loss is occasionally slightly higher than the training loss, which could indicate minor overfitting or some variability in the data, with some validation subsets possibly containing more challenging samples. In summary, the CNN model with extracted features samples exhibits a solid training process with good convergence between training and validation loss, making it the cleanest of all sections so far.

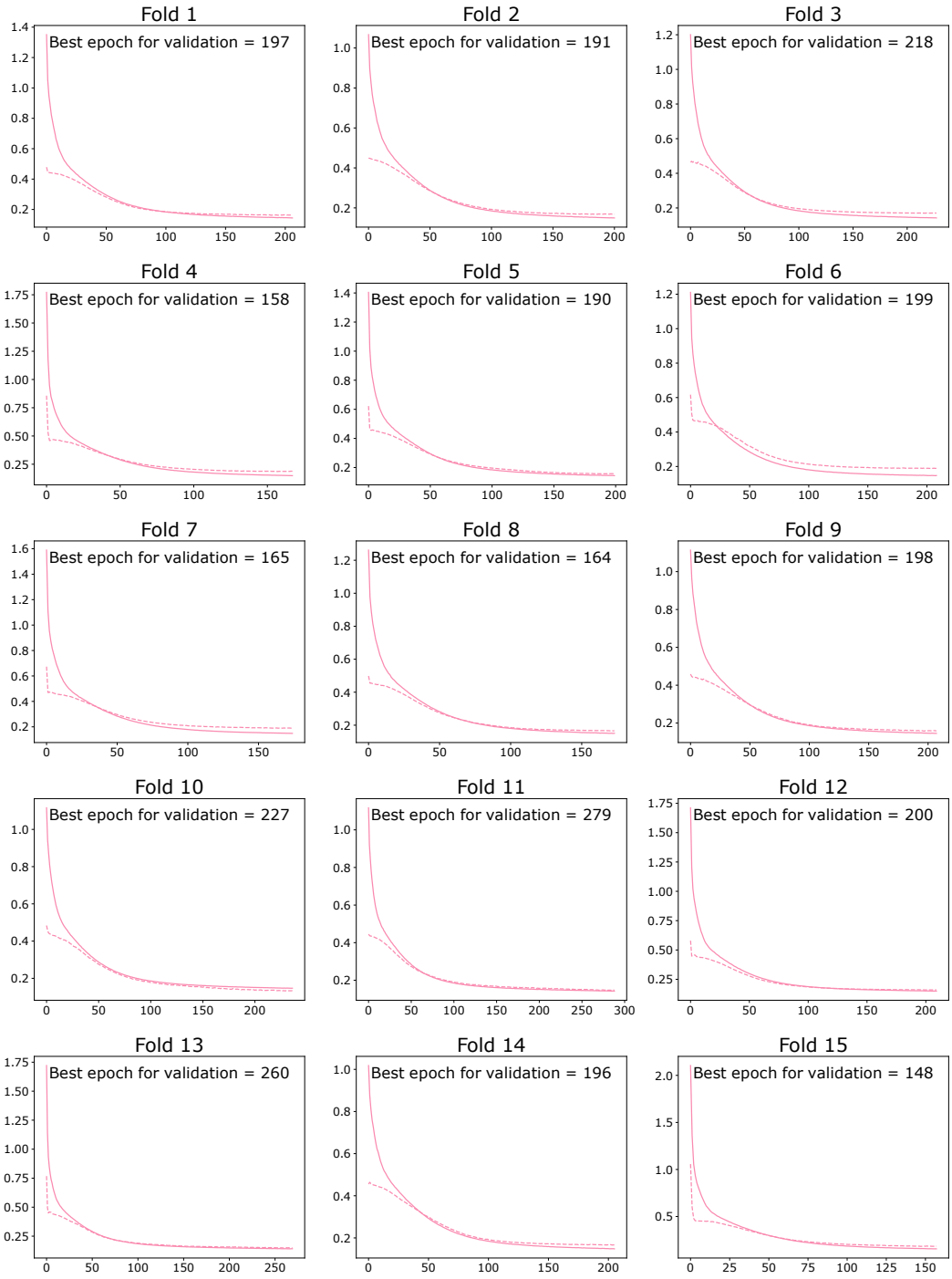


Figure 4.20 – **Learning curves for the 15 folds of the CNN approach with extracted features samples as input.** The training loss is shown as a solid line, while the validation loss is displayed as a dotted line. The abscissa shows the number of epochs, and the ordinate shows the loss value (MSE). The best epoch in terms of early stopping (i.e., validation) loss is indicated. As an early stopping strategy with patience and restoration of the best weights is employed, this number indicates the number of epochs for which each model was trained.

Same as previous sections, Figure 4.21 presents four key comparisons using mean squared error (MSE), based on predictions from all 15 trained models on their respective test subjects. It includes predicted fMRI NF scores versus true fMRI NF scores (corresponding to the pink (left) boxplot), fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the yellow (center left) boxplot), EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the purple (center right) boxplot, also referred to as baseline n°1), and true fMRI NF scores means averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the blue (right) boxplot, also referred to as baseline n°2).

Firstly, we look at the predictions from the models: the mean MSE between predicted fMRI NF scores and true fMRI NF scores (pink boxplot) is $0.1586(\pm 0.0212)$. Secondly, our final results: the mean MSE between fMRI NF predictions averaged with EEG NF scores and true bi-modal EEG-fMRI NF scores (yellow boxplot) is $0.0397(\pm 0.0053)$. Thirdly, for baseline n°1: the mean MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores (purple boxplot) is $0.0829(\pm 0.0198)$. The MSE values for predictions and final results are the lowest so far. Like the LSTM approaches, the predictions averaged with EEG NF scores significantly ($p = 3.40e-20 < 0.05$ using a paired t-test) outperform baseline n°1. Fourthly, for baseline n°2 (blue boxplot), the mean MSE between the mean of true fMRI scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores is $0.0384(\pm 0.0037)$. Here, as the p-value computed using a paired t-test is $p = 0.204 > 0.05$, we cannot reject the null hypothesis of identical average scores, meaning that our predictions averaged with EEG NF scores have a similar accuracy as the mean of true fMRI scores averaged with EEG NF scores.

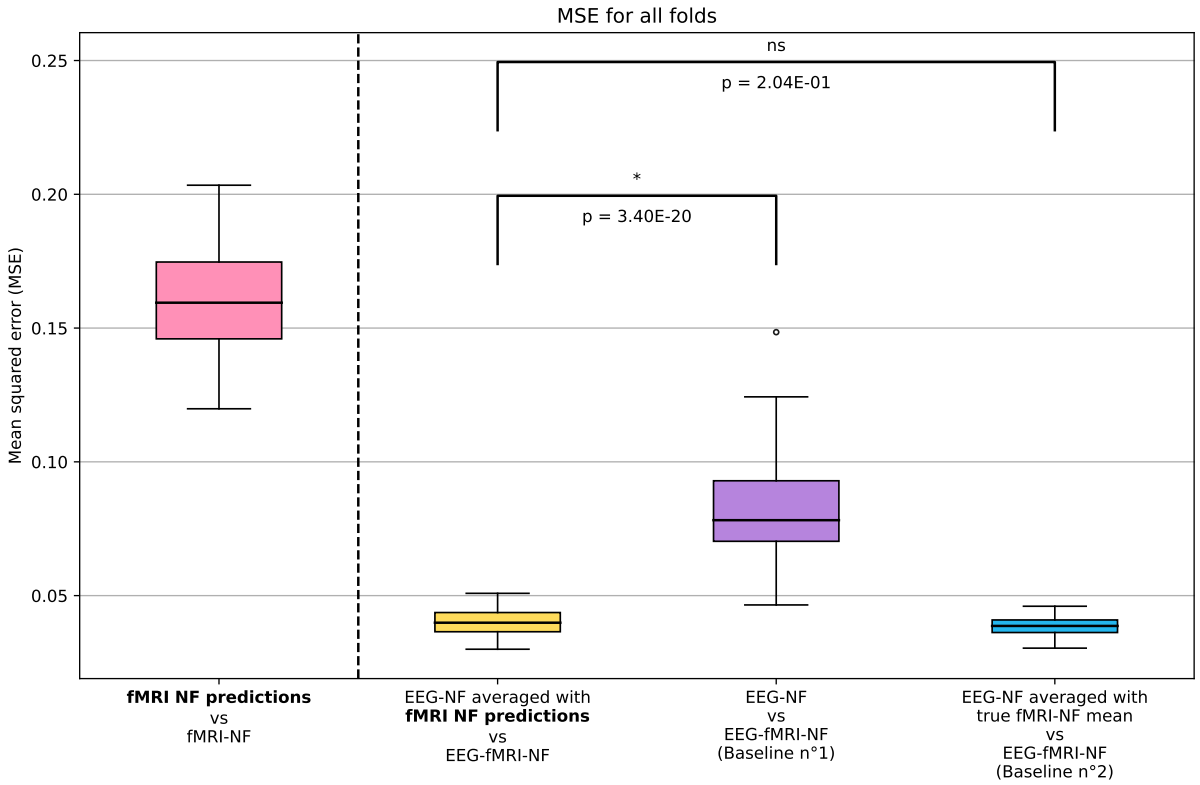


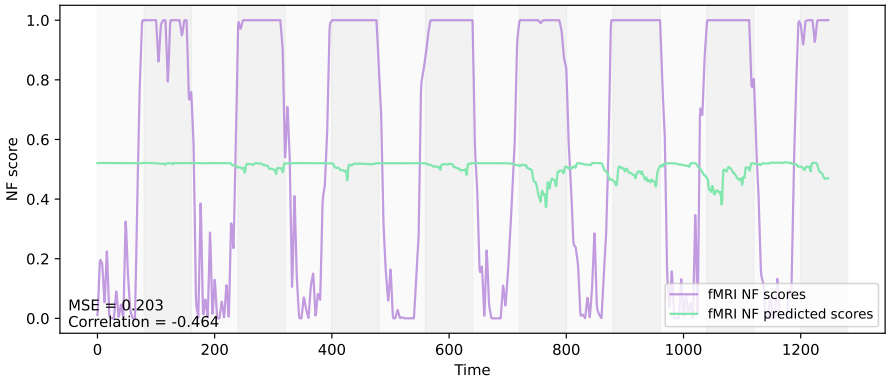
Figure 4.21 – **Results for all test subjects across all folds using the mean squared error (MSE) metric for the CNN with extracted features samples as input.** From left to right: MSE between the predictions of fMRI NF scores directly from the models and true fMRI NF scores (pink). MSE between fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (yellow). MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores, representing baseline n°1 (purple). MSE between the mean of true fMRI NF scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores, representing baseline n°2 (blue).

To better comprehend the results, we provide examples of predictions made using these models. We selected predictions from subjects with the highest and lowest mean squared error (MSE) in comparison with true fMRI NF scores across all folds. Figure 4.22 illustrates the comparison between the predicted and true fMRI NF scores for sub-xp211 run 3, which represents the highest error, and sub-xp221 run 3, which represents the lowest error.

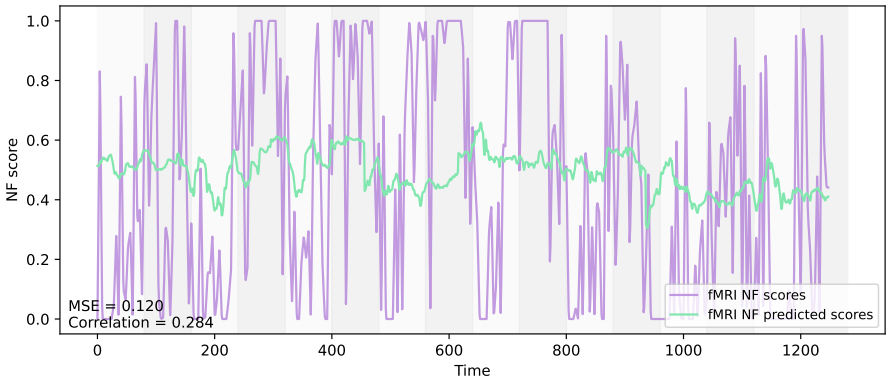
For the highest MSE example (a), the true fMRI NF scores are remarkably regular and clean, representing an ideal signal pattern. Despite this, the model's predictions are nearly flat, with occasional spikes that often move in the opposing direction during task blocks, leading to a mean MSE of 0.203. It is surprising that such perfect true scores

result in the highest error across all folds. This discrepancy suggests that factors beyond the quality of the true fMRI NF scores might be influencing the model's performance, such as the quality of the EEG signals used as input.

For the lowest MSE example (b) with a value of 0.120, the true fMRI NF scores align roughly with the expected rest/task trend but remain quite spiky. The model's predictions follow the true scores somewhat closely, displaying a similar spikiness, and appear to be approximately centered around the mean of the true scores. This suggests that the model was able to capture information from the EEG inputs, even though it struggled to fully match the amplitude of the true scores.



(a) Prediction for sub-xp211 run 3



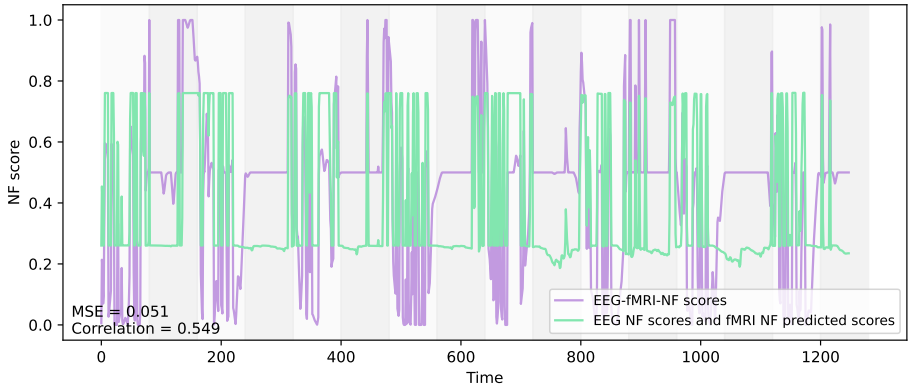
(b) Prediction for sub-xp221 run 3

Figure 4.22 – **Examples of predictions made using a CNN model with extracted features samples as input.** The green lines represent the model predictions, while the purple lines represent true fMRI NF scores. (a) illustrates the comparison between the prediction and the true scores for sub-xp211 run 3, representing the highest error across all folds. (b) illustrates the comparison between the prediction and the true scores for sub-xp221 run 3, representing the lowest error across all folds.

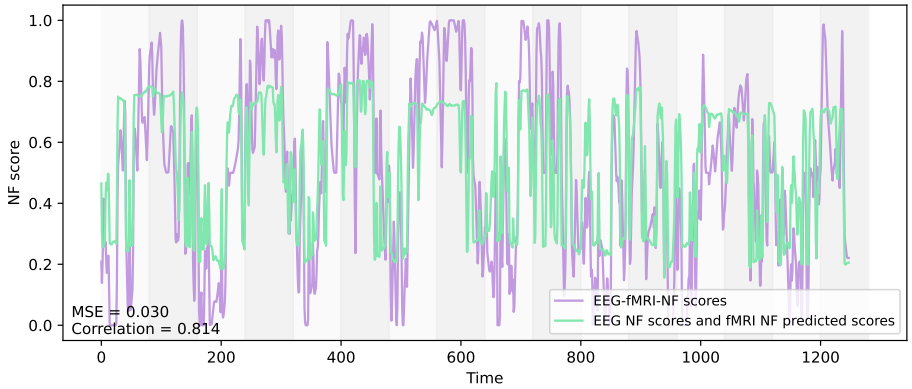
As described in section 4.3.3.2, we combined the EEG NF scores to the fMRI NF predictions to compare them with the true bi-modal EEG-fMRI NF scores, in order to get closer to the practical use of this method. To continue the illustration, Figure 4.23 presents the predictions for the same subjects used in Figure 4.22, averaged with the calibrated EEG NF scores, for comparison with the true bi-modal EEG-fMRI NF scores.

For the highest MSE example (a) with a value of 0.051, the true bi-modal EEG-fMRI NF scores exhibit a problematic pattern, with many plateaus at 0.5 during the task blocks. This indicates that when the true fMRI NF scores were at 1 (as seen in the preceding figure), the EEG NF scores were at 0. This discrepancy suggests potential issues with the quality of the EEG NF scores or possibly the EEG signals themselves. As the model's predictions in this case were almost flat, the final result show clearly the EEG NF scores with slight variations introduced by the fMRI NF predictions. The fact that it is a mean leads to reduced amplitude in this case.

For the lowest MSE example (b), the true bi-modal EEG-fMRI NF scores align more closely with the expected rest/task trend, though they are not entirely ideal, especially towards the end of the run. The final result here seems to track the true scores well. The mean MSE of 0.030 reflects this closer alignment, though the model still struggles to fully match the true scores' amplitude. These examples further highlight the influence of the quality of EEG NF scores, likely reflecting the quality of EEG signals, in achieving good prediction performance.



(a) Prediction for sub-xp211 run 3



(b) Prediction for sub-xp221 run 3

Figure 4.23 – **Examples of final results made using a CNN model with extracted features samples as input.** The green lines represent the model's fMRI NF predictions averaged with EEG NF scores, while the purple lines represent the true bi-modal EEG-fMRI NF scores.

Finally, we investigate why our method performs differently across subjects. As previously mentioned, a high t-statistic indicates that NF scores during task blocks are significantly higher than those during rest blocks, suggesting a strong neurofeedback response from the participant and good signal quality. Conversely, a t-statistic value below zero indicates that the participant responded better to neurofeedback during rest than during the task, which could imply lower EEG signal quality and/or a misunderstanding of the instructions. Figure 4.24 shows the performance of fMRI NF predictions compared to the t-statistic calculated on the EEG NF scores of the corresponding run for each fold (i.e., 1 fold is 1 subject with 3 runs tested).

The results regarding the t-statistic between the task and rest blocks of the EEG NF scores are the same as the previous sections. In summary, a vast majority of runs have a positive t-statistic, although a significant number are close to zero, and sub-xp211 appears to be a clear outlier.

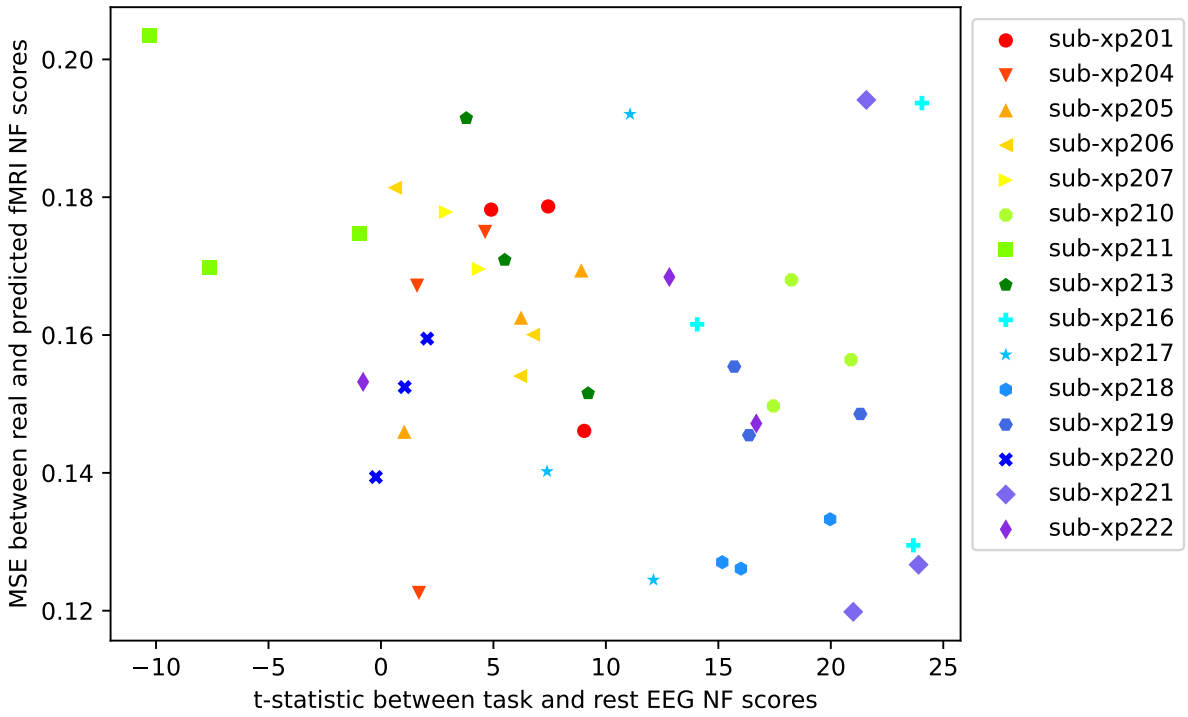


Figure 4.24 – **Analysis for the CNN approach with extracted features samples as input.** Mean squared error (MSE) between fMRI NF predictions and true fMRI NF scores, in contrast to the t-statistic between task and rest blocks of the EEG NF scores which indirectly represents the quality of the EEG signal, for all test subjects across all folds. Each test subject with its 3 runs is represented by 3 points of different shape and color. The correlation coefficient between the two variables is -0.358.

Now, regarding the MSE between predicted and true fMRI NF scores, we observe a general trend where a higher t-statistic is associated with a lower MSE. For instance, the highest MSE example run we observed earlier, from sub-xp211, corresponds to the lowest t-statistic value (around -10). While the lowest MSE example run (from sub-xp221) does not have the absolute highest t-statistic value, it still ranks among the highest (around 21). However, there are a few counterexamples: for instance, the run from sub-xp204 in the bottom left corner shows a very low MSE despite a t-statistic close to 0. Conversely, the runs from sub-xp221 and sub-xp216 in the top right corner have t-statistics close to

25 but very high MSEs. This indicates that good predictive performance can sometimes occur even when the neurofeedback response and signal quality (as measured by the t-statistic) are weak. To conclude, the correlation coefficient value of -0.358 suggests that, while there seems to be a link between the t-statistic and MSE, it is not the only factor influencing the model's performance.

4.4.5 CNN model with raw signal samples as inputs

In this final section, we present the results of our method applied to the 1D CNN type with raw signal-based input data. After running the genetic algorithm, the architecture hyperparameters found include four convolutional layers, starting with 128 filters in the first layer, all using a kernel size of 3. The dense layer contains 64 neurons. Regularization values were set to 0.01. Additionally, a spatial dropout rate of 0.6 was used for the convolutional layers, while the dense layer had a dropout rate of 0.4. An illustration of this network is available in Figure 4.25.

Similarly to the LSTM approach, the architecture identified through the genetic search for raw signal inputs is larger than the one found for extracted features inputs. The presence of four convolutional layers, with the first one already having 128 filters and subsequent layers doubling that number, makes this a relatively large model. Notably, the dense layer has the same number of neurons as in the previous section, which is modest in this context. This suggests that the model may have "won" the genetic search by striking a balance: it has substantial capacity for extracting meaningful patterns in the convolutional layers, while the relatively small dense layer helps prevent overfitting. Regularization also plays a critical role here, with average values for kernel regularizers and dropout rates indicating that the network required it to perform well. The slight increase in error, 0.166, compared to the extracted features data model (0.159), supports the interpretation given for the LSTM approach that this model is more complex because it faces greater challenges in extracting meaningful patterns directly from the raw signal inputs.

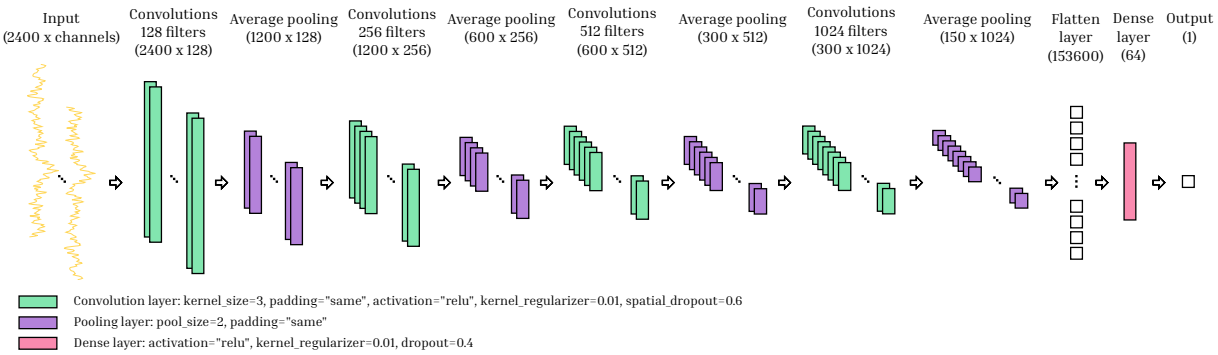


Figure 4.25 – **Architecture of the CNN found through the genetic search, using raw signal samples as input.**

Figure 4.26 displays the loss curves for the 15 CNN models trained on raw signal-based samples, as described in the preceding sections.

The near-complete overlap of the training and validation loss curves throughout the training process signifies that the model's learning was highly consistent across both datasets. This close alignment is an indicator that the model is generalizing well to unseen data, as there is little to no sign of overfitting. However, the flatness of both curves towards the end of training suggests that the model reached a plateau where further training did not lead to significant improvements. This stability could either indicate that the model quickly learned the patterns in the data, or that there is limited information in the data, making additional epochs unlikely to yield further performance gains. The overall short number of epochs, ranging between 38 and 89 before early stopping, suggests that the model quickly reached its optimal performance. In summary, the CNN model with raw signal inputs demonstrates an efficient training process, characterized by rapid convergence and stable overlapping loss curves.

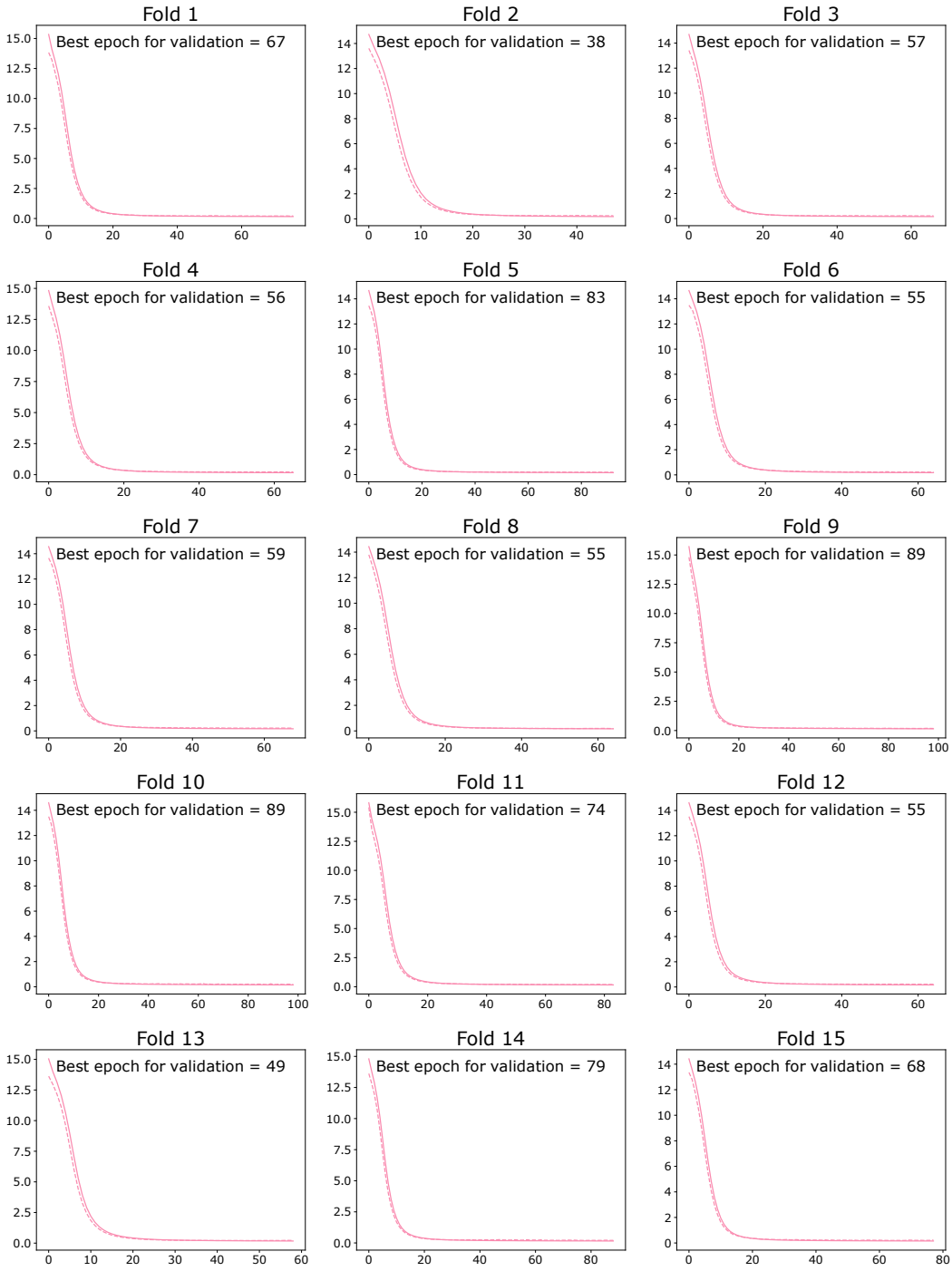


Figure 4.26 – **Learning curves for the 15 folds of the CNN approach with raw signal samples as input.** The training loss is shown as a solid line, while the validation loss is displayed as a dotted line. The abscissa shows the number of epochs, and the ordinate shows the loss value (MSE). The best epoch in terms of early stopping (i.e., validation) loss is indicated. As an early stopping strategy with patience and restoration of the best weights is employed, this number indicates the number of epochs for which each model was trained.

Same as preceding sections, Figure 4.27 presents four key comparisons using mean squared error (MSE), based on predictions from all 15 trained models on their respective test subjects. It includes predicted fMRI NF scores versus true fMRI NF scores (corresponding to the pink (left) boxplot), fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the yellow (center left) boxplot), EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the purple (center right) boxplot, also referred to as baseline n°1), and true fMRI NF scores means averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (corresponding to the blue (right) boxplot, also referred to as baseline n°2).

Firstly, we look at the predictions from the models: the mean MSE between predicted fMRI NF scores and true fMRI NF scores (pink boxplot) is $0.1692(\pm 0.0299)$. Secondly, our final results: the mean MSE between fMRI NF predictions averaged with EEG NF scores and true bi-modal EEG-fMRI NF scores (yellow boxplot) is $0.0423(\pm 0.0075)$. Thirdly, for baseline n°1 (purple boxplot): the mean MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores is $0.0831(\pm 0.0196)$. This fourth approach also significantly ($p = 1.15e-19 < 0.05$ using a paired t-test) outperforms baseline n°1. However, for baseline n°2 (blue boxplot), the mean MSE between the mean of true fMRI scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores is $0.0388(\pm 0.0037)$. Our predictions averaged with EEG NF scores are thus significantly ($p = 8.32e{-}3 < 0.05$ using a paired t-test) less accurate than the mean of true fMRI scores averaged with EEG NF scores.

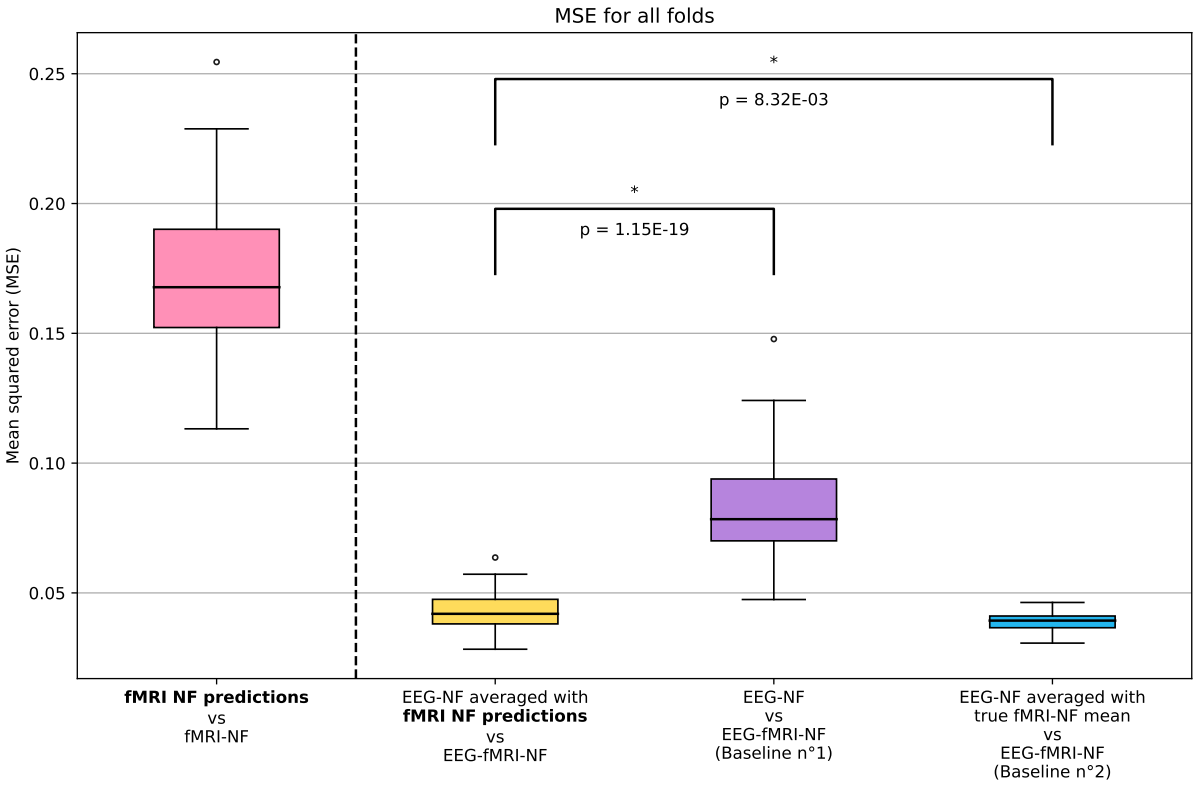


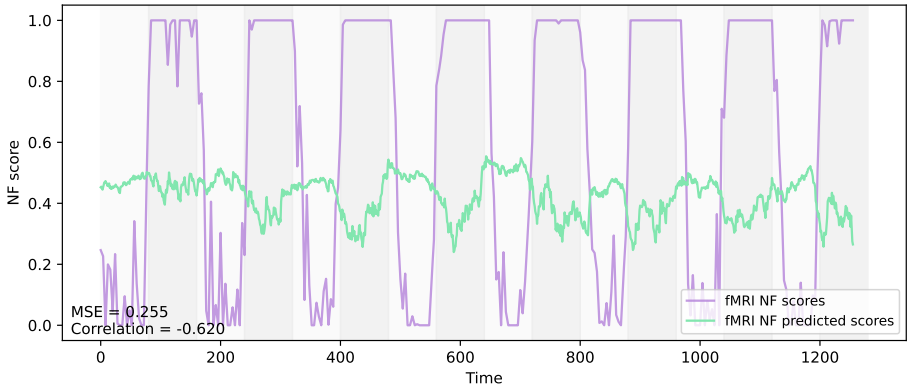
Figure 4.27 – **Results for all test subjects across all folds using the mean squared error (MSE) metric for the CNN with raw signal samples as input.** From left to right: MSE between the predictions of fMRI NF scores directly from the models and true fMRI NF scores (pink). MSE between fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (yellow). MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores, representing baseline n°1 (purple). MSE between the mean of true fMRI NF scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores, representing baseline n°2 (blue).

To better comprehend the results, we provide examples of predictions made using these models. We selected predictions from subjects with the highest and lowest mean squared error (MSE) in comparison with true fMRI NF scores across all folds. Figure 4.28 illustrates the comparison between the predicted and true fMRI NF scores for sub-xp211 run 3, which represents the highest error, and sub-xp210 run 3, which represents the lowest error.

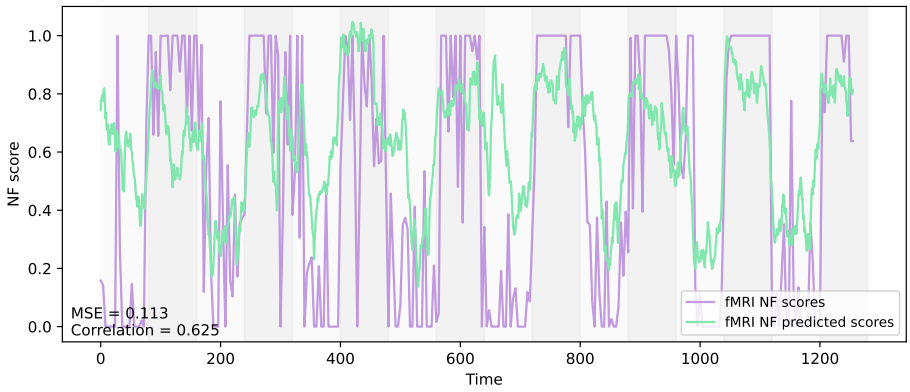
For the highest MSE example (a) with a value of 0.255, we observe that it is the same subject and run as in the previous section. Once again, the true fMRI NF scores are remarkably regular and clean. Despite this, the model's predictions are only somewhat centered around the mean of the true scores and often move in the opposite direction

during both task and rest blocks. This outcome further supports the notion raised in the previous section: even when the true scores are nearly ideal, the model struggles, likely due to issues related to the EEG signal inputs rather than the true fMRI NF scores themselves.

For the lowest MSE example (b), the true fMRI NF scores follow the expected rest/task trend with some slight spikiness. The model's predictions appear to be the best among all section examples, aligning well with the true scores and showing good amplitude, resulting in a mean MSE of 0.113.



(a) Prediction for sub-xp211 run 3



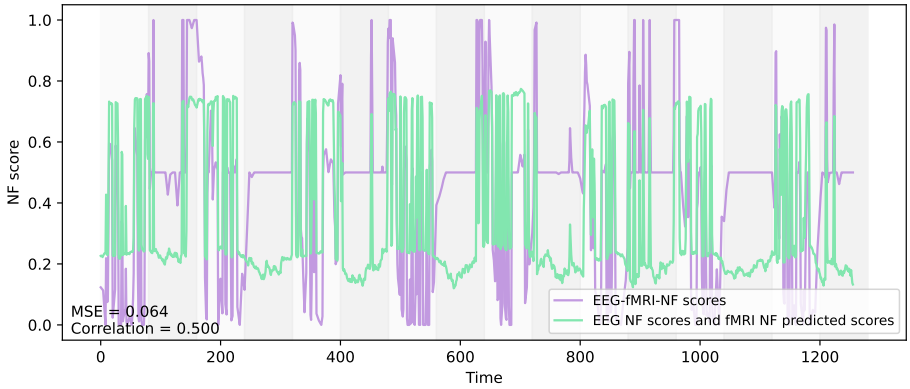
(b) Prediction for sub-xp210 run 3

Figure 4.28 – **Examples of predictions made using a CNN model with raw signal samples as input.** The green lines represent the model predictions, while the purple lines represent true fMRI NF scores. (a) illustrates the comparison between the prediction and the true scores for sub-xp211 run 3, representing the highest error across all folds. (b) illustrates the comparison between the prediction and the true scores for sub-xp210 run 3, representing the lowest error across all folds.

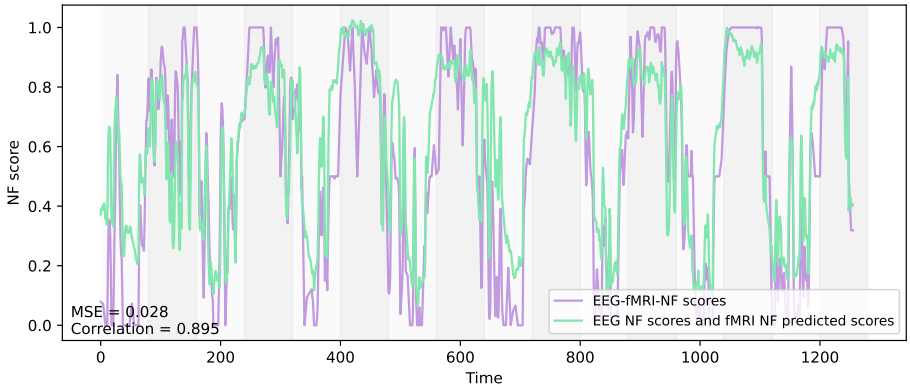
As described in section 4.3.3.2, we combined the EEG NF scores to the fMRI NF predictions to compare them with the true bi-modal EEG-fMRI NF scores, in order to get closer to the practical use of this method. To continue the illustration, Figure 4.29 presents the predictions for the same subjects used in Figure 4.28, averaged with the calibrated EEG NF scores, for comparison with the true bi-modal EEG-fMRI NF scores.

For the highest MSE example (a), the true bi-modal EEG-fMRI NF scores display, as in the previous section, significant flat sections at 0.5 during the task blocks. This flatness suggests again that when the true fMRI NF scores were at 1, the EEG NF scores were at 0, indicating potential issues with the EEG NF scores and even EEG signals. As the model's predictions in this case were somewhat flat, except during the task blocks where they tended to move in the opposite direction of the true scores, the final outcome resembles the EEG NF scores but with reduced amplitude and further misalignment during task blocks. Consequently, this leads to a high mean MSE of 0.064.

In contrast, the lowest MSE example (b) shows true bi-modal EEG-fMRI NF scores that follow the expected rest/task trend quite well. The model's predictions in this case were well-aligned with the true scores, resulting in a final output that exhibits good amplitude and accurately captures the overall trend. This example has a mean MSE of 0.028, the lowest across all sections. The close match between the final result and the true bi-modal scores in this instance may be due to the model's ability to generalize well when the input signals are of higher quality.



(a) Prediction for sub-xp211 run 3



(b) Prediction for sub-xp210 run 3

Figure 4.29 – **Examples of final results made using a CNN model with raw signal samples as input.** The green lines represent the model's fMRI NF predictions averaged with EEG NF scores, while the purple lines represent the true bi-modal EEG-fMRI NF scores.

Finally, we investigate why our method performs differently across subjects. As previously mentioned, a high t-statistic indicates that NF scores during task blocks are significantly higher than those during rest blocks, suggesting a strong neurofeedback response from the participant and good signal quality. Conversely, a t-statistic value below zero indicates that the participant responded better to neurofeedback during rest than during the task, which could imply lower EEG signal quality and/or a misunderstanding of the instructions. Figure 4.30 shows the performance of fMRI NF predictions compared to the t-statistic calculated on the EEG NF scores of the corresponding run for each fold (i.e., 1 fold is 1 subject with 3 runs tested).

The results regarding the t-statistic between the task and rest blocks of the EEG NF scores are the same as the previous sections. In summary, a vast majority of runs have a positive t-statistic, although a significant number are close to zero, and sub-xp211 appears to be a clear outlier.

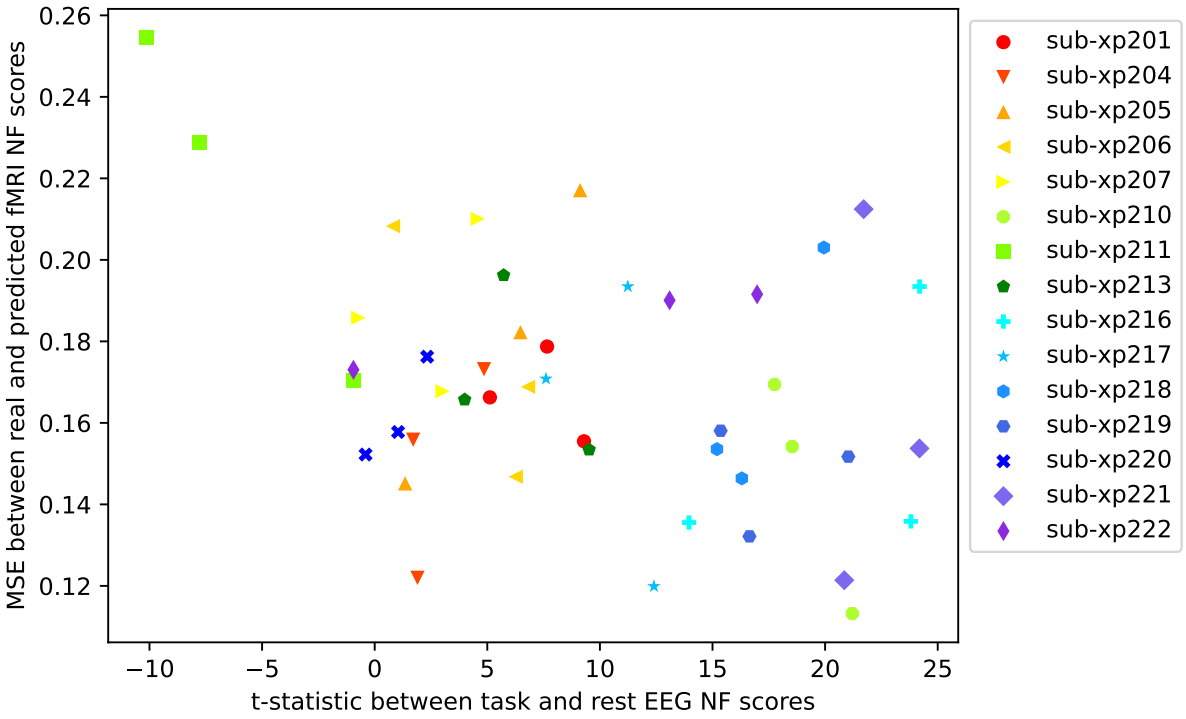


Figure 4.30 – **Analysis for the CNN approach with raw signal samples as input.** Mean squared error (MSE) between fMRI NF predictions and true fMRI NF scores, in contrast to the t-statistic between task and rest blocks of the EEG NF scores which indirectly represents the quality of the EEG signal, for all test subjects across all folds. Each test subject with its 3 runs is represented by 3 points of different shape and color. The correlation coefficient between the two variables is -0.365.

Regarding the MSE between predicted and true fMRI NF scores, we observe the same trend as the previous section, where a higher t-statistic is generally associated with a lower MSE. For example, the highest MSE run we observed earlier, from sub-xp211, corresponds to the lowest t-statistic value (around -10). While the lowest MSE run (from sub-xp210) does not have the absolute highest t-statistic value, it still ranks among the highest (around 21). As with the previous section, there are a few counterexamples. For instance, the run from sub-xp204 in the bottom left corner shows a very low MSE despite having a t-statistic close to 0. Conversely, the runs from sub-xp221, sub-xp218, and sub-xp216 in the top right corner have t-statistics between 20 and 25 but quite high MSEs. It

is interesting to note that the correlation coefficient (-0.365) and the outliers are similar to those in the previous section, despite differences in the architecture and the formatting of the input data. This consistency suggests that, at least for the CNN approaches, some additional factors influencing the results may be linked to inherent information within the raw EEG data and true fMRI NF scores. We will discuss it a bit further in the next section.

4.5 Discussion

The use of a genetic algorithm was primarily motivated by the complexity of our application, mainly due to the nature of our input and output data. Despite thorough preprocessing, EEG input data remains noisy, influenced both externally by MRI recording conditions and internally by the brain's electrical activity. Additionally, as previously mentioned, although EEG and fMRI measure the same physical phenomenon, the relationship between these two modalities remain indirect and not well understood. This lack of prior expertise made it difficult to select an appropriate model architecture. Moreover, on a more conceptual level, we wanted to avoid proposing an architecture based on arbitrary choices without concrete justification. As a result, the genetic algorithm method allowed us to automatically move closer to an optimal architecture.

The main limitation of this approach is the computation time. Each individual in the genetic population requires a model to be trained, which is time-consuming. A larger population size and a greater number of generations would likely result in an architecture closer to the theoretical global optimum. Moreover, the long computation time also influenced the initial choices we made. For instance, we limited the number of convolutional layers in the 1D CNN to 4 because a more complex model would take too long to train, and we aimed to limit the genetic search to about one week per case. As a result, more ambitious architectures were not explored using this method. However, we can take some reassurance from the fact that we had previously tested larger model architectures outside of the genetic algorithm framework, which did not yield better performance. This allowed us to empirically choose the initial pre-selected values for the genetic search.

Initially, we expected LSTMs to outperform 1D CNNs in predicting fMRI NF scores due to their reputed strength in handling sequential data. However, our experiments revealed that 1D CNNs and LSTMs were similar in terms of performance, with 1D CNN

outperforming LSTM only within the extracted features approach. One possible explanation for why LSTMs did not outperform 1D CNNs could be the limited quantity and representativity of our dataset, which included only 15 subjects with 3 runs each. LSTMs typically excel with datasets that exhibit clear trends, such as those found in weather forecasting or sales predictions. However, the EEG data, particularly in simultaneous EEG-fMRI acquisitions, may have been too challenging for the LSTM to handle effectively. Additionally, complex architectures like LSTMs may struggle to generalize well to unseen data when working with a dataset that is not sufficiently representative. This could explain why a simpler architecture, like the 1D CNN, slightly outperformed LSTMs in this particular application.

On the other hand, we anticipated that the extracted features samples approach would yield better results than the raw signal samples approach, and our experiments confirmed this for the CNN configurations. Extracting bandpowers in the alpha and beta ranges from the raw signals appears to assist the models during the feature extraction phase of the network done in the convolutional layers. In contrast, using raw signals directly adds complexity for the model, as it essentially requires an additional step of extracting meaningful features. The reasoning behind trying raw signals was to simplify the pipeline for real-life applications, where the model could be used directly on raw signals from the subject to predict fMRI NF scores. Additionally, it served to test the neural networks' core strength in autonomously extracting interesting features. However, based on our results, incorporating an extra processing step to compute bandpowers before using the model in a neurofeedback protocol is a viable option, as it is not a costly operation at all.

Overall, developing a model that can predict fMRI NF scores from EEG signals for any subject (what we refer to as a global model, as opposed to an individual model) is challenging. The preceding work [4], from which this research stems, involved individualized sparse regression models designed to exploit EEG data alone to predict fMRI NF scores. That work demonstrated that such an endeavor seemed possible. However, it is difficult to directly compare our results with this previous work due to several key differences, such as our focus on developing a global model rather than one model per subject, and our use of MSE as the main metric instead of Pearson's correlation. Nevertheless, the earlier study showed that a simple machine learning approach with sparse models could achieve promising results, outperforming what we called baseline n°1 in this work. This motivated us to attempt creating a global model, which would be more practical in a clinical setup.

This exploration was particularly challenging due to the stark differences between

the two modalities. Nonetheless, all our models outperformed baseline n°1, where our predictions, when averaged with EEG NF scores, were closer in terms of MSE to the true bi-modal EEG-fMRI NF scores than EEG NF scores alone. This indicates that we successfully enhanced the EEG NF scores. However, the mean correlation between our fMRI NF predictions and the true fMRI NF scores was quite low, and the shapes of some result runs did not perfectly match. It could be interpreted that the EEG NF scores and fMRI NF scores are so distinct that even predicting something vaguely close to the fMRI NF score was enough to surpass baseline n°1. The LSTM with raw signal-based samples presented an interesting case, producing almost flat predictions with a mean MSE similar to those with more amplitude. This made us consider another baseline, referred to as baseline n°2. As we observed, the models did not outperform baseline n°2, indicating that the models provided at best fMRI predictions that had as much error with the true fMRI NF scores as the mean of these true scores. Since these fMRI predictions are derived from an entirely different modality (EEG), this remains a interesting result, but there is still much to understand and improve.

To begin with areas for improvement, let's talk about neural networks. While re-framing the problem could enable the use of fine-tuning to create individualized models, we will focus here on improving global models. The current trend in deep learning leans towards larger architectures, which require a significant amount of data to train. This could still be a viable path if we can acquire more data, either through extensive data collection efforts or by pursuing data sharing and open data initiatives. However, since the simpler 1D CNN models in our study performed slightly better, it is worth considering a contradicting idea in the field: exploring smaller models. These models require less data to train and are less prone to overfitting, making them also an interesting direction for further investigation.

Next, regarding the data itself: there is an understanding that comes after working in the machine learning field for a short while that significant performance improvements often come not from changing the model, but from cleaning, organizing, and understanding the data. With simultaneous EEG-fMRI acquisitions, there is certainly correlated noise (especially when the subject is moving, which is inevitable) between the electrodes, that may have been captured by the model. To our knowledge, there are currently no methods capable of fully correcting these residual noises, highlighting a need for further research in this area.

It would have been valuable to allocate more time to the better understanding the

EEG signal inputs. The t-statistic measure of EEG NF scores, which we used as an approximate indicator of EEG signal quality (as well as neurofeedback response), revealed significant variability between participants. While this variability is advantageous for training a global model with representativity, it also makes the relationship between inputs and scores more challenging for the model to learn. In our experiments, it might have been worth considering the exclusion of sub-xp211, as an outlier like this could potentially confuse the model during training. Although to be precise, we can note that the training losses and results from fold 6, where this subject was in the validation dataset and therefore not used for training or testing, remained consistent with other folds.

Overall, this work provided in-depth exploration of the possibility of predicting fMRI information for any subject using EEG signals acquired from multiple participants. Moving forward, we believe that performance enhancements may still be achieved by developing another modeling approach, but more importantly, by gaining a deeper understanding of the data. Improving noise correction techniques for EEG signals and better characterizing variability between subjects could lead to more robust models. On that note, a first attempt at correcting individual differences between subjects will be presented in the next chapter.

4.6 Data, code, and model availability

— Data: The pseudonymized data are available in BIDS format on the OpenNeuro platform at `https://openneuro.org/datasets/ds002338`. It is described in [40] as the XP2 protocol.

— Code: The code developed for this research is available on Gitlab Inria at `https://gitlab.inria.fr/cpinte/prediction-of-fmri-neurofeedback-scores-from-eeg-signals`. Additionally, the code has been archived with Software Heritage to ensure long-term preservation at `https://archive.softwareheritage.org/swh:1:dir:a651db8d1934543cac321a1723cdf404348e2156;origin=https://gitlab.inria.fr/cpinte/prediction-of-fmri-neurofeedback-scores-from-eeg-signals;visit=swh:1:snp:7004a52e6d9c3e956d761b71bdb3ec6e90a99d8e;anchor=swh:1:rev:f62cd5eff14180de3167bea085892e34a0f3a517`. All implementations were made on an Nvidia RTX A3000 GPU. The genetic search takes between 1 and 2 weeks, while post-genetic training takes between 1 and 2

days, depending on the neural network type and hyperparameters chosen. The predictions on the test dataset are almost instantaneous.

— Model: Since the data used to train the models are open, we share the trained models weights for all configurations and all folds in the same repository as the code.

4.7 Conclusion

We have presented a method for searching model architecture hyperparameters using a genetic algorithm in the context of predicting fMRI NF scores from EEG signals. This method is flexible and can be easily adapted to different model types. We used our genetic algorithm to converge towards four configurations, using LSTM and CNN types, both with two different data formats: extracted feature-based samples and raw signal-based samples. The CNN with extracted features approach demonstrated slightly superior performance in terms of mean squared error (MSE) compared to the other tested architectures. Our results showed that fMRI NF predictions, when averaged with EEG NF scores, align significantly closer to the true bi-modal EEG-fMRI NF scores than the EEG NF scores alone. This approach can enrich EEG NF scores by incorporating fMRI predictions derived from EEG, offering an improved NF score that leverages multi-modal information. However, our fMRI NF predictions have at best the same MSE with true fMRI NF scores as the mean of these true scores. So, we believe that the models developed using this method are not yet suitable for unimodal EEG neurofeedback applications and still require further improvements. The next chapter will explore one potential area for improvements, focusing on correcting individual differences between subjects.

Chapter highlights

- We investigated the possibility of predicting fMRI NF scores from EEG signals with a generalized model approach (as opposed to subject-specific models), aiming to reduce reliance on MRI in the context of bi-modal EEG-fMRI neurofeedback.

- To determine the model architectures, we introduced a hyperparameter search method based on a genetic algorithm, which can be applied across different types of neural networks.

- We evaluated four configurations: LSTM with extracted features-based samples, LSTM with raw signal-based samples, CNN with extracted features-based samples, and CNN with raw signal-based samples.

- All configurations outperformed our first baseline (n°1), symbolizing that our fMRI NF predictions successfully enhanced the EEG NF scores. The CNN with extracted features inputs achieved the lowest mean squared error (MSE).

- However, none of the configurations surpassed the second baseline (n°2), meaning that the models produced fMRI predictions with errors comparable to the mean of the true fMRI NF scores, highlighting the need for improvements.

Chapter 5

Euclidean space data alignment applied to EEG signals

In this final chapter, I introduce a preliminary work following the study presented in Chapter 4. Here, the focus is on reducing inter-subject variability in our EEG data using a method called Euclidean space alignment (EA), before applying the same methods and analysis as in the previous chapter.

5.1 Introduction

This work was conducted as part of a two-month international mobility program, funded by the Collège doctoral de Bretagne (with contributions from the Région Bretagne, Rennes Métropole, EUR Caps, and the Collège doctoral itself), obtained through an application process. During this period, I joined the Decoded Neurofeedback (DecNef) department of the Advanced Telecommunications Research Institute (ATR) in Kyoto, Japan. I was under the responsibility of Aurelio Cortese, whom I would like to thank once again for his warm welcome. I also had valuable exchanges with Reinmar Kobler, who introduced me to the field of domain adaptation that I will describe in this chapter, and I would like to express my gratitude to him as well. Out of all the ways of improving the research question presented in Chapter 4 that I have considered and explored throughout my thesis, this mobility experience allowed me to discover and investigate a particularly promising research field, which I will elaborate on in this chapter.

My contribution focuses on applying the Euclidean space alignment (EA) method [145], originating from the work of He He and Dongrui Wu, to our EEG data. After aligning the data, I applied the same method described in Chapter 4 to explore the potential of EA as a promising direction for future development.

In this chapter, we will discuss the field of domain adaptation. Domain adaptation refers to the process of adapting machine learning models trained on data from one domain (often referred to as the "source domain") to perform well on data from another domain (the "target domain") within the same task. It can be considered a sub-part of transfer learning (TL), which is a broader concept where knowledge learned from one task (the source) is used to improve performance on a different but related task (the target). In the context of domain adaptation in EEG, the term "domain" refers to a distribution of EEG data that shares specific characteristics. As outlined in the article [146], differences in these characteristics between domains can be attributed to inter-subject variability, which includes individual structural and functional differences in brain networks, as well as variations in the task being performed. It can also be attributed to intra-subject variability, as EEG signals from the same subject can vary across recording sessions due to changes in mental state, fatigue, or recording conditions such as the relative positioning and impedance of the electrodes.

To address this problem, a large number of methods are available, a categorization of which is proposed in this review [147]. Firstly, sample-based methods focus on selecting or weighting samples from the source domain to better match the target domain's data distribution. Overall, the goal is to give importance to the source samples that are most similar to the target domain, reducing the impact of domain differences. Secondly, inference-based methods seek to adapt the model itself to account for domain differences, for instance by incorporating constraints during the optimization process. Lastly, feature-based methods attempt to create a shared feature space between the source and target domains by aligning or transforming features to minimize the difference between their distributions.

This last category includes the Euclidean space alignment (EA) [145] method we are applying here, and which we will present in section 5.3. The article introducing EA also provides a brief overview of preceding methods, including one key method to which they compare themselves: Riemannian Alignment - Minimum Distance to Riemannian Mean (RA-MDRM) [148]. To understand this method, it is important to know that covariance matrices, which represent the relationships between different EEG channels, are symmetric positive definite (SPD) and lie on a Riemannian manifold. The Minimum Distance to Riemannian Mean (MDRM) approach [149, 150] consists in considering EEG covariance matrices as points in Riemannian space, and using their distance to the Riemannian mean

(called geodesic) as features in classification or regression tasks. Building on this, the authors of [148] proposed RA-MDRM, a transfer learning framework that applies affine transformations to the covariance matrices of each session/subject in order to center them with respect to a reference covariance matrix, utilizing the information of the resting state.

The authors of the EA method [145], inspired by the RA-MDRM method, present three limitations of RA-MDRM that they propose to address. Firstly, RA-MDRM aligns covariance matrices in the Riemannian space, which means that the subsequent model we wish to use must be able to operate on the covariance matrices directly. In contrast, EA aligns the EEG signals directly in the Euclidean space, meaning that any subsequent processing and machine learning methods can be applied directly on the aligned data. Secondly, using RA-MDRM in the context of event-related potentials BCIs (as opposed to motor imagery BCIs) requires some labeled data, making it a supervised learning approach. In any contexts, EA does not need any label information, making it a completely unsupervised learning approach. Lastly, EA can be computed faster than RA-MDRM, as EA uses the arithmetic mean as the reference matrix, whereas RA-MDRM requires the computationally more intensive Riemannian mean. Out of the three, our interest here lies in the first advantage, alignment in Euclidean space, since it allows us to apply the same method described in Chapter 4 following alignment.

To sum up, we present in this chapter the application of the Euclidean space alignment (EA) method to our EEG data. After alignment, we proceeded with the same hyperparameter genetic search and evaluation process across 15 folds detailed in Chapter 4. The goal was to begin exploring the domain adaptation field with a simple method to see if it can mitigate distribution shifts in our EEG data and assess whether the results are promising for future developments.

5.2 Materials

The data used are exactly the same as those described in section 4.2 of Chapter 4. The important detail to recall here is the structure of the data, presented in Figure 4.1, where each subject completes 3 neurofeedback runs, and each run consists of 8 blocks of rest and 8 blocks of task, with each block lasting 20 seconds.

5.3 Methods

To introduce the Euclidean space alignment (EA) method, I use the equations taken from the original article [145]. It should be noted that the alignment is performed independently for each subject. The method relies on a reference matrix $\bar{R}$ calculated according to the following formula:

$$\bar{R} = \frac{1}{n} \sum_{i=1}^{n} X_i X_i^T$$

where n is the number of samples X_i from a subject. $\bar{R}$ thus represents the arithmetic mean of all covariance matrices from one subject.

Then, the alignment is performed on each data point following the formula:

$$\tilde{X}_i = \bar{R}^{-1/2} X_i$$

The article demonstrates that after alignment, *"the mean covariance matrices of all subjects are equal to the identity matrix, and hence the distributions of the covariance matrices from different subjects are more similar"*. Applying this alignment to our EEG data should therefore attenuate distribution shifts across subjects.

To adapt this method to the specifics of our EEG data, recorded during neurofeedback sessions consisting of both rest and task blocks, we first selected only the 25 motor channels that we use in the method described in Chapter 4 (illustrated in Figure 4.3). Then, the reference matrix $\bar{R}$ for each subject was computed using only the data points from the rest blocks of all runs, even though the alignment was applied to the entire data. The reasoning behind this decision is that data from the rest blocks are expected to be more homogeneous across subjects than the data from task blocks.

5.4 Results

5.4.1 Impact of EA on the EEG data

To visualize the effect of the alignment applied to our data, the preferred method in this field is to use the t-distributed stochastic neighbor embedding (t-SNE) [151]. It is a nonlinear dimensionality reduction technique used to project high-dimensional data (such as EEG) into a lower-dimensional space, typically 2D or 3D. t-SNE works by grouping similar points and separating dissimilar points to reflect differences. This projection makes it easier to identify clusters or patterns, such as subject-specific differences, which can be useful in our case for comparing EEG data before and after applying EA.

We used the t-SNE implementation from scikit-learn with default parameters to generate the projections. As explained in the documentation, preparing the data in a compact and representative manner is recommended for better visualization and reduced computational costs. Therefore, for each subject, each run, and each block (rest or task, with 8 blocks of each per run), we grouped the data into chunks of 2000 points, representing 10-second segments (out of the 20-second blocks). Then, we computed the covariance matrices for each chunk, applied a logarithmic transformation, and converted them into their upper triangular form, for better visualization. The result is a compact representation of each chunks's covariance matrix, which we used as input for t-SNE.

We will now apply this technique to visualize our data before and after alignment. Figures 5.1, 5.2, and 5.3 represent the same projections but highlight different properties of the points using different color mappings. Specifically, they show respectively which subject, run, and block (rest or task) the data points belong to.

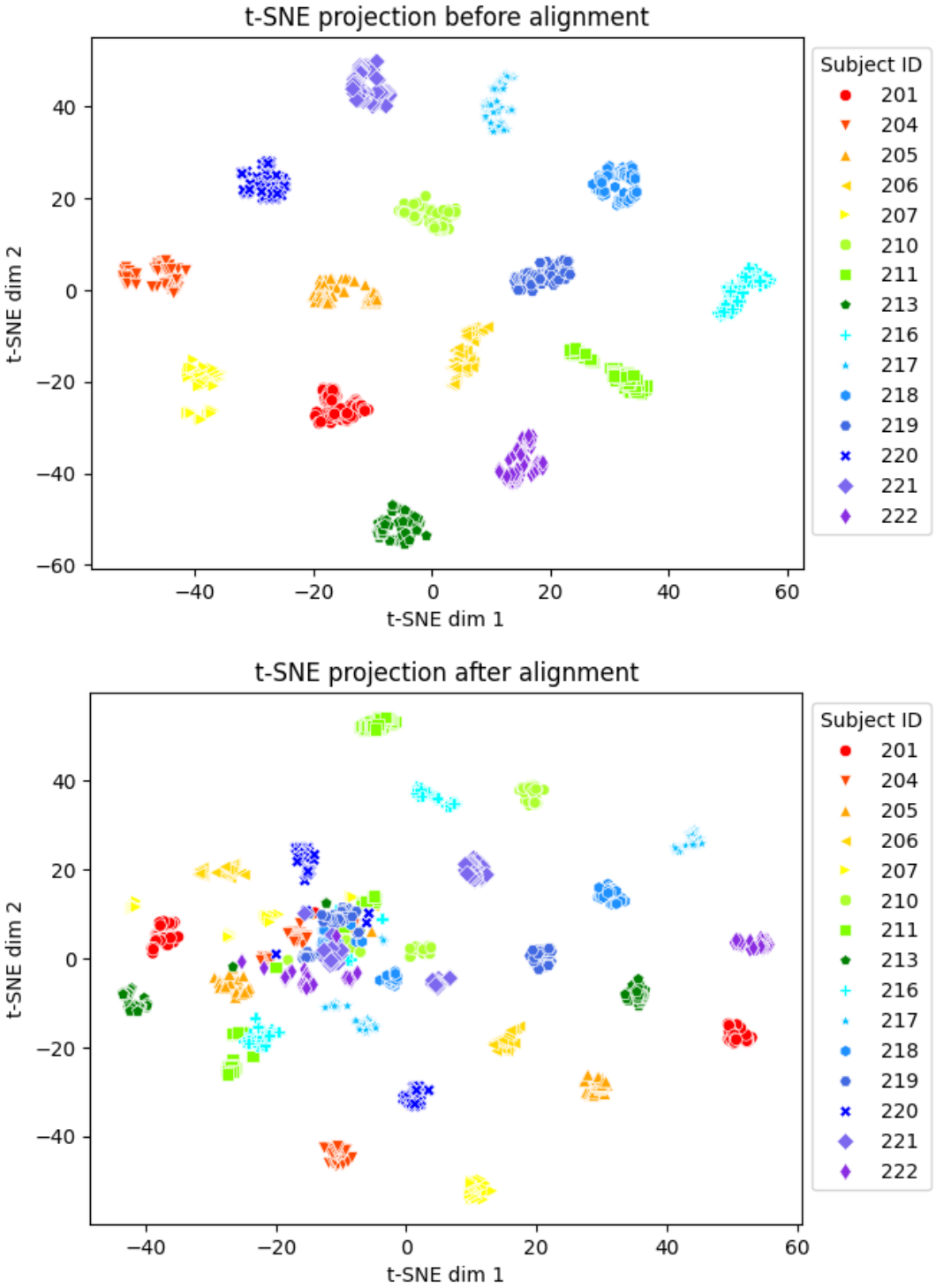


Figure 5.1 – **t-SNE projection before and after EA focusing on subject differences.** The t-distributed stochastic neighbor embedding (t-SNE) technique was used to visualize EEG data before (top) and after (bottom) Euclidean space alignment (EA). Each point represents a 10-second chunk of data points from the same block (rest or task), meaning each of the 3 runs from each subject is represented by 32 points. Here, each subject with its 3 runs is represented by a different shape and color.

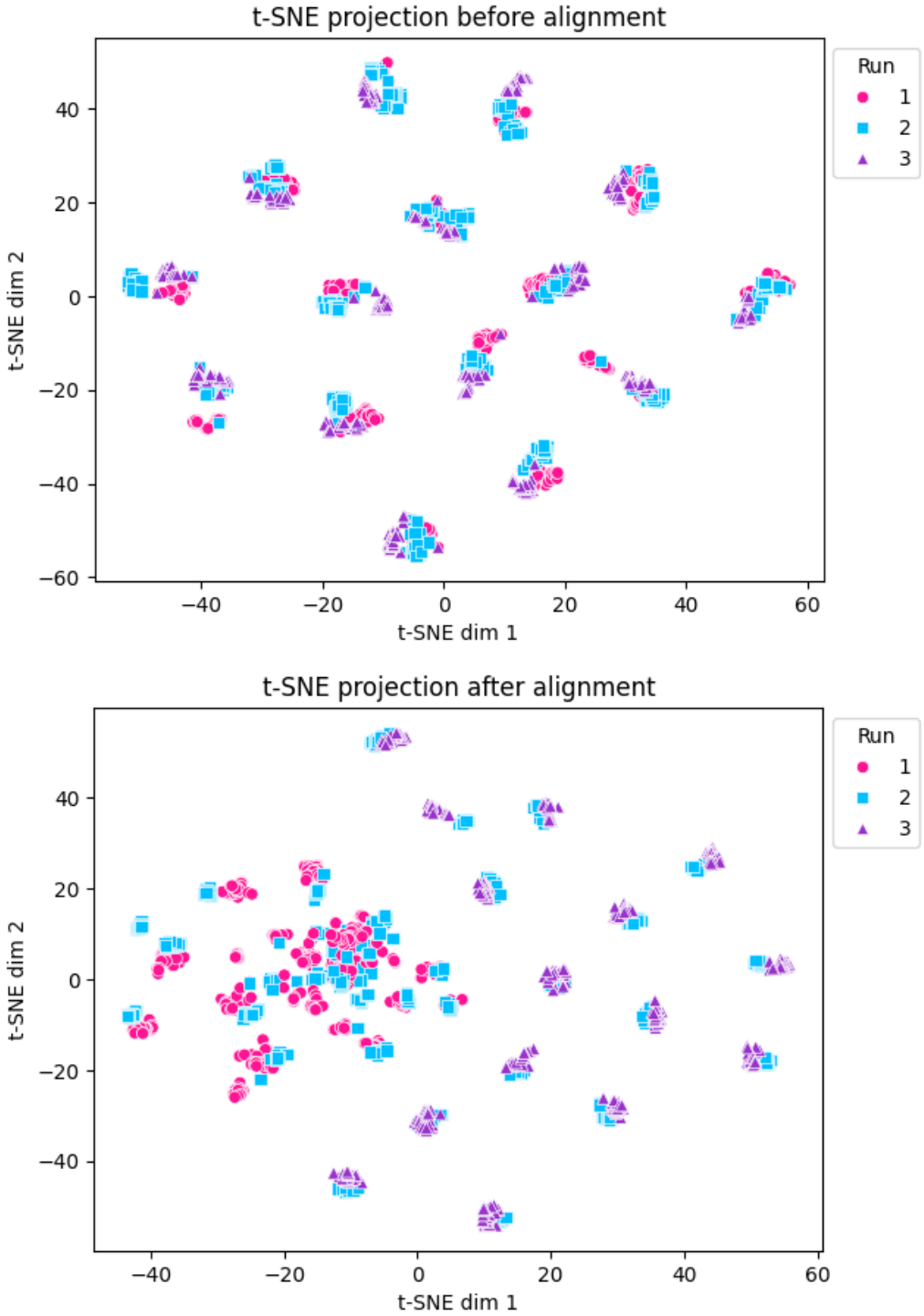


Figure 5.2 – **t-SNE projection before and after EA focusing on run differences.** The t-distributed stochastic neighbor embedding (t-SNE) technique was used to visualize EEG data before (top) and after (bottom) Euclidean space alignment (EA). Each point represents a 10-second chunk of data points from the same block (rest or task), meaning each of the 3 runs from each subject is represented by 32 points. Here, the first, second and third runs of every subjects are represented by a different shape and color.

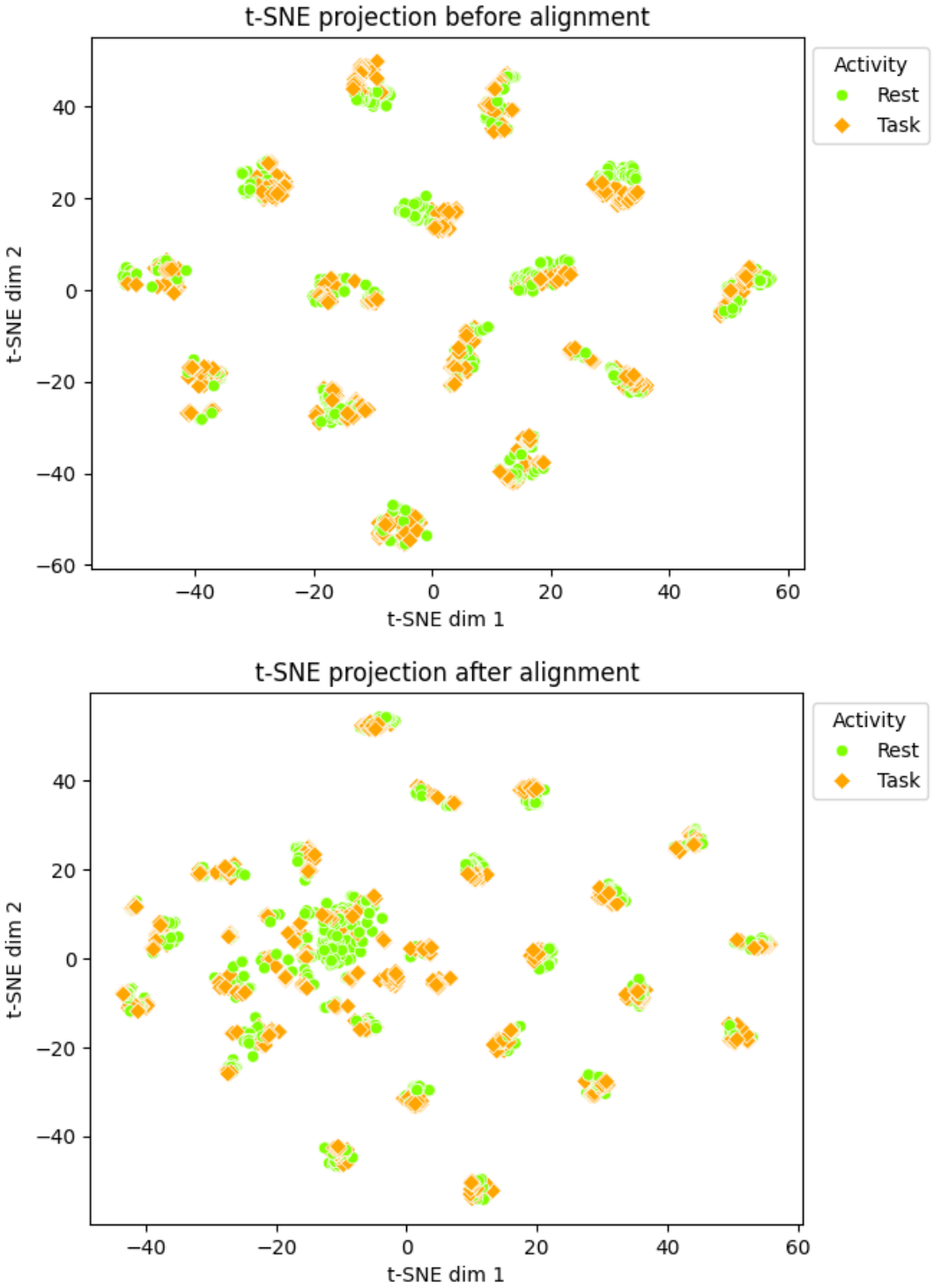


Figure 5.3 – **t-SNE projection before and after EA focusing on rest/task differences.** The t-distributed stochastic neighbor embedding (t-SNE) technique was used to visualize EEG data before (top) and after (bottom) Euclidean space alignment (EA). Each point represents a 10-second chunk of data points from the same block (rest or task), meaning each of the 3 runs from each subject is represented by 32 points. Here, the rest and task blocks are represented by a different shape and color.

Interpreting a t-SNE projection can be difficult because the distances between points and their values do not have direct interpretable meanings. Since t-SNE uses non-linear transformations to artificially spread out clusters in the low-dimensional space to enhance separation and make the plot visually interpretable, the distances between clusters or points in t-SNE plots do not correspond to real distances. The values on t-SNE plot axes represent relative positions in the 2D (or 3D) embedding created by t-SNE, which is not an interpretable measure. These distances and values come from the t-SNE algorithm's embedding process that attempts to place points in a 2D (or 3D) plane such that similar points are close together.

Let's start with Figure 5.1. In the first t-SNE plot, representing data before alignment, subjects are distinctly separated into small clusters. This indicates that t-SNE has successfully identified distinct patterns in the data that correspond to different subjects. As each point represents a 10-second chunk of data from a rest or task block, the formation of such distinct clusters suggests that the data varies significantly across subjects. After alignment, the t-SNE projection looks quite different. Firstly, we can notice a larger cluster on the center-left of the plot (which we will name center-left cluster), grouping together points from different subjects. This suggests that after EA, the EEG data from different subjects has become more similar. However, there are also some smaller clusters of points associated with only one subject, all around the center-left cluster. We can think that those chunks still have subject-specific patterns that the alignment process did not remove. One interesting thing to note is that for the majority of subjects, we can see three distinct clusters which we will refer to as: one in the center-left cluster, one far away, and one in between. Since we know that each subject encompasses 3 runs, we can hypothesize that these 3 clusters correspond to the 3 runs.

Following this hypothesis, we now look at the same t-SNE projection, focusing on runs instead of subjects (Figure 5.2). Before alignment, we can observe again the distinct clusters of subjects, this time seeing that they indeed consist of points from the 3 runs. After alignment however, our previous hypothesis appears incorrect. In fact, the far subject-specific clusters consist of all the chunks from the runs n°3 and approximately half of the chunks from the runs n°2. In contrast, the larger center-left cluster and the "in between" clusters consist of all the chunks from the runs n°1 and the remaining chunks from the runs n°2. Interestingly, the runs n°1 seem to align more effectively across subjects after EA, while the runs n°2 show mixed alignment success. The runs n°3 seem to

retain more specific characteristics. This projection suggests that there may be inherent subject-specific differences between runs (e.g., participant movement, fatigue, or learning strategies evolving along runs) that increase with each run and persist after EA.

Finally, we take another look at the same projection, this time observing the difference between chunks from rest and task blocks (Figure 5.3). Since our Euclidean space alignment was applied on all data using a reference matrix based solely on rest blocks, it's worth investigating how this affects the result. Before alignment, we can see that each cluster corresponding to a subject contains both rest and task blocks, though the neatness of the separation varies slightly between subjects. After alignment, the chunks corresponding to the rest blocks do not appear to be better aligned than those corresponding to the task blocks. To sum up, the t-SNE projection before alignment highlights that the most important differences are between subjects. The projection after alignment shows that EA reduces the differences between subjects, especially for the first run of each subject, with no noticeable impact on the separation of rest and task blocks. Nevertheless, it is important to note that the alignment seems not perfect, which is to be expected from a rather simple method like EA.

5.4.2 Impact of EA on model performance

After applying the EA method, we proceed to use the same approach on our aligned data as the one presented in Chapter 4. As a reminder, this involves formatting the dataset, running the genetic search algorithm to find architecture hyperparameters, and then training and evaluating the resulting models across 15 different folds. To carry out this experiment with all those steps, we chose to use the configuration that previously yielded the best performance: the CNN approach with extracted features-based samples.

Without going into as much detail as for the analyses in Chapter 4, the genetic research resulted in the following architecture: four convolutional layers, with the first having 16 filters (subsequent layers doubling that number from the preceding one), all with a kernel size of 3. The dense layer contains 256 neurons. Kernel regularizers (applied to the convolutional and dense layers) were set to 0.005. Moreover, a spatial dropout rate of 0.4 was employed for the convolutional layers, while the dropout rate for the dense layer was set to 0.5. Overall, and after analyzing the four architectures in the previous chapter, these seem to be well-suited values.

However, after training the 15 folds models, it turned out that the performance of the approach with Euclidean space alignment was surprisingly worse than the approach without it. A representation of the results is available in Figures 5.4 and 5.5 (with p-values from paired t-tests). Specifically, the average MSE between predicted and true fMRI NF scores for the approach without EA (the results of which are detailed in section 4.4.4) is $0.1586(\pm 0.0212)$ whereas it is $0.1686(\pm 0.0310)$ for the approach with EA. Like the other configurations, the approach with EA passes baseline n°1 but not n°2.

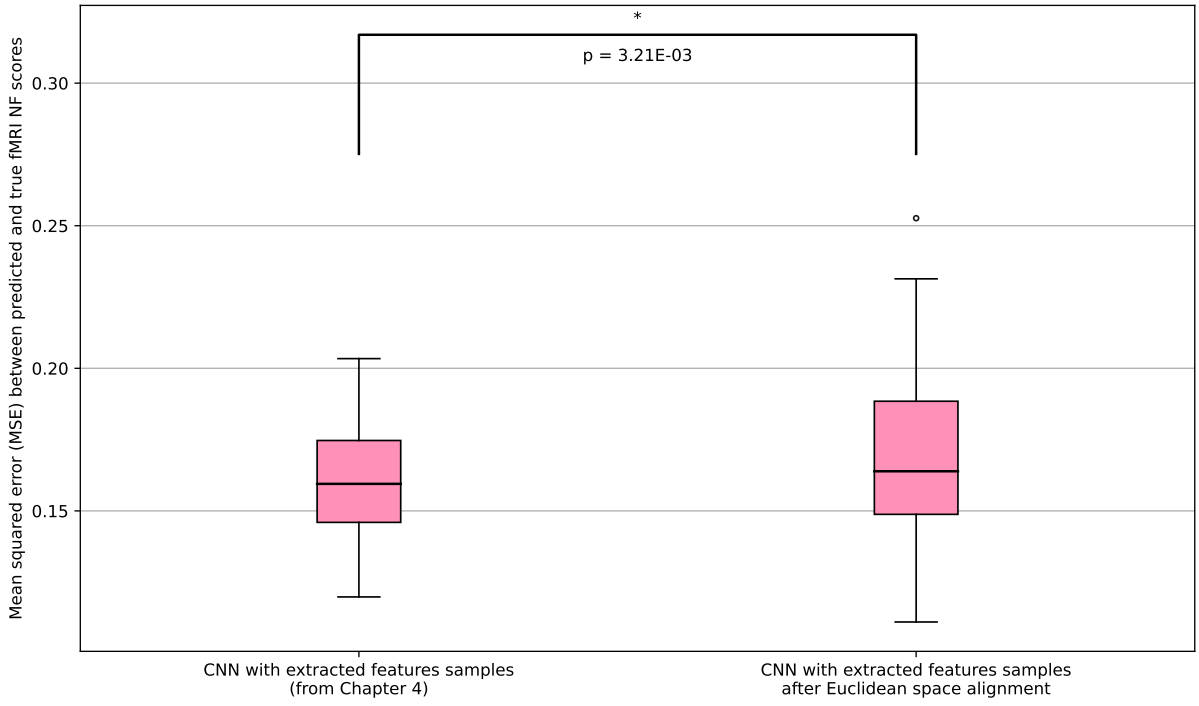


Figure 5.4 – **Results without and with EA for all test subjects across all folds using the mean squared error (MSE) metric.** Each boxplot represents the MSE between the predictions of fMRI NF scores directly from the models and true fMRI NF scores. The p-value from a paired t-test is displayed (*: significant, ns: not significant).

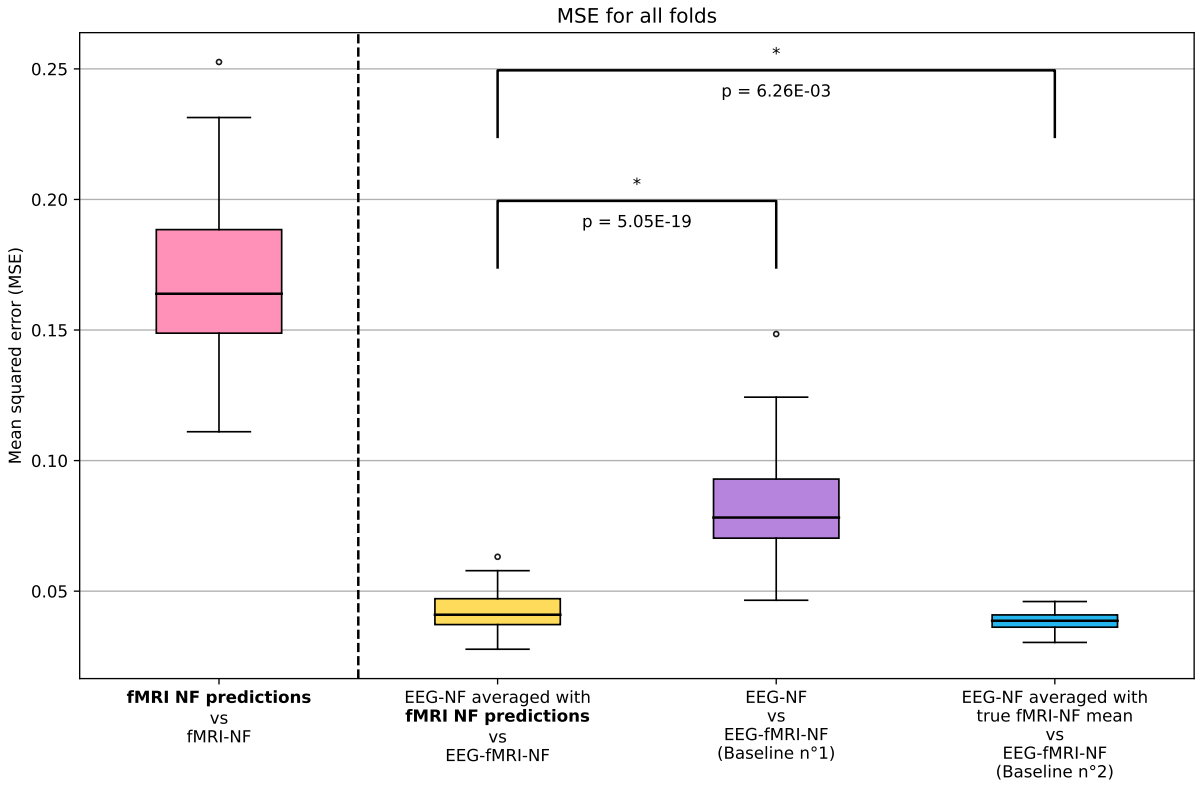


Figure 5.5 – **Results for all test subjects across all folds after EA using the mean squared error (MSE) metric for the CNN with extracted features samples as input.** From left to right: MSE between the predictions of fMRI NF scores directly from the models and true fMRI NF scores (pink). MSE between fMRI NF predictions averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores (yellow). MSE between EEG NF scores and true bi-modal EEG-fMRI NF scores, representing baseline n°1 (purple). MSE between the mean of true fMRI NF scores averaged with EEG NF scores versus true bi-modal EEG-fMRI NF scores, representing baseline n°2 (blue).

5.5 Discussion

Applying the Euclidean space alignment method to our data did not yield satisfactory results. This outcome does not appear to be due to the architecture found by the genetic algorithm, as preliminary tests indicated that even worse performance was obtained using the CNN architecture presented in section 4.4.4. Indeed, given that the input data has been modified, it seems necessary to re-run the genetic search algorithm to find an architecture more capable of producing desirable performance.

Still, I believe that this should not mark the end of exploring the field of domain adaptation for our goal of predicting fMRI NF scores from EEG signals in a global model

setting. As the t-SNE visualizations suggest, the data appear to be highly subject-specific, and the Euclidean space alignment achieves only partial alignment. So, given that these preliminary results were obtained using a simple (subject-by-subject alignment with reference matrices) and practical (directly applicable in Euclidean space) method, which serves as a first step into the field, I still encourage my successors in this research question to continue pursuing this direction.

5.6 Data, code, and model availability

— Data: Just like Chapter 4, the pseudonymized data are available in BIDS format on the OpenNeuro platform at `https://openneuro.org/datasets/ds002338`. It is described in [40] as the XP2 protocol.

— Code: The code developed for this research is also available on Gitlab Inria at `https://gitlab.inria.fr/cpinte/prediction-of-fmri-neurofeedback-scores-from-eeg-signals`. Additionally, the code has been archived with Software Heritage to ensure long-term preservation at `https://archive.softwareheritage.org/swh:1:dir:a651db8d1934543cac321a1723cdf404348e2156;origin=https://gitlab.inria.fr/cpinte/prediction-of-fmri-neurofeedback-scores-from-eeg-signals;visit=swh:1:snp:7004a52e6d9c3e956d761b71bdb3ec6e90a99d8e;anchor=swh:1:rev:f62cd5eff14180de3167bea085892e34a0f3a517`. All implementations were made on an Nvidia RTX A3000 GPU. The Euclidean space alignment takes a few minutes. The genetic search takes between 1 and 2 weeks, while post-genetic training takes between 1 and 2 days, depending on the neural network type and hyperparameters chosen. The predictions on the test dataset are almost instantaneous.

— Model: The trained models weights for all configurations and all folds are shared in the same repository as the code.

5.7 Conclusion

In this final chapter, we applied a feature-based domain adaptation method called Euclidean space alignment (EA) to our EEG data. The aim was to explore this field using a simple, practical method to see whether it could improve performance in predicting fMRI neurofeedback (NF) scores from EEG signals with a global (as opposed to individual) model. The EA method used here aligns each subject's data using a reference matrix, aiming to reduce variability between subjects and make the data more comparable for analysis or machine learning tasks. The effects of this alignment can be observed using t-distributed stochastic neighbor embedding (t-SNE), a nonlinear dimensionality reduction technique that projects high-dimensional data (such as EEG) into a lower-dimensional space (2D or 3D) by grouping similar points and separating dissimilar points to reflect underlying differences. With t-SNE, we observed some reduction in variability between subjects after alignment, although the alignment remains imperfect. Next, we evaluated the impact of this alignment using the best-performing configuration from the previous chapter: the CNN with extracted features-based samples. After running the genetic algorithm for architecture search, followed by training and evaluating the models across 15 folds, the performance of the approach with EA turned out to be worse than that without EA. Despite this, we believe the field of domain adaptation remains promising for advancing towards a global model. Achieving better results may require more in-depth exploration of more advanced methods and better understanding of individual differences.

Chapter highlights

- The Euclidean space alignment (EA) [145] is a domain adaptation method that aligns each subject's data using a reference matrix, aiming to reduce variability between subjects, used here to align our EEG data.
- We visualized the EEG data before and after alignment using the t-distributed stochastic neighbor embedding (t-SNE) [151], a nonlinear dimensionality reduction technique that projects high-dimensional data into a lower-dimensional space to reflect underlying differences.
- To carry out this experiment, we chose to use the configuration that previously yielded the best performance: the CNN approach with extracted features-based samples.
- After applying the same methods as in Chapter 4, we showed that the performance of the approach using Euclidean space alignment was worse than the approach without it.

Conclusion

This thesis has presented and addressed three major questions within the context of EEG-fMRI bi-modality. Firstly, how to automatically and accurately determine the position and label of each EEG electrode within an MRI volume? Secondly, is it possible to predict fMRI NF scores using only EEG signals through multi-subject modeling? Thirdly, can reducing inter-subject variability improve the performance of these fMRI NF score prediction models? For each of these questions, we will summarize the work conducted, discuss the limitations, and mention future directions, whether they are short-term projects that were not pursued due to prioritization and time constraints, or medium to long-term goals that will require further investigation.

In Chapter 3, we presented a new method for detecting and labeling EEG electrodes within an MRI volume. The method involves training a U-Net model on a training dataset along with the associated ground truths, then using this model to obtain a segmentation map, and finally applying a refinement step based on the ICP method to improve the detection and the labeling. This automatic segmentation method is easy to implement, requires very few steps, and provides excellent results. For all these reasons, we believe it could be highly beneficial for all protocols involving simultaneous EEG-MRI acquisitions.

However, this method has some limitations, which were addressed in the discussion of this chapter. To begin with, this method has been tested on PETRA volumes, which is a type of UTE sequence that offers a clearer visualization of the electrodes than more conventional sequences. Although the robustness of the method was evaluated using volumes from another sequence, it also belonged to the UTE category. Therefore, it would be interesting to explore the use of more common sequences, such as T1-weighted or T2-weighted MRI, to make the method more accessible, or to apply it to existing data that did not use UTE sequences.

Additionally, the method was tested only in cases where the goal was to detect electrodes from a 64-channel EEG device. It would be conceivable to adapt the method to other EEG devices with a different number of channels, or even to other modalities such as NIRS (near infrared spectroscopy). In fact, a preliminary test showed that models trained on volumes containing EEG electrodes could detect the position of NIRS optodes

in MRI volumes. Thus, the method could be adapted to other contexts, for example by fine-tuning the models trained on EEG electrode detection for other tasks like detecting NIRS optodes.

On that note, it is important to consider that these models require a training dataset with ground truths, which could limit their application in cases where such data is not available. Even though the additional analysis presented in the chapter showed that the method works even with a relatively small amount of data (9 training volumes), acquiring the data and creating the ground truth is still very costly. This makes it all the more regrettable that we are currently unable to share these models. It would be beneficial for the community to solve the question of whether it is possible to share the weights of a model trained with non-open data, given the risk of data reconstruction. If sharing is permitted, we would naturally proceed to share these weights. However, if it is not, it would be highly advantageous to start the process of making this data open, or apply the method to open data, thereby enabling more accessible use of these models and facilitating fine-tuning.

In Chapter 4, we presented a method for hyperparameter search for model architecture using a genetic algorithm, in the context of predicting fMRI NF scores from EEG signals. We used this genetic algorithm to explore four configurations, involving LSTM and CNN network types, and data formats based on extracted features on one hand and raw signals on the other. Our results showed that the predicted fMRI NF scores, when averaged with EEG NF scores, aligned significantly closer to the true bi-modal EEG-fMRI NF scores than the EEG NF scores alone. This could mean that this approach can enrich EEG NF scores by incorporating fMRI predictions derived from EEG, offering an improved NF score that leverages multi-modal information. However, our fMRI NF predictions had, at best, the same MSE with the true fMRI NF scores as the average of these true scores. Although this requires access to the participant's fMRI data (where our model relies only on EEG signals), we believe that it is not yet possible to use the models developed with this method in the context of unimodal EEG neurofeedback sessions and that they still need improvements.

First, regarding the genetic search algorithm, its main limitation is its computational cost. By making a few optimizations, for example, using parallelization on a computing grid, the architecture search could be performed on more individuals and more generations, using a wider set of hyperparameter values.

This search could also be applied to other types of models, such as the currently

popular transformers. However, our experience with LSTMs suggests that it is necessary to increase both the amount and the representativeness of the data if we wish to use such deep learning models.

We might also consider re-framing the problem altogether. A simplification would be to shift from a regression task to a classification task. Indeed, participants receive visual feedback in the form of a ball moving up and down on a bar with several levels. The problem would then be reduced to predicting the level of the bar, which might be easier. Another potential direction would be to go back to individualized models for subjects. We could start with a global model and then add a fine-tuning step to create a participant-specific model. In a clinical setup, the protocol could thus include an EEG-fMRI neurofeedback session, whose data would be used for fine-tuning, allowing subsequent sessions to occur without the need for MRI.

Another direction I explored, though I did not have enough time to investigate fully, is the field of AI explainability. Specifically, I was interested in feature importance methods, which aim to identify which input features have the most impact on the model output. For example, in our case, these methods could help identifying which channels, power bands, or timesteps of the samples have the most influence. These methods can be useful both for interpreting results and for improving models by providing a better understanding of the patterns learnt.

Finally, it is important to consider that, in the field of machine learning, understanding and cleaning the data is just as, if not more, important than the network design. The t-statistic measure of EEG NF scores, which we used as an approximate indicator of the quality of the EEG NF training, revealed significant variability between participants. Therefore, correcting or reducing individual differences between subjects could help in the training of a global model. This is what we began to investigate in the final chapter of the thesis.

In Chapter 5, we applied a domain adaptation method called Euclidean space alignment (EA) to our EEG data. The goal was to explore this field using a simple and practical method, to see if it could improve the performance of predicting fMRI NF scores from EEG signals using a global model. The EA method used here aligns each subject's data using a reference matrix, aiming to reduce variability between subjects. The effects of this alignment can be observed using t-distributed stochastic neighbor embedding (t-SNE), a nonlinear dimensionality reduction technique that groups similar points and separates dissimilar points to reflect underlying differences.

By analyzing the t-SNE plots that we generated before and after alignment, we observed some reduction in the variability between subjects. However, the alignment did not seem perfect, especially regarding the second and third runs of the subjects. This was also reflected during the evaluation of the full application of the method described in Chapter 4. Indeed, the performance of the approach with EA turned out to be worse than the approach without EA. To further investigate EA, we could consider changing the alignment strategy, for instance by calculating the reference matrix on the entire signal instead of just the rest blocks (though preliminary tests suggested that the rest block approach was more effective), or calculating this reference matrix run by run rather than subject by subject.

However, this method remains particularly simple in the field of domain adaptation. There are numerous methods and frameworks available to reduce this variability, which could be promising despite the results of this initial attempt. In the long term, I encourage further exploration of more advanced methods in this field, which could help advance towards the goal of a global model.

As a final note, I hope to have succeeded in presenting the work I carried out during my thesis in a clear and understandable way. Despite the challenges and results not always satisfactory, I greatly enjoyed conducting this research. I learned a lot, both in the field of machine learning, which I began exploring during my engineering studies, and in the field of neuroimaging and neurofeedback, which I discovered within the Empenn team. I hope that my exploration of these topics will be useful to the next people interested in these questions!

PUBLICATIONS

Peer-reviewed journal articles:

— **Deep learning-based localization of EEG electrodes within MRI acquisitions**, Caroline Pinte, Mathis Fleury, Pierre Maurel. Frontiers in Neurology 12 (2021): 644278. ⟨https://doi.org/10.3389/fneur.2021.644278⟩

Communications:

— **Symposium: Multi-modal neurofeedback methods for post-stroke rehabilitation: Genetic algorithm applied to hyperparameter selection for fMRI neurofeedback score prediction from EEG signals**, oral presentation at the rtFIN conference in November 2024.

— **Poster: Algorithme génétique appliqué à la sélection d'hyperparamètres pour la prédiction des scores neurofeedback IRMf à partir des signaux EEG**, Caroline Pinte, Claire Cury, Pierre Maurel. IABM 2024 - Colloque Français d' Intelligence Artificielle en Imagerie Biomédicale, March 2024, Grenoble, France. ⟨https://inria.hal.science/hal-04529036⟩

— **Poster: Improving portability of bimodal neurofeedback: predicting NF-fMRI scores from EEG signals**, Caroline Pinte, Claire Cury, Pierre Maurel. OHBM 2023 - Organization for Human Brain Mapping, July 2023, Montréal, Canada. ⟨https://inria.hal.science/hal-04168694⟩

— **Poster: Améliorer la portabilité du neurofeedback bimodal : prédire par apprentissage automatique les scores NF-IRMf à partir des signaux EEG**, Caroline Pinte, Claire Cury, Pierre Maurel. IABM 2023 - Colloque Français d' Intelligence Artificielle en Imagerie Biomédicale, March 2023, Paris, France. ⟨https://inria.hal.science/hal-04040883⟩

— **Poster: RNN-LSTM neural network for predicting fMRI neurofeedback scores from EEG signals**, Caroline Pinte, Claire Cury, Pierre Maurel. rtFIN 2022 - Real-Time Functional Imaging and Neurofeedback meeting, October 2022, New Haven, United States. ⟨https://inria.hal.science/hal-03798824⟩

Work in progress:

— **Investigating fMRI neurofeedback score prediction from EEG signals: genetic algorithm applied to hyperparameter selection**, Caroline Pinte, Claire Cury, Pierre Maurel, submitted to MELBA (Machine Learning for Biomedical Imaging) journal in November 2024.

BIBLIOGRAPHY

[1] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel, « Backpropagation applied to handwritten zip code recognition », *in*: *Neural computation* 1.*4* (1989), pp. 541–551.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, « U-net: Convolutional networks for biomedical image segmentation », *in*: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, Springer, 2015, pp. 234–241.

[3] Sepp Hochreiter and Jürgen Schmidhuber, « Long short-term memory », *in*: *Neural computation* 9.*8* (1997), pp. 1735–1780.

[4] Claire Cury, Pierre Maurel, Rémi Gribonval, and Christian Barillot, « A sparse EEG-informed fMRI model for hybrid EEG-fMRI neurofeedback prediction », *in*: *Frontiers in neuroscience* 13 (2020), p. 1451.

[5] Michal Teplan et al., « Fundamentals of EEG measurement », *in*: *Measurement science review* 2.*2* (2002), pp. 1–11.

[6] Richard Caton, « Electrical currents of the brain », *in*: *The Journal of Nervous and Mental Disease* 2.*4* (1875), p. 610.

[7] Priyanka A Abhang, Bharti W Gawali, and Suresh C Mehrotra, *Introduction to EEG-and speech-based emotion recognition*, Academic Press, 2016.

[8] Peter A Bandettini, « Twenty years of functional MRI: the science and the stories », *in*: *Neuroimage* 62.*2* (2012), pp. 575–588.

[9] Stefano Sandrone, Marco Bacigaluppi, Marco R Galloni, Stefano F Cappa, Andrea Moro, Marco Catani, Massimo Filippi, Martin M Monti, Daniela Perani, and Gianvito Martino, « Weighing brain activity with the balance: Angelo Mosso's original manuscripts come to light », *in*: *Brain* 137.*2* (2014), pp. 621–633.

[10] Seiji Ogawa, Tso-Ming Lee, Alan R Kay, and David W Tank, « Brain magnetic resonance imaging with contrast dependent on blood oxygenation. », *in*: *proceedings of the National Academy of Sciences* 87.*24* (1990), pp. 9868–9872.

[11] Seiji Ogawa, « Finding the BOLD effect in brain images », *in*: *Neuroimage* 62.*2* (2012), pp. 608–609.

[12] Kenneth K Kwong, « Record of a single fMRI experiment in May of 1991 », *in*: *Neuroimage* 62.*2* (2012), pp. 610–612.

[13] Nikos K Logothetis, « What we can do and what we cannot do with fMRI », *in*: *Nature* 453.*7197* (2008), pp. 869–878.

[14] Stephen M Smith, « Overview of fMRI analysis », *in*: *The British Journal of Radiology* 77.*suppl_2* (2004), S167–S175.

[15] Massimo Filippi and Filippi, *fMRI techniques and protocols*, vol. 830, Springer, 2016.

[16] Melanie Boly, Olivia Gosseries, Marcello Massimini, and Mario Rosanova, « Functional neuroimaging techniques », *in*: *The Neurology of Conciousness*, Elsevier, 2016, pp. 31–47.

[17] Helmut Laufs, « A personalized history of EEG–fMRI integration », *in*: *Neuroimage* 62.*2* (2012), pp. 1056–1067.

[18] John R Ives, Steven Warach, Franz Schmitt, RR Edelman, and Donald L Schomer, « Monitoring the patient's EEG during echo planar MRI », *in*: *Electroencephalography and clinical neurophysiology* 87.*6* (1993), pp. 417–420.

[19] S Warach, JR Ives, G Schlaug, MR Patel, DG Darby, V Thangaraj, RR Edelman, and DL Schomer, « EEG-triggered echo-planar functional MRI in epilepsy », *in*: *Neurology* 47.*1* (1996), pp. 89–93.

[20] Robin I Goldman, John M Stern, Jerome Engel Jr, and Mark S Cohen, « Acquiring simultaneous EEG and functional MRI », *in*: *Clinical neurophysiology* 111.*11* (2000), pp. 1974–1980.

[21] Louis Lemieux, Philip J Allen, Florence Franconi, Mark R Symms, and David K Fish, « Recording of EEG during fMRI experiments: patient safety », *in*: *Magnetic Resonance in Medicine* 38.*6* (1997), pp. 943–952.

[22] Rodolfo Abreu, Alberto Leal, and Patrıcia Figueiredo, « EEG-informed fMRI: a review of data analysis methods », *in*: *Frontiers in human neuroscience* 12 (2018), p. 29.

[23] Frédéric Grouiller, Laurent Vercueil, Alexandre Krainik, Christoph Segebarth, Philippe Kahane, and Olivier David, « A comparative study of different artefact removal algorithms for EEG signals acquired during functional MRI », *in*: *Neuroimage* 38.*1* (2007), pp. 124–137.

[24] Sven Rothlübbers, Vânia Relvas, Alberto Leal, Teresa Murta, Louis Lemieux, and Patrıcia Figueiredo, « Characterisation and reduction of the EEG artefact caused by the helium cooling pump in the MR environment: validation in epilepsy patient data », *in*: *Brain topography* 28 (2015), pp. 208–220.

[25] Till Nierhaus, Christopher Gundlach, Dominique Goltz, Sabrina D Thiel, Burkhard Pleger, and Arno Villringer, « Internal ventilation system of MR scanners induces specific EEG artifact during simultaneous EEG-fMRI », *in*: *Neuroimage* 74 (2013), pp. 70–76.

[26] Maximilien Chaumon, Dorothy VM Bishop, and Niko A Busch, « A practical guide to the selection of independent components of the electroencephalogram for artifact correction », *in*: *Journal of neuroscience methods* 250 (2015), pp. 47–63.

[27] Karen Mullinger and Richard Bowtell, « Combining EEG and fMRI », *in*: *Magnetic Resonance Neuroimaging: Methods and Protocols* (2011), pp. 303–326.

[28] Jonathan CW Brooks, Olivia K Faull, Kyle TS Pattinson, and Mark Jenkinson, « Physiological noise in brainstem FMRI », *in*: *Frontiers in human neuroscience* 7 (2013), p. 623.

[29] César Caballero-Gaudes and Richard C Reynolds, « Methods for cleaning the BOLD fMRI signal », *in*: *Neuroimage* 154 (2017), pp. 128–149.

[30] Ranganatha Sitaram, Tomas Ros, Luke Stoeckel, Sven Haller, Frank Scharnowski, Jarrod Lewis-Peacock, Nikolaus Weiskopf, Maria Laura Blefari, Mohit Rana, Ethan Oblak, et al., « Closed-loop brain training: the science of neurofeedback », *in*: *Nature Reviews Neuroscience* 18.*2* (2017), pp. 86–100.

[31] Robert T Thibault, Amanda MacPherson, Michael Lifshitz, Raquel R Roth, and Amir Raz, « Neurofeedback with fMRI: A critical systematic review », *in*: *Neuroimage* 172 (2018), pp. 786–807.

[32] Simon H Kohl, David MA Mehler, Michael Lührs, Robert T Thibault, Kerstin Konrad, and Bettina Sorger, « Corrigendum: The Potential of Functional Near-Infrared Spectroscopy-Based Neurofeedback—A Systematic Review and Recommendations for Best Practice », *in*: *Frontiers in Neuroscience* 16 (2022), p. 907941.

[33] Lorraine Perronnet, « Combining EEG and FMRI for Neurofeedback », PhD thesis, University of Rennes I, 2017.

[34] John H Gruzelier, « EEG-neurofeedback for optimising performance. I: A review of cognitive and affective outcome in healthy participants », *in*: *Neuroscience & Biobehavioral Reviews* 44 (2014), pp. 124–141.

[35] John H Gruzelier, « EEG-neurofeedback for optimising performance. II: creativity, the performing arts and ecological validity », *in*: *Neuroscience & Biobehavioral Reviews* 44 (2014), pp. 142–158.

[36] D Corydon Hammond, « What is neurofeedback: An update », *in*: *Journal of neurotherapy* 15.*4* (2011), pp. 305–336.

[37] Martijn Arns, Sabine De Ridder, Ute Strehl, Marinus Breteler, and Anton Coenen, « Efficacy of neurofeedback treatment in ADHD: the effects on inattention, impulsivity and hyperactivity: a meta-analysis », *in*: *Clinical EEG and neuroscience* 40.*3* (2009), pp. 180–189.

[38] Kymberly D Young, Vadim Zotev, Raquel Phillips, Masaya Misaki, Wayne C Drevets, and Jerzy Bodurka, « Amygdala real-time functional magnetic resonance imaging neurofeedback for major depressive disorder: A review », *in*: *Psychiatry and clinical neurosciences* 72.*7* (2018), pp. 466–481.

[39] Toshinori Chiba, Tetsufumi Kanazawa, Ai Koizumi, Kentarou Ide, Vincent Taschereau-Dumouchel, Shuken Boku, Akitoyo Hishimoto, Miyako Shirakawa, Ichiro Sora, Hakwan Lau, et al., « Current status of neurofeedback for post-traumatic stress disorder: a systematic review and the possibility of decoded neurofeedback », *in*: *Frontiers in human neuroscience* 13 (2019), p. 233.

[40] Giulia Lioi, Claire Cury, Lorraine Perronnet, Marsel Mano, Elise Bannier, Anatole Lécuyer, and Christian Barillot, « Simultaneous EEG-fMRI during a neurofeedback task, a brain imaging dataset for multimodal data integration », *in*: *Scientific data* 7.*1* (2020), p. 173.

[41] Vadim Zotev, Raquel Phillips, Han Yuan, Masaya Misaki, and Jerzy Bodurka, « Self-regulation of human brain activity using simultaneous real-time fMRI and EEG neurofeedback », *in*: *NeuroImage* 85 (2014), pp. 985–995.

[42] Marsel Mano, Anatole Lécuyer, Elise Bannier, Lorraine Perronnet, Saman Noorzadeh, and Christian Barillot, « How to build a hybrid neurofeedback platform combining EEG and fMRI », *in*: *Frontiers in neuroscience* 11 (2017), p. 140.

[43] Lorraine Perronnet, Anatole Lécuyer, Marsel Mano, Elise Bannier, Fabien Lotte, Maureen Clerc, and Christian Barillot, « Unimodal versus bimodal EEG-fMRI neurofeedback of a motor imagery task », *in*: *Frontiers in Human Neuroscience* 11 (2017), p. 193.

[44] Jean-Gabriel Ganascia, « Epistemology of AI Revisited in the Light of the Philosophy of Information », *in*: *Knowledge, Technology & Policy* 23 (2010), pp. 57–73.

[45] Bruce G Buchanan, « A (very) brief history of artificial intelligence », *in*: *Ai Magazine* 26.*4* (2005), pp. 53–53.

[46] Arthur L Samuel, « Some studies in machine learning using the game of checkers », *in*: *IBM Journal of research and development* 3.*3* (1959), pp. 210–229.

[47] Rina Dechter, « Learning while searching in constraint-satisfaction problems », *in*: *AAAI National Conference on Artificial Intelligence* (1986).

[48] Sagar Sharma, Simone Sharma, and Anidhya Athaiya, « Activation functions in neural networks », *in*: *Towards Data Sci* 6.*12* (2017), pp. 310–316.

[49] Frank Rosenblatt, *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*, tech. rep., Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[50] Seppo Linnainmaa, « The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors », PhD thesis, Master's Thesis (in Finnish), Univ. Helsinki, 1970.

[51] Paul J Werbos, « Applications of advances in nonlinear sensitivity analysis », *in*: *System Modeling and Optimization: Proceedings of the 10th IFIP Conference New York City, USA, August 31–September 4, 1981*, Springer, 1982, pp. 762–770.

[52] Li Yang and Abdallah Shami, « On hyperparameter optimization of machine learning algorithms: Theory and practice », *in*: *Neurocomputing* 415 (2020), pp. 295–316.

[53] Rajendra V Patil and Renu Agggarwal, « Comprehensive Review on Image Segmentation Applications », *in*: *Sci. Int.(Lahore)* 35.*5* (2023), pp. 573–579.

[54] Yucheng Song, Shengbing Ren, Yu Lu, Xianghua Fu, and Kelvin KL Wong, « Deep learning-based automatic segmentation of images in cardiac radiography: a promising challenge », *in*: *Computer Methods and Programs in Biomedicine* 220 (2022), p. 106821.

[55] Dinh Van Chi Mai, Ioanna Drami, Edward T Pring, Laura E Gould, Phillip Lung, Karteek Popuri, Vincent Chow, Mirza F Beg, Thanos Athanasiou, John T Jenkins, et al., « A systematic review of automated segmentation of 3D computed-tomography scans for volumetric body composition analysis », *in*: *Journal of Cachexia, Sarcopenia and Muscle* 14.*5* (2023), pp. 1973–1986.

[56] Qinghua Huang, Yaozhong Luo, and Qiangzhi Zhang, « Breast ultrasound image segmentation: a survey », *in*: *International journal of computer assisted radiology and surgery* 12 (2017), pp. 493–507.

[57] Zeynettin Akkus, Alfiia Galimzianova, Assaf Hoogi, Daniel L Rubin, and Bradley J Erickson, « Deep learning for brain MRI segmentation: state of the art and future directions », *in*: *Journal of digital imaging* 30 (2017), pp. 449–459.

[58] Azimah Ajam, Azrina Abd Aziz, Vijanth Sagayan Asirvadam, Ahmad Sobri Muda, Ibrahima Faye, and S Jamal Safdar Gardezi, « A review on segmentation and modeling of cerebral vasculature for surgical planning », *in*: *IEEE Access* 5 (2017), pp. 15222–15240.

[59] Huiyan Jiang, Zhaoshuo Diao, and Yu-Dong Yao, « Deep learning techniques for tumor segmentation: a review », *in*: *The Journal of Supercomputing* 78.*2* (2022), pp. 1807–1851.

[60] Mahender Kumar Singh and Krishna Kumar Singh, « A review of publicly available automatic brain segmentation methodologies, machine learning models, recent advancements, and their comparison », *in*: *Annals of Neurosciences* 28.*1-2* (2021), pp. 82–93.

[61] Intisar Rizwan I Haque and Jeremiah Neubert, « Deep learning approaches to biomedical image segmentation », *in*: *Informatics in Medicine Unlocked* 18 (2020), p. 100297.

[62] Minjie Fan and Thomas CM Lee, « Variants of seeded region growing », *in*: *IET image processing* 9.*6* (2015), pp. 478–485.

[63] Jinping Fan, Ruichun Wang, Shiguo Li, and Chunxiao Zhang, « Automated cervical cell image segmentation using level set based active contour model », *in*: *2012 12th International Conference on Control Automation Robotics & Vision (ICARCV)*, IEEE, 2012, pp. 877–882.

[64] Uroosa Sehar and Muhammad Luqman Naseem, « How deep learning is empowering semantic segmentation: Traditional and deep learning techniques for semantic segmentation: A comparison », *in*: *Multimedia Tools and Applications* 81.*21* (2022), pp. 30519–30544.

[65] Arsen Plaksyvyi, Maria Skublewska-Paszkowska, and Paweł Powroźnik, « A comparative analysis of image segmentation using classical and deep learning approach », *in*: *Advances in Science and Technology. Research Journal* 17.*6* (2023).

[66] Kunihiko Fukushima, « Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position », *in*: *Biological cybernetics* 36.*4* (1980), pp. 193–202.

[67] Jonathan Long, Evan Shelhamer, and Trevor Darrell, « Fully convolutional networks for semantic segmentation », *in*: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[68] Nahian Siddique, Sidike Paheding, Colin P Elkin, and Vijay Devabhaktuni, « U-net and its variants for medical image segmentation: A review of theory and applications », *in*: *IEEE access* 9 (2021), pp. 82031–82057.

[69] Shanwen Zhang and Chuanlei Zhang, « Modified U-Net for plant diseased leaf image segmentation », *in*: *Computers and Electronics in Agriculture* 204 (2023), p. 107511.

[70] Sania Gul and Muhammad Salman Khan, « A survey of audio enhancement algorithms for music, speech, bioacoustics, biomedical, industrial and environmental sounds by image U-Net », *in*: *IEEE Access* (2023).

[71] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger, « 3D U-Net: learning dense volumetric segmentation from sparse annotation », *in*: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*, Springer, 2016, pp. 424–432.

[72] Fabian Isensee, Jens Petersen, Andre Klein, David Zimmerer, Paul F Jaeger, Simon Kohl, Jakob Wasserthal, Gregor Koehler, Tobias Norajitra, Sebastian Wirkert, et al., « nnu-net: Self-adapting framework for u-net-based medical image segmentation », *in*: *arXiv preprint arXiv:1809.10486* (2018).

[73] Fabian Isensee, Jens Petersen, Simon AA Kohl, Paul F Jäger, and Klaus H Maier-Hein, « nnu-net: Breaking the spell on successful medical image segmentation », *in*: *arXiv preprint arXiv:1904.08128* 1.*1-8* (2019), p. 2.

[74] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein, « nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation », *in*: *Nature methods* 18.*2* (2021), pp. 203–211.

[75] Robert H Shumway, David S Stoffer, and David S Stoffer, *Time series analysis and its applications*, vol. 3, Springer, 2000.

[76] Dominik S Meier, Howard L Weiner, and Charles RG Guttmann, « Time-series modeling of multiple sclerosis disease activity: a promising window on disease progression and repair potential? », *in*: *Neurotherapeutics* 4 (2007), pp. 485–498.

[77] Asal Asgari, « Clustering of clinical multivariate time-series utilizing recent advances in machine-learning », *in*: *Thesis* (2023).

[78] Robert B Penfold and Fang Zhang, « Use of interrupted time series analysis in evaluating health care quality improvements », *in*: *Academic pediatrics* 13.*6* (2013), S38–S44.

[79] John J Hopfield, « Neural networks and physical systems with emergent collective computational abilities. », *in*: *Proceedings of the national academy of sciences* 79.*8* (1982), pp. 2554–2558.

[80] Yoshua Bengio, Patrice Simard, and Paolo Frasconi, « Learning long-term dependencies with gradient descent is difficult », *in*: *IEEE transactions on neural networks* 5.*2* (1994), pp. 157–166.

[81] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, « Learning phrase representations using RNN encoder-decoder for statistical machine translation », *in*: *arXiv preprint arXiv:1406.1078* (2014).

[82] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, « Empirical evaluation of gated recurrent neural networks on sequence modeling », *in*: *arXiv preprint arXiv:1412.3555* (2014).

[83] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, « Attention is all you need », *in*: *Advances in neural information processing systems* 30 (2017).

[84] Aysu Ezen-Can, « A Comparison of LSTM and BERT for Small Corpus », *in*: *arXiv preprint arXiv:2009.05451* (2020).

[85] Safwan Mahmood Al-Selwi, Mohd Fadzil Hassan, Said Jadid Abdulkadir, Amgad Muneer, Ebrahim Hamid Sumiea, Alawi Alqushaibi, and Mohammed Gamal Ragab, « RNN-LSTM: From applications to modeling techniques and beyond—Systematic review », *in*: *Journal of King Saud University-Computer and Information Sciences* (2024), p. 102068.

[86] Christopher Olah, « Understanding lstm networks », *in*: *https://colah.github.io/posts/2015-08-Understanding-LSTMs/* (2015).

[87] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J Inman, « 1D convolutional neural networks and applications: A survey », *in*: *Mechanical systems and signal processing* 151 (2021), p. 107398.

[88] Caroline Pinte, Mathis Fleury, and Pierre Maurel, « Deep learning-based localization of EEG electrodes within MRI acquisitions », *in*: *Frontiers in Neurology* 12 (2021), p. 644278.

[89] Owen J. Arthurs and Simon Boniface, « How well do we understand the neural origins of the fMRI BOLD signal? », *in*: *Trends in Neurosciences* 25.*1* (2002), pp. 27–31, ISSN: 0166-2236, DOI: https://doi.org/10.1016/S0166-2236(00)01995-0, URL: http://www.sciencedirect.com/science/article/pii/S0166223600019950.

[90] Michal Teplan et al., « Fundamentals of EEG measurement », *in*: *Measurement science review* 2.*2* (2002), pp. 1–11.

[91] Giulia Mele, Carlo Cavaliere, Vincenzo Alfano, Mario Orsini, Marco Salvatore, and Marco Aiello, « Simultaneous EEG-fMRI for Functional Neurological Assessment », *in*: *Frontiers in Neurology* 10 (2019), p. 848, ISSN: 1664-2295, DOI: 10.3389/fneur.2019.00848, URL: https://www.frontiersin.org/article/10.3389/fneur.2019.00848.

[92] R.D. Pascual-Marqui, C.M. Michel, and D. Lehmann, « Low resolution electromagnetic tomography: a new method for localizing electrical activity in the brain », *in*: *International Journal of Psychophysiology* 18.*1* (1994), pp. 49–65, ISSN: 0167-8760, DOI: https://doi.org/10.1016/0167-8760(84)90014-X, URL: http://www.sciencedirect.com/science/article/pii/016787608490014X.

[93] Zeynep Akalin Acar and Scott Makeig, « Effects of Forward Model Errors on EEG Source Localization », *in*: *Brain topography* 26 (Jan. 2013), DOI: 10.1007/s10548-012-0274-6.

[94] Deepak Khosla, Manuel Don, and Betty Kwong, « Spatial mislocalization of EEG electrodes – effects on accuracy of dipole estimation », *in*: *Clinical Neurophysiology* 110.*2* (1999), pp. 261–271, ISSN: 1388-2457, DOI: https://doi.org/10.1016/S0013-4694(98)00121-7, URL: http://www.sciencedirect.com/science/article/pii/S0013469498001217.

[95] L. Koessler, L. Maillard, A. Benhadid, J.-P. Vignal, M. Braun, and H. Vespignani, « Spatial localization of EEG electrodes », *in*: *Neurophysiologie Clinique/Clinical Neurophysiology* 37.*2* (2007), pp. 97–102, ISSN: 0987-7053, DOI: https://doi.org/10.1016/j.neucli.2007.03.002, URL: http://www.sciencedirect.com/science/article/pii/S0987705307000354.

[96] JC De Munck, PCM Vijn, and Henk Spekreijse, « A practical method for determining electrode positions on the head », *in*: *Electroencephalography and clinical Neurophysiology* 78.*1* (1991), pp. 85–87.

[97] Jian Le, Min Lu, Emiliana Pellouchoud, and Alan Gevins, « A rapid method for determining standard 10/10 electrode positions for high resolution EEG studies », *in*: *Electroencephalography and clinical neurophysiology* 106.*6* (1998), pp. 554–558.

[98] S Steddin and K Bötzel, « A new device for scalp electrode localization with unrestrained head », *in*: *J. Neurol* 242 (1995), p. 65.

[99] Jan C. de Munck, Petra J. van Houdt, Ruud M. Verdaasdonk, and Pauly P.W. Ossenblok, « A semi-automatic method to determine electrode positions and labels from gel artifacts in EEG/fMRI-studies », *in*: *NeuroImage* 59.*1* (2012), pp. 399–403, ISSN: 1053-8119, DOI: https://doi.org/10.1016/j.neuroimage.2011.07.021, URL: http://www.sciencedirect.com/science/article/pii/S1053811911007828.

[100] P Adjamian, GR Barnes, A Hillebrand, IE Holliday, Krish Devi Singh, Paul Lawrence Furlong, E Harrington, CW Barclay, and PJG Route, « Co-registration of magnetoencephalography with magnetic resonance imaging using bite-bar-based fiducials and surface-matching », *in*: *Clinical Neurophysiology* 115.*3* (2004), pp. 691–698.

[101] Christopher Whalen, Edward L Maclin, Monica Fabiani, and Gabriele Gratton, « Validation of a method for coregistering scalp recording locations with 3D structural MR images », *in*: *Human brain mapping* 29.*11* (2008), pp. 1288–1301.

[102] Russell Butler, Guillaume Gilbert, Maxime Descoteaux, Pierre-Michel Bernier, and Kevin Whittingstall, « Application of polymer sensitive MRI sequence to localization of EEG electrodes », *in*: *Journal of Neuroscience Methods* 278 (2017), pp. 36–45, ISSN: 0165-0270, DOI: https://doi.org/10.1016/j.jneumeth.2016.12.013, URL: http://www.sciencedirect.com/science/article/pii/S0165027016302953.

[103] Marco Marino, Quanying Liu, Silvia Brem, Nicole Wenderoth, and Dante Mantini, « Automated detection and labeling of high-density EEG electrodes from structural MR images », *in*: *Journal of neural engineering* 13.*5* (2016), p. 056003.

[104] Joanne E. Holmes and Graeme M. Bydder, « MR imaging with ultrashort TE (UTE) pulse sequences: Basic principles », *in*: *Radiography (London 1995)* 11.*3* (2005), RADIOLOGY AND NUCLEAR MEDICINE, pp. 163–174, ISSN: 1078-8174, URL: http://inis.iaea.org/search/search.aspx?orig_q=RN:37006330.

[105] V. Keereman, Y. Fierens, T. Broux, Y. De Deene, M. Lonneux, and S. Vandenberghe, « MRI-Based Attenuation Correction for PET/MRI Using Ultrashort Echo Time Sequences », *in*: *Journal of Nuclear Medicine* 51.*5* (May 2010), pp. 812–818, DOI: 10.2967/jnumed.109.065425, URL: https://doi.org/10.2967/jnumed.109.065425.

[106] Mathis Fleury, Christian Barillot, Marsel Mano, Elise Bannier, and Pierre Maurel, « Automated Electrodes Detection During Simultaneous EEG/fMRI », *in*: *Frontiers in ICT* 5 (2019), p. 31, ISSN: 2297-198X, DOI: 10.3389/fict.2018.00031, URL: https://www.frontiersin.org/article/10.3389/fict.2018.00031.

[107] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, « U-net: Convolutional networks for biomedical image segmentation », *in*: *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[108] David M. Grodzki, Peter M. Jakob, and Bjoern Heismann, « Ultrashort echo time imaging using pointwise encoding time reduction with radial acquisition (PETRA) », *in*: *Magnetic Resonance in Medicine* 67.*2* (2012), pp. 510–518, DOI: 10.1002/mrm.23017, eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/mrm.23017, URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.23017.

[109] P. J. Besl and N. D. McKay, « A method for registration of 3-D shapes », *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.*2* (1992), pp. 239–256.

[110] Mark Jenkinson, Christian F. Beckmann, Timothy E.J. Behrens, Mark W. Woolrich, and Stephen M. Smith, « FSL », *in*: *NeuroImage* 62.*2* (2012), 20 YEARS OF fMRI, pp. 782–790, ISSN: 1053-8119, DOI: https://doi.org/10.1016/j.neuroimage.2011.09.015, URL: http://www.sciencedirect.com/science/article/pii/S1053811911010603.

[111] Russell Butler, *Electrode hand labeling and segmentation based off of UTE image intensity*, [Online]. Available: https://github.com/russellu/ute_git/. [Accessed: Aug. 19, 2020]., 2017.

[112] Fabian Isensee, Paul F Jäger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein, « Automated design of deep learning methods for biomedical image segmentation », *in*: *arXiv preprint arXiv:1904.08128* (2019).

[113] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger, « 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation », *in*: *CoRR* abs/1606.06650 (2016), arXiv: 1606.06650, URL: http://arxiv.org/abs/1606.06650.

[114] Ranganatha Sitaram, Tomas Ros, Luke Stoeckel, Sven Haller, Frank Scharnowski, Jarrod Lewis-Peacock, Nikolaus Weiskopf, Maria Laura Blefari, Mohit Rana, Ethan Oblak, et al., « Closed-loop brain training: the science of neurofeedback », *in*: *Nature Reviews Neuroscience* 18.*2* (2017), pp. 86–100.

[115] Tian Renton, Alana Tibbles, and Jane Topolovec-Vranic, « Neurofeedback as a form of cognitive rehabilitation therapy following stroke: A systematic review », *in*: *PloS one* 12.*5* (2017), e0177290.

[116] Tianlu Wang, Dante Mantini, and Celine R Gillebert, « The potential of real-time fMRI neurofeedback for stroke rehabilitation: A systematic review », *in*: *cortex* 107 (2018), pp. 148–165.

[117] J-M Batail, Stéphanie Bioulac, François Cabestaing, Christophe Daudet, Dominique Drapier, Mélanie Fouillen, Thomas Fovet, Aurore Hakoun, Renaud Jardri, Camille Jeunet, et al., « EEG neurofeedback research: A fertile ground for psychiatry? », *in*: *L'encephale* 45.*3* (2019), pp. 245–255.

[118] James Sulzer, Sven Haller, Frank Scharnowski, Nikolaus Weiskopf, Niels Birbaumer, Maria Laura Blefari, Annette B Bruehl, Leonardo G Cohen, R Christopher DeCharms, Roger Gassert, et al., « Real-time fMRI neurofeedback: progress and challenges », *in*: *Neuroimage* 76 (2013), pp. 386–399.

[119] Robert T Thibault, Amanda MacPherson, Michael Lifshitz, Raquel R Roth, and Amir Raz, « Neurofeedback with fMRI: A critical systematic review », *in*: *Neuroimage* 172 (2018), pp. 786–807.

[120] Melanie Boly, Olivia Gosseries, Marcello Massimini, and Mario Rosanova, « Functional neuroimaging techniques », *in*: *The Neurology of Conciousness*, Elsevier, 2016, pp. 31–47.

[121] Giuseppina Ciccarelli, Giovanni Federico, Giulia Mele, Angelica Di Cecca, Miriana Migliaccio, Ciro Rosario Ilardi, Vincenzo Alfano, Marco Salvatore, and Carlo Cavaliere, « Simultaneous real-time EEG-fMRI neurofeedback: A systematic review », *in*: *Frontiers in Human Neuroscience* 17 (2023), p. 1123014.

[122] Jan C de Munck, Sonia I Gonçalves, L Huijboom, Joost PA Kuijer, Petra JW Pouwels, Rob M Heethaar, and FH Lopes da Silva, « The hemodynamic response of the alpha rhythm: an EEG/fMRI study », *in*: *Neuroimage* 35.*3* (2007), pp. 1142–1151.

[123] René Scheeringa, Pascal Fries, Karl-Magnus Petersson, Robert Oostenveld, Iris Grothe, David G Norris, Peter Hagoort, and Marcel CM Bastiaansen, « Neuronal dynamics underlying high-and low-frequency EEG oscillations contribute independently to the human BOLD signal », *in*: *Neuron* 69.*3* (2011), pp. 572–583.

[124] Cesare Magri, Ulrich Schridde, Yusuke Murayama, Stefano Panzeri, and Nikos K Logothetis, « The amplitude and timing of the BOLD signal reflects the relationship between local field potential power at different frequencies », *in*: *Journal of Neuroscience* 32.*4* (2012), pp. 1395–1407.

[125] Galina V Portnova, Alina Tetereva, Vladislav Balaev, Mikhail Atanov, Lyudmila Skiteva, Vadim Ushakov, Alexey Ivanitsky, and Olga Martynova, « Correlation of BOLD signal with linear and nonlinear patterns of EEG in resting state EEG-informed fMRI », *in*: *Frontiers in human neuroscience* 11 (2018), p. 654.

[126] Claire Cury, Giulia Lioi, Lorraine Perronnet, Anatole Lécuyer, Pierre Maurel, and Christian Barillot, « Impact of 1d and 2d visualisation on eeg-fmri neurofeedback training during a motor imagery task », *in*: *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, IEEE, 2020, pp. 1018–1021.

[127] Yoshua Bengio, Patrice Simard, and Paolo Frasconi, « Learning long-term dependencies with gradient descent is difficult », *in*: *IEEE transactions on neural networks* 5.*2* (1994), pp. 157–166.

[128] Sepp Hochreiter and Jürgen Schmidhuber, « Long short-term memory », *in*: *Neural computation* 9.*8* (1997), pp. 1735–1780.

[129] Klaus Greff, Rupesh K Srivastava, Jan Koutnık, Bas R Steunebrink, and Jürgen Schmidhuber, « LSTM: A search space odyssey », *in*: *IEEE transactions on neural networks and learning systems* 28.*10* (2016), pp. 2222–2232.

[130] Antonio Rafael Sabino Parmezan, Vinicius MA Souza, and Gustavo EAPA Batista, « Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model », *in*: *Information sciences* 484 (2019), pp. 302–337.

[131] Xiang Zhang, Junbo Zhao, and Yann LeCun, « Character-level convolutional networks for text classification », *in*: *Advances in neural information processing systems* 28 (2015).

[132] Aslıhan Cura, Haluk Küçük, Erdem Ergen, and İsmail Burak Öksüzoğlu, « Driver profiling using long short term memory (LSTM) and convolutional neural network (CNN) methods », *in*: *IEEE Transactions on Intelligent Transportation Systems* 22.*10* (2020), pp. 6572–6582.

[133] Felipe P Marinho, Paulo AC Rocha, Ajalmar RR Neto, and Francisco DV Bezerra, « Short-term solar irradiance forecasting using CNN-1D, LSTM, and CNN-LSTM deep neural networks: A case study with the Folsom (USA) dataset », *in*: *Journal of Solar Energy Engineering* 145.*4* (2023), p. 041002.

[134] John H Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, MIT press, 1992.

[135] Geoffrey F Miller, Peter M Todd, and Shailesh U Hegde, « Designing Neural Networks Using Genetic Algorithms. », *in*: *ICGA*, vol. 89, 1989, pp. 379–384.

[136] PG Benardos and G-C Vosniakos, « Optimizing feedforward artificial neural network architecture », *in*: *Engineering applications of artificial intelligence* 20.*3* (2007), pp. 365–382.

[137] Lingxi Xie and Alan Yuille, « Genetic cnn », *in*: *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1379–1388.

[138] Yanan Sun, Bing Xue, Mengjie Zhang, Gary G Yen, and Jiancheng Lv, « Automatically designing CNN architectures using the genetic algorithm for image classification », *in*: *IEEE transactions on cybernetics* 50.*9* (2020), pp. 3840–3854.

[139] Nikolaos Gorgolis, Ioannis Hatzilygeroudis, Zoltan Istenes, and Lazlo–Grad Gyenne, « Hyperparameter optimization of LSTM network models through genetic algorithm », *in*: *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, IEEE, 2019, pp. 1–4.

[140] James V Hansen, James B McDonald, and Ray D Nelson, « Time series prediction with Genetic-Algorithm designed neural networks: An empirical comparison with modern statistical models », *in*: *Computational Intelligence* 15.*3* (1999), pp. 171–184.

[141] Salah Bouktif, Ali Fiaz, Ali Ouni, and Mohamed Adel Serhani, « Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches », *in*: *Energies* 11.*7* (2018), p. 1636.

[142] C Erden, « Genetic algorithm-based hyperparameter optimization of deep learning models for PM2. 5 time-series prediction », *in*: *International Journal of Environmental Science and Technology* 20.*3* (2023), pp. 2959–2982.

[143] Hyejung Chung and Kyung-shik Shin, « Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction », *in*: *Neural Computing and Applications* 32.*12* (2020), pp. 7897–7914.

[144] Lorraine Perronnet, Anatole Lécuyer, Marsel Mano, Mathis Fleury, Giulia Lioi, Claire Cury, Maureen Clerc, Fabien Lotte, and Christian Barillot, « Learning 2-in-1: towards integrated EEG-fMRI-neurofeedback », *in*: *BioRxiv* (2018), p. 397729.

[145] He He and Dongrui Wu, « Transfer learning for brain–computer interfaces: A Euclidean space data alignment approach », *in*: *IEEE Transactions on Biomedical Engineering* 67.*2* (2019), pp. 399–410.

[146] Reinmar Kobler, Jun-ichiro Hirayama, Qibin Zhao, and Motoaki Kawanabe, « SPD domain-specific batch normalization to crack interpretable unsupervised domain adaptation in EEG », *in*: *Advances in Neural Information Processing Systems* 35 (2022), pp. 6219–6235.

[147] Wouter M Kouw and Marco Loog, « A review of domain adaptation without target labels », *in*: *IEEE transactions on pattern analysis and machine intelligence* 43.*3* (2019), pp. 766–785.

[148] Paolo Zanini, Marco Congedo, Christian Jutten, Salem Said, and Yannick Berthoumieu, « Transfer learning: A Riemannian geometry framework with applications to brain–computer interfaces », *in*: *IEEE Transactions on Biomedical Engineering* 65.*5* (2017), pp. 1107–1116.

[149] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten, « Multiclass brain–computer interface classification by Riemannian geometry », *in*: *IEEE Transactions on Biomedical Engineering* 59.*4* (2011), pp. 920–928.

[150] Florian Yger, Maxime Berar, and Fabien Lotte, « Riemannian approaches in brain-computer interfaces: a review », *in*: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.*10* (2016), pp. 1753–1762.

[151] Laurens Van der Maaten and Geoffrey Hinton, « Visualizing data using t-SNE. », *in*: *Journal of machine learning research* 9.*11* (2008).

COLLEGE DOCTORAL BRETAGNE

MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

Titre : Apprentissage automatique pour le neurofeedback bi-modal EEG-IRMf : localisation des électrodes EEG et prédiction des scores NF IRMf

Mot clés : EEG, IRMf, neurofeedback, apprentissage automatique

Résumé : Cette thèse explore l'apport des méthodes d'apprentissage automatique dans le contexte de la bi-modalité EEG-IRMf, avec pour objectif de localiser automatiquement et précisément les électrodes EEG dans un volume IRM et de prédire des scores neurofeedback IRMf à partir de signaux EEG. La première partie présente le contexte et les outils utilisés, en abordant les modalités EEG et IRMf ainsi que leur combinaison, le neurofeedback, les réseaux de neurones artificiels, la segmentation d'images et la régression de séries temporelles. La deuxième partie comprend trois contributions principales. La première décrit le développement d'une méthode permettant détecter automatiquement la position et l'étiquetage des électrodes EEG dans un volume IRM à l'aide d'une séquence IRM spécifique. La deuxième contribution propose une méthode de recherche d'hyperparamètres d'architecture de modèles basée sur un algorithme génétique. Ces modèles sont ensuite entraînés sur plusieurs sujets afin de prédire des scores neurofeedback IRMf à partir de signaux EEG. Cette étude compare différentes architectures issues de deux catégories de réseaux neuronaux : les LSTMs et les CNNs. Enfin, la troisième contribution consiste à étudier une piste d'amélioration de ces modèles. Ce travail évalue l'impact de la réduction de la variabilité inter-sujet sur les performances, en appliquant un alignement dans l'espace euclidien aux données EEG.

Title: Machine learning for bi-modal EEG-fMRI neurofeedback: EEG electrodes localization and fMRI NF scores prediction

Keywords: EEG, fMRI, neurofeedback, machine learning

Abstract: This thesis explores the impact of machine learning methods in the context of bi-modal EEG-fMRI, with the goal of automatically and accurately localizing EEG electrodes in an MRI volume and predicting fMRI neurofeedback scores from EEG signals. The first part presents the context and tools used, covering EEG and fMRI modalities as well as their combination, neurofeedback, artificial neural networks, image segmentation and time series regression. The second part contains three main contributions. The first one describes the development of a method for automatically detecting the position and labeling of EEG electrodes in an MRI volume using a specific MRI sequence. The second contribution proposes a method for finding model architecture hyperparameters based on a genetic algorithm. These models are then trained on several subjects to predict fMRI neurofeedback scores from EEG signals. This study compares different architectures from two categories of neural networks: LSTMs and CNNs. Finally, the third contribution consists in investigating an area of improvement for these models. This work evaluates the impact on model performance of reducing inter-subject variability, by applying an alignment in Euclidean space to EEG data.